



**HAL**  
open science

# Very high-order accurate finite volume scheme for the convection-diffusion equation with general boundary conditions on arbitrary curved boundaries

Ricardo Costa, João Nóbrega, Stéphane Clain, Gaspar Machado, Raphaël Loubère

## ► To cite this version:

Ricardo Costa, João Nóbrega, Stéphane Clain, Gaspar Machado, Raphaël Loubère. Very high-order accurate finite volume scheme for the convection-diffusion equation with general boundary conditions on arbitrary curved boundaries. *International Journal for Numerical Methods in Engineering*, 2019, 117 (2), pp.188-220. 10.1002/nme.5953 . hal-02488969

**HAL Id: hal-02488969**

**<https://hal.science/hal-02488969>**

Submitted on 22 Dec 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Very high-order accurate finite volume scheme for the convection-diffusion equation with general boundary conditions on arbitrary curved boundaries

Ricardo Costa<sup>a,\*</sup>, João M. Nóbrega<sup>a</sup>, Stéphane Clain<sup>b</sup>, Gaspar J. Machado<sup>b</sup>, Raphaël Loubère<sup>c</sup>

<sup>a</sup>IPC/i3N – Instituto de Polímeros e Compósitos, Universidade do Minho, Campus de Azurém, 4804-533 Guimarães, Portugal

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

<sup>c</sup>IMB – Institut de Mathématiques de Bordeaux, Université de Bordeaux, 33405 Talence, France

---

## Abstract

Obtaining very high-order accurate solutions in curved domains is a challenging task as the accuracy of numerical methods may dramatically reduce without an appropriate treatment of the boundary condition. The classical techniques to preserve the optimal order of accuracy, proposed in the context of Finite Element and Finite Volume methods, rely on curved mesh elements to fit the curved boundaries. Such techniques often demand sophisticated meshing algorithms, cumbersome quadrature rules for integration, and complex nonlinear transformations to map the locally curved element onto the reference polygonal mesh elements. In this regard, the *Reconstruction for Off-site Data* method was proposed in Costa et al. (2018) to provide very high-order accurate polynomial reconstructions for curved boundaries, enabling the integration of the governing equations on polygonal mesh elements, and, therefore, without the need of complex integration quadrature rules or nonlinear transformations. The method was introduced with Dirichlet boundary conditions and the present article proposes an extension for general boundary conditions, which represents an important advance for real context applications. To achieve that, a generic framework to compute the polynomial reconstructions is also proposed based on the Least-squares method, enabling the method to handle general constraints and further improving the algorithm. A comprehensive numerical benchmark test suite is provided to validate and assess the accuracy, convergence rates, robustness, and efficiency, which proves that boundary conditions for curved domains are properly satisfied and the optimal very high-order convergence rates are successfully achieved.

**Keywords:** Very high-order accurate Finite Volume scheme, Arbitrary curved domains, General boundary conditions, Polynomial reconstructions, Least-squares method, *Reconstruction for Off-site Data* method, Convection-diffusion equation

---

## 1. Introduction

Nowadays the numerical simulations demand an increasing computational effort as models become far more complex than what they used to be. In that regard, computational efficiency, usually defined as the ratio between solution quality and computational effort, implies a trade-off between the order of accuracy of the method and mesh size or the number of degrees of freedom. To improve this ratio for smooth solutions, increasing the order of accuracy of the method is clearly preferable compared to mesh refinement. Although such approach improves the computational efficiency, obtaining very high-order accurate approximations is a current and challenging task.

The majority of very high-order accurate methods are designed specifically for polygonal (or polyhedral) domains, and therefore numerical difficulties to obtain the optimal convergence order of accuracy arise when dealing with boundary conditions prescribed on curved boundaries. For a short literature review on this topic, the reader is referred to the introduction in [1], which is summarized in the following. The classical approach to deal with curved boundary conditions is based on the Isoparametric Element method [2, 3], which requires, on one hand, the introduction of curved mesh elements and, on the other hand, a nonlinear transformation to map the local curved element onto the reference polygonal mesh element. An alternative approach, dedicated to the Finite Volume method, was initially proposed by Ollivier-Gooch et al. [4]. The technique does not require nonlinear transformations but the main shortcoming remains, being mainly related to the meshing algorithm needed to generate curved mesh elements fitting

---

\*Corresponding author

Email address: [rcosta@ipc.uminho.pt](mailto:rcosta@ipc.uminho.pt) (Ricardo Costa)

the boundary in addition to the high-order accurate quadrature rules for integration applied on the nonpolygonal mesh elements. Handling complex two- or three-dimensional curved elements turns out to be a cumbersome task that results into significant computational costs [5, 6, 7].

In [1], the authors introduced a new approach in the Finite Volume context, called the *Reconstruction for Off-site Data* method (shortened to ROD method), which is capable of handling complex curved boundaries with a very high-order accuracy. The strength and the novelty of the method rely in the fact that only polygonal mesh elements are used, even those generated to fit at best the curved boundaries, for which a mismatch between the mesh boundary and the domain boundary exists. The method integrates the prescribed boundary conditions into the scheme using polynomial reconstructions, which are computed based on the least-squares method for which the parameterized boundary is needed. Moreover, the governing equations are integrated on the polygonal mesh elements and, consequently, the fluxes are defined on the boundaries of the polygonal control volumes. Therefore, sophisticated meshing algorithms with curved elements are no more required, nor nonlinear transformations, nor cumbersome quadrature rules for integration for curved elements are then demanded. **There are very few methods capable of handling curved domains with polygonal meshes and most of them limited to first- or second-order of accuracy. Recently extension of the immersed boundary method to the fourth-order has been proposed in the framework of the Fourier spectral method [8, 9].**

The ROD method was developed in [1] for the steady-state two-dimensional convection-diffusion problem with Dirichlet boundary conditions, and subsequently improvements are reported in this article, namely: (i) the introduction of Neumann and Robin boundary conditions, which represents an important advance for real context applications; and (ii) the development of a generic framework to compute the polynomial reconstructions based on the least-squares method, which enables to consider general constraints and improves upon the algorithm.

The remaining sections of the article are organized as follows. Section 2 presents the model, the mesh, and the basic assumptions and notations. Section 3 introduces the generic framework to compute the polynomial reconstructions based on the least-squares method. Section 4 is dedicated to describe the ROD method based on the previously introduced polynomial reconstructions and the Dirichlet, Neumann, and Robin boundary conditions on curved boundaries are addressed. Section 5 presents the very high-order accurate Finite Volume scheme comprising the polynomial reconstructions and the ROD method. Section 6 provides a comprehensive numerical benchmark test suite that validates the proposed method, and the article is completed in Section 7 with the conclusions and some perspectives for future work.

## 2. Problem formulation, mesh, and notation

The goal is to solve the two-dimensional steady-state convection-diffusion problem, in Cartesian coordinates  $\mathbf{x} := (x, y)$ , for curved domains with boundaries given by regular Jordan curves that admit a local parameterization. To this end, let  $\Omega$  be an open bounded domain of  $\mathbb{R}^2$  with boundary  $\Gamma$ , partitioned into three non-overlapping and possibly empty subsets,  $\Gamma_D$ ,  $\Gamma_N$ , and  $\Gamma_R$ , such that  $\Gamma = \Gamma_D \cup \Gamma_N \cup \Gamma_R$ . Vector  $\mathbf{n} = (n_x, n_y) := (n_x(\mathbf{x}), n_y(\mathbf{x}))$  stands for the outward unit normal vector to  $\Gamma$  at point  $\mathbf{x}$  (see Fig. 1).

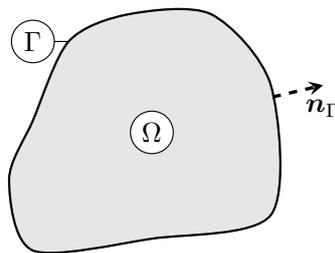


Figure 1: Arbitrary curved domain.

### 2.1. Convection-diffusion problem

The governing equation for unknown function  $\phi := \phi(\mathbf{x})$  is given by

$$\nabla \cdot (\mathbf{u}\phi - \kappa \nabla \phi) = f, \quad \text{in } \Omega, \quad (1)$$

where  $\mathbf{u} = (u_x, u_y) := (u_x(\mathbf{x}), u_y(\mathbf{x}))$  is a velocity field in  $\mathbb{R}^2$ ,  $\kappa := \kappa(\mathbf{x})$  is a non-negative diffusion coefficient, and  $f := f(\mathbf{x})$  is a given source term function. All functions are assumed to be regular in  $\Omega$ . To complete Equation (1), boundary partitions  $\Gamma_D$ ,  $\Gamma_N$ , and  $\Gamma_R$  are prescribed with the following boundary conditions:

- On partition  $\Gamma_D$  a Dirichlet boundary condition is prescribed with equation

$$\phi = g_D, \text{ in } \Gamma_D, \quad (2)$$

where  $g_D := g_D(\mathbf{x})$  is a given regular function on  $\Gamma_D$ ;

- On partition  $\Gamma_N$  a Neumann boundary condition is prescribed with equation

$$-\kappa \nabla \phi \cdot \mathbf{n} = g_N, \text{ in } \Gamma_N, \quad (3)$$

where  $g_N := g_N(\mathbf{x})$  is a given regular function on  $\Gamma_N$ ;

- On partition  $\Gamma_R$  a Robin boundary condition is prescribed with equation

$$\sigma \phi + \rho \nabla \phi \cdot \mathbf{n} = g_R, \text{ in } \Gamma_R, \quad (4)$$

where  $g_R := g_R(\mathbf{x})$  is a given regular function on  $\Gamma_R$ , and  $\sigma := \sigma(\mathbf{x})$  and  $\rho := \rho(\mathbf{x})$  are given regular real-valued coefficient functions on  $\Gamma_R$ ; for instance, if  $\sigma := \mathbf{u} \cdot \mathbf{n}$  and  $\rho := -\kappa$ , then the Robin boundary condition represents a total flux boundary condition.

## 2.2. Mesh and notations

Domain  $\Omega$  is discretized into a mesh  $\mathcal{M}$  comprising  $N_C$  non-overlapping convex and polygonal cells without gap and hole, denoted by  $c_i, i \in C_M = \{1, \dots, N_C\}$ . Two adjacent cells  $c_i$  and  $c_j, i, j \in C_M$ , share a common edge denoted by  $e_{ij}$  and named inner edge (notice that  $e_{ji}$  denotes the same edge as  $e_{ij}$ ). On the mesh boundary, the edges are denoted by  $e_{iB}$  and named boundary edges if they belong to cell  $c_i, i \in C_M$  and discretize boundary partition  $\Gamma_B$ , where the index  $B$  is replaced by letter D, N, or R accordingly to the boundary partition discretized. That is, if the edge discretizes boundary partition  $\Gamma_D, \Gamma_N$ , or  $\Gamma_R$ , then edge  $e_{iB}$  is after all denoted by  $e_{iD}, e_{iN}$ , or  $e_{iR}$ , respectively. For the sake of simplicity and without loss of generality, each mesh boundary edge is associated to only one boundary partition and the mesh boundary vertices are assumed to belong to physical boundary  $\Gamma$ .

To simplify, let  $\mathcal{B}_M$  be the subset of  $C_M$  gathering the indices of the cells with an edge on the mesh boundary, that is, all  $i$  indices such that  $e_{iB}, B \in \{D, N, R\}$ , is an edge in  $\mathcal{M}$ . Subsets of  $\mathcal{B}_M$  denoted as  $\mathcal{B}_{M,D}, \mathcal{B}_{M,N}$ , and  $\mathcal{B}_{M,R}$  are also consider to gather the indices of the cells having an edge on  $\Gamma_D, \Gamma_N$ , and,  $\Gamma_R$ , respectively, that is, any index  $i$  such that  $e_{iD}, e_{iN}$ , or  $e_{iR}$  is an edge in  $\mathcal{M}$ , respectively.

In the scope of this article, keywords *physical* and *computational* are used to distinguish the real domain from the discretized domain, respectively. In this way, the following terms and definitions are used:

- The computational domain gathers all the cells and is defined by  $\Omega_\Delta = \bigcup_{i \in C_M} c_i$ , which stands for a representative approximation of the physical domain defined by  $\Omega$ ;
- The computational boundary gathers all the boundary edges and is defined by  $\Gamma_\Delta = \bigcup_{i \in \mathcal{B}_M, B \in \{D, N, R\}} e_{iB}$ , which stands for a representative approximation of the physical boundary defined by  $\Gamma$ ;
- The computational boundary partition gathers all the boundary partition edges and is defined by  $\Gamma_{B,\Delta} = \bigcup_{i \in \mathcal{B}_{M,B}} e_{iB}, B \in \{D, N, R\}$ , which stands for a representative approximation of the physical boundary partition defined by  $\Gamma_B$ .

**Remark 1.** Assuming that domain  $\Omega$  is nonpolygonal and bounded by a regular Jordan curve, the corresponding polygonal approximation,  $\Omega_\Delta$ , does not fully overlap the physical domain and a mismatch of order  $O(h^2)$  is expected between the physical and the computational boundaries, where  $h$  is the characteristic mesh length. Such mismatch represents a potential accuracy degradation for any more than second-order accurate scheme.

In Table 1, additional notation is introduced (see also Figure 2).

## 3. Least-squares method and polynomial reconstructions

The least-squares method for Finite Volume method, described in [10, 11, 12], is a powerful tool to compute local representations of the underlying solution from scattered pointwise data associated to the cells. The underlying solution is assumed to be continuous and smooth enough in domain  $\Omega$ . Hence, no limiting procedure must be designed to avoid parasitical oscillations resulting from Gibbs phenomenon, such as slope limiting for piecewise linear functions [13, 14, 15], essentially non-oscillatory ((W)ENO) techniques [16, 17, 18, 19, 20, 21, 22], or advanced hierarchical

Table 1: Notation associated to the edges, cells, reference and quadrature points.

Element type	Notation	Definition	Choice
Cells $c_i, i \in \mathcal{C}_M$	$\partial c_i$	Boundary	
	$ c_i $	Area	
	$\mathbf{m}_i = (m_{i,x}, m_{i,y})$	Reference point, can be any point in $c_i$	Centroid
	$\mathbf{q}_{i,r} = (q_{i,r,x}, q_{i,r,y})$	Quadrature points	Gaussian
Inner edges $e_{ij}, i \in \mathcal{C}_M,$ $j \in \mathcal{N}_i$	$ e_{ij} $	Length	
	$\mathbf{n}_{ij} = (n_{ij,x}, n_{ij,y}) = -\mathbf{n}_{ji}$	Outward unit normal from $c_i$ to $c_j$	
	$\mathbf{m}_{ij} = (m_{ij,x}, m_{ij,y}) = \mathbf{m}_{ji}$	Reference point, can be any point in $e_{ij}$	Midpoint
	$\mathbf{q}_{ij,r} = (q_{ij,r,x}, q_{ij,r,y}) = \mathbf{q}_{ji,r}$	Quadrature points with $r = 1, \dots, R$	Gaussian
Boundary edges $e_{iB}, i \in \mathcal{B}_{M,B},$ $B \in \{D, N, R\}$	$ e_{iB} $	Length	
	$\mathbf{n}_{iB} = (n_{iB,x}, n_{iB,y})$	Outward unit normal of $e_{iB}$	
	$\mathbf{m}_{iB} = (m_{iB,x}, m_{iB,y})$	Reference point, can be any point in $e_{iB}$	Midpoint
	$\mathbf{q}_{iB,r} = (q_{iB,r,x}, q_{iB,r,y})$	Quadrature points with $r = 1, \dots, R$	Gaussian

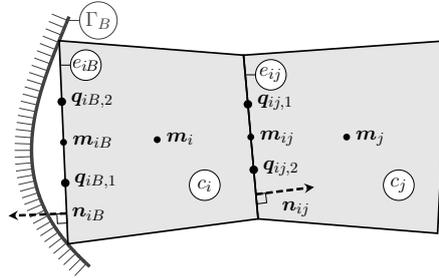


Figure 2: Mesh notation for the edges and cells, the quadrature and reference points, and the outward unit normal vectors.

limiting strategies [23, 24] for quadratic or higher polynomials. The method consists in computing a local least-squares approximation of function  $\phi(\mathbf{x})$  for each mesh element, given in the general form as  $\varphi(\mathbf{x}) = \boldsymbol{\eta}^T \mathbf{p}(\mathbf{x})$ , where  $\boldsymbol{\eta}$  is a vector of coefficients and  $\mathbf{p}(\mathbf{x})$  is a basis function vector. These local approximations are used to compute accurate flux approximations and arbitrary high-order accurate convergence rates can be achieved under mesh refinement. Although any type of basis function vectors can be considered, polynomial basis functions present a high flexibility and are easy to construct [1, 25, 27]. Their associated local approximations are then called polynomial reconstructions.

In [25], the least-squares method for polynomial reconstructions is employed to design a very high-order accurate Finite Volume scheme for convection-diffusion problems. The scheme uses several types of polynomial reconstructions to compute the different types of numerical fluxes, namely boundary or inner fluxes and diffusive or convective ones. In particular, the authors introduced a conservative polynomial reconstruction associated to the boundary elements in order to handle the Dirichlet boundary condition. To achieve that, the polynomial reconstructions representation are rewritten to force the prescribed Dirichlet boundary condition value at the quadrature points for flux integration on the boundary edges. As such, the least-squares procedure corresponds to a simple and straightforward unconstrained minimization problem. To handle Neumann boundary conditions, no polynomial reconstructions are, in fact, required as the corresponding boundary fluxes at the edge quadrature points are assigned with the provided Neumann boundary condition functions. Despite the effectiveness of the method to achieve arbitrary high-order convergence rates under mesh refinement, the scheme was designed for polygonal domain and polygonal mesh that fit each others.

The method introduced in [25] was later extended in [1] for curved domains with prescribed Dirichlet boundary conditions. In [25] the method adapts the polynomial reconstructions associated to the boundary edges to force the boundary condition at the collocation points, successfully restoring the optimal convergence rates. Since the mesh elements are assumed to be polygonal, Neumann and Robin boundary conditions on curved boundaries cannot be assigned to the edge quadrature points without accuracy loss. On the other hand, the procedure developed for the Dirichlet boundary edges can be adapted to the cases of Neumann and Robin boundary conditions. That is, the polynomial reconstructions can be transformed into a different form such that the Neumann or Robin boundary conditions are forced at the collocation points, which means that polynomial  $\varphi(\mathbf{x})$  is not expressed in the form  $\varphi(\mathbf{x}) = \boldsymbol{\eta}^T \mathbf{p}(\mathbf{x})$  but instead in some other way. In practice, such approach turns out to be cumbersome as the transformation

yields very complex polynomial reconstructions. Contrarily, this article proposes a reformulation of the polynomial reconstructions framework and associated least-squares procedure. The new approach improves the versatility of the local approximations without loss of efficiency and can handle genuine general boundary conditions, including the ones of Neumann and Robin types.

The main idea to compute a polynomial reconstruction is to collect a stencil of cells in the vicinity of the current mesh element. The associated pointwise data of the stencil is used to compute the unknown coefficients vector  $\boldsymbol{\eta}$  of the polynomial reconstruction in the least-squares sense. To handle the boundary conditions, some collocation points are defined near the boundary mesh elements, and the associated data (extracted from the prescribed boundary conditions functions) is used to constrain the least-squares method. Notice that, as the boundary conditions are forced via the least-squares procedure to compute coefficients vector  $\boldsymbol{\eta}$ , polynomial  $\varphi(\mathbf{x})$  is always expressed in the form  $\varphi(\mathbf{x}) = \boldsymbol{\eta}^\top \mathbf{p}(\mathbf{x})$  and no cumbersome transformations are needed. In this way, one of the following two types of polynomial reconstructions are employed for the current mesh element depending on its location. An unconstrained polynomial reconstruction is employed when no additional data, apart from the stencil, is provided. A constrained polynomial reconstruction is employed when data extracted from the prescribed boundary conditions (or other condition that has to be satisfied) is provided. Both types of reconstructions are obtained by means of a weighted, possibly constrained, least-squares procedure.

### 3.1. Polynomial representation

Polynomial basis functions are considered in this article to represent the local approximations of the underlying function. A local polynomial approximation of degree  $d > 0$  can be written in a compacted form as

$$\varphi(\mathbf{x}; \mathbf{m}, \boldsymbol{\eta}) = \boldsymbol{\eta}^\top \mathbf{p}_d(\mathbf{x} - \mathbf{m}) = \sum_{\alpha=0}^d \sum_{\beta=0}^{d-\alpha} \eta_{(\alpha,\beta)} (x - m_x)^\alpha (y - m_y)^\beta, \quad (5)$$

where vector  $\mathbf{p}_d(\mathbf{x})$  is a two-dimensional polynomial basis function vector of degree  $d$  (see some examples in Table 2), and  $\mathbf{m} = (m_x, m_y)$  is a reference point used for conditioning purposes. Vector  $\boldsymbol{\eta} = [\eta_{(\alpha,\beta)}]$  gathers the unknown polynomial coefficients to be computed. For the sake of simplicity and when it is unmistakable, the notation  $\varphi(\mathbf{x}) := \varphi(\mathbf{x}; \mathbf{m}, \boldsymbol{\eta})$  is adopted to represent the polynomial.

Table 2: Monomial terms in polynomial basis function vector  $\mathbf{p}_d(\mathbf{x})$ .

	1	$x$	$y$	$x^2$	$xy$	$y^2$	$x^3$	$x^2y$	$xy^2$	$y^3$	$x^4$	$x^3y$	$x^2y^2$	$xy^3$	$y^4$
$\mathbf{p}_1(\mathbf{x})$	✓	✓	✓												
$\mathbf{p}_2(\mathbf{x})$	✓	✓	✓	✓	✓	✓									
$\mathbf{p}_3(\mathbf{x})$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
$\mathbf{p}_4(\mathbf{x})$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

### 3.2. Data, stencils, and weights

To provide an approximation of underlying solution  $\phi(\mathbf{x})$ ,  $N_C$  mean-value approximations over the cells of the mesh are considered, given as

$$\phi_i \approx \frac{1}{|c_i|} \int_{c_i} \phi(\mathbf{x}) \, d\mathbf{x}, \quad i \in C_M, \quad (6)$$

that is,  $\phi_i$  corresponds to an approximation of the mean-value of function  $\phi(\mathbf{x})$  over cell  $c_i$ . All mean-values are gathered in a vector  $\boldsymbol{\Phi} = (\phi_i)_{i \in C_M}$ .

For a given mesh element, a stencil  $\mathcal{S}$  is a collection of  $s$  cell indices generally located in its vicinity.<sup>1</sup> The goal of a stencil is to use the associated data,  $\phi_k$ ,  $k \in \mathcal{S}$ , to compute vector  $\boldsymbol{\eta}$  of the local polynomial reconstruction  $\varphi(\mathbf{x})$  from (5). The number of cell indices in the stencil depends on the polynomial basis function vector degree and, eventually, on the type of the polynomial reconstruction to be computed, as seen later. Generally speaking, a polynomial reconstruction of degree  $d$  requires  $n = (d+1)(d+2)/2$  coefficients and thus a minimum of  $n$  cells has to be collected in stencil  $\mathcal{S}$ . In accordance to the common practice,  $s \approx 1.5n$  is chosen to exceed that minimum

<sup>1</sup>A simple algorithm based on cells proximity can be implemented to collect  $\mathcal{S}$ , but other possibilities can also be considered to deal with complex or particular situations.

giving some freedom to compute the polynomial reconstruction, decreasing conditioning numbers and, consequently, improving its robustness [4].

A positive weight  $\omega_k$  is associated to each cell index  $k \in \mathcal{S}$ , in the stencil. Those weights are gathered in vector  $\boldsymbol{\omega} \in \mathbb{R}^s$ . The weighting procedure is reported in Barth in [26] to compute fluid dynamics solutions with unstructured meshes. The goal is to reduce the influence of data farther from the local reconstruction where neglected high-order terms will have the greatest effect. In this article, weights  $\omega_k := \omega(d_k; \delta, \gamma)$  are computed as the inverse of distance to a power of  $\gamma \in \mathbb{R}$ , given as

$$\omega(d_k; \delta, \gamma) = \frac{1}{(\delta d_k)^\gamma + 1}, \quad k \in \mathcal{S}, \quad (7)$$

where  $\delta \in \mathbb{R}_0^+$  is a sensibility factor and  $d_k = |\mathbf{m}_k - \mathbf{m}|$  is the Euclidean distance between the cell reference point  $\mathbf{m}_k$  (the cell centroid in this article) and the chosen polynomial reconstruction reference point  $\mathbf{m}$ . The values of  $\delta$  and  $\gamma$  will be specified later.

### 3.3. Unconstrained polynomial reconstructions

Unconstrained polynomial reconstructions approximate the underlying solution in the vicinity of any inner edge (see also Table 3). As the name suggests, no constraint is applied and the polynomial reconstruction is straightforwardly computed in the approximated sense, by solving a weighted cost functional from  $\mathbb{R}^n$  to  $\mathbb{R}$  given as

$$\widetilde{\mathcal{F}}(\boldsymbol{\eta}; \mathcal{S}, \boldsymbol{\Phi}, \boldsymbol{\omega}) = \sum_{k \in \mathcal{S}} \omega_k \left[ \frac{1}{|c_k|} \int_{c_k} \varphi(\mathbf{x}) \, d\mathbf{x} - \phi_k \right]^2, \quad (8)$$

also denoted as  $\widetilde{\mathcal{F}}(\boldsymbol{\eta}) := \widetilde{\mathcal{F}}(\boldsymbol{\eta}; \mathcal{S}, \boldsymbol{\Phi}, \boldsymbol{\omega})$  for the sake of simplicity.

The approximation procedure consists in seeking the unique vector  $\widetilde{\boldsymbol{\eta}} \in \mathbb{R}^n$  that minimizes  $\widetilde{\mathcal{F}}(\boldsymbol{\eta})$  in the least-squares sense, that is

$$\widetilde{\boldsymbol{\eta}} = \arg \min_{\boldsymbol{\eta}} [\widetilde{\mathcal{F}}(\boldsymbol{\eta})]. \quad (9)$$

An overdetermined system of  $s$  linear equations to solve for  $n$  unknowns arises, which can be written in the matrix form as  $\mathbf{W}\mathbf{A}\boldsymbol{\eta} = \mathbf{W}\mathbf{b}$ . Matrix  $\mathbf{W} \in \mathbb{R}^{s \times s}$  is the diagonal matrix  $\mathbf{W} = \text{diag}(\boldsymbol{\omega})$  and vector  $\mathbf{b} \in \mathbb{R}^s$  is obtained from the mean-values  $\boldsymbol{\Phi}$  with index in  $\mathcal{S}$ . Matrix  $\mathbf{A} \in \mathbb{R}^{s \times n}$  is composed of entries

$$a_{k,l} = \frac{1}{|c_k|} \int_{c_k} (x - m_x)^\alpha (y - m_y)^\beta \, d\mathbf{x}, \quad k = 1, \dots, s, \quad l = 1, \dots, n, \quad (10)$$

where for each index  $l$  a pair  $(\alpha, \beta)$ ,  $\alpha \in \{1, \dots, d\}$ ,  $\beta \in \{1, \dots, d - \alpha\}$ , is associated for a polynomial reconstruction of degree  $d$ . Assuming a local indexation of the stencil  $\mathcal{S} = \{1, 2, \dots, s\}$  and assuming that vector  $\boldsymbol{\eta}$  writes as  $\boldsymbol{\eta} = [\eta_1 \ \eta_2 \ \dots \ \eta_n]^\top$ , the system of linear equations in matrix form then writes

$$\underbrace{\begin{bmatrix} \omega_1 & 0 & \cdots & 0 \\ 0 & \omega_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \omega_s \end{bmatrix}}_{\mathbf{W}} \underbrace{\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{s,1} & a_{s,2} & \cdots & a_{s,n} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_n \end{bmatrix}}_{\boldsymbol{\eta}} = \underbrace{\begin{bmatrix} \omega_1 & 0 & \cdots & 0 \\ 0 & \omega_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \omega_s \end{bmatrix}}_{\mathbf{W}} \underbrace{\begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_s \end{bmatrix}}_{\mathbf{b}}. \quad (11)$$

Finally, the solution of the unconstrained least-squares problem (9) provides vector  $\widetilde{\boldsymbol{\eta}}$  that minimizes the cost functional (8). The associated unconstrained polynomial reconstruction is denoted by  $\widetilde{\varphi}(\mathbf{x}) = \widetilde{\boldsymbol{\eta}}^\top \mathbf{p}_d(\mathbf{x} - \mathbf{m})$ .

Several minimization procedures can be derived to found the least-squares solution. In this article, the normal equations method is applied.

Following this procedure, unconstrained polynomial reconstruction for inner edge  $e_{ij}$ ,  $i \in C_M$ ,  $j \in N_i$ , is computed and denoted as  $\varphi_{ij}(\mathbf{x}) = \boldsymbol{\eta}_{ij}^\top \mathbf{p}_d(\mathbf{x} - \mathbf{m}_{ij})$ . The reference point corresponds to the edge midpoint,  $\mathbf{m} := \mathbf{m}_{ij}$ , and the stencil,  $\mathcal{S} := \mathcal{S}_{ij}$ , gathers  $s$  cells in the vicinity of the edge. The solution of the unconstrained least-squares problem associated to the polynomial reconstructions for the inner edges, as given in (9), provides vector  $\widetilde{\boldsymbol{\eta}}_{ij}$  that minimizes cost functional (8). The associated unconstrained polynomial reconstruction is denoted as  $\widetilde{\varphi}_{ij}(\mathbf{x}) = \widetilde{\boldsymbol{\eta}}_{ij}^\top \mathbf{p}_d(\mathbf{x} - \mathbf{m}_{ij})$ .

### 3.4. Constrained polynomial reconstructions

Constrained polynomial reconstructions are computed to approximate the underlying function in the vicinity of the boundary cells and edges (see Table 3). This situations arise when it is necessary to include local boundary conditions, conservation properties, etc., which are easily handled in an approximated sense. As the name suggests, in addition to the linear equations to be approximated in the least-squares sense,  $p$  linear constraints, where  $0 < p < n$ , with respect to vector  $\boldsymbol{\eta}$  must be fulfilled. For this purpose, a constraint functional from  $\mathbb{R}^n$  to  $\mathbb{R}^p$  is introduced and denoted as  $\widehat{\mathcal{G}}(\boldsymbol{\eta})$ . Mimicking the unconstrained polynomial reconstruction case, the same weighted cost functional from  $\mathbb{R}^n$  to  $\mathbb{R}$  is considered

$$\widehat{\mathcal{F}}(\boldsymbol{\eta}; \mathcal{S}, \boldsymbol{\Phi}, \omega) = \sum_{k \in \mathcal{S}} \omega_k \left[ \frac{1}{|c_k|} \int_{c_k} \varphi(\mathbf{x}) \, d\mathbf{x} - \phi_k \right]^2. \quad (12)$$

also denoted as  $\widehat{\mathcal{F}}(\boldsymbol{\eta}) := \widehat{\mathcal{F}}(\boldsymbol{\eta}; \mathcal{S}, \boldsymbol{\Phi}, \omega)$  for the sake of simplicity.

The approximation procedure consists in seeking unique vector  $\widehat{\boldsymbol{\eta}} \in \mathbb{R}^n$  that minimizes  $\widehat{\mathcal{F}}(\boldsymbol{\eta})$  in the least-squares sense and, at the same time, exactly satisfies equation  $\widehat{\mathcal{G}}(\boldsymbol{\eta}) = 0$ , that is

$$\widehat{\boldsymbol{\eta}} = \arg \min_{\boldsymbol{\eta}} \left[ \widehat{\mathcal{F}}(\boldsymbol{\eta}) \right] \quad \text{subject to} \quad \widehat{\mathcal{G}}(\boldsymbol{\eta}) = 0. \quad (13)$$

The least-squares method yields an overdetermined system of  $s$  linear equations for  $n$  unknowns, which again can be written in the matrix form as  $\mathbf{W}\mathbf{A}\boldsymbol{\eta} = \mathbf{W}\mathbf{b}$ . Matrices  $\mathbf{W} \in \mathbb{R}^{s \times s}$  and  $\mathbf{A} \in \mathbb{R}^{s \times n}$ , and vector  $\mathbf{b} \in \mathbb{R}^s$  are given as for the unconstrained polynomial reconstruction in Equation (11). Moreover, the solution is obtained applying the linear constraints in the matrix form  $\mathbf{C}\boldsymbol{\eta} = \mathbf{d}$  with  $p$  equations for  $n$  unknowns. Matrix  $\mathbf{C} \in \mathbb{R}^{p \times n}$  gathers the coefficients of the linear constraints, assumed to be full rank, while vector  $\mathbf{d} \in \mathbb{R}^p$  is the right-hand side.

Finally, the solution of the constrained least-squares problem (13), provides vector  $\widehat{\boldsymbol{\eta}}$  that minimizes the cost functional (12) and exactly satisfies the linear constraints. The associated constrained polynomial reconstruction writes as  $\widehat{\varphi}(\mathbf{x}) = \widehat{\boldsymbol{\eta}}^T \mathbf{p}_d(\mathbf{x} - \mathbf{m})$ .

Several minimization procedures fulfilling linear constraints can be found in the literature, and this article considers the so-called linearly constrained Lagrange Multipliers method. This topic is not elaborated and the reader is referred to [28] for more details.

#### 3.4.1. Constrained polynomial reconstructions for the cells

Constrained polynomial reconstruction for cell  $c_i$ ,  $i \in C_M$ , is computed and denoted as  $\varphi_i(\mathbf{x}) = \boldsymbol{\eta}_i^T \mathbf{p}(\mathbf{x} - \mathbf{m}_i)$ . The unknown polynomial coefficients vector is  $\boldsymbol{\eta} := \boldsymbol{\eta}_i$ , the reference point  $\mathbf{m} = \mathbf{m}_i$  is chosen to be the cell centroid, and stencil  $\mathcal{S} = \mathcal{S}_i$  gathers  $s$  cells in the vicinity of the cell (excluding  $c_i$ ). The corresponding mean-value,  $\phi_i$  is an approximation of function  $\phi(\mathbf{x})$  over cell  $c_i$ , and to enforce the conservation of  $\phi_i$  in polynomial  $\varphi_i(\mathbf{x})$ , the constraint functional  $\widehat{\mathcal{G}}(\boldsymbol{\eta}) := \widehat{\mathcal{G}}_i(\boldsymbol{\eta}_i)$  must have the form

$$\widehat{\mathcal{G}}_i(\boldsymbol{\eta}) = \frac{1}{|c_i|} \int_{c_i} \varphi_i(\mathbf{x}) \, d\mathbf{x} - \phi_i = 0. \quad (14)$$

Assuming that vector  $\boldsymbol{\eta}$  writes as  $\boldsymbol{\eta} = [\eta_1 \ \eta_2 \ \dots \ \eta_n]^T$ , the linear constraint (14) in matrix form writes

$$\underbrace{\begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} \eta_1 & \eta_2 & \dots & \eta_n \end{bmatrix}^T}_{\boldsymbol{\eta}} = \underbrace{\begin{bmatrix} \phi_i \end{bmatrix}}_{\mathbf{d}}, \quad (15)$$

with the coefficients  $c_l$  given as

$$c_l = \frac{1}{|c_i|} \int_{c_i} (x - m_{i,x})^\alpha (y - m_{i,y})^\beta \, d\mathbf{x}, \quad l = 1, \dots, n, \quad (16)$$

where for each index  $l$  a pair  $(\alpha, \beta)$ ,  $\alpha \in \{1, \dots, d\}$ ,  $\beta \in \{1, \dots, d - \alpha\}$ , is associated for a polynomial reconstruction of degree  $d$ .

The solution of the constrained least-squares problem (13) associated to the polynomial reconstructions in the cells, provides the vector  $\widehat{\boldsymbol{\eta}}_i$  that minimizes the cost functional (12) and exactly satisfies linear constraint (14). The associated constrained polynomial reconstruction writes as  $\widehat{\varphi}_i(\mathbf{x}) = \widehat{\boldsymbol{\eta}}_i^T \mathbf{p}_d(\mathbf{x} - \mathbf{m}_i)$ .

### 3.4.2. Constrained polynomial reconstructions for the boundary edges

Constrained polynomial reconstruction for boundary edge  $e_{iB}$ ,  $i \in \mathcal{B}_{M,B}$ , on computational boundary  $\Gamma_{\Delta,B}$ ,  $B \in \{D, N, R\}$ , with prescribed Dirichlet, Neumann, or Robin boundary conditions is computed and denoted as  $\varphi_{iB}(\mathbf{x}) = \boldsymbol{\eta}_{iB}^T \mathbf{p}(\mathbf{x} - \mathbf{m}_{iB})$ . The unknown polynomial coefficients vector is  $\boldsymbol{\eta} := \boldsymbol{\eta}_{iB}$ , the reference point  $\mathbf{m} := \mathbf{m}_{iD}$  is chosen to be the edge midpoint, and stencil  $\mathcal{S} := \mathcal{S}_{iD}$  gathers  $s$  cells in the vicinity of the edge. Assume that  $\mathbf{p}_{iB} = (p_{iB,x}, p_{iB,y})$  is a point in the vicinity of edge  $e_{iB}$  with an associated outward unit normal vector  $\mathbf{v}_{iB} = (v_{iB,x}, v_{iB,y})$ , to be assigned later.

Constraint functional  $\widehat{\mathcal{G}}(\boldsymbol{\eta}_{iB})$  is defined according to the prescribed boundary condition at boundary partition  $\Gamma_B$ , as follows:

- In case edge  $e_{iB}$  is on boundary partition  $\Gamma_D$  (index  $B$  is replaced by D), to enforce the Dirichlet boundary condition  $\phi(\mathbf{p}_{iD}) = g_D(\mathbf{p}_{iD})$  within polynomial  $\varphi_{iD}(\mathbf{x})$ , constraint functional  $\widehat{\mathcal{G}}_{iB}(\boldsymbol{\eta}) := \widehat{\mathcal{G}}_{iD}(\boldsymbol{\eta}_{iD})$  must have the form

$$\widehat{\mathcal{G}}_{iD}(\boldsymbol{\eta}_{iD}) = \varphi_{iD}(\mathbf{p}_{iD}) - g_{D,iD} = 0, \quad (17)$$

where the boundary condition function value is defined as  $g_{D,iD} := g_D(\mathbf{p}_{iD})$ ;

- In case edge  $e_{iB}$  is on boundary partition  $\Gamma_N$  (index  $B$  is replaced by N), to enforce the Neumann boundary condition  $-\kappa(\mathbf{p}_{iN})\nabla\phi(\mathbf{p}_{iN}) \cdot \mathbf{v}_{iN} = g_N(\mathbf{p}_{iN})$  within polynomial  $\varphi_{iN}(\mathbf{x})$ , the constraint functional  $\widehat{\mathcal{G}}_{iB}(\boldsymbol{\eta}) := \widehat{\mathcal{G}}_{iN}(\boldsymbol{\eta}_{iN})$  must have the form

$$\widehat{\mathcal{G}}_{iN}(\boldsymbol{\eta}_{iN}) = -\kappa_{iN}\nabla\varphi_{iN}(\mathbf{p}_{iN}) \cdot \mathbf{v}_{iN} - g_{N,iN} = 0, \quad (18)$$

where the coefficient is defined as  $\kappa_{iN} := \kappa(\mathbf{p}_{iN})$  and the boundary condition function value is defined as  $g_{N,iN} := g_N(\mathbf{p}_{iN})$ ;

- In case edge  $e_{iB}$  is on boundary partition  $\Gamma_R$  (index  $B$  is replaced by R), to enforce the Robin boundary condition  $\sigma(\mathbf{p}_{iR})\phi(\mathbf{p}_{iR}) + \rho(\mathbf{p}_{iR})\nabla\phi(\mathbf{p}_{iR}) \cdot \mathbf{v}_{iR} = g_R(\mathbf{p}_{iR})$  within polynomial  $\varphi_{iR}(\mathbf{x})$ , the constraint functional  $\widehat{\mathcal{G}}_{iB}(\boldsymbol{\eta}) := \widehat{\mathcal{G}}_{iR}(\boldsymbol{\eta}_{iR})$  must have the form

$$\widehat{\mathcal{G}}_{iR}(\boldsymbol{\eta}_{iR}) = \sigma_{iR}\varphi_{iR}(\mathbf{p}_{iR}) + \rho_{iR}\nabla\varphi_{iR}(\mathbf{p}_{iR}) \cdot \mathbf{v}_{iR} - g_{R,iR} = 0, \quad (19)$$

where the coefficients are defined as  $\sigma_{iR} := \sigma(\mathbf{p}_{iR})$  and  $\rho_{iR} := \rho(\mathbf{p}_{iR})$  and the boundary condition function value is defined as  $g_{R,iR} := g_R(\mathbf{p}_{iR})$ .

Assuming that vector  $\boldsymbol{\eta}$  writes as  $\boldsymbol{\eta} = [\eta_1 \ \eta_2 \ \dots \ \eta_n]^T$ , the linear constraints (17-19) in matrix form writes

$$\underbrace{\begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix}}_C \underbrace{\begin{bmatrix} \eta_1 & \eta_2 & \dots & \eta_n \end{bmatrix}^T}_\boldsymbol{\eta} = \underbrace{\begin{bmatrix} g_{B,iB} \end{bmatrix}}_d, \quad (20)$$

with the coefficients  $c_l$ ,  $l = 1, \dots, n$ , for the Dirichlet, Neumann, and Robin cases given respectively as

$$c_l = (p_{iD,x} - m_{iD,y})^\alpha (p_{iD,y} - m_{iD,y})^\beta, \quad (21)$$

$$c_l = -\kappa_{iN} \begin{bmatrix} \alpha(p_{iN,x} - m_{iN,x})^{\alpha-1} (p_{iN,y} - m_{iN,y})^\beta \\ \beta(p_{iN,x} - m_{iN,x})^\alpha (p_{iN,y} - m_{iN,y})^{\beta-1} \end{bmatrix} \cdot \mathbf{v}_{iN}, \quad (22)$$

$$c_l = \sigma_{iR} (p_{iR,x} - m_{iR,y})^\alpha (p_{iR,y} - m_{iR,y})^\beta + \rho_{iR} \begin{bmatrix} \alpha(p_{iR,x} - m_{iR,x})^{\alpha-1} (p_{iR,y} - m_{iR,y})^\beta \\ \beta(p_{iR,x} - m_{iR,x})^\alpha (p_{iR,y} - m_{iR,y})^{\beta-1} \end{bmatrix} \cdot \mathbf{v}_{iR}, \quad (23)$$

where for each index  $l$  a pair  $(\alpha, \beta)$ ,  $\alpha \in \{1, \dots, d\}$ ,  $\beta \in \{1, \dots, d - \alpha\}$ , is associated for a polynomial reconstruction of degree  $d$ .

The solution of the constrained least-squares problem (13) associated to the polynomial reconstructions for the boundary edges, provides the vector  $\widehat{\boldsymbol{\eta}}_{iB}$  that minimizes the cost functional (12) and exactly satisfies the linear constraint (17), (18), or (19) for a Dirichlet, Neumann, or Robin boundary condition, respectively. The associated constrained polynomial reconstruction writes as  $\widehat{\varphi}_{iB}(\mathbf{x}) = \widehat{\boldsymbol{\eta}}_{iB}^T \mathbf{p}_d(\mathbf{x} - \mathbf{m}_{iB})$ .

Table 3 summarizes the different polynomial reconstructions and their associated constraints (if applicable) considered in this article.

Table 3: Edge and cell based polynomial reconstructions under constraints.

<i>Mesh Element</i>	<i>Polynomial</i>	<i>Reference Point</i>	<i>Type</i>	<i>Least-squares constraint</i>
Inner edge $e_{ij}, i \in C_M, j \in N_i$	$\tilde{\varphi}_{ij}(\mathbf{x})$	$\mathbf{m}_{ij}$	Unconstrained	—
Cell $c_i, i \in C_M$	$\hat{\varphi}_i(\mathbf{x})$	$\mathbf{m}_i$	Constrained (Mean-value)	(14)
Dirichlet edge $e_{iD}, i \in \mathcal{B}_{M,D}$	$\hat{\varphi}_{iD}(\mathbf{x})$	$\mathbf{m}_{iD}$	Constrained (Dirichlet)	(17)
Neumann edge $e_{iN}, i \in \mathcal{B}_{M,N}$	$\hat{\varphi}_{iN}(\mathbf{x})$	$\mathbf{m}_{iN}$	Constrained (Neumann)	(18)
Robin edge $e_{iR}, i \in \mathcal{B}_{M,R}$	$\hat{\varphi}_{iR}(\mathbf{x})$	$\mathbf{m}_{iR}$	Constrained (Robin)	(19)

#### 4. Curved boundary treatment

The discretization of regular curved boundaries by polygons yields second-order accurate geometrical approximations [6]. Therefore, if no specific treatment is used to accurately handle the boundary conditions, the scheme converges with a maximal second-order of accuracy. In this regard, developing techniques dedicated to boundary conditions which are prescribed on curved boundaries is of paramount importance to achieve arbitrary high-order accurate convergence rates. The classical technique is based on the Isoparametric Elements method [2, 3], but similar techniques have been specifically designed for the Finite Volume method. For example, the seminal paper of Ollivier-Gooch et al. [4] introduces a technique based on the constrained least-squares reconstruction method for curved cells fitting the physical boundary. The proposed method enforces the local polynomial reconstructions associated to the boundary mesh elements to satisfy the boundary conditions at the collocation points. Such points are extracted from the physical boundary [29, 30, 31] and have to correspond to quadrature points  $\mathbf{q}_{iB,r}$ ,  $r = 1, \dots, R$ , on the mesh elements of the boundary which are used to numerically integrate the fluxes. Consequently, the boundary mesh elements have to be curved in order to fit the physical boundary (see Figure 3), which requires complex and computational intensive meshing algorithms [6]. Moreover, extracting collocation/quadrature points  $\mathbf{q}_{iB,r}$ ,  $r = 1, \dots, R$ , and associated outward unit normal vectors  $\mathbf{n}_{iB,r}$ ,  $r = 1, \dots, R$ , is not trivial for the general case [27, 29, 30, 31]. Contrary to the Isoparametric Elements method, the Ollivier-Gooch technique does not require nonlinear transformations but the principal shortcoming remains: the meshing algorithm to generate curved mesh elements fitting the boundary, and, in addition the associated high-order accurate quadrature rules for integration applied on those nonpolygonal mesh elements. Handling complex two- or three-dimensional curved elements turns to be a cumbersome task that may result into significant computational costs [5, 6, 7].

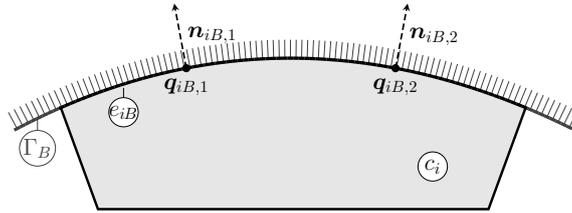


Figure 3: Representation of physical boundary  $\Gamma$  and fitting curved boundary edge  $e_{iB}$  with quadrature points  $\mathbf{q}_{iB,1}$  and  $\mathbf{q}_{iB,2}$  (eventually less or more depending on the order of the quadrature rule used) and associated outward unit normal vectors  $\mathbf{n}_{iB,1}$  and  $\mathbf{n}_{iB,2}$ .

To avoid the use of curved mesh elements, a new technique proposed in [1] deals solely with polygonal representative approximations of the curved boundaries. As introduced in Section 3, for given boundary edge  $e_{iB}$ ,  $i \in \mathcal{B}_{M,B}$ ,  $B \in \{D, N, R\}$ , the prescribed boundary condition is evaluated at the collocation point  $\mathbf{p}_{iB}$  with outward unit normal vector  $\mathbf{v}_{iB}$ . Then, a linear constraint is defined at point  $\mathbf{p}_{iB}$  that is embedded in the least-squares method that computes the polynomial reconstruction  $\hat{\varphi}_{iB}(\mathbf{x})$  associated to edge  $e_{iB}$ . For the Dirichlet, Neumann, and Robin boundary edges, linear constraint  $\hat{\mathcal{G}}(\boldsymbol{\eta}) = 0$  is defined in Equations (17), (18), and (19), respectively. Notice that only one collocation point per boundary edge is required and no exact location for  $\mathbf{p}_{iB}$  was yet defined but, as stated before, it will be sought in the vicinity of the corresponding boundary edge. In the proposed method,  $\mathbf{q}_{iB,r}$ ,  $r = 1, \dots, R$ , are the quadrature points on edge  $e_{iB}$  with outward unit normal vector  $\mathbf{n}_{iB}$ . Moreover, the key-point to handle curved boundaries with

polygonal mesh elements is to carefully determine the appropriate constraints, such that the polynomial reconstructions satisfy the boundary conditions and provides an arbitrary high-order accurate approximation of the underlying solution. The following subsections introduce several of such strategies.

#### 4.1. Naive method

A first and straightforward approach consists in constraining the least-squares methods with the boundary condition on the computational boundary represented by edge  $e_{iB}$  (see Figure 4). That is, one can choose in Equations (17), (18), and (19) the edge midpoint as the collocation point,  $\mathbf{p}_{iB} := \mathbf{m}_{iB}$ , and for Neumann and Robin boundary conditions, one can use the edge outward unit normal vector,  $\mathbf{v}_{iB} := \mathbf{n}_{iB}$ , and define coefficients  $\kappa_{iB} := \kappa(\mathbf{m}_{iB})$ ,  $\sigma_{iB} := \sigma(\mathbf{m}_{iB})$ , and  $\rho_{iB} := \rho(\mathbf{m}_{iB})$ . The right-hand side values in Equations (17), (18), and (19) are defined as  $g_{D,iD} := g_D(\mathbf{m}_{iD})$ ,  $g_{N,iN} := g_N(\mathbf{m}_{iN})$ , and  $g_{R,iR} := g_R(\mathbf{m}_{iR})$ , respectively. Such approach provides a maximal second-order convergence rate since collocation point  $\mathbf{m}_{iB}$  and outward unit normal vector  $\mathbf{n}_{iB}$  only represent second-order approximations with respect to physical boundary  $\Gamma$ . Notice that, in this approach, collocation point  $\mathbf{m}_{iB}$  does not belong to boundary  $\Gamma$ , hence an extension of function  $g_B(\mathbf{x})$ ,  $B \in \{D, N, R\}$ , in the neighborhood of the edge is required to guarantee that the associated boundary condition makes sense. This approach can assure an arbitrary high-order of accuracy only in the case the computational domain exactly fits the physical one, in other words when the physical domain is polygonal.

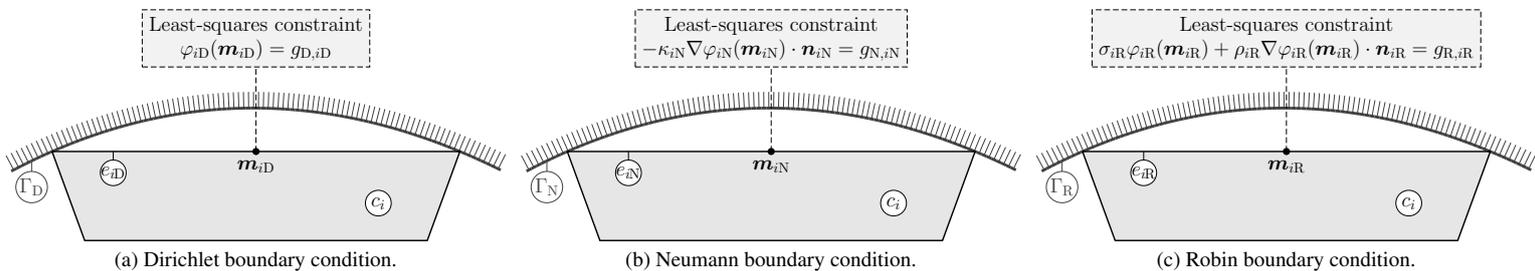


Figure 4: Representation of the least-squares constraints and the collocation points for the Naive method from Section 4.1.

#### 4.2. ROD – Reconstruction for Off-site Data method

The *Reconstruction for Off-site Data* method (shortened to ROD method) was introduced in [1] to restore the very high-order convergence rates for Dirichlet boundary conditions on curved domains. In this work, the method is extended to more general boundary conditions such as the Neumann and Robin ones. The basic principle is based on an evaluation of the boundary condition with collocation points on the physical boundary while the flux computation is carried out on the computational boundary. The term *Off-site Data* reminds the difference between the data location (boundary condition on physical boundary  $\Gamma$ ) and its use (flux computation on computational boundary  $\Gamma_\Delta$ ). Therefore all computations are performed on the polygonal edges but taking into account the information located on the physical boundary using, for that purpose, the previously introduced constrained polynomial reconstructions for the boundary edges. The main advantages of the ROD method are the following ones:

- Numerical integration of the fluxes and source terms are only carried out on the polygonal computational domain and not on the curved physical domain;
- No curved mesh element is required;
- No geometrical transformation is required;
- No quadrature points for integration on the physical boundary need to be determined;
- The method is independent of the spacial dimension and the shape of the mesh elements.

To recover the optimal order of convergence and accuracy, the ROD method constraints the least-squares methods by the boundary condition at the true physical boundary (see Figures 5 and 6 and compare with Figures 3 and 4) instead of the computational boundary, as in the Naive method. That is, one can choose in Equations (17), (18), and (19) the collocation points as points extracted from the physical boundary  $\Gamma$ ,  $\mathbf{p}_{iB} := \mathbf{b}_{iB}$ , and for Neumann and Robin boundary conditions, one can use the outward unit normal vector to the physical boundary at the collocation point,  $\mathbf{v}_{iB} := \mathbf{r}_{iB}$ , and define coefficients  $\kappa_{iB} := \kappa(\mathbf{b}_{iB})$ ,  $\sigma_{iB} := \sigma(\mathbf{b}_{iB})$ , and  $\rho_{iB} := \rho(\mathbf{b}_{iB})$ . The right-hand side values in Equations (17), (18), and (19) are  $g_{D,iD} := g_D(\mathbf{b}_{iD})$ ,  $g_{N,iN} := g_N(\mathbf{b}_{iN})$ , and  $g_{R,iR} := g_R(\mathbf{b}_{iR})$ , respectively. In practice,

for the sake of simplicity, the collocation point  $\mathbf{b}_{iB}$  can be the orthogonal projection of edge midpoint  $\mathbf{m}_{iB}$  onto the associated physical boundary  $\Gamma$  (as illustrated in Figure 5). Notice that quadrature points  $\mathbf{q}_{iB,r}$ ,  $r = 1, \dots, R$ , do belong to the straight edge of the boundary element where all the fluxes are computed, as will be presented in Section 5.1. This last point definitively distinguishes the ROD method from the one proposed by Ollivier-Gooch et. al [4], where in the latter the authors propose to use the same points (quadrature points on the curved edges) both for the boundary condition collocation and for the flux integration.

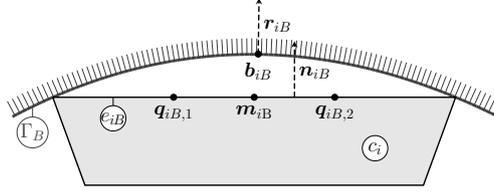


Figure 5: Representation of physical (curved) boundary  $\Gamma$  and computational (polygonal) boundary edge  $e_{iB}$  with quadrature points  $\mathbf{q}_{iB,1}$  and  $\mathbf{q}_{iB,2}$  (eventually less or more depending on the quadrature rule used) and the associated outward unit normal vector  $\mathbf{n}_{iB}$ . On the physical boundary are represented collocation point  $\mathbf{b}_{iB}$  and associated outward unit normal vector  $\mathbf{r}_{iB}$ .

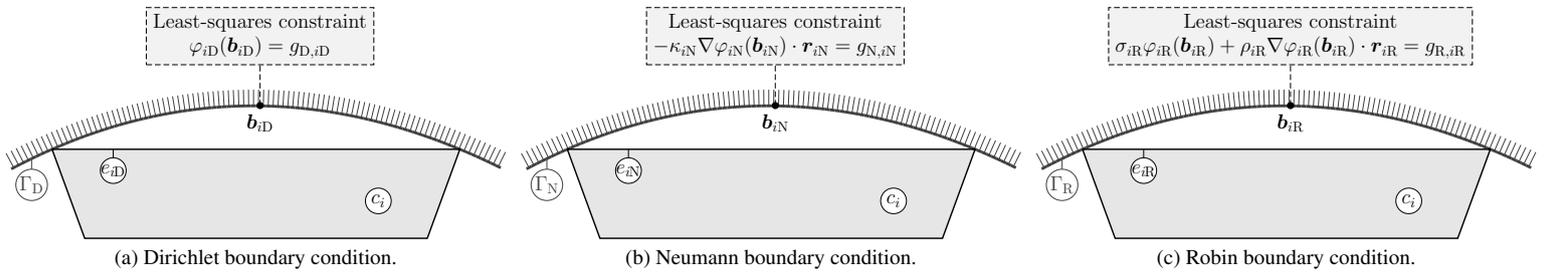


Figure 6: Representation of the least-squares constraints and the collocation points for the ROD method from Section 4.2.

### 4.3. AROD – Adaptive Reconstruction for Off-site Data method

For the ROD method implies the entries of the linear constraints in Equations (17), (18), and (19) explicitly depend, by construction, on the physical boundary location via collocation points  $\mathbf{b}_{iB}$ . An alternative method, called *Adaptive Reconstruction for Off-site Data* method (shortened to AROD method), is introduced to treat boundary conditions on curved boundaries. Contrarily to the ROD method, in the AROD method the physical boundary location is decoupled from the least-squares procedure.

**Remark 2.** *One of the advantages of the AROD method over the ROD method lies in the treatment of unsteady and time-dependent problems with moving boundaries/interfaces or tracking boundaries/interfaces problems. In these situations, the displacement of the physical boundary or interface can deteriorate the accuracy of the previously computed polynomial reconstructions associated to the boundary edges and, therefore, limit the convergence orders under mesh refinement. As instance, if the boundary is parameterized by time,  $\Gamma(t)$ , then collocation points  $\mathbf{b}_{iB}(t_1)$ , used in the ROD method to impose the boundary conditions at time-step  $t = t_1$ , do not necessarily exactly represent the physical boundary at time-step  $t = t_2$ . Therefore, if the ROD method is applied at time-step  $t = t_1$  and if, hypothetically,  $\mathbf{b}_{iB}(t_2) - \mathbf{b}_{iB}(t_1) = \mathcal{O}(h^2)$ , then the convergence order at time-step  $t = t_2$  is limited to a maximal second-order. In fact, as the performed least-squares procedure depends on the physical boundary, collocation points  $\mathbf{b}_{iB}(t_1)$  do not represent exactly the boundary condition at the current time-step, unless this procedure is recomputed for the new collocation points,  $\mathbf{b}_{iB}(t_2)$ . The recomputation of the least-squares procedure is, therefore, required to recover the accuracy and the convergence order in the ROD method. On the other hand, the AROD method can avoid this cost for small perturbations of the physical boundary.*

The three main ingredients are:

- A constrained polynomial reconstruction,  $\varphi_{iB}(\mathbf{x})$ , for boundary edge  $e_{iB}$ ,  $i \in \mathcal{B}_{M,B}$ ,  $B \in \{D, N, R\}$ , with linear constraints as given in Equations (17-19), depending on the type of the boundary condition (Dirichlet, Neumann, or Robin);

- A collocation point,  $\mathbf{b}_{iB} \in \Gamma_B$ , where the fitting condition is prescribed, and the associated outward unit normal vector,  $\mathbf{r}_{iB}$  (for the cases of Neumann and Robin boundary conditions); the coefficients in Equations (17-19) are defined as  $\kappa_{iB} := \kappa(\mathbf{b}_{iB})$ ,  $\sigma_{iB} := \sigma(\mathbf{b}_{iB})$ , and  $\rho_{iB} := \rho(\mathbf{b}_{iB})$ , and the boundary condition functions values are defined as  $g_{D,iD} := g_D(\mathbf{b}_{iD})$ ,  $g_{N,iN} := g_N(\mathbf{b}_{iN})$ , and  $g_{R,iR} := g_R(\mathbf{b}_{iR})$ , respectively;
- A real free-parameter,  $g$ , associated to edge midpoint  $\mathbf{m}_{iB}$ , where the least-squares constraint is imposed, and the associated outward unit normal vector,  $\mathbf{n}_{iB}$  (for the cases of Neumann and Robin boundary conditions); the coefficients in Equations (17-19) are defined as  $\kappa_{iB} := \kappa(\mathbf{m}_{iB})$ ,  $\sigma_{iB} := \sigma(\mathbf{m}_{iB})$ , and  $\rho_{iB} := \rho(\mathbf{m}_{iB})$ , and the boundary condition functions values are defined as  $g_{D,iD} := g_D(\mathbf{m}_{iD})$ ,  $g_{N,iN} := g_N(\mathbf{m}_{iN})$ , and  $g_{R,iR} := g_R(\mathbf{m}_{iR})$ , respectively.

The term *fitting condition* refers to the condition imposed at collocation point  $\mathbf{b}_{iB}$  while, on the other hand, term *least-squares constraint* refers to the constraint imposed at edge midpoint  $\mathbf{m}_{iB}$  (see Figure 7). Both conditions are of the same type as the boundary condition type associated to the edge but, while the fitting condition takes the value of the boundary condition function, the least-squares constraint takes the value of free-parameter  $g$ . The basic idea of AROD method, described as follows, is to constrain the least-squares method with the least-squares constraint (which does not exactly satisfy the boundary condition at the physical boundary) and then to compute free-parameter  $g$  to satisfy the fitting condition.

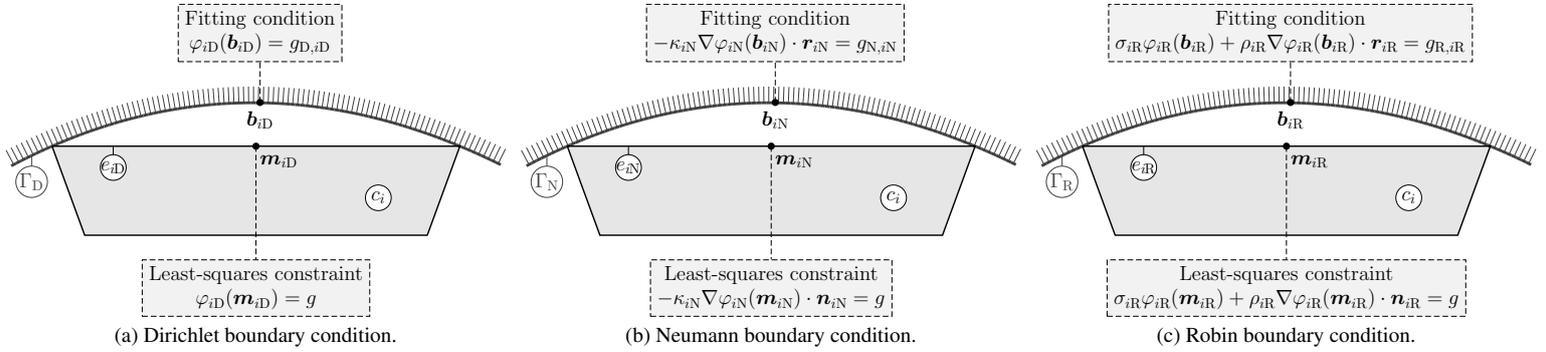


Figure 7: Representation of the least-squares constraints and the collocation points for the AROD method from Section 4.3.

For a given edge  $e_{iB}$ ,  $i \in \mathcal{B}_{M,B}$ , on boundary partition  $\Gamma_B$ ,  $B \in \{D, N, R\}$ , polynomial  $\varphi_{iB}(\mathbf{x})$  is computed solving the least-squares problem (13) with the linear constraint  $\widehat{\mathcal{G}}_{iB}(\boldsymbol{\eta}_{iB})$ , which is defined according to the prescribed boundary condition at boundary partition  $\Gamma_B$ , as follows:

- In case edge  $e_{iB}$  is on boundary partition  $\Gamma_D$  (index  $B$  is replaced by  $D$ ), condition  $\phi(\mathbf{m}_{iD}) = g$  is enforced within polynomial  $\varphi_{iD}(\mathbf{x})$  and constraint functional  $\widehat{\mathcal{G}}_{iB}(\boldsymbol{\eta}) := \widehat{\mathcal{G}}_{iD}(\boldsymbol{\eta}_{iD})$  must have the form

$$\widehat{\mathcal{G}}_{iD}(\boldsymbol{\eta}_{iD}) = \varphi_{iD}(\mathbf{m}_{iD}) - g, \quad (24)$$

which corresponds to Equation (17) where collocation point  $\mathbf{p}_{iD} := \mathbf{m}_{iD}$  corresponds to the edge midpoint, and the boundary condition function value is replaced by free-parameter  $g$  (see Figure 7(a));

- In case edge  $e_{iB}$  is on boundary partition  $\Gamma_N$  (index  $B$  is replaced by  $N$ ), condition  $-\kappa(\mathbf{m}_{iN})\nabla\phi(\mathbf{m}_{iN}) \cdot \mathbf{n}_{iN} = g_N(\mathbf{m}_{iN})$  is enforced within polynomial  $\varphi_{iN}(\mathbf{x})$  and constraint functional  $\widehat{\mathcal{G}}_{iB}(\boldsymbol{\eta}) := \widehat{\mathcal{G}}_{iN}(\boldsymbol{\eta}_{iN})$  must have the form

$$\widehat{\mathcal{G}}_{iN}(\boldsymbol{\eta}_{iN}) = -\kappa_{iN} \nabla\varphi_{iN}(\mathbf{m}_{iN}) \cdot \mathbf{n}_{iN} - g, \quad (25)$$

which corresponds to Equation (18) where collocation point  $\mathbf{p}_{iN} := \mathbf{m}_{iN}$  corresponds to the edge midpoint, vector  $\mathbf{v}_{iN} := \mathbf{n}_{iN}$  corresponds to the edge outward unit normal vector, the coefficient is defined as  $\kappa_{iN} := \kappa(\mathbf{m}_{iN})$ , and the boundary condition function value is replaced by free-parameter  $g$  (see Figure 7(b));

- In case edge  $e_{iB}$  is on boundary partition  $\Gamma_R$  (index  $B$  is replaced by  $R$ ), condition  $\sigma(\mathbf{m}_{iR})\phi(\mathbf{m}_{iR}) + \rho(\mathbf{m}_{iR})\nabla\phi(\mathbf{m}_{iR}) \cdot \mathbf{n}_{iR} = g_R(\mathbf{m}_{iR})$  is enforced within polynomial  $\varphi_{iR}(\mathbf{x})$  and constraint functional  $\widehat{\mathcal{G}}_{iB}(\boldsymbol{\eta}) := \widehat{\mathcal{G}}_{iR}(\boldsymbol{\eta}_{iR})$  must have the form

$$\widehat{\mathcal{G}}_{iR}(\boldsymbol{\eta}_{iR}) = \sigma_{iR} \varphi_{iR}(\mathbf{m}_{iR}) + \rho_{iR} \nabla\varphi_{iR}(\mathbf{m}_{iR}) \cdot \mathbf{n}_{iR} - g, \quad (26)$$

which corresponds to Equation (19) where collocation point  $\mathbf{p}_{iN} := \mathbf{m}_{iN}$  corresponds to the edge midpoint, vector  $\mathbf{v}_{iN} := \mathbf{n}_{iN}$  corresponds to the edge outward unit normal vector, the coefficients are defined as  $\sigma_{iN} := \sigma(\mathbf{m}_{iN})$  and  $\rho_{iN} := \rho(\mathbf{m}_{iN})$  and the boundary condition function value is replaced by free-parameter  $g$  (see Figure 7(c)).

The polynomial reconstruction coefficients in vector  $\widehat{\boldsymbol{\eta}}_{iD}$  computed by the least-square procedure, linearly depends on free-parameter  $g$ . Notice that the polynomial reconstruction procedure does not depend on the physical boundary but, instead, on the edge midpoint. The fitting condition is introduced as an affine real valued functional  $\widehat{\mathcal{B}}_{iB}(g)$  depending on parameter  $g$ , and is defined according to the prescribed boundary condition at boundary partition  $\Gamma_B$ , as follows:

- In case edge  $e_{iB}$  is on boundary partition  $\Gamma_D$  (index  $B$  is replaced by  $D$ ), the Dirichlet boundary condition  $\phi(\mathbf{b}_{iD}) = g_D(\mathbf{b}_{iD})$ , is enforced within polynomial  $\varphi_{iD}(\mathbf{x})$  and functional  $\widehat{\mathcal{B}}_{iB}(g) := \widehat{\mathcal{B}}_{iD}(g)$  must have the form

$$\widehat{\mathcal{B}}_{iD}(g) = \varphi_{iD}(\mathbf{b}_{iD}) - g_{D,iD}, \quad (27)$$

which corresponds to Equation (17) where  $\mathbf{p}_{iD} := \mathbf{b}_{iD}$  is a collocation point close to edge  $e_{iD}$  extracted from the physical boundary, and the boundary condition function value is defined as  $g_{D,iD} := g_D(\mathbf{b}_{iD})$  (see Figure 7(a));

- In case edge  $e_{iB}$  is on boundary partition  $\Gamma_N$  (index  $B$  is replaced by  $N$ ), the Neumann boundary condition  $-\kappa(\mathbf{b}_{iN})\nabla\phi(\mathbf{b}_{iN}) \cdot \mathbf{r}_{iN} = g_N(\mathbf{b}_{iN})$  is enforced within polynomial  $\varphi_{iN}(\mathbf{x})$  and functional  $\widehat{\mathcal{B}}_{iB}(g) := \widehat{\mathcal{G}}_{iN}(g)$  must have the form

$$\widehat{\mathcal{B}}_{iN}(g) = -\kappa_{iN}\nabla\varphi_{iN}(\mathbf{b}_{iN}) \cdot \mathbf{r}_{iN} - g_{N,iN}, \quad (28)$$

which corresponds to Equation (18) where  $\mathbf{p}_{iN} := \mathbf{b}_{iN}$  is a collocation point close to edge  $e_{iN}$  extracted from the physical boundary, vector  $\mathbf{v}_{iN} := \mathbf{r}_{iN}$  is the associated outward unit normal vector to partition  $\Gamma_N$  at point  $\mathbf{b}_{iN}$ , the coefficient is defined as  $\kappa_{iN} := \kappa(\mathbf{b}_{iN})$ , and the boundary condition function value is defined as  $g_{N,iN} := g_N(\mathbf{b}_{iN})$  (see Figure 7(b));

- In case edge  $e_{iB}$  is on boundary partition  $\Gamma_R$  (index  $B$  is replaced by  $R$ ), condition  $\sigma(\mathbf{b}_{iR})\phi(\mathbf{b}_{iR}) + \rho(\mathbf{b}_{iR})\nabla\phi(\mathbf{b}_{iR}) \cdot \mathbf{r}_{iR} = g_R(\mathbf{b}_{iR})$  is enforced within polynomial  $\varphi_{iR}(\mathbf{x})$  and functional  $\widehat{\mathcal{B}}_{iB}(g) := \widehat{\mathcal{G}}_{iR}(g)$  must have the form

$$\widehat{\mathcal{B}}_{iR}(g) = \sigma_{iR}\varphi_{iR}(\mathbf{b}_{iR}) + \rho_{iR}\nabla\varphi_{iR}(\mathbf{b}_{iR}) \cdot \mathbf{r}_{iR} - g_{R,iR}, \quad (29)$$

which corresponds to Equation (19)  $\mathbf{p}_{iR} := \mathbf{b}_{iR}$  is a collocation point close to edge  $e_{iR}$  extracted from the physical boundary, vector  $\mathbf{v}_{iR} := \mathbf{r}_{iR}$  is the associated outward unit normal vector to partition  $\Gamma_R$  at point  $\mathbf{b}_{iR}$ , the coefficient are defined as  $\sigma_{iR} := \sigma(\mathbf{b}_{iR})$  and  $\rho_{iR} := \rho(\mathbf{b}_{iR})$  and the boundary condition function value is defined as  $g_{R,iR} := g_R(\mathbf{b}_{iR})$  (see Figure 7(c)).

Functional  $\widehat{\mathcal{B}}_{iB}(g)$  depends on free-parameter  $g$  since polynomial reconstruction  $\varphi_{iD}(\mathbf{x})$  also depends on  $g$ . Notice that the least-squares procedure is constrained by one of Equations (24-26) and does not involve  $\widehat{\mathcal{B}}_{iD}(g)$ .

In order to exactly satisfy the boundary condition, a parameter  $g^*$  is sought such that the employed least-squares constraint (24-26) and equation  $\widehat{\mathcal{B}}_{iB}(g) = 0$  are satisfied simultaneously. Due to the linearity of functional  $\widehat{\mathcal{B}}_{iB}(g)$  with respect to free-parameter  $g$ , parameter  $g^*$  is sought under the form

$$g^* = g_0 - \varepsilon \frac{\widehat{\mathcal{B}}_{iB}(g_0)}{\widehat{\mathcal{B}}_{iB}(g_0 + \varepsilon) - \widehat{\mathcal{B}}_{iB}(g_0)}, \quad (30)$$

where  $g_0$  and  $\varepsilon \neq 0$  are scalars. In other words, the free-parameter in the least-squares constraint is freely adjusted such that  $\widehat{\mathcal{B}}_{iB}(g) = 0$  and therefore  $\widehat{\varphi}_{iB}(\mathbf{b}_{iB}) = g_{B,iB}$ . Although it seems that two constraints are being applied to the least-squares problems, in fact only one constraint, whose right-hand side (the value of  $g$ ) serves as an intermediate value to satisfy the fitting condition, is genuinely active. Here, in practice scalar  $g_0 = g_{B,iB}$  is chosen for robustness since parameter  $g^*$  is relatively close to  $g_{B,iB}$  as only smooth solutions are computed. Moreover, scalar  $\varepsilon = 1$  is chosen in this article as only normalized solutions are computed but, for the general case, values of the same order of magnitude as the solution are preferable for robustness purposes.

Contrarily to the previous methods, here, the polynomial reconstruction  $\varphi_{iB}(\mathbf{x})$  does not depend on the boundary position. Hence, its coefficients are given by a matrix-vector product between the mesh-based structural matrix and the vector of local values  $\phi_k$ ,  $k \in \mathcal{S}_{iB}$ , and depends on the free-parameter  $g$ . Therefore, the structural matrix is evaluated only during the preprocessing procedure and the computational effort is reduced. Consequently, the evaluation of functional  $\widehat{\mathcal{B}}_{iB}(g)$  is rather fast leading to an efficient computation of  $g^*$ . More importantly, AROD method generates a polynomial reconstruction  $\widehat{\varphi}_{iB}(\mathbf{x})$  which satisfies the prescribed boundary condition at the collocation point on the physical boundary.

**Remark 3.** *The AROD method has been designed for situation when the physical boundary and its associated collocation points evolve in time. Compared to ROD method, AROD method requires an additional step to compute parameter  $g^*$ . For such situations, the ROD method requires to recompute the structural matrix and to solve the associated least-squares problem, while AROD method re-uses the initial and precomputed structural matrix.*

#### 4.4. AROD–XY – Adaptive Reconstruction for Off-site Data method

As seen before, the least-squares constraint and the fitting condition for the AROD method have different collocation points, outward unit normal vectors, and coefficients, nonetheless, both are of the same type. The fitting conditions have to respect the type of the prescribed boundary condition, Dirichlet, Neumann, or Robin. To ensure an arbitrary high-order of accuracy, AROD methods assumes that the collocation points, outward unit normal vectors, coefficients, and right-hand side values, are extracted from the exact physical boundary and its associated boundary conditions functions. Remind that there is no reason for the least-squares constraint in Equations (24-26) to respect the prescribed boundary condition. Although it is the natural choice to choose both of the same type, a different approach called AROD–XY method was found to be more stable, as will be numerically proven in Section 6. Character X recall the fitting condition type while Y refers to the least-squares constraint type, and will be replaced by D, N, or R, standing for Dirichlet, Neumann, and Robin types, respectively.

The formulation of the AROD–XY method strictly follows the AROD method (see Subsection 4.3), where the only change occurs for the least-squares constraint. Consequently, only this difference is addressed. Table 4 summarizes the constraints and fitting conditions used for each of the AROD–XY methods considered.

Table 4: Description of the constraints and fitting conditions used for AROD–XY methods.

Method	Fitting condition		Least-squares constraint	
	Type	Equation	Type	Equation
AROD–DN AROD–DR	Dirichlet	(27)	Neumann	$\widehat{\mathcal{G}}_{iD}(\boldsymbol{\eta}_{iD}) = -\kappa(\mathbf{m}_{iD})\nabla\varphi_{iD}(\mathbf{m}_{iD}) \cdot \mathbf{n}_{iD} - g$
			Robin	$\widehat{\mathcal{G}}_{iD}(\boldsymbol{\eta}_{iD}) = \rho(\mathbf{m}_{iD})\varphi_{iD}(\mathbf{m}_{iD}) + \sigma(\mathbf{m}_{iD})\nabla\varphi_{iD}(\mathbf{m}_{iD}) \cdot \mathbf{n}_{iD} - g$
AROD–ND AROD–NR	Neumann	(28)	Dirichlet	$\widehat{\mathcal{G}}_{iN}(\boldsymbol{\eta}_{iN}) = \varphi_{iN}(\mathbf{m}_{iN}) - g$
			Robin	$\widehat{\mathcal{G}}_{iN}(\boldsymbol{\eta}_{iN}) = \rho(\mathbf{m}_{iN})\varphi_{iD}(\mathbf{m}_{iN}) + \sigma(\mathbf{m}_{iN})\nabla\varphi_{iN}(\mathbf{m}_{iN}) \cdot \mathbf{n}_{iN} - g$
AROD–RD AROD–RN	Robin	(29)	Dirichlet	$\widehat{\mathcal{G}}_{iR}(\boldsymbol{\eta}_{iR}) = \varphi_{iR}(\mathbf{m}_{iR}) - g$
			Neumann	$\widehat{\mathcal{G}}_{iR}(\boldsymbol{\eta}_{iR}) = -\kappa(\mathbf{m}_{iR})\nabla\varphi_{iR}(\mathbf{m}_{iR}) \cdot \mathbf{n}_{iR} - g$

**Remark 4.** *When the fitting condition is of Dirichlet or Neumann types but the least-squares constraint is of Robin type, as in methods AROD–DR and AROD–NR, functions  $\sigma(\mathbf{x})$  and  $\rho(\mathbf{x})$  are not available for that boundary partition and some definition is required. In this article, functions  $\sigma(\mathbf{x}) := \mathbf{u}(\mathbf{x}) \cdot \mathbf{n}$  and  $\rho(\mathbf{x}) := -\kappa(\mathbf{x})$  are chosen such that the constrained Robin condition represents a total flux condition and has a physical meaning.*

**Remark 5.** *This approach can improve robustness to handle boundary conditions on curved boundaries compared with the AROD method, as will be numerically proven in Section 6. On the other hand, the AROD–XY method can handle situations where the boundary condition type changes, as instance, from one time-step to another in a time-dependent or unsteady problem, therefore avoiding to recompute the least-squares procedure.*

**Remark 6.** *In the proposed methods, a single collocation point is considered to constrain the boundary conditions polynomial reconstructions associated to the boundary edges. Contrarily, the common practice, as in curved mesh approaches, consists in finding multiple collocation points, usually corresponding to integration points on the physical boundary. Given that, one may question the the subsequent impact of a single collocation point on the quality of the high-degree polynomial reconstructions to represent boundary conditions.*

*The first attempt for the proposed method included a formulation with multiple collocation points on the curved boundary per polynomial reconstruction associated to the boundary edges. An exhaustive numerical verification were performed varying the number of collocation points from one up to three - notice that a sixth-order quadrature on the curved boundary would require three points. From the obtained results, conclusions were drawn showing that a single collocation point per polynomial reconstruction is the optimal choice, that is, no significant accuracy, robustness, or convergence rates gains were consistently observed with multiple collocation points, regardless the scheme order. In fact, a slight accuracy deterioration (and consequently of efficiency) occurred in the generality of the cases when considering multiple collocation points, particularly when constraining gradients. This results may be related to the over-fitted or over-constrained polynomial reconstructions, which increases least-squares matrices*

condition number and, consequently, has a negative impact on solution accuracy that is not counterbalanced by the additional information.

On the other side, one must keep in mind that each boundary polynomial reconstruction includes, not only a single collocation point to constrain the boundary condition, but also the mean-values of the neighbor cells included in the stencil. In turn, each of these mean-values takes part in the computation of several boundary polynomial reconstructions since the associated cell is shared among many stencils. Also, notice that, when higher polynomial degrees are used, larger stencils are required and, therefore, each cell is shared among more stencils. Hence, in a intrinsic way, each polynomial reconstruction associated to the boundary edges depends on several collocation points via other polynomial reconstructions.

## 5. Finite volume scheme

Generic Finite Volume discretization derives from applying the divergence theorem to Equation (1) over each cell  $c_i$ ,  $i = 1, \dots, N_C$ , providing equation

$$\int_{\partial c_i} (\mathbf{u}\phi - \kappa\nabla\phi) \cdot \mathbf{n}_i ds = \int_{c_i} f d\mathbf{x}, \quad (31)$$

where  $\partial c_i$  stands for the cell boundary and  $\mathbf{n}_i$  the associated outward unit normal vector. An  $R$ -points Gaussian quadrature rule with weights  $\xi_r$ ,  $r = 1, \dots, R$ , is considered for the line integration, which provides a residual expression of order  $2R$  for Equation (31), given as

$$\sum_{j \in \mathcal{N}_i} |e_{ij}| \left[ \sum_{r=1}^R \zeta_r (\mathbb{F}_{ij,r}^C + \mathbb{F}_{ij,r}^D) \right] - f_i |c_i| = \mathcal{O}(h_i^{2R}), \quad (32)$$

where  $h_i = \max_{j \in \mathcal{N}_i} |e_{ij}|$  and  $f_i$  stands for an approximation of order  $2R$  of the mean-value of the source term function  $f$  over cell  $c_i$ . Notice that, if cell  $c_i$  is not triangular, a splitting into sub-triangles which share the cell centroid as a common vertex is used, and the quadrature rule is then applied in each sub-triangle, as in described in [32]. Physical diffusive flux  $\mathbb{F}_{ij,r}^D$ , and physical convective flux  $\mathbb{F}_{ij,r}^C$ , at quadrature point  $\mathbf{q}_{ij,r}$  are given by

$$\mathbb{F}_{ij,r}^C = (\mathbf{u}(\mathbf{q}_{ij,r}) \cdot \mathbf{n}_{ij}) \phi(\mathbf{q}_{ij,r}), \quad \mathbb{F}_{ij,r}^D = -\kappa(\mathbf{q}_{ij,r}) \nabla\phi(\mathbf{q}_{ij,r}) \cdot \mathbf{n}_{ij}. \quad (33)$$

Notice that  $\mathbb{F}_{ij,r}^C = \mathbb{F}_{ji,r}^C$  and  $\mathbb{F}_{ij,r}^D = \mathbb{F}_{ji,r}^D$ .

### 5.1. Numerical fluxes

Given the polynomial reconstructions, see Table 3, the approximations to the physical fluxes are then computed with respect to the edges, as follows: For inner edge  $e_{ij}$ ,  $i \in \mathcal{C}_M$ ,  $j \in \mathcal{N}_i \setminus \{\text{D, NR}\}$ , the numerical diffusive and convective fluxes at quadrature points  $\mathbf{q}_{ij,r}$ ,  $r = 1, \dots, R$ , write

$$\mathcal{F}_{ij,r}^C = \mathcal{F}_{ji,r}^C = [\mathbf{u}(\mathbf{q}_{ij,r}) \cdot \mathbf{n}_{ij}]^+ \widehat{\varphi}_i(\mathbf{q}_{ij,r}) + [\mathbf{u}(\mathbf{q}_{ij,r}) \cdot \mathbf{n}_{ij}]^- \widehat{\varphi}_j(\mathbf{q}_{ij,r}), \quad (34)$$

$$\mathcal{F}_{ij,r}^D = \mathcal{F}_{ji,r}^D = -\kappa(\mathbf{q}_{ij,r}) \nabla\widehat{\varphi}_{ij}(\mathbf{q}_{ij,r}) \cdot \mathbf{n}_{ij}, \quad (35)$$

where  $[a]^+ = \max(0, a)$  and  $[a]^- = \min(0, a)$  for any scalar  $a \in \mathbb{R}$ . For boundary edge  $e_{iB}$ ,  $i \in \mathcal{B}_M$ ,  $B \in \{\text{D, N, R}\}$ , with a prescribed boundary conditions, the numerical diffusive and convective fluxes at quadrature points  $\mathbf{q}_{iB,r}$ ,  $r = 1, \dots, R$ , write

$$\mathcal{F}_{iB,r}^C = [\mathbf{u}(\mathbf{q}_{iB,r}) \cdot \mathbf{n}_{iB}]^+ \widehat{\varphi}_i(\mathbf{q}_{iB,r}) + [\mathbf{u}(\mathbf{q}_{iB,r}) \cdot \mathbf{n}_{iB}]^- \widehat{\varphi}_{iB}(\mathbf{q}_{iB,r}), \quad (36)$$

$$\mathcal{F}_{iB,r}^D = -\kappa(\mathbf{q}_{iB,r}) \nabla\widehat{\varphi}_{iB}(\mathbf{q}_{iB,r}) \cdot \mathbf{n}_{iB}. \quad (37)$$

Notice that, in both cases, for the convective fluxes the constrained polynomial reconstruction associated to the adjacent cell and the constrained polynomial reconstruction associated to the edge are used while, for the diffusive fluxes, the edge-based unconstrained polynomial reconstruction is used.

The constrained polynomial reconstructions associated to the boundary edges might be computed with the Naive method (see Section 4.1), ROD method (see Section 4.2), AROD method (see Section 4.3), or AROD-XY method (see Section 4.4).

Notice that all numerical fluxes are computed on the polygonal boundary edges of the computational domain without any explicit reference to the physical domain. Similarly, the prescribed boundary condition is taken into account via polynomial reconstructions  $\widehat{\varphi}_{iB}$  and, therefore, no explicit reference to the boundary condition does appear in the numerical scheme, which deals with only two situations: inner or boundary edges.

## 5.2. Residual operator

For any vector  $\Phi$  in  $\mathbb{R}^{N_C}$  of mean-value approximations, the residual operator for each cell  $c_i$ ,  $i = 1, \dots, N_C$ , defined as

$$\mathcal{R}_i(\Phi) = \sum_{j \in \mathcal{N}_i} |e_{ij}| \left[ \sum_{r=1}^R \zeta_r (\mathcal{F}_{ij,r}^C + \mathcal{F}_{ij,r}^D) \right] - f_i |c_i|, \quad (38)$$

corresponds to the Finite Volume scheme (32) cast in residual form. Gathering all the residual operators, a global affine operator  $\Phi \rightarrow \mathcal{R}(\Phi) = (\mathcal{R}_i(\Phi))_{i \in \mathcal{C}_M}$  is defined. Vector  $\Phi^* \in \mathbb{R}^{N_C}$ , solution the system of linear equations  $\mathcal{R}(\Phi) = 0$ , provides the approximate mean-values of the original convection-diffusion problem (1). A GMRES method, supplemented by a preconditioning matrix, is used to compute an approximation of vector  $\Phi^*$ , similarly to the proposed methods in [25, 27].

## 6. Numerical benchmark

In order to validate the implementation of the proposed methods and assess their performance in terms of accuracy and convergence rates, manufactured solutions are computed for specific curved domains for which the associated source term has to satisfy Equation (1), given the convective and diffusive coefficients. Notice that only smooth solutions of the steady-state two-dimensional convection-diffusion equation are considered to achieve the optimal order. To complete Equation (1), Dirichlet, Neumann, or Robin boundary conditions are prescribed on the corresponding boundary partitions, satisfying the associated analytical solution. The method of validation consists in assessing the rates of the error convergence under mesh refinement, applying the different techniques and methods proposed in this article to compute the approximate solution. Structured and unstructured triangular meshes are considered in this work.

Assume that vector  $\Phi^* = (\phi_i^*)_{i \in \mathcal{C}_M}$  gathers the numerical approximated mean values while vector  $\bar{\Phi} = (\bar{\phi}_i)_{i \in \mathcal{C}_M}$  gathers exact mean-values of function  $\phi(\mathbf{x})$ , that is  $\bar{\phi}_i = (1/|c_i|) \int_{c_i} \phi \, d\mathbf{x}$ . Relative  $L^1$ - and  $L^\infty$ -norm errors, denoted as  $E_1$  and  $E_\infty$ , respectively, are evaluated as

$$E_1(\mathcal{M}) = \frac{\sum_{i=1}^{N_C} (|\phi_i^* - \bar{\phi}_i| |c_i|)}{\sum_{i=1}^{N_C} (|\bar{\phi}_i| |c_i|) \sum_{i=1}^{N_C} |c_i|} \quad \text{and} \quad E_\infty(\mathcal{M}) = \frac{\max_{i=1}^{N_C} |\phi_i^* - \bar{\phi}_i|}{\sum_{i=1}^{N_C} (|\bar{\phi}_i| |c_i|)}. \quad (39)$$

Convergence rates for relative  $L^1$ - and  $L^\infty$ -norm errors between two different meshes,  $\mathcal{M}_1$  and  $\mathcal{M}_2$ , with  $DOF_1$  and  $DOF_2$  degrees of freedom (equal to the number of cells), respectively, where  $DOF_1 \neq DOF_2$ , are estimated by

$$O_k(\mathcal{M}_1, \mathcal{M}_2) = 2 \frac{|\log(E_k(\mathcal{M}_1)/E_k(\mathcal{M}_2))|}{|\log(DOF_1/DOF_2)|}, \quad k \in \{1, \infty\}. \quad (40)$$

The Naive, ROD, AROD, and AROD–XY methods are tested and compared using  $\mathbb{P}_d$  polynomial reconstructions degrees, with  $d \in \{1, 3, 5\}$ . For the Naive method, the same degree  $d$  is considered for all the elements (see Table 3). Contrarily, for the ROD and AROD methods, either (i) the same degree  $d$  is considered for all the elements or (ii) a degree  $d + 1$  is considered for the Neumann and Robin boundary edges (see Table 3) while degree  $d$  is considered for the others, in this case we adopt the notation  $\mathbb{P}_d/\mathbb{P}_{d+1}$ .

All the simulations are carried out considering the weighting function (7) with  $\delta = 5h$  and  $\gamma = 2$ , where  $h$  is the characteristic length of the reference mesh edge or cell. Two non-polygonal physical domains are considered: a simple annulus domain, in Subsection 6.1, and a rose-shaped one, in Subsection 6.2.

**Remark 7.** *For the presented geometries, all the mesh boundary edges intersect with the physical curved boundary at only two points corresponding to the edge ends. For the general case, there is no particular limitation and the mesh boundary edges may intersect one, several, or have no intersections with the physical boundary. The key point is that the collocation point for boundary condition constraining is extracted from the surrounding physical curved boundary. Notice, however, that only smooth curved boundaries are handled in this benchmark and, therefore, more than two intersections on each mesh boundary edge are not expected for refined enough grids.*

### 6.1. Annulus domain

A simple annulus geometry is first addressed, consisting of an interior,  $\Gamma_I$ , and exterior,  $\Gamma_E$ , circumferences centered at point  $(0, 0)$  with radius  $r_I = 0.5$  and  $r_E = 1$ , respectively. For this domain, the assigned manufactured solution (see Figure 8(a)), invariant by rotation, is expressed in terms of polar coordinates  $(r, \theta)$  as

$$\phi(r) = a (\exp(R(r)) + \exp(-R(r)) + b) + 1, \quad R \equiv R(r) = \frac{2r - (r_E + r_I)}{r_E - r_I}, \quad (41)$$

where  $r^2 = x^2 + y^2$  such that  $R \in [-1, 1]$ , and coefficients  $a, b \in \mathbb{R}$ . Notice that, although the solution is given in terms of  $r$  and  $\theta$ , the problem is numerically solved in Cartesian coordinates. In order to guarantee property  $\phi(\mathbf{x}) \in [1, 2]$  in  $\Omega$ , the coefficients  $a$  and  $b$  are deduced as  $a = 1/(2 - \exp(1) - \exp(-1))$  and  $b = -\exp(1) - \exp(-1)$ . The associated source term function is obtained after substituting Equation (41) into Equation (1) (see Figure 8(b)).

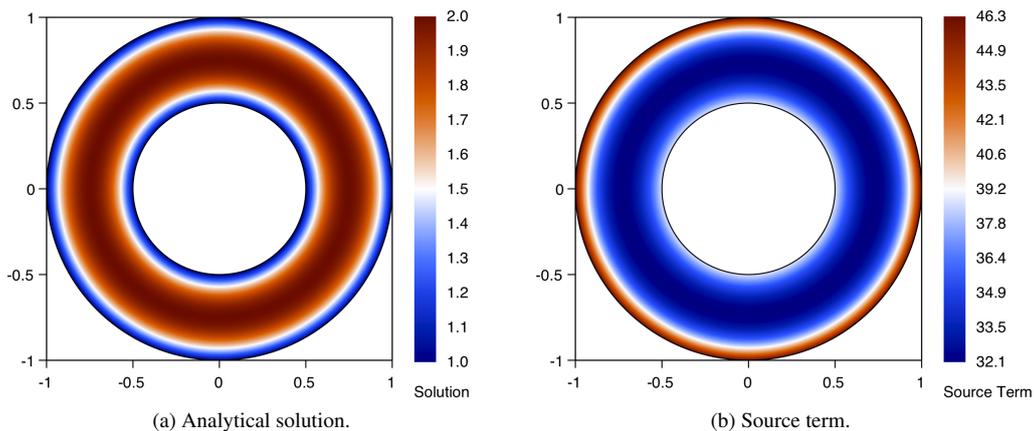


Figure 8: Analytical solution and source term in a very fine mesh for the annulus domain (for interpretation of the color in this figure, the reader is referred to the electronic version of this article).

Successive refined uniform triangular Delaunay meshes are generated for the annulus domain (see Figure 9(a)) and the Naive and ROD methods with  $\mathbb{P}_d$ ,  $d \in \{1, 3, 5\}$ , polynomial reconstructions are tested. Let emphasize that the boundary vertices belong to the physical boundary and the maximum gap between the physical and the computational boundaries is of order  $O(h^2)$  (see Figure 9(b)). Pure convective and pure diffusive situations, prescribing different boundary conditions, are addressed in this test case assigning convection and diffusion coefficients accordingly.

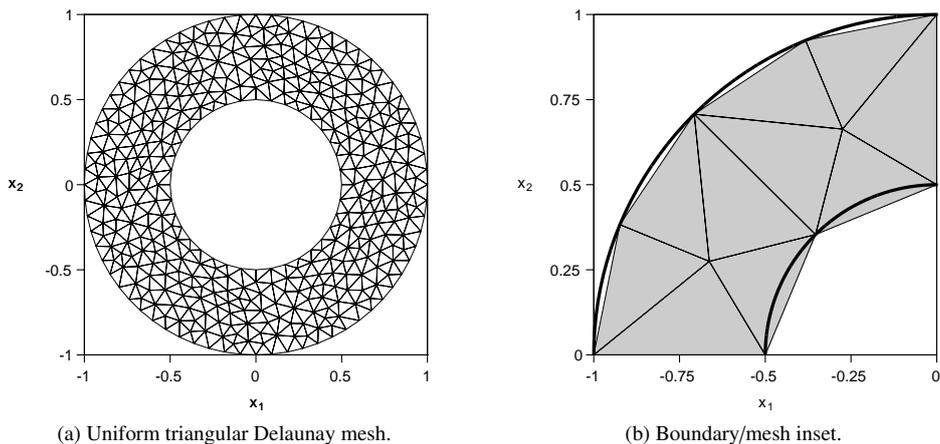


Figure 9: Coarse uniform triangular Delaunay mesh and inset of a very coarse mesh (gray filling) and the curved physical boundaries (thick lines) for the annulus domain.

#### 6.1.1. Pure convective case

A radial velocity field  $\mathbf{u}(r, \theta) = (\cos(\theta), \sin(\theta))$  is assigned while no diffusion is considered, that is  $\kappa = 0$ . Inner boundary  $\Gamma_I$  is prescribed with constant Dirichlet boundary condition  $g_D(\mathbf{x}) = 1$ ,  $\mathbf{x} \in \Gamma_I$ , while no boundary condition

is required for outer boundary  $\Gamma_E$  since an upwind scheme is applied to discretize the hyperbolic term in Equation (1).

In Figure 10 are shown the individual errors distribution obtained from the Naive and ROD methods with  $\mathbb{P}_d$ ,  $d \in \{1, 3, 5\}$ , polynomial reconstructions with a mesh made of 736 cells. As observed, both the Naive and ROD methods seem to provide comparable errors distribution when  $\mathbb{P}_1$  polynomial reconstructions are used. In fact, the geometrical mismatch between the boundary collocation points, located on the computational boundary, and the physical boundary is of order  $O(h^2)$ , and the same size error is expected for the inner fluxes approximations. On the contrary, when  $\mathbb{P}_3$  and  $\mathbb{P}_5$  polynomial reconstructions are used, the Naive method noticeably provides a stalled error resulting from the treatment of the inner boundary. Notice that, as there is no diffusion term in this test case and a radial velocity field is prescribed, the outer boundary represents an outflow condition and, therefore, no boundary condition and curved boundary treatment is, in fact, applied here. As predicted, the solution accuracy is restored for high-degree polynomial reconstructions when the ROD method is used since the boundary condition is appropriately handled on the inner curved boundary.

Table 5 reports the obtained errors and corresponding convergence rates. As observed, the Naive method provides at most a second-order convergence rate for both error norms, regardless of the polynomial degree considered. Such results are expected attending to the geometrical mismatch of order  $O(h^2)$  between the boundary collocation points on the computational boundary and the true physical boundary. The ROD method, contrarily, restores the optimal effective second-, fourth-, and sixth-order convergence rates for polynomial degrees of  $d \in \{1, 3, 5\}$ , respectively, while no oscillations are reported. The results support the capability of the ROD method to overcome the second-order limitation expected when dealing with curved boundaries, without the use of curved cells to fit the boundaries. Notice that the hyperbolic term is treated with no difficulty, no nonphysical oscillations appear due to the upwind flux used for the convective contribution, and the steady-state scheme is unconditionally stable with respect to the mesh parameter.

Table 5: Pure convective case with Dirichlet–Dirichlet boundary conditions – relative errors and convergence rates obtained from the Naive and ROD methods with uniform triangular Delaunay meshes.

DOF	$d = 1$				$d = 3$				$d = 5$			
	$E_1$	$O_1$	$E_\infty$	$O_\infty$	$E_1$	$O_1$	$E_\infty$	$O_\infty$	$E_1$	$O_1$	$E_\infty$	$O_\infty$
<b>Naive method with <math>\mathbb{P}_d</math></b>												
736	3.65E-03	—	1.48E-02	—	2.72E-03	—	4.11E-03	—	2.78E-03	—	4.05E-03	—
2828	8.96E-04	2.09	5.25E-03	1.54	7.73E-04	1.87	1.15E-03	1.89	7.79E-04	1.89	1.15E-03	1.87
11500	2.35E-04	1.91	1.87E-03	1.47	1.94E-04	1.97	2.89E-04	1.97	1.95E-04	1.98	2.90E-04	1.97
45248	6.37E-05	1.90	6.47E-04	1.55	5.01E-05	1.98	7.49E-05	1.97	5.02E-05	1.98	7.49E-05	1.98
177880	1.69E-05	1.94	2.21E-04	1.57	1.27E-05	2.00	1.91E-05	2.00	1.27E-05	2.00	1.91E-05	2.00
<b>ROD method with <math>\mathbb{P}_d</math></b>												
736	2.96E-03	—	1.19E-02	—	6.64E-05	—	2.12E-04	—	1.15E-05	—	4.04E-05	—
2828	7.66E-04	2.01	4.66E-03	1.39	7.19E-06	3.30	2.52E-05	3.17	4.34E-07	4.87	1.25E-06	5.17
11500	2.00E-04	1.92	1.73E-03	1.42	4.82E-07	3.85	1.95E-06	3.65	7.73E-09	5.74	2.36E-08	5.66
45248	5.43E-05	1.90	6.09E-04	1.52	3.65E-08	3.77	2.22E-07	3.18	1.71E-10	5.57	6.18E-10	5.32
177880	1.48E-05	1.90	2.02E-04	1.61	2.38E-09	3.99	1.33E-08	4.11	2.98E-12	5.91	1.41E-11	5.52

### 6.1.2. Pure diffusive case

For this test case, a constant diffusion coefficient  $\kappa = 1$  is assigned while no velocity field is considered, that is  $\mathbf{u} = 0$ . Inner boundary  $\Gamma_I$  is prescribed with constant Dirichlet boundary condition  $g_D(\mathbf{x}) = 1$ ,  $\mathbf{x} \in \Gamma_I$ , and  $\Gamma_I \subseteq \Gamma_D$ , while outer boundary  $\Gamma_E$  is prescribed according to one of the following cases:

- Dirichlet–Dirichlet: a constant Dirichlet boundary condition  $g_D(\mathbf{x}) = 1$ ,  $\mathbf{x} \in \Gamma_E$ , and  $\Gamma_E \subset \Gamma_D$ ;
- Dirichlet–Neumann: a constant Neumann boundary condition  $g_N(\mathbf{x}) = 4a(\exp(1) - \exp(-1))$ ,  $\mathbf{x} \in \Gamma_E$ , and  $\Gamma_E = \Gamma_N$ ;
- Dirichlet–Robin: a constant Robin boundary condition, providing constant coefficients  $\alpha(\mathbf{x}) = 1$  and  $\beta(\mathbf{x}) = 1$ , and  $g_R(\mathbf{x}) = 1 + 4a(\exp(1) - \exp(-1))$ ,  $\mathbf{x} \in \Gamma_E$ , and  $\Gamma_E = \Gamma_R$ .

*Dirichlet–Dirichlet case.* The Dirichlet boundary conditions on both boundaries are firstly addressed and Table 6 reports the errors and the corresponding convergence rates obtained from the Naive and ROD methods with  $\mathbb{P}_d$ ,  $d \in \{1, 3, 5\}$ , polynomial reconstructions. Similarly to the pure convective case, the Naive method provides at most a second-order convergence rate for both error norms, regardless of the polynomial degree, while the ROD method

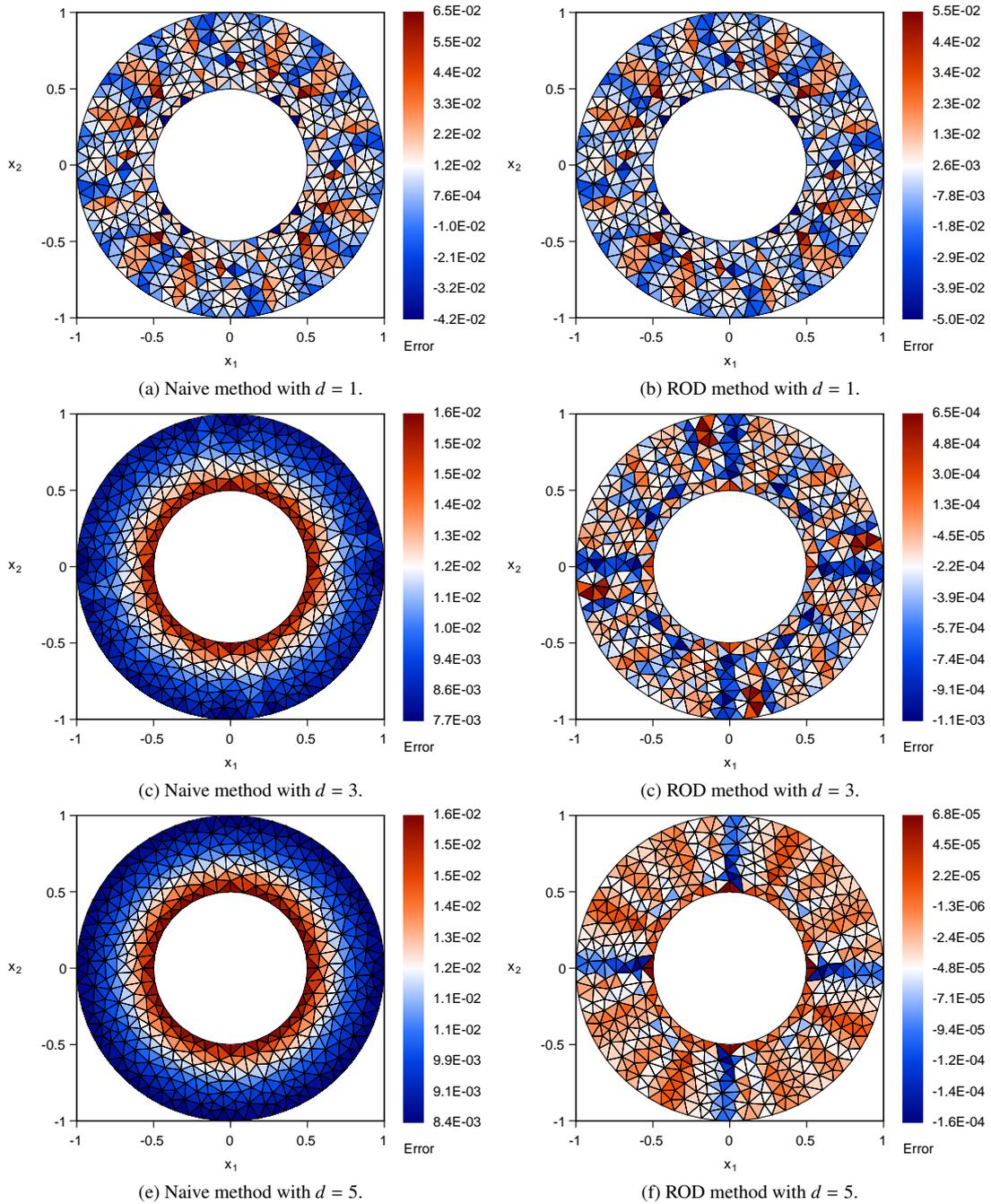


Figure 10: Pure convective case with Dirichlet–Dirichlet boundary conditions – individual errors distribution obtained from the Naive and ROD methods with  $\mathbb{P}_d$  polynomial reconstructions and a uniform triangular Delaunay mesh with 736 cells (for interpretation of the color in this figure, the reader is referred to the electronic version of this article).

restores the optimal orders while no oscillations are reported. The results highlight, once again, the capability of the proposed method to overcome the second-order limitation, in that case for the elliptic term.

Table 6: Pure diffusive case with Dirichlet–Dirichlet boundary conditions – relative errors and convergence rates obtained from the Naive and ROD methods with uniform triangular Delaunay meshes.

DOF	$d = 1$				$d = 3$				$d = 5$			
	$E_1$	$O_1$	$E_\infty$	$O_\infty$	$E_1$	$O_1$	$E_\infty$	$O_\infty$	$E_1$	$O_1$	$E_\infty$	$O_\infty$
<b>Naive method with <math>\mathbb{P}_d</math></b>												
736	7.71E-03	—	1.40E-02	—	1.49E-03	—	3.84E-03	—	1.50E-03	—	3.90E-03	—
2828	2.31E-03	1.79	4.15E-03	1.80	4.06E-04	1.93	1.13E-03	1.82	4.06E-04	1.94	1.13E-03	1.84
11500	5.90E-04	1.95	1.11E-03	1.89	1.03E-04	1.96	2.87E-04	1.95	1.03E-04	1.96	2.87E-04	1.95
45248	1.50E-04	2.00	3.09E-04	1.86	2.63E-05	1.99	7.45E-05	1.97	2.63E-05	1.99	7.46E-05	1.97
177880	3.54E-05	2.11	8.37E-05	1.91	6.65E-06	2.01	1.90E-05	2.00	6.65E-06	2.01	1.90E-05	2.00
<b>ROD method with <math>\mathbb{P}_d</math></b>												
736	7.52E-03	—	1.29E-02	—	3.03E-05	—	1.06E-04	—	3.92E-06	—	1.93E-05	—
2828	2.21E-03	1.82	3.73E-03	1.84	2.00E-06	4.04	1.24E-05	3.19	1.29E-07	5.07	9.48E-07	4.48
11500	5.68E-04	1.94	1.01E-03	1.86	2.55E-07	2.94	1.30E-06	3.21	1.16E-09	6.72	9.83E-09	6.51
45248	1.44E-04	2.00	2.64E-04	1.95	1.84E-08	3.84	8.70E-08	3.95	2.55E-11	5.57	2.74E-10	5.23
177880	3.38E-05	2.12	7.35E-05	1.87	8.63E-10	4.47	5.53E-09	4.02	5.36E-13	5.64	6.55E-12	5.46

*Dirichlet–Neumann case.* The Neumann boundary condition is now prescribed for outer boundary  $\Gamma_E$  and Table 7 reports the errors and the corresponding convergence rates obtained from the Naive method, ROD method with  $\mathbb{P}_d$ , and  $\mathbb{P}_d/\mathbb{P}_{d+1}$   $d \in \{1, 3, 5\}$ , polynomial reconstructions. Inexorably the Naive method provides at most a second-order convergence rate for both error norms, regardless of the polynomial degree. On the other hand, the ROD method with  $\mathbb{P}_d$ ,  $d \in \{1, 3, 5\}$ , polynomial reconstructions achieves second-, third-, and fifth-order convergence rates, respectively, while remind that second-, fourth-, and sixth-order convergence rates were achieved for the previous Dirichlet–Dirichlet case. To overcome this problem, the scheme is slightly modified considering  $(d + 1)$ -degree polynomial reconstructions for the Neumann boundary edges while  $d$ -degree polynomial reconstructions are maintained for the remaining mesh edges and cells. Doing so, a small performance penalty is accepted since additional coefficients for the  $\mathbb{P}_{d+1}$  polynomial reconstructions have to be included in the associated least-squares procedure. In practice, this penalty is acceptable according to the relative small number of  $\mathbb{P}_{d+1}$  reconstructions – around 3–4% of the total number of reconstructions for uniform triangular Delaunay meshes generated for the annulus domain. As can be seen in Table 7, applying the ROD method with  $\mathbb{P}_d/\mathbb{P}_{d+1}$ ,  $d \in \{1, 3, 5\}$ , polynomial reconstructions, restore the optimal convergence rates. Notice that the IEEE 754 floating-point precision is attained with  $d = 5$  for the last mesh and, therefore, no convergence rate was computed.

*Dirichlet–Robin case.* Last, a Robin boundary condition is prescribed on outer boundary  $\Gamma_E$ , and Table 8 reports the errors and corresponding convergence rates obtained from the Naive method, ROD method with  $\mathbb{P}_d$ , and  $\mathbb{P}_d/\mathbb{P}_{d+1}$ ,  $d \in \{1, 3, 5\}$ , polynomial reconstructions. The results are consistent with the conclusions previously drawn: the Naive method is limited to a maximal second-order convergence rate and the ROD method with  $\mathbb{P}_d$ ,  $d \in \{3, 5\}$ , polynomial reconstructions does not achieve the optimal convergence rates. On the other hand, the ROD method with  $\mathbb{P}_d/\mathbb{P}_{d+1}$ ,  $d \in \{3, 5\}$ , polynomial reconstructions seems to effectively increase the approximated solution accuracy and achieves the expected fourth- and sixth-order convergence rates, respectively.

## 6.2. Rose-shaped domain

The convection-diffusion problem is now addressed in a complex shape the boundaries of which cannot be parameterized by polynomial terms. For this purpose, a diffeomorphism is applied to the annulus geometry previously introduced. This consists in a periodic transformation of its boundaries given in polar coordinates  $(r, \theta)$  by

$$\Gamma_I : \begin{bmatrix} x \\ y \end{bmatrix} = R_I(\theta; r_I, \alpha_I) \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}, \quad \Gamma_E : \begin{bmatrix} x \\ y \end{bmatrix} = R_E(\theta; r_E, \alpha_E) \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}, \quad (42)$$

Functions  $R_I(\theta) := R_I(\theta; r_I, \alpha_I)$  and  $R_E(\theta; r_E) := R_E(\theta; r_E, \alpha_E)$  represent the transformed rays of the inner and outer boundaries, respectively, having initial constant rays  $r_I, r_E \in \mathbb{R}$  and coefficients  $\alpha_I, \alpha_E \in \mathbb{R}$ , and are given, respectively, by

$$R_I(\theta; r_I, \alpha_I) = r_I \left( 1 + \frac{1}{20} \sin(\alpha_I \theta) \right), \quad R_E(\theta; r_E, \alpha_E) = r_E \left( 1 + \frac{1}{20} \sin(\alpha_E \theta) \right). \quad (43)$$

Table 7: Pure diffusive case with Dirichlet–Neumann boundary conditions – relative errors and convergence rates obtained from the Naive and ROD methods with uniform triangular Delaunay meshes.

DOF	$d = 1$				$d = 3$				$d = 5$			
	$E_1$	$O_1$	$E_\infty$	$O_\infty$	$E_1$	$O_1$	$E_\infty$	$O_\infty$	$E_1$	$O_1$	$E_\infty$	$O_\infty$
<b>Naive method with <math>\mathbb{P}_d</math></b>												
736	5.85E-03	—	1.35E-02	—	2.15E-03	—	3.94E-03	—	2.24E-03	—	4.14E-03	—
2828	2.16E-03	1.48	4.12E-03	1.76	5.11E-04	2.14	1.12E-03	1.87	5.21E-04	2.16	1.12E-03	1.94
11500	4.46E-04	2.25	1.09E-03	1.89	1.35E-04	1.90	2.86E-04	1.95	1.37E-04	1.91	2.86E-04	1.95
45248	1.55E-04	1.54	3.10E-04	1.84	3.42E-05	2.00	7.44E-05	1.96	3.44E-05	2.01	7.44E-05	1.96
177880	5.24E-05	1.58	1.03E-04	1.60	8.60E-06	2.02	1.90E-05	1.99	8.63E-06	2.02	1.90E-05	1.99
<b>ROD method with <math>\mathbb{P}_d</math></b>												
736	2.27E-03	—	9.39E-03	—	3.27E-05	—	1.15E-04	—	1.93E-05	—	3.65E-05	—
2828	1.02E-03	1.19	2.96E-03	1.71	1.06E-05	1.67	2.13E-05	2.50	3.86E-07	5.81	8.84E-07	5.53
11500	1.87E-04	2.42	8.03E-04	1.86	2.39E-06	2.13	4.02E-06	2.38	1.56E-08	4.58	2.86E-08	4.89
45248	7.99E-05	1.24	2.35E-04	1.80	3.22E-07	2.92	5.33E-07	2.95	5.12E-10	4.99	9.08E-10	5.04
177880	3.33E-05	1.28	8.41E-05	1.50	3.68E-08	3.17	6.34E-08	3.11	1.64E-11	5.03	3.64E-11	4.70
<b>ROD method with <math>\mathbb{P}_d/\mathbb{P}_{d+1}</math></b>												
736	6.07E-03	—	1.07E-02	—	3.35E-05	—	1.06E-04	—	1.03E-05	—	2.04E-05	—
2828	2.14E-03	1.54	3.19E-03	1.80	2.44E-06	3.89	1.25E-05	3.18	3.38E-07	5.07	9.78E-07	4.51
11500	4.56E-04	2.21	9.21E-04	1.77	6.31E-07	1.93	1.33E-06	3.20	3.36E-09	6.57	9.97E-09	6.54
45248	1.57E-04	1.55	2.97E-04	1.65	5.01E-08	3.70	8.95E-08	3.94	6.93E-11	5.67	2.76E-10	5.24
177880	5.29E-05	1.59	1.15E-04	1.38	1.39E-09	5.24	5.51E-09	4.07	5.64E-12	—	6.30E-11	—

Table 8: Pure diffusive case with Dirichlet–Robin boundary conditions – relative errors and convergence rates obtained from the Naive and ROD methods with uniform triangular Delaunay meshes.

DOF	$d = 1$				$d = 3$				$d = 5$			
	$E_1$	$O_1$	$E_\infty$	$O_\infty$	$E_1$	$O_1$	$E_\infty$	$O_\infty$	$E_1$	$O_1$	$E_\infty$	$O_\infty$
<b>Naive method with <math>\mathbb{P}_d</math></b>												
736	6.44E-03	—	1.36E-02	—	1.85E-03	—	3.80E-03	—	1.90E-03	—	3.85E-03	—
2828	2.16E-03	1.62	4.12E-03	1.78	4.62E-04	2.06	1.12E-03	1.81	4.67E-04	2.08	1.12E-03	1.83
11500	4.89E-04	2.12	1.10E-03	1.89	1.20E-04	1.92	2.86E-04	1.95	1.21E-04	1.93	2.86E-04	1.95
45248	1.49E-04	1.74	3.09E-04	1.85	3.05E-05	2.00	7.45E-05	1.96	3.06E-05	2.00	7.45E-05	1.97
177880	4.44E-05	1.77	9.17E-05	1.78	7.69E-06	2.01	1.90E-05	2.00	7.71E-06	2.01	1.90E-05	2.00
<b>ROD method with <math>\mathbb{P}_d</math></b>												
736	3.96E-03	—	9.97E-03	—	3.00E-05	—	1.10E-04	—	1.08E-05	—	2.24E-05	—
2828	1.44E-03	1.51	3.03E-03	1.77	6.87E-06	2.19	1.52E-05	2.94	2.19E-07	5.80	9.10E-07	4.76
11500	3.09E-04	2.19	8.14E-04	1.87	1.51E-06	2.16	2.59E-06	2.52	8.95E-09	4.56	1.75E-08	5.63
45248	1.02E-04	1.62	2.35E-04	1.81	1.98E-07	2.97	3.25E-07	3.03	2.95E-10	4.98	5.52E-10	5.05
177880	3.25E-05	1.67	8.43E-05	1.50	2.25E-08	3.18	3.82E-08	3.13	9.84E-12	4.97	4.62E-11	3.62
<b>ROD method with <math>\mathbb{P}_d/\mathbb{P}_{d+1}</math></b>												
736	5.44E-03	—	1.04E-02	—	3.03E-05	—	1.06E-04	—	8.15E-06	—	2.00E-05	—
2828	1.83E-03	1.62	3.08E-03	1.80	1.88E-06	4.13	1.24E-05	3.19	2.58E-07	5.13	9.67E-07	4.50
11500	4.09E-04	2.13	8.42E-04	1.85	4.30E-07	2.11	1.31E-06	3.20	2.54E-09	6.59	9.92E-09	6.53
45248	1.28E-04	1.70	2.47E-04	1.79	3.39E-08	3.71	8.74E-08	3.96	5.26E-11	5.66	2.76E-10	5.23
177880	3.91E-05	1.73	9.53E-05	1.39	5.55E-10	6.01	5.54E-09	4.03	1.73E-11	—	1.11E-10	—

The resulting geometry is called rose-shaped domain because of its resemblance to rose petals. In this test case, initial rays  $r_I = 0.5$  and  $r_E = 1$  and coefficients  $\alpha_I = 5$  and  $\alpha_E = 8$  are assigned. Notice that the annulus geometry is recovered assigning  $\alpha_I = \alpha_E = 0$ . Outward unit normal vectors on the inner and outer boundaries,  $\mathbf{n}_I := \mathbf{n}_I(\theta)$  and  $\mathbf{n}_E := \mathbf{n}_E(\theta)$ , respectively, are obtained applying the chain rule and are given by

$$\mathbf{n}_I(\theta) = \frac{1}{(R_I(\theta))^2 + (\partial_\theta R_I(\theta))^2} \begin{bmatrix} -R_I(\theta) \cos(\theta) - \partial_\theta R_I(\theta) \sin(\theta) \\ -R_I(\theta) \sin(\theta) + \partial_\theta R_I(\theta) \cos(\theta) \end{bmatrix}, \quad (44)$$

$$\mathbf{n}_E(\theta) = \frac{-1}{(R_E(\theta))^2 + (\partial_\theta R_E(\theta))^2} \begin{bmatrix} -R_E(\theta) \cos(\theta) - \partial_\theta R_E(\theta) \sin(\theta) \\ -R_E(\theta) \sin(\theta) + \partial_\theta R_E(\theta) \cos(\theta) \end{bmatrix}, \quad (45)$$

where  $\partial_\theta R_I(\theta)$  and  $\partial_\theta R_E(\theta)$  are the derivatives of  $R_I(\theta)$  and  $R_E(\theta)$  in order to variable  $\theta$ , respectively.

For this domain, the manufactured solution (see Figure 11(a)) is expressed in terms of polar coordinates as

$$\phi(r, \theta) = a \left( \exp(R(r, \theta)) + \exp(-R(r, \theta)) + b \right) + 1, \quad R(r, \theta) = \frac{2r - (R_E(\theta) + r_I)}{R_E(\theta) - r_I}, \quad (46)$$

where  $r^2 = x^2 + y^2$  such that  $R \in [-1, 1]$ , and coefficients  $a, b \in \mathbb{R}$ . Notice that, although the solution is given in terms of  $r$  and  $\theta$ , the problem is numerically solved using the Cartesian geometry. In order to guarantee the property  $\phi(\mathbf{x}) \in [1, 2]$  in  $\Omega$ , the coefficients are deduced as  $a = 1/(2 - \exp(1) - \exp(-1))$  and  $b = -\exp(1) - \exp(-1)$ . The associated source term function is obtained after substituting manufactured solution (46) into Equation (1) (see Figure 11(b)). A convective-diffusive situation is considered setting a constant diffusion coefficient  $\kappa = 1$  and a constant radial velocity field  $\mathbf{u}(r, \theta) = (\cos(\theta), \sin(\theta))$ . Inner boundary  $\Gamma_I$  is prescribed with a nonconstant Dirichlet boundary condition, while outer boundary  $\Gamma_E$  is prescribed with either: a constant Dirichlet boundary condition (Dirichlet–Dirichlet case), a nonconstant Neumann one (Dirichlet–Neumann case), or a nonconstant Robin one (Dirichlet–Robin case) with coefficients  $\sigma(\theta) = (\theta + \pi)(\theta - \pi)/\pi^2$  and  $\rho(\theta) = 1 - \sigma(\theta)$  given in terms of polar coordinates. The boundary conditions functions computed from the analytical solution are represented by long expressions and therefore, for the sake of simplicity, are omitted from this article.

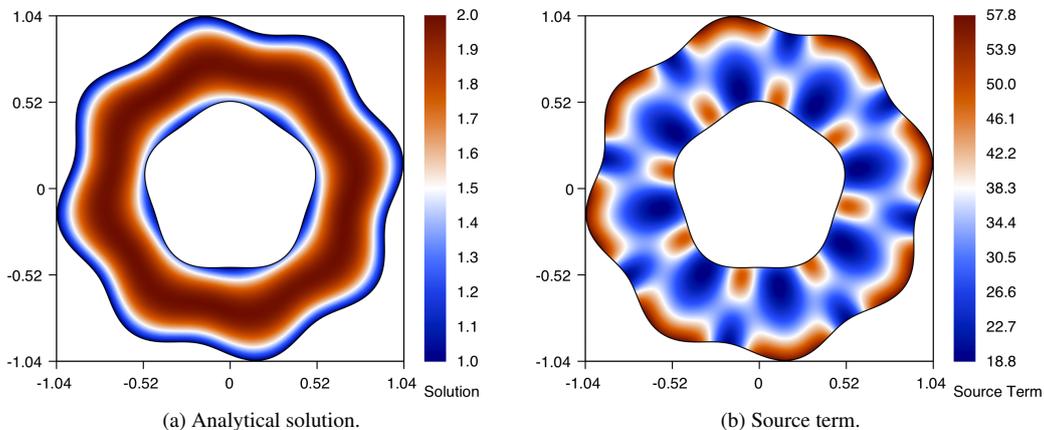


Figure 11: Analytical solution and source term in a very fine mesh for the rose-shaped domain (for interpretation of the color in this figure, the reader is referred to the electronic version of this article).

All the simulations were carried out either with successive refined uniform triangular Delaunay or quadrilateral meshes generated for the rose-shaped domain (see Figure 12(a) and 12(b)). Observe that the boundary vertices belong to the physical boundary and the maximum gap between the physical and the computational boundaries is of order  $\mathcal{O}(h^2)$  (see Figure 12(c)). The ROD and AROD methods are tested while the Naive method is not considered here since it can not provide more than a second-order convergence rate. When Neumann or Robin boundary conditions are prescribed, then  $\mathbb{P}_d/\mathbb{P}_{d+1}$ ,  $d \in \{1, 3, 5\}$ , polynomial reconstructions are used to achieve the optimal orders of accuracy, as proceed previously.

*Dirichlet–Dirichlet case.* Both boundaries  $\Gamma_I$  and  $\Gamma_E$  are prescribed with a Dirichlet boundary condition and Table 9 reports the errors and corresponding convergence rates obtained from the ROD and AROD methods when uniform triangular Delaunay meshes are used. As observed, the optimal convergence orders are obtained for this complex geometry and the ROD and AROD methods perform very similarly in terms of accuracy.

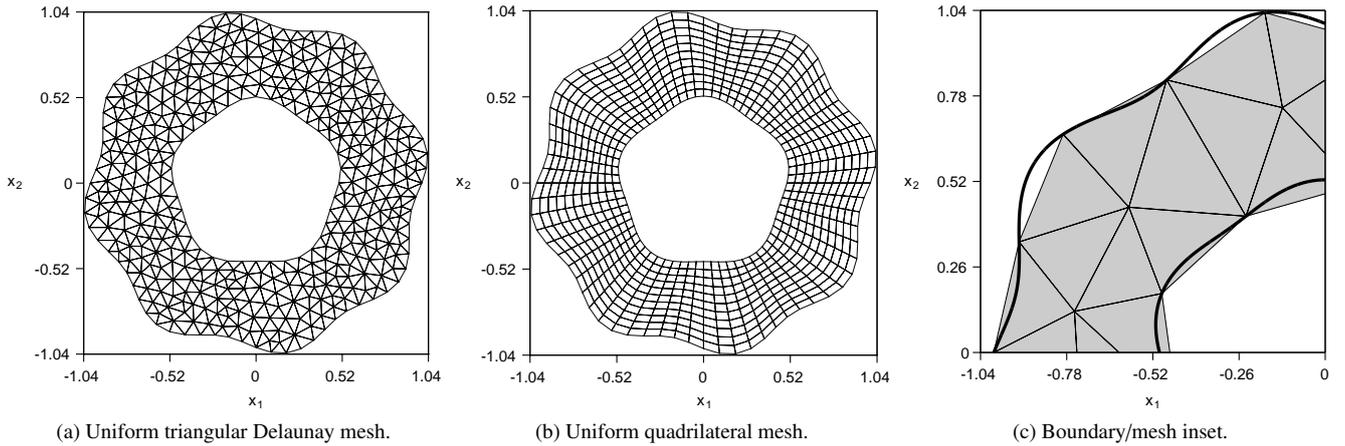


Figure 12: Coarse uniform triangular Delaunay and quadrilateral meshes and inset of a very coarse mesh (gray filling) and the curved physical boundaries (thick lines) for the rose-shaped domain.

To investigate the AROD–DN and AROD–DR methods, the least-squares constraint to compute the polynomial reconstructions associated to the boundary edges, is considered to be of Neumann and Robin types, respectively, while the fitting condition remains of Dirichlet type. Only the boundary edges associated to the outer boundary are treated with the AROD–DN or AROD–DR methods while those associated to the inner boundary are treated with the AROD method. When the AROD–DN and AROD–DR methods are used, no residual convergence was obtained with the last two meshes and  $d = 5$  for the applied stopping criteria. On the other hand, for the cases where residual convergence is achieved, the AROD–DN and AROD–DR methods provide similar accuracy compared to the ROD and AROD methods.

To deeper investigate this issue, Table 10 reports the GMRES iterations count in the ROD, AROD, AROD–DN, and AROD–DR methods. A comparable number of GMRES iterations between the ROD and AROD methods is always observed regardless of the polynomial degree or mesh used. However, when the polynomial degree increases and the mesh characteristic length decreases, the AROD–DN and AROD–DR methods provide a larger number of iterations. Such results can be explained since for each boundary edge, a polynomial reconstruction with a least-squares constraint of Neumann or Robin types, defined in terms of gradient  $\nabla\varphi(\mathbf{x})$ , is used to satisfy a fitting condition of Dirichlet type, defined in terms of function  $\phi(\mathbf{x})$ . Such choice increases the condition number of the implicit system of linear equations, slowing down or destroying the residual convergence.

*Dirichlet–Neumann case.* Outer boundary  $\Gamma_E$  is now prescribed with a Neumann boundary condition and Table 11 reports the errors and corresponding convergence rates obtained from ROD and AROD methods with  $\mathbb{P}_d/\mathbb{P}_{d+1}$ ,  $d \in \{1, 3, 5\}$ , polynomial reconstructions and uniform triangular Delaunay meshes. As previously observed, the optimal convergence orders are obtained for this complex geometry and the ROD and AROD methods perform very similarly, but for the latter one no residual convergence was obtained with the last meshes and  $d = 5$ .

Similarly to the previous test case, to investigate the AROD–ND and AROD–NR methods, the least-squares constraint to compute the polynomial reconstructions associated to the boundary edges, is considered to be of Dirichlet and Robin types, respectively, while the fitting condition is of Neumann type. Only the boundary edges associated to the outer boundary are treated with the AROD–ND or AROD–NR methods. As for the AROD method, when the AROD–NR method is applied no residual convergence was obtained with the last mesh and  $d = 5$  although, for the cases where residual convergence is achieved, similar accuracy is obtained compared to the ROD and AROD methods. Moreover, the residual convergence is achieved with all meshes and degrees when the AROD–ND method is used.

In Table 12, the GMRES iterations counts are reported. The results confirm that different number of iterations are counted when the polynomial degree increases or the mesh characteristic length decreases. Concerning to the AROD–NR method, the number of GMRES iterations to achieve residual convergence is comparable to the results obtained in the AROD method. The AROD–ND method GMRES iterations counts are always comparable to the ROD method ones and smaller than for the AROD and AROD–NR methods. In conclusion, considering a least-squares constraint of Dirichlet type instead of Neumann or Robin types, seems to perform better and provides better conditioning, even when the fitting and the boundary conditions are of Neumann type.

*Dirichlet–Robin case.* Last, outer boundary  $\Gamma_E$  is prescribed with a Robin boundary condition and Table 13 reports the errors and corresponding convergence rates obtained from the ROD and AROD methods with  $\mathbb{P}_d/\mathbb{P}_{d+1}$ ,  $d \in \{1, 3, 5\}$ ,

Table 9: Convective-diffusive case with Dirichlet–Dirichlet boundary conditions – relative errors and convergence rates obtained from the ROD, AROD, AROD–DN, and AROD–DR methods with uniform triangular Delaunay meshes.

DOF	$d = 1$				$d = 3$				$d = 5$			
	$E_1$	$O_1$	$E_\infty$	$O_\infty$	$E_1$	$O_1$	$E_\infty$	$O_\infty$	$E_1$	$O_1$	$E_\infty$	$O_\infty$
<b>ROD method with <math>\mathbb{P}_d</math></b>												
750	7.18E-03	—	1.31E-02	—	7.61E-05	—	4.95E-04	—	3.64E-05	—	3.00E-04	—
3144	1.85E-03	1.89	3.76E-03	1.75	5.03E-06	3.79	3.55E-05	3.68	9.28E-07	5.12	1.62E-05	4.07
12482	4.77E-04	1.97	1.15E-03	1.71	2.94E-07	4.12	3.46E-06	3.38	1.60E-08	5.89	4.04E-07	5.36
50102	1.15E-04	2.05	3.03E-04	1.92	2.05E-08	3.84	2.71E-07	3.66	2.85E-10	5.80	1.22E-08	5.03
199636	2.91E-05	1.99	8.68E-05	1.81	1.28E-09	4.01	1.88E-08	3.86	3.89E-12	6.21	1.75E-10	6.14
<b>AROD method with <math>\mathbb{P}_d</math></b>												
750	7.19E-03	—	1.32E-02	—	7.51E-05	—	4.94E-04	—	3.57E-05	—	3.00E-04	—
3144	1.85E-03	1.89	3.77E-03	1.75	5.02E-06	3.78	3.51E-05	3.69	9.20E-07	5.10	1.61E-05	4.09
12482	4.77E-04	1.97	1.16E-03	1.71	2.94E-07	4.12	3.46E-06	3.36	1.60E-08	5.87	4.04E-07	5.34
50102	1.15E-04	2.05	3.03E-04	1.93	2.05E-08	3.83	2.71E-07	3.66	2.85E-10	5.80	1.22E-08	5.04
199636	2.91E-05	1.99	8.68E-05	1.81	1.28E-09	4.00	1.88E-08	3.86	3.93E-12	6.20	1.75E-10	6.14
<b>AROD–DN method with <math>\mathbb{P}_d</math></b>												
750	6.58E-03	—	1.24E-02	—	6.97E-05	—	4.79E-04	—	3.48E-05	—	2.91E-04	—
3144	1.67E-03	1.92	3.49E-03	1.76	4.92E-06	3.70	3.29E-05	3.74	8.84E-07	5.13	1.52E-05	4.11
12482	4.31E-04	1.96	1.07E-03	1.72	2.81E-07	4.15	3.27E-06	3.35	1.67E-08	5.76	4.28E-07	5.18
50102	1.03E-04	2.06	2.71E-04	1.97	1.97E-08	3.82	2.57E-07	3.66	—	—	—	—
199636	2.63E-05	1.98	7.74E-05	1.82	1.27E-09	3.97	1.91E-08	3.76	—	—	—	—
<b>AROD–DR method with <math>\mathbb{P}_d</math></b>												
750	6.58E-03	—	1.24E-02	—	6.97E-05	—	4.79E-04	—	3.48E-05	—	2.91E-04	—
3144	1.67E-03	1.92	3.50E-03	1.76	4.92E-06	3.70	3.29E-05	3.74	8.84E-07	5.13	1.52E-05	4.11
12482	4.31E-04	1.96	1.07E-03	1.72	2.81E-07	4.15	3.27E-06	3.35	1.67E-08	5.76	4.28E-07	5.18
50102	1.03E-04	2.06	2.71E-04	1.97	1.97E-08	3.82	2.57E-07	3.66	—	—	—	—
199636	2.63E-05	1.98	7.74E-05	1.82	1.27E-09	3.97	1.91E-08	3.76	—	—	—	—

Table 10: Convective-diffusive case with Dirichlet–Dirichlet boundary conditions – GMRES iterations counts in the ROD, AROD, AROD–DN, and AROD–DR methods with  $\mathbb{P}_d$  reconstructions and uniform triangular Delaunay meshes.

DOF	$d = 1$				$d = 3$				$d = 5$			
	ROD	AROD	AROD–DN	AROD–DR	ROD	AROD	AROD–DN	AROD–DR	ROD	AROD	AROD–DN	AROD–DR
750	80	40	80	80	80	80	120	120	120	80	280	240
3144	80	80	120	120	120	120	200	200	200	160	400	400
12482	120	120	200	200	200	200	440	440	440	320	1360	1440
50102	240	240	400	400	440	440	920	920	600	600	NC	NC
199636	520	520	800	800	960	960	2240	2240	1200	1160	NC	NC

Table 11: Convective-diffusive case with Dirichlet–Neumann boundary conditions – relative errors and convergence rates obtained from the ROD, AROD, AROD–ND, and AROD–NR methods with uniform triangular Delaunay meshes.

DOF	$d = 1$				$d = 3$				$d = 5$			
	$E_1$	$O_1$	$E_\infty$	$O_\infty$	$E_1$	$O_1$	$E_\infty$	$O_\infty$	$E_1$	$O_1$	$E_\infty$	$O_\infty$
<b>ROD method with <math>\mathbb{P}_d/\mathbb{P}_{d+1}</math></b>												
750	3.33E-03	—	8.50E-03	—	9.41E-05	—	5.36E-04	—	7.51E-05	—	5.35E-04	—
3144	9.34E-04	1.77	2.72E-03	1.59	5.29E-06	4.02	3.66E-05	3.74	1.73E-06	5.27	1.63E-05	4.87
12482	2.31E-04	2.02	8.20E-04	1.74	3.83E-07	3.81	3.44E-06	3.43	2.58E-08	6.10	4.04E-07	5.36
50102	5.78E-05	2.00	2.28E-04	1.84	3.98E-08	3.26	2.72E-07	3.65	5.07E-10	5.66	1.22E-08	5.04
199636	1.52E-05	1.93	8.29E-05	1.46	3.00E-09	3.74	1.88E-08	3.86	6.99E-12	6.20	1.75E-10	6.14
<b>AROD method with <math>\mathbb{P}_d/\mathbb{P}_{d+1}</math></b>												
750	3.30E-03	—	8.40E-03	—	9.38E-05	—	5.34E-04	—	7.57E-05	—	5.40E-04	—
3144	9.30E-04	1.77	2.71E-03	1.58	5.29E-06	4.01	3.65E-05	3.74	1.72E-06	5.28	1.62E-05	4.90
12482	2.31E-04	2.02	8.20E-04	1.73	3.82E-07	3.81	3.45E-06	3.42	2.59E-08	6.09	4.05E-07	5.35
50102	5.77E-05	2.00	2.27E-04	1.85	3.98E-08	3.25	2.72E-07	3.66	5.06E-10	5.66	1.22E-08	5.04
199636	1.52E-05	1.93	8.28E-05	1.46	3.01E-09	3.73	1.88E-08	3.86	—	—	—	—
<b>AROD–ND method with <math>\mathbb{P}_d/\mathbb{P}_{d+1}</math></b>												
750	3.31E-03	—	8.40E-03	—	1.04E-04	—	5.83E-04	—	7.76E-05	—	5.37E-04	—
3144	9.34E-04	1.77	2.71E-03	1.58	5.83E-06	4.02	4.02E-05	3.73	1.74E-06	5.30	1.62E-05	4.89
12482	2.32E-04	2.02	8.20E-04	1.73	3.57E-07	4.05	3.44E-06	3.57	2.53E-08	6.13	4.05E-07	5.35
50102	5.80E-05	1.99	2.27E-04	1.85	2.45E-08	3.86	2.71E-07	3.66	5.20E-10	5.59	1.22E-08	5.04
199636	1.53E-05	1.93	8.29E-05	1.46	1.66E-09	3.89	1.88E-08	3.86	6.82E-12	6.27	1.75E-10	6.14
<b>AROD–NR method with <math>\mathbb{P}_d/\mathbb{P}_{d+1}</math></b>												
750	3.30E-03	—	8.40E-03	—	9.38E-05	—	5.34E-04	—	7.57E-05	—	5.40E-04	—
3144	9.30E-04	1.77	2.71E-03	1.58	5.29E-06	4.01	3.65E-05	3.74	1.72E-06	5.28	1.62E-05	4.90
12482	2.31E-04	2.02	8.20E-04	1.73	3.82E-07	3.81	3.45E-06	3.42	2.59E-08	6.09	4.05E-07	5.35
50102	5.77E-05	2.00	2.27E-04	1.85	3.98E-08	3.25	2.72E-07	3.66	5.06E-10	5.66	1.22E-08	5.04
199636	1.52E-05	1.93	8.28E-05	1.46	3.01E-09	3.73	1.88E-08	3.86	—	—	—	—

Table 12: Convective-diffusive case with Dirichlet–Neumann boundary conditions – GMRES iterations counts in the ROD, AROD, AROD–ND, and AROD–NR methods with  $\mathbb{P}_d/\mathbb{P}_{d+1}$  reconstructions and uniform triangular Delaunay meshes.

DOF	$d = 1$				$d = 3$				$d = 5$			
	ROD	AROD	AROD–ND	AROD–NR	ROD	AROD	AROD–ND	AROD–NR	ROD	AROD	AROD–ND	AROD–NR
750	80	80	80	80	120	80	120	80	160	200	200	200
3144	120	120	120	120	240	200	200	200	280	280	320	280
12482	240	240	240	240	360	360	400	360	720	920	640	920
50102	440	440	440	440	800	840	800	840	1480	4520	1240	4640
199636	840	840	800	840	1880	1960	1800	1960	5600	NC	3240	NC

polynomial reconstructions and uniform triangular Delaunay meshes. The results are in accordance with the ones obtained for the previous Dirichlet–Dirichlet and Dirichlet–Neumann cases, achieving the optimal convergence order for the ROD and AROD methods with comparable accuracy.

Table 13: Convective-diffusive case with Dirichlet–Robin boundary conditions – relative errors and convergence rates obtained from the ROD, AROD, AROD–RD, and AROD–RN methods with uniform triangular Delaunay meshes.

DOF	$d = 1$				$d = 3$				$d = 5$			
	$\bar{E}_1$	$O_1$	$\bar{E}_\infty$	$O_\infty$	$\bar{E}_1$	$O_1$	$\bar{E}_\infty$	$O_\infty$	$\bar{E}_1$	$O_1$	$\bar{E}_\infty$	$O_\infty$
<b>ROD method with <math>\mathbb{P}_d/\mathbb{P}_{d+1}</math></b>												
750	3.50E-03	—	8.55E-03	—	8.71E-05	—	4.95E-04	—	6.89E-05	—	4.86E-04	—
3144	9.64E-04	1.80	2.72E-03	1.60	5.00E-06	3.99	3.50E-05	3.70	1.60E-06	5.25	1.63E-05	4.74
12482	2.38E-04	2.03	8.21E-04	1.74	3.69E-07	3.78	3.44E-06	3.36	2.44E-08	6.07	4.04E-07	5.36
50102	5.87E-05	2.02	2.28E-04	1.84	3.68E-08	3.32	2.72E-07	3.66	4.67E-10	5.70	1.22E-08	5.04
199636	1.54E-05	1.94	8.15E-05	1.49	2.71E-09	3.77	1.88E-08	3.86	1.63E-11	4.85	1.75E-10	6.14
<b>AROD method with <math>\mathbb{P}_d/\mathbb{P}_{d+1}</math></b>												
750	3.47E-03	—	8.45E-03	—	8.65E-05	—	4.93E-04	—	6.93E-05	—	4.91E-04	—
3144	9.60E-04	1.79	2.71E-03	1.59	5.00E-06	3.98	3.49E-05	3.69	1.59E-06	5.27	1.62E-05	4.76
12482	2.38E-04	2.02	8.21E-04	1.73	3.68E-07	3.78	3.45E-06	3.36	2.45E-08	6.06	4.05E-07	5.35
50102	5.86E-05	2.02	2.28E-04	1.85	3.67E-08	3.32	2.72E-07	3.66	4.65E-10	5.70	1.22E-08	5.04
199636	1.50E-05	1.97	8.13E-05	1.49	2.70E-09	3.77	1.88E-08	3.86	—	—	—	—
<b>AROD–RD method with <math>\mathbb{P}_d/\mathbb{P}_{d+1}</math></b>												
750	3.47E-03	—	8.45E-03	—	9.48E-05	—	5.37E-04	—	7.08E-05	—	4.88E-04	—
3144	9.61E-04	1.79	2.71E-03	1.59	5.37E-06	4.01	3.83E-05	3.69	1.61E-06	5.28	1.61E-05	4.76
12482	2.38E-04	2.02	8.21E-04	1.73	3.45E-07	3.98	3.44E-06	3.50	2.41E-08	6.10	4.05E-07	5.35
50102	5.87E-05	2.02	2.28E-04	1.85	2.38E-08	3.85	2.71E-07	3.66	4.77E-10	5.64	1.22E-08	5.04
199636	1.51E-05	1.97	8.15E-05	1.49	1.58E-09	3.93	1.88E-08	3.86	6.31E-12	6.26	1.75E-10	6.14
<b>AROD–RN method with <math>\mathbb{P}_d/\mathbb{P}_{d+1}</math></b>												
750	3.47E-03	—	8.45E-03	—	8.64E-05	—	4.92E-04	—	6.93E-05	—	4.91E-04	—
3144	9.60E-04	1.79	2.71E-03	1.59	5.00E-06	3.98	3.49E-05	3.69	1.59E-06	5.27	1.62E-05	4.76
12482	2.38E-04	2.02	8.21E-04	1.73	3.68E-07	3.78	3.45E-06	3.36	2.45E-08	6.06	4.05E-07	5.35
50102	5.86E-05	2.02	2.28E-04	1.85	3.67E-08	3.32	2.72E-07	3.66	4.66E-10	5.70	1.22E-08	5.04
199636	1.50E-05	1.97	8.13E-05	1.49	2.70E-09	3.77	1.88E-08	3.86	—	—	—	—

In Table 14 the GMRES iterations counts are reported. As already observed, when the polynomial degree increases or the mesh characteristic length decreases, the AROD–RD method performs always comparably to the ROD method whereas the AROD–RN and AROD methods require more GMRES iterations to achieve a residual convergence.

To conclude the analysis, the ROD, AROD, and AROD–XY methods provide comparable accuracy although differences in terms of residual convergence are reported when the polynomial degree increases or the mesh characteristic length decreases. For Dirichlet boundary conditions, the GMRES method with the ROD and AROD methods perform better compared with the AROD–DN and AROD–DR methods, which require in general a larger number of iterations to converge. For Neumann and Robin boundary conditions, the ROD and AROD–ND/AROD–RD methods perform better compared with the AROD and AROD–NR/AROD–RN methods, which also require in general more iterations to converge. Given these statements, the ROD method provides in general the first or second best performance in terms of residual convergence regardless of the boundary condition. On the other hand, for the AROD and AROD–XY methods, the polynomial reconstructions associated to the boundary edges provide better residual convergence constraining the associated least-squares problem with a Dirichlet type equation, regardless of the type of the boundary condition. In other words, the AROD, AROD–ND, and AROD–RD for the cases of a Dirichlet, Neumann, and Robin boundary conditions, respectively, perform better compared with the others.

*Quadrilateral meshes.* To conclude this benchmark, the previous Dirichlet–Dirichlet, Dirichlet–Neumann, and Dirichlet–Robin cases for the rose-shaped domain, are simulated with uniform quadrilateral meshes. Only the ROD method with  $\mathbb{P}_d$  or  $\mathbb{P}_d/\mathbb{P}_{d+1}$ ,  $d \in \{1, 3, 5\}$ , polynomial reconstructions is addressed whereas the other methods (AROD and AROD–XY) perform as observed previously for uniform triangular meshes. The obtained errors and corresponding convergence rates are reported in Table 15. The results support the capability of the method to achieve the optimal convergence rates with quadrilateral meshes and with an essentially non-oscillatory behavior.

Table 14: Convective-diffusive case with Dirichlet–Robin boundary conditions – GMRES iterations counts in the ROD, AROD, AROD–RD, and AROD–RN methods with  $\mathbb{P}_d/\mathbb{P}_{d+1}$  reconstructions and uniform triangular Delaunay meshes.

<i>DOF</i>	<i>d</i> = 1				<i>d</i> = 3				<i>d</i> = 5			
	ROD	AROD	AROD–RD	AROD–RN	ROD	AROD	AROD–RD	AROD–RN	ROD	AROD	AROD–RD	AROD–RN
750	80	80	80	80	80	80	120	80	160	200	160	200
3144	200	120	120	120	200	160	200	160	280	280	280	280
12482	200	200	240	200	320	360	360	360	1080	840	600	880
50102	440	400	440	400	680	800	760	800	1440	4320	1200	4280
199636	840	960	960	960	1600	2040	1880	2040	4640	NC	2840	NC

Table 15: Convective-diffusive case with Dirichlet–Dirichlet, Dirichlet–Neumann, and Dirichlet–Robin boundary conditions – relative errors and convergence rates obtained from the ROD method with  $\mathbb{P}_d/\mathbb{P}_{d+1}$  polynomial reconstructions and uniform quadrilateral meshes.

<i>DOF</i>	<i>d</i> = 1				<i>d</i> = 3				<i>d</i> = 5			
	$\bar{E}_1$	$O_1$	$E_\infty$	$O_\infty$	$\bar{E}_1$	$O_1$	$E_\infty$	$O_\infty$	$\bar{E}_1$	$O_1$	$E_\infty$	$O_\infty$
<b>Dirichlet–Dirichlet case</b>												
748	6.89E–03	—	1.19E–02	—	2.08E–04	—	7.85E–04	—	1.28E–04	—	7.96E–04	—
3128	1.71E–03	1.94	3.15E–03	1.85	1.77E–05	3.44	7.87E–05	3.22	5.03E–06	4.52	7.17E–05	3.36
12784	4.24E–04	1.99	7.97E–04	1.95	1.28E–06	3.74	5.41E–06	3.80	9.23E–08	5.68	1.28E–06	5.72
52224	1.04E–04	1.99	1.98E–04	1.98	8.18E–08	3.90	3.31E–07	3.97	1.33E–09	6.02	1.64E–08	6.19
209984	2.61E–05	1.99	4.96E–05	1.99	5.05E–09	4.00	2.10E–08	3.96	2.63E–11	5.64	2.66E–10	5.92
<b>Dirichlet–Neumann case</b>												
748	2.27E–03	—	9.47E–03	—	4.79E–04	—	2.78E–03	—	2.70E–04	—	2.05E–03	—
3128	5.88E–04	1.89	2.47E–03	1.88	6.41E–05	2.81	2.73E–04	3.24	4.34E–05	2.56	1.59E–04	3.58
12784	1.51E–04	1.93	6.30E–04	1.94	4.76E–06	3.69	2.20E–05	3.58	1.28E–06	5.01	3.47E–06	5.43
52224	3.79E–05	1.96	1.56E–04	1.98	2.59E–07	4.14	1.91E–06	3.48	5.02E–09	7.87	4.19E–08	6.28
209984	9.41E–06	2.00	3.99E–05	1.96	1.77E–08	3.86	1.91E–07	3.31	3.43E–11	7.17	6.65E–10	5.95
<b>Dirichlet–Robin case</b>												
748	2.36E–03	—	8.76E–03	—	4.33E–04	—	2.67E–03	—	2.62E–04	—	2.00E–03	—
3128	6.10E–04	1.89	2.31E–03	1.86	5.59E–05	2.86	2.64E–04	3.23	3.66E–05	2.75	1.47E–04	3.64
12784	1.56E–04	1.94	5.90E–04	1.94	4.15E–06	3.69	2.10E–05	3.60	1.07E–06	5.02	3.28E–06	5.41
52224	3.90E–05	1.97	1.48E–04	1.97	2.33E–07	4.09	1.88E–06	3.43	4.39E–09	7.81	4.03E–08	6.25
209984	9.61E–06	2.01	3.77E–05	1.97	1.68E–08	3.77	1.86E–07	3.32	3.54E–11	6.93	6.55E–10	5.92

## 7. Conclusions

A very high-order finite volume scheme for the steady-state two-dimensional convection-diffusion problem with curved boundaries has been constructed based on a new framework to compute polynomial reconstructions. The scheme is equipped with the *Reconstruction for Off-site Data* method (shortened to ROD method) to handle complex curved boundaries prescribed with general boundary conditions, such as the Dirichlet, Neumann, and Robin. The method does not require curved elements to fit the boundary, nor complex quadrature rules for integration, nor complex nonlinear transformations, which represents an important achievement to the existing techniques in the Finite Volume context. An *Adaptive Reconstruction for Off-site Data* method (AROD and its variants AROD-XY methods, were also proposed in order to handle evolving boundaries (with respect to time or an iterative process). This approach avoids recomputing the least-squares procedure for small boundary displacements, avoiding in fact a waste of computer resources. A comprehensive numerical benchmark test suit, using the method of manufactured solutions, validated the implementation and did assess the performance of the methods with simple and complex curved domains. The following conclusions were drawn:

- Very high accurate convergence rates were achieved for the two-dimensional steady-state pure convection, pure diffusion, and convection-diffusion problems, either with triangular or quadrilateral meshes;
- To achieve the optimal convergence rates when Neumann and Robin boundary conditions are prescribed,  $\mathbb{P}_{d+1}$  polynomial reconstructions are required for the associated boundary edges while  $\mathbb{P}_d$  polynomial reconstructions can be used for the remaining boundary and inner edges and cells; since only  $\mathbb{P}_d$  polynomial reconstructions are required to treat Dirichlet boundary conditions, a performance penalty is expected when treating Neumann and Robin boundary conditions; in practice, such penalty is relatively small as it only affects the polynomial reconstructions associated to the Neumann or Robin boundary edges; for the three-dimensional case, such penalty would be negligible as the ratio of the number of boundary faces by the number of inner faces and cells decreases, compared to the two-dimensional case;
- Both the ROD, AROD, and AROD-XY methods provide comparable and optimal convergence rates when dealing with Dirichlet boundary conditions and no oscillations or instabilities were reported;  $\mathbb{P}_d$  polynomial reconstructions for the edges and cells provide  $(d+1)$ th-order accurate convergence rates under mesh refinement;
- Although all the methods provide comparable accuracy, a performance degradation, in terms of GMRES convergence, occurs in some cases when the polynomial reconstructions degree increases and the characteristic mesh length decreases; to summarize, the ROD method (regardless of the boundary condition) and the AROD, AROD-ND, and AROD-RD methods (for Dirichlet, Neumann, and Robin boundary conditions, respectively) have the best performance.

The future plans within this research include to apply the proposed ROD and AROD methods to complex systems (*e.g.* Navier-Stokes equations and Euler equations) in three-dimensional geometries with structured and unstructured meshes. Additionally, unsteady problems with time evolving domains (*e.g.* piston problems, pulsating interfaces problems, and tracking interfaces problems) will be investigated aiming towards complex physical and industrial applications. Moreover, code and algorithms performance improvements, validation, and verification will be required.

## Acknowledgements

R. Costa acknowledges the financial support by Portugal 2020 and FSE – Fundo Social Europeu, through NORTE 2020 – Programa Operacional Regional do Norte, project no. NORTE-08-5369-FSE-000034, under program IMPULSE – Polímeros e Compósitos: Drivers da Inovação Tecnológica e da Competitividade Industrial. R. Costa and J.M. Nóbrega acknowledge the financial support by FEDER – Fundo Europeu de Desenvolvimento Regional, through COMPETE 2020 – Programa Operacional Fatores de Competitividade, and the National Funds through FCT– Fundação para a Ciência e a Tecnologia, project no. UID/CTM/50025/2013. S. Clain and G.J. Machado acknowledge the financial support by FEDER – Fundo Europeu de Desenvolvimento Regional, through COMPETE 2020 – Programa Operacional Fatores de Competitividade, and the National Funds through FCT – Fundação para a Ciência e a Tecnologia, project no. UID/FIS/04650/2013. R. Costa, J.M. Nóbrega, S. Clain and G.J. Machado acknowledge the financial support by FEDER – Fundo Europeu de Desenvolvimento Regional, through COMPETE 2020 – Programa Operacional Fatores de Competitividade, and the National Funds through FCT – Fundação para a Ciência e a Tecnologia, project no. POCI-01-0145-FEDER-028118.

## References

- [1] Costa R, Clain S, Loubère R, Machado GJ. Very high-order accurate finite volume scheme on curved boundaries for the two-dimensional steady-state convection-diffusion equation with Dirichlet condition. *Applied Mathematical Modeling* 2018; **54**:752-767.
- [2] Lehrenfeld C. High order unfitted finite element methods on level set domains using isoparametric mappings. *Computer Methods in Applied Mechanics and Engineering* 2016; **300**:716-733.
- [3] Lehrenfeld C, Reusken A. Analysis of a high order unfitted finite element method for elliptic interface problems. *IMA Journal on Numerical Analysis*; drx041.
- [4] Ollivier-Gooch C, Van Alena M. A high-order accurate unstructured mesh finite-volume scheme for the advection-diffusion equation. *Journal of Computational Physics* 2002; **181**(2):729-752.
- [5] Geuzaine C, Remacle JF. Gmsh: a three-dimensional Finite Element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering* 2009; **79**:1309-1331.
- [6] Wang ZJ. High-order computational fluid dynamics tools for aircraft design. *Philosophical Transactions of the Royal Society A* 2014; **372**:20130318.
- [7] Moxey D, Green MD, Sherwin SJ, Peiró J. An isoparametric approach to high-order curvilinear boundary-layer meshing. *Computer Methods in Applied Mechanics and Engineering* 2015; **283**:636-650.
- [8] Stein DB, Guy RD, Thomases B. Immersed boundary smooth extension: A high-order method for solving PDE on arbitrary smooth domains using Fourier spectral methods. *Journal of Computational Physics* 2016; **304**:252-274.
- [9] Stein DB, Guy RD, Thomases B. Immersed Boundary Smooth Extension (IBSE): A high-order method for solving incompressible flows in arbitrary smooth domains. *Journal of Computational Physics* 2017; **335**:155-178.
- [10] Cournede P-H, Debiez C, Dervieux A. A positive MUSCL scheme for triangulations. *INRIA Report* 1998; 3465.
- [11] Barth TJ, Frederickson PO. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. *AIAA Paper* 1990; 90-0013.
- [12] Barth TJ. Recent developments in high order k-exact reconstruction on unstructured meshes. *AIAA Paper* 1993; 93-0668.
- [13] Leer B. Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second-order scheme. *Journal of Computational Physics* 1974; **14**(4):361-370.
- [14] Venkatakrishnan V. On the Accuracy of Limiters and Convergence to Steady-State Solutions. *AIAA paper* 1993; 93-0880.
- [15] Hubbard ME. Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids. *Journal of Computational Physics* 1999; **155**:54-74.
- [16] Abgrall R. On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. *Journal of Computational Physics* 1994; **114**:45-58.
- [17] Balsara D, Shu C-W. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *Journal of Computational Physics* 2000; **160**:405-452.
- [18] Dumbser M, Käser M. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics* 2007; **221**:693-723.
- [19] Shu C-W. High-order weighted essentially non-oscillatory schemes for convection dominated problems. *SIAM Review* 2009; **51**:82-126.
- [20] Ollivier-Gooch C. High-Order ENO schemes for unstructured meshes based on least-squares reconstruction. *AIAA Paper* 1997; 97-0540.
- [21] Qiu J, Shu C-W. Runge-Kutta discontinuous Galerkin method using WENO limiters. *SIAM Journal on Scientific Computing* 2005; **26**:907-929.
- [22] Abgrall R, Sonar T. On the use of Mühlbach expansions in the recovery step of ENO methods. *Numerische Mathematik* 1997; **76**:1-25.
- [23] Colella P, Sekora MD. A limiter for PPM that preserves accuracy at smooth extrema. *Journal of Computational Physics* 2008; **227**:7069-7076.
- [24] Kuzmin D. A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods. *Journal of Computational and Applied Mathematics* 2010; **233**(12):3077-3085.
- [25] Clain S, Machado GJ, Nóbrega JM, Pereira RMS. A sixth-order finite volume method for the convection-diffusion problem with discontinuous coefficients. *Computer Methods in Applied Mechanics and Engineering* 2013; **267**:43-64.
- [26] Barth TJ. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations. *AGARD Report* 1992; 787.
- [27] Boularas A, Clain S, Baudoin F. A sixth-order finite volume method for diffusion problem with curved boundaries. *Applied Mathematical Modelling* 2017; **42**:401-422.
- [28] Bertsekas DP. Constrained optimization and lagrange multiplier methods. 1st Edition. Academic Press; 1982.
- [29] Ollivier-Gooch C, Nejat A, Michalak C. On obtaining high-order finite-volume solutions to the Euler equations on unstructured meshes. *AIAA Paper* 2007; 2007-4464.
- [30] Ollivier-Gooch C, Nejat A, Michalak C. Obtaining and verifying high-order unstructured finite volume solutions to the Euler equations. *AIAA Journal* 2009; **47**(9):2105-2120.
- [31] Michalak C, Ollivier-Gooch C. Unstructured high-order accurate finite volume solutions of the Navier-Stokes equations. *AIAA Paper* 2009; 2009-954.
- [32] Ern A, Guermond J-L. Theory and practice of finite elements. 1st Edition. Springer; 2004.