RESEARCH ARTICLE

For submission to International Journal for Numerical Methods in Engineering

The shifted boundary collocation immersogeometric method: An application to Poisson problems

Vedrana Kozulić[1], Blaž Gotovac[1], Hrvoje Gotovac[1,*]
[1]University of Split, Faculty of Civil Engineering, Architecture and Geodesy, Matice hrvatske 15, 21000 Split, Croatia.
[*]Corresponding author: Tel: +38521303354, Fax: +38521465117
E-mails: kozulic@gradst.hr, gotovac@gradst.hr, hrvoje.gotovac@gradst.hr

Abstract

Despite the wide application and successful solving of many engineering and physical problems, classical mesh-based numerical methods, such as the finite element methods via the weak Galerkin formulation, suffer from complex and computationally expensive meshing procedures, especially for irregular geometries with holes, trimmed boundary surfaces, and singularities. The domain is usually defined by a boundary representation since a numerical analysis requires a volumetric conforming mesh. This gap has been bridged by IsoGeometric Analysis (IGA) linking CAD (usually NURBS) boundary representations with the numerical analysis on multipatch subdomains, eliminating the classical meshing step and enabling a meshless approach. However, this is true unless trimmed and complex boundary curves/surfaces are not present. In the last two decades, enormous attention has been given to immersed boundary methods that use background regular mesh, for geometry and solution approximation, which do not fit boundary representations, completely eliminating classical conforming meshing. A problem arises for cut finite elements due to the demand for numerical integration, the imposition of essential Dirichlet boundary conditions and the stabilization of the small cut elements or the support of the used basis functions. These drawbacks are removed with the application of the shifted boundary finite element method, where the cut elements are discarded from the analysis, but imposition of boundary conditions are replaced with internal surrogate boundaries. In this paper, we propose an alternative collocation shifted boundary method, which is a meshless approach without any numerical integrations. Moreover, the differential equation is satisfied for internal Greville collocation points on a regular background grid, and all the boundary conditions are satisfied only for the internal collocation points of surrogate internal boundaries via a Taylor series expansion or an equivalent 2D cut element polynomial approximation up to the order of the spline basis functions used. Therefore, the collocation procedure is very simple, utilizing only the internal collocation points and boundary point values of the Dirichlet or Neuman conditions, and presenting an immersogeometric type of analysis because the original CAD or another chosen boundary representation is retained. The methodology is demonstrated on 2-D Poisson examples, attaining the same convergence rate as IGA collocation, which proves that a shifted satisfaction of boundary conditions does not reduce the accuracy and efficiency of the proposed collocation procedure.

Keywords: immersogeometric analysis, immersed boundary method, collocation, shifted boundary method, 2-D Poisson problem.

1. INTRODUCTION

Modern industrial revolutions would have been impossible without the finite element method (FEM), which is the central numerical methodology for many engineering and scientific analyses but also is used in many industrial and applied software packages (see fundamentals in, among others, reviews of [1,2,3]). FEM, in its basic isoparametric form, uses the same set of Lagrangian basis functions (usually low first and second order) for geometry and solution descriptions, including the prescribed essential Dirichlet, natural Neumann or mixed Robin boundary conditions. The finite element mesh is conforming, i.e., it fits the boundary of the domain. In the case of complex irregular geometries containing many patches/subdomains with holes, trimmed boundary surfaces/curves or singularities, the meshing procedure requires 70-90% of the total processing time obtaining less accurate solutions and poorly described geometries. At the local level, finite elements become ill-conditioned due to their poorly degenerated shapes and practically singular finite element Jacobian matrix. Remeshing, refinements and other patched works are very intensive, or in some complex industrial real-life applications, almost impossible. This FEM bottleneck is widely recognized, although its application has solved many problems and has significantly advanced the computational and the overall sciences over the last five decades.

The success of FEM machinery lies in the methodology potential for many new variants and approaches. Among others, IsoGeometric Analysis (IGA) links CAD (usually NURBS) boundary domain representations with numerical analyses of multipatch subdomains (see [4, 5]). Essentially, the geometry domain is defined by its boundary representation (B-Rep), but the classical FEM solution requires a volumetric mesh fitting to the boundary. IGA uses spline-based basis functions for CAD description of geometry but also for the solutions defined on that geometry. In this way, the geometry is exactly defined in the CAD sense, especially for industrial applications such as in the avio or car industry, and a solution is defined on exact geometry, eliminating additional errors due to geometry approximations. Moreover, for instance, NURBS control points enable elimination of the classical meshing step and enable a meshless approach. The IGA becomes a new FEM with spline basis functions, enabling more accurate and continuous solutions, the application of high-order basis functions, easier adaptive procedures, no classical meshing and exact CAD geometry[4]. However, IGA extends the classical FEM problems from the local finite element to the subdomain patch level. In the classical CAD format, a patch is a subdomain described by curved quadrilaterals for 2D or cubes for 3D simulations. For complex domains with holes, trimmed boundary surfaces/curves

or singularities, many patches are needed; therefore, ill-conditioned patch geometry and interpatch solution continuity problems become similar to those in classical FEMs at the local level.

Immersed boundary methods (IBMs), first introduced by [6], are attractive alternatives to FEM or the IGA boundary fitting methods. The geometry and solution are divided such that the geometry is completely defined by B-Rep, while the solution is defined in the physical domain and immersed in a background regular/structured grid or mesh that is not fit with geometry B-Rep. The background mesh is easy to generate, which implies elimination of the classical FEM meshing preprocessing step. However, problems arise with cut finite elements (background finite elements cut by boundaries) due to the demand for numerical integration, the introduction of essential Dirichlet boundary conditions, and the stabilization of small cut elements or the support of used basis functions (among others, see discussions of [7,8]). It is very difficult to mention and review all the immersed methods, especially because in the last two decades, the number of IBM publications has drastically increased. Nevertheless, a few variants need to be emphasized.

At approximately the same time Rvachev extended the Kantorovich solution for Dirichlet boundary conditions and developed the so-called Rvachev solution structure method[9]. An irregular geometry is described by a weighting or level set function (also called the distance function), which is zero on the boundary, greater than zero inside the domain, and less than zero outside the domain. He developed special Rvachev algebra for finding a level set function for different boundary geometries[10]. Using the level set function, the solution structure method is determined, which exactly describes all the boundary conditions. An unknown part of the solution needs to satisfy the differential equation using the Galerkin weak formulation[11] or collocation strong formulation[12,13]. The methodology is analytically very sound, but the solution structure becomes quite demanding for dynamic and mixed boundary conditions.

Hollig approach[14] utilized a background mesh with equally distributed B-splines. He recognized that basis functions can be divided into internal and external basis functions, while external basis functions cut geometry boundaries and eventually could have very small support inside the domain, which causes computational instabilities. Therefore, external unknown spline coefficients are described with the help of internal coefficients using the development of polynomial expansion near the boundary up to the spline order used. In this way, the internal basis functions are changed and extended near the boundary and cut elements by adding an appropriate portion of the external basis functions. Then, numerical analysis is performed on all the internal and cut finite elements. An extended basis results in the stabilization of the cut

elements, resulting in a smaller matrix condition number and optimal convergence rates (i.e., [7,15]). The essential Dirichlet boundary conditions are satisfied using Rvachev algebra and the solution structure using the weighting level set function, while the Neumann boundary conditions are naturally added as known fluxes, such as in the classical FEM. The Hollig procedure is therefore called WEB splines or weighted extended B-splines. A similar approach was presented by the i-spline FEM, where the B-spline basis functions near the boundary in the zone of the cut elements are modified using a weighting level set function that also exactly satisfies the essential boundary conditions[16].

The essential boundary conditions in other IBM algorithms are usually satisfied using the Nitsche weak formulation, such as in the finite cell method (FCM, among others[17,18]) and the immersed FEM[19] or in the CutFEM/CutIGA[20,21] using ghost penalty parameters. For the numerical integration of cut elements, most authors used several types of cut element triangulation and tessellation using the well-known Gauss–Legendre quadrature rules (i.e.[14]), or quadtrees in 2D and octrees in 3D adaptive numerical integration, dividing the cut elements up to the desired refinement level of accuracy. Both schemes suffer from inaccuracies or a large number of quadrature points on the cut elements. Optimal numerical integration of the cut elements is still an open problem[8]. Stabilization of the cut elements can be established in many ways using the abovementioned extensions and boundary modifications of the basis functions, preconditioning of the stiffness matrix, aggregation of small cut elements with larger ones, adding new terms in the penalty and Nitsche weak formulation, and many others. Please refer to the comprehensive review of [8], for open questions about the treatment of IBM and its three main drawbacks: cut element integration, stabilization and the imposition of essential boundary conditions.

Most of the aforementioned IBMs affect geometry B-rep through solution solving on a background mesh due to, for instance, numerical cut integration, approximation of the level set function, or triangulation and tessellation of the cut elements. Another variant of IBM was called an immersogeometric analysis (IMGA) for solving fluid–structure interaction (FSI) problems to accurately capture the solid immersed boundary to the background fluid mesh[22]. Two crucial steps of IMGA are needed, imposition of essential boundary conditions directly on the surface of the B-rep, and selection of adaptive quadrature rules for accurate capturing of the boundary geometry in the cut elements.

Recently developed immersed boundary finite elements that escape all three drawbacks mentioned by IBMs are the shifted boundary method (SBM), where the cut elements are discarded from analysis, but imposition of boundary conditions are replaced to internal

surrogate boundaries[23,24,25]. Therefore, the main problem is how to accurately project boundary conditions from the real boundary to the surrogate boundary using a Taylor expansion[23] or novel solutions with discrete extension operators[26] to obtain the optimal convergence rate.

Most of the aforementioned IBMs used the Galerkin weak formulation. In contrast, the collocation procedure enables a simpler and truly meshless approach without any numerical integration. A simple example is the well-known Kansa collocation[27] with radial basis functions on irregular geometries using internal collocation points for differential equations and boundary collocation points for the imposition of boundary conditions. Rare examples of collocation IBMs are solutions with Rvachev solution structure[12,13], with WEB splines[28], or hybrid immersed techniques using collocation at internal collocation points, and the Galerkin formulation on cut elements for the imposition of boundary conditions[29].

In this paper, we propose an alternative collocation shifted boundary method, which is a truly meshless approach without any numerical integration. The differential equation is satisfied in internal Greville collocation points on a regular background grid, and all the boundary conditions are satisfied in the internal collocation points of the surrogate internal boundaries via designated polynomials (related to Taylor series expansions) up to the order of the spline basis functions used. The collocation procedure is therefore very simple, utilizing only the internal collocation points and boundary point values of the Dirichlet or Neumann conditions, and presenting an immersogeometric type of analysis because the original CAD or another chosen boundary representation is retained. The methodology is presented on 2-D Poisson examples, and attains the same convergence rate as IGA collocation, which proves that a shifted satisfaction of boundary conditions does not reduce the accuracy and efficiency of the proposed collocation procedure.

2. METHODOLOGY

In this section, we describe all parts of the proposed shifted boundary collocation immersogeometric analysis (SBC-IMGA), which is a truly meshless approach without any numerical integrations. The methodology finds a solution on a regular background grid consisting of equally distributed Fup_2 basis functions. Subsection 2.1 describes these basis functions, their main properties, and their close relationships to the B-splines. The 2D domain geometry is set only by the boundary representation (B-Rep), which will be presented in Subsection 2.2. The collocation procedure through the classical strong formulation based on

the Greville internal collocation points with a shifted imposition of boundary conditions is presented in Subsection 2.3.

2.1. FUP BASIS FUNCTIONS

Fup basis functions belong to the class of atomic basis functions[30,31] spanning the vector space of algebraic polynomials. The simplest and basic atomic function is *up(x)*, which can be obtained by an infinite number of convolutions of zero-order B-splines - $B_0(x)$ with compact support 2^{-k} and values 2^k, $k \in \mathrm{N}$:

$$up(x) = B_0(x) * B_0(2x) * \ldots * B_0(2^k x) * \ldots * B_0(2^\infty x) \tag{1}$$

Equation (1) implies that every next $B_0(x)$ in the convolution procedure has two times less compact support but retains the same unit integral value. When *k* approaches infinity, $B_0(x)$ approaches the Dirac delta function. This means that the length of the *up(x)* compact support is finite despite the infinite number of convolutions in (1):

$$h_{up} = \sum_{k=0}^{\infty} \frac{1}{2^k} = 2 \qquad \rightarrow \qquad \operatorname{supp} up(x) = \left[-1, 1\right] \tag{2}$$

Convolution procedure (1) causes *up(x)* to contain polynomials of all orders by parts of the compact support. Therefore, *up(x)* has an infinite number of derivatives and can be regarded as a perfect spline.

The values of the function *up(x)* and its derivatives can be exactly calculated in the form of rational numbers at the binary-rational points of the compact support:

$$x_{br} = -1 + k \cdot 2^{-m}, \quad m \in \mathrm{N}, \quad k = 1, \ldots, 2^{m+1}. \tag{3}$$

At all other points of compact support, it is possible to calculate the function *up(x)* and its derivatives up to machine precision, but this approach is sufficient for all practical purposes[31].

For an exact description of algebraic polynomials up to the *n*-order on the characteristic interval $\Delta x_n = 2^{-n}$, 2^{n+1} shifted *up(x)* basis functions are needed for Δx_n. Due to the relatively large number of used basis functions, *up(x)* has low approximation properties. For a more efficient numerical implementation, $Fup_n(x)$ basis functions are needed.

Fup$_n$(x) are also compactly supported atomic basis functions, but they need only (n+2) *Fup$_n$(x)* basis functions to exactly describe all the polynomials up to the *n*-order. The compact support of function *Fup$_n$(x)* contains (n+2) characteristic intervals $\Delta x_n = 2^{-n}$:

$$\text{supp } Fup_n(x) = \left[-(n+2)\cdot 2^{-n-1},\ (n+2)\cdot 2^{-n-1} \right]. \tag{4}$$

For $n = 0$, we have $Fup_0(x) = up(x)$.

The function *Fup$_n$(x)* can be obtained by convolution of the compressed B_n - spline and *up* basis function:

$$Fup_n(x) = B_n(2^n x) * up(2^{n+1} x) \tag{5}$$

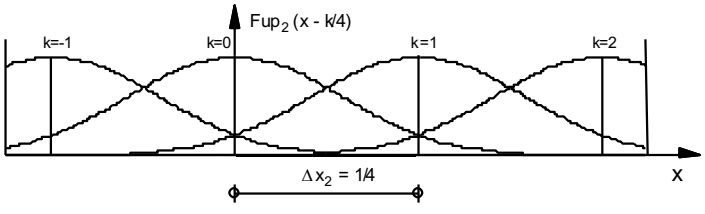
This means that *Fup$_n$(x)* is closely related to *B$_n$(x),* and that these two basis functions have the same convergence rate for the same order. However, due to the convolution with *up(x)*, the function *Fup$_n$(x)* has better approximation properties and higher continuity than *B$_n$(x)*. Therefore, *Fup$_n$(x)* is an infinitely derivable basis function. Figure 1 shows the distribution of the *Fup$_2$* basis functions for the exact description of the polynomials up to the second order.

Atomic basis functions have a solid mathematical background because they represent general solutions of differential functional equations[30]. For *Fup$_n$(x)*, this equation has the following form:

$$Fup'_n(x) = 2\sum_{k=0}^{n+2} \left(C_n^k - C_n^{k-2} \right) \cdot Fup_n\left(2x - \frac{k}{2^n} + \frac{n+2}{2^{n+1}} \right) \tag{6}$$

where C_n^k are binomial coefficients defined as:

$$C_n^k = \binom{n}{k} = \frac{n!}{(n-k)!\cdot k!} \tag{7}$$



$$x^0 = 1 = 2^{-2} \sum_{k=-1}^{2} Fup_2(x - k/4)$$

$$x^1 = 2^{-4} \sum_{k=-1}^{2} k \cdot Fup_2(x - k/4)$$

$$x^2 = 2^{-6} \sum_{k=-1}^{2} \left(k^2 - 5/18 \right) Fup_2(x - k/4)$$

Figure 1. An exact description of the polynomials up to the second order by a linear combination of *Fup$_2$* basis functions.

Equation (6) shows the so-called atomic structure of these basis functions because their derivatives are described by linear combinations of the same functions[30]. *Fup$_n$(x)* can be easily calculated with respect to the linear combination of *up(x)* basis functions shifted for 2^{-n}:

$$Fup_n(x) = \sum_{k=0}^{\infty} C_k(n)\, up\left(x - 1 - \frac{k}{2^n} + \frac{n+2}{2^{n+1}} \right) \tag{8}$$

where $C_0(n) = 2^{C_{n+1}^2} = 2^{n(n+1)/2}$. The remaining coefficients are $C_k(n) = C_0(n) \cdot C'_k(n)$, where the coefficients $C'_k(n)$ are obtained using the following recursion formulas:

$$\begin{aligned} C'_0(n) &= 1 \\ C'_k(n) &= (-1)^k C_{n+1}^k - \sum_{j=1}^{\min\left\{k;\,2^{n+1}-1\right\}} C'_{k-j}(n) \cdot \delta_{j+1} \end{aligned} \tag{9}$$

Figure 2 shows *Fup$_2$(x)* and its first three derivatives.

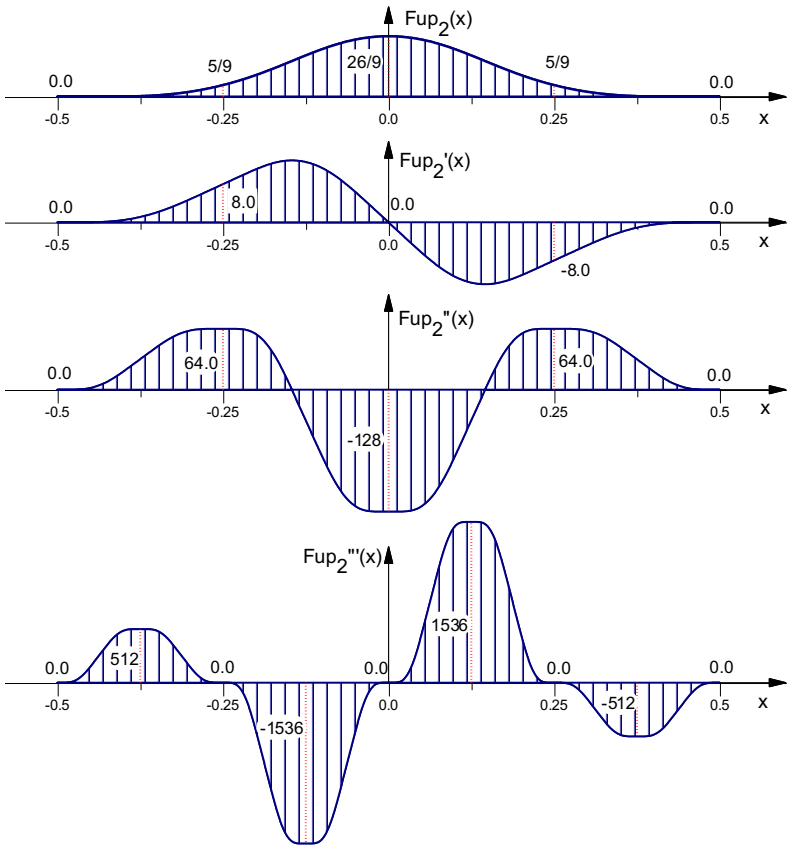


Figure 2. *Fup$_2$(x)* and its first three derivatives.

2-D Fup basis functions are obtained by the Cartesian product of two 1-D basis functions, defined for each direction. In this paper, *Fup$_2$(x,y)* basis functions are used. Our numerical

group has developed many collocation algorithms based on Fup basis functions, for instance, collocation on regular geometry or using Coons patches[32], for initial value problems[33], adaptive collocation methods using classic time marching schemes[34], or directly in the space-time domain[35] as well as using the Rvachev structure solution[13]. Recently, CV-IGA was developed as an IGA with control volumes presenting a weak collocation subdomain formulation[36]. Moreover, novel hierarchical Fup basis functions have been developed with an efficient h–p adaptive procedure[37,38]. In [39], three IGA formulations are compared, namely, Galerkin, collocation and control volumes, with Fup basis functions and B-splines. Both functions have the same convergence order for the same order of basis functions and for all three formulations. Fup basis functions yield greater accuracy due to their greater continuity and better approximation properties. These consequences are visible in adaptive procedures with hierarchical Fup's enabling h–p adaptation and exponential convergence, contrary to the hierarchical B-splines that enable h adaptation[38].

2.2. B-Rep GEOMETRY REPRESENTATION

The proposed collocation methodology SBC-IMGA only requires that the boundary curves be set as boundary geometry representations (B-Rep) to define the 2-D domain. According to the CAD standard, it is possible to create NURBS or parametrically defined curves such as lines or parabolic and circular arcs. Although NURBS is the usual CAD standard, in this paper, without loss of generality, we choose B-Rep with parametrically defined curves. Figure 3 shows the triangle domain defined by three boundary lines (1-D entities) and three corners (0-D entities).

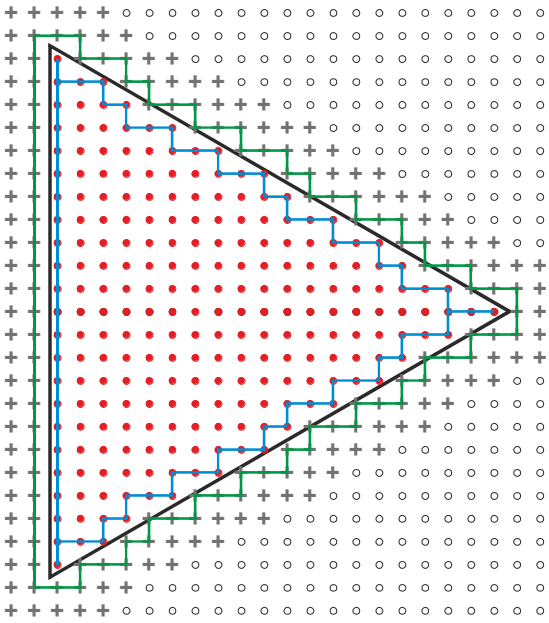


Figure 3. The B-rep of the triangle domain is defined by three boundary lines (1-D entities) and three corners (0-D entities). The geometry boundary is immersed in the background uniform collocation grid. The red points represent the Greville internal collocation points, which coincide with the vertices of the basis functions. Black crosses represent external basis functions that cut B-Rep, while black circles represent basis functions discarded from the approximation because they do not cut boundaries. The blue internal polygon represents the internal surrogate boundary needed for the imposition of boundary conditions. A green external polygon represents the union of external basis functions for which the boundary conditions need to be prescribed, using an internal blue polygon.

Figure 3 clearly shows that B-Rep defines the boundary of the domain in which direct imposition of all kinds of boundary conditions (Dirichlet, Neumann and Robin) can be defined in a classical functional way (see next Subsection 2.3). 1-D boundary curves are defined in the parametric domain:

$$x = x(\xi),\ y = y(\xi);\quad \xi \in [0,1] \tag{10}$$

External domain boundary is defined only by one set of connected external 1-D boundary curves that are oriented in the positive geometric counterclockwise direction. The number of holes can be arbitrary. Each hole is represented by one set of connected internal 1-D boundary curves that are oriented in the negative geometric clockwise direction. Orientation is important due to the proper definition of the outward normal along the boundary. 0-D entities present the joint points of two neighboring 1-D boundary curves. These 0-D entities could have two different boundary values if the boundary conditions are different in the neighboring curves or if there are jumps in the boundary values between them.

Moreover, B-Rep enables the definition of a background uniform collocation grid, which is generally defined by the origin point and discretization size in both directions. The B-rep is immersed in the background grid, and it is simple to find the internal and external basis functions. Figure 3 shows the symbols that represent the vertices of the basis functions. Internal basis functions are those whose vertices lie inside the domain surrounded by the external 1-D boundary curves. The union of vertices of internal basis functions (red points in Figure 3) represents the set of internal Greville collocation points needed for satisfying the differential equation. All the other symbols represent external basis functions. The surrogate internal boundary needed for the imposition of boundary conditions consists of a union of internal collocation points that are not surrounded by the maximum number of internal collocation points or basis functions. The $Fup_n(x,y)$ basis function has the (n+1)×(n+1) maximum internal basis functions. This means that for the $Fup_2(x,y)$ basis functions, there are 9 internal basis functions or Greville collocation points. In Figure 3, for the triangle domain, the surrogate internal boundary is a blue internal polygon. A green external polygon is a union of external

basis functions that cut boundaries and has a nonzero value at the collocation points of the blue internal polygon. The number of external basis functions in the green polygon is equal to the number of boundary conditions that we need to impose in the internal blue polygon. The black crosses represent external basis functions that cut B-Rep but do not contribute to the internal collocation points of the blue polygon. The black circles represent external basis functions discarded from the approximation because they do not cut boundaries and have no contribution. Obviously, finding internal collocation points enables a quick calculation of all the entities of the background uniform collocation grid needed for a solution approximation. Therefore, B-rep enables the definition of the geometry as well as the background collocation grid needed for the solution approximation, which will be presented in the sequel.

2.3. SHIFTED BOUNDARY COLLOCATION METHOD

The proposed strong collocation formulation is presented for a two-dimensional domain-Ω solving boundary-value problem defined with a differential equation and related boundary conditions:

$$\begin{aligned} &L(u) = f \quad in \ \Omega, \\ &u = u_D \quad on \ \Gamma_D, \\ &\frac{\partial u}{\partial n} = q_N \quad on \ \Gamma_N, \end{aligned} \tag{11}$$

where *L* is the differential operator (here, we consider different 2-D Poisson problems defined by well-known diffusion or second-order Laplace operators), $u = u(x, y)$ is the unknown solution, $f(x, y)$ is the known external loading, u_D is the function that defines the essential Dirichlet boundary conditions on Γ_D, and q_N is the flux function oriented normal to the boundary defining the natural Neumann boundary conditions on Γ_N. The outward normal is denoted by ***n***. The solution approximation u_N is sought with respect to the following linear combination:

$$u_N(x,y) = \sum_{i=1}^{n} \sum_{j=1}^{m} C_{ij} \cdot \varphi_{ij}(x,y) \tag{12}$$

where $\underline{\varphi_{i,j}}$ are *Fup₂* basis functions, and $\underline{C_{i,j}}$ are unknown *Fup* coefficients obtained from a properly defined linear system due to solving the 2-D Poisson problem. Figure 3 indicates that

the collocation uniform grid has four types of collocation points (related to the basis functions and corresponding conditions). External basis functions that do not cut boundaries (black circles in Figure 3) have no contribution to the solution, and their coefficients are directly set to zero in (12) and discarded from the approximation. As usual in collocation strong point formulations, the differential equation is satisfied in internal collocation points (red points in Figure 3):

$$\sum_{i=1}^{n}\sum_{j=1}^{m} C_{ij}\cdot L\left[\varphi_{ij}(x_k,y_{\mathbf{l}})\right]=f(x_k,y_{\mathbf{l}})\ , \qquad 1\le k\le n\ ,\ 1\le \mathbf{l}\le m; \quad k,\mathbf{l}\in N_{in} \tag{13}$$

where $(x_k, y_{\mathbf{l}})$ are internal collocation points located inside the domain surrounded by 1-D external boundary curves (Figure 3). N_{in} represents the union of all the internal collocation points. These collocation points are usually called the Greville collocation points, which coincides with the vertices of the basis functions. It is well known in the IGA community that IGA collocation has no optimal convergence, unlike its Galerkin counterpart[39,40,41]. Among others, other more efficient collocation point locations have been analyzed[42]. However, this is a very simple and efficient collocation algorithm with Greville points, and the application of hierarchical B-splines or Fup basis functions enables enhanced convergence of the collocation algorithm[37,38]. Therefore, we retain this classical and simple Greville choice while implementation of efficient adaptive collocation algorithm with a higher-order Fup or B-splines will be left for future publications. Furthermore, it is necessary to resolve the role of the remaining external basis functions (black crosses in Figure 3).

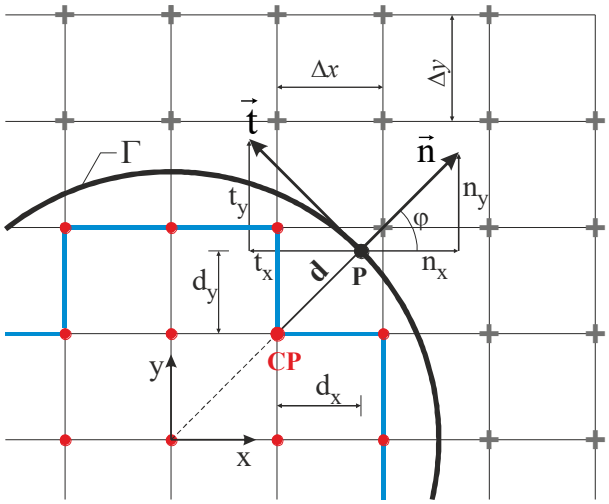
External basis functions whose vertices belong to the external green polygon cut boundary and have nonzero values in the internal blue polygon and its internal collocation points. The union of these basis functions determines the number of collocation points for the imposition of boundary conditions (N_{BC}). According to the properties of the shifted boundary finite element method[23,24], here, we propose that for each external basis function from the external green polygon, corresponding internal collocation points need to be selected in the blue internal polygon.

For each internal collocation point of the blue polygon, a Taylor series expansion can be used for the imposition of boundary conditions from the internal collocation point to the boundary. For the $Fup_2(x,y)$ basis functions used, all derivatives in both directions greater than two are

zero at the collocation point (x_k, y_l), as shown in Figure 2. For example, the formula for describing the Dirichlet boundary condition has the following form:

$$
\begin{aligned}
&u(x_k,y_l)+\frac{d}{1!}\left(\frac{\partial u(x_k,y_l)}{\partial x}n_x+\frac{\partial u(x_k,y_l)}{\partial y}n_y\right)+\\
&\frac{d^2}{2!}\left(\frac{\partial^2 u(x_k,y_l)}{\partial x^2}{n_x}^2+2\frac{\partial^2 u(x_k,y_l)}{\partial x\partial y}n_x n_y+\frac{\partial^2 u(x_k,y_l)}{\partial y^2}{n_y}^2\right)+\\
&\frac{d^3}{3!}\left(3\frac{\partial^3 u(x_k,y_l)}{\partial x^2\partial y}{n_x}^2 n_y+3\frac{\partial^3 u(x_k,y_l)}{\partial x\partial y^2}n_x{n_y}^2\right)+\\
&\frac{d^4}{4!}\left(6\frac{\partial^4 u(x_k,y_l)}{\partial x^2\partial y^2}{n_x}^2{n_y}^2\right)=u_D(x_k+d_x,y_l+d_y);\ k,l\in N_{BC}
\end{aligned}
\tag{14}
$$

where $u(x_k, y_l)$ is the solution value at the internal collocation point, u_D denotes the boundary value at point **P** on Γ_D, i.e., $u(x_k + d_x, y_l + d_y)$, *d* is the distance between these two points, and n_x and n_y are outward normal components (see Figure 4).



$$\vec{n} = n_x\vec{i} + n_y\vec{j}$$

$$\vec{t} = t_x\vec{i} + t_y\vec{j}$$

$$|t| = \sqrt{x'(\xi)^2 + y'(\xi)^2}$$

$$t_x = \frac{x'(\xi)}{|t|} \quad ; \quad t_y = \frac{y'(\xi)}{|t|}$$

$$n_x = \cos\varphi = t_y \; ; \; n_y = \sin\varphi = -t_x$$

$$d_x = d \cdot n_x \quad ; \quad d_y = d \cdot n_y$$

$$d = \sqrt{{d_x}^2 + {d_y}^2}$$

Figure 4. Taylor series of a solution expansion from the internal collocation point in the normal direction to the boundary.

Expression (14) shows that the internal collocation points of the internal blue polygon are multiple collocation points (Figure 6), while for instance, the boundary Dirichlet values are sampled exactly on predefined boundary Dirichlet curves. In that sense, the proposed collocation algorithm is a shifted boundary method due to the imposition of essential boundary conditions on the surrogate boundary (internal blue polygon in Figure 3 or 4) rather

than on the actual boundary, but it is also immersogeometric method because there is no integration or approximation of the boundary and/or Dirichlet values; instead, only the values are sampled exactly on the immersed boundary.

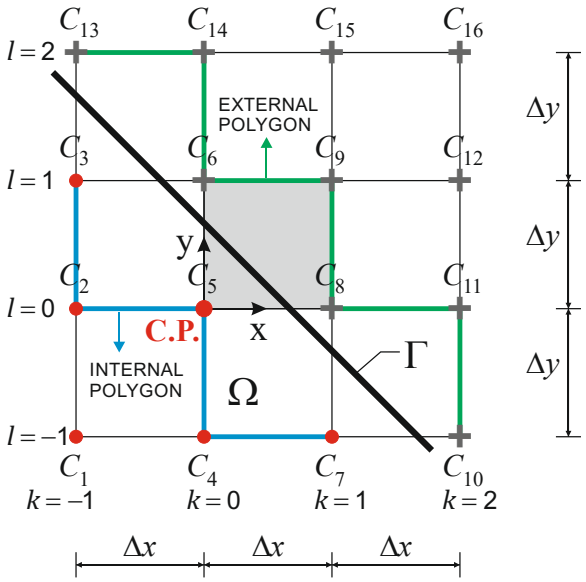


Figure 5. Solution approximation of the "cut element", which is located between the internal blue polygon and the external green polygon

However, expression (14) and direct application of the Taylor truncated series introduce higher-order mixed derivatives and are not computationally efficient, as occurs in the Rvachev structure solution[9,10]. On the other hand, it is more convenient to construct a second-order polynomial from the internal collocation point on the "cut element". This application is also related to recent work on the shifted boundary finite element method, where the Taylor series is replaced by a local discrete operator[26]. Figure 5 shows the "cut element", i.e., the rectangle between the internal blue and external green polygons where the internal collocation point is located at the origin of the local cut element coordinate system. The second-order 2-D polynomials are obtained by equalizing the 2-D polynomials calculated at the origin internal collocation point and the known Fup approximation solution (12) at that point where 9 Fup_2 basis functions have nonzero values (from 1 to 9 in the local cut element; see Figure 5):

$$
\begin{aligned}
P_2(x,y) = \frac{1}{81\Delta x^2\Delta y^2}\Big\{ & (18y^2 - 18y\Delta y + 5\Delta y^2)(18x^2 - 18x\Delta x + 5\Delta x^2)\cdot C_1 + \\
& 2(13\Delta y^2 - 18y^2)(18x^2 - 18x\Delta x + 5\Delta x^2)\cdot C_2 + \\
& (18y^2 + 18y\Delta y + 5\Delta y^2)(18x^2 - 18x\Delta x + 5\Delta x^2)\cdot C_3 + \\
& 2(18y^2 - 18y\Delta y + 5\Delta y^2)(13\Delta x^2 - 18x^2)\cdot C_4 + \\
& 4(18y^2 - 13\Delta y^2)(18x^2 - 13\Delta x^2)\cdot C_5 + \\
& 2(18y^2 + 18y\Delta y + 5\Delta y^2)(13\Delta x^2 - 18x^2)\cdot C_6 + \\
& (18x^2 + 18x\Delta x + 5\Delta x^2)(18y^2 - 18y\Delta y + 5\Delta y^2)\cdot C_7 + \\
& 2(18x^2 + 18x\Delta x + 5\Delta x^2)(13\Delta y^2 - 18y^2)\cdot C_8 + \\
& (18x^2 + 18x\Delta x + 5\Delta x^2)(18y^2 + 18y\Delta y + 5\Delta y^2)\cdot C_9\Big\}
\end{aligned}
\tag{15}
$$

Therefore, the 2-D polynomial consists of 9 Fup coefficients and 9 second-order polynomials, representing the solution approximation of the cut element (its complete calculation is given in Appendix A.2). Now, a 2-D polynomial (15) can be used for the imposition of boundary conditions instead of the Taylor series (14). This is an equivalent procedure because the 2-D polynomial (15) is also obtained at the internal collocation point and describes the solution approximation (12) on the cut element. The Dirichlet boundary values are satisfied directly in the immersed boundary as known 2-D polynomial values:

$$
P_2(x_k + d_x, y_{\mathbf{l}} + d_y) = u_D(x_k + d_x, y_{\mathbf{l}} + d_y), \qquad 1 \le k \le n\ ,\ 1 \le \mathbf{l} \le m; \quad k, \mathbf{l} \in N_{BC} \tag{16}
$$

Analogously, the Neumann boundary conditions are satisfied as known normal derivatives in the immersed boundary:

$$
\frac{\partial P_2(x_k + d_x, y_{\mathbf{l}} + d_y)}{\partial n} = q_N(x_k + d_x, y_{\mathbf{l}} + d_y), \qquad 1 \le k \le n\ ,\ 1 \le \mathbf{l} \le m; \quad k, \mathbf{l} \in N_{BC} \tag{17}
$$

The locations of the boundary values in (16) and (17) are found in the normal direction from the internal collocation point to the boundary according to Figure 4. The number of external basis functions is usually not the same (green external polygon) as the number of internal basis functions and corresponding collocation points in the blue internal polygon. It is necessary to computationally calculate accurate statistics of the number of internal and external basis functions in both polygons. The difference in the number of boundary conditions between them needs to be satisfied around edge points or 0-D entities, or along 1-D curves (Figure 6).

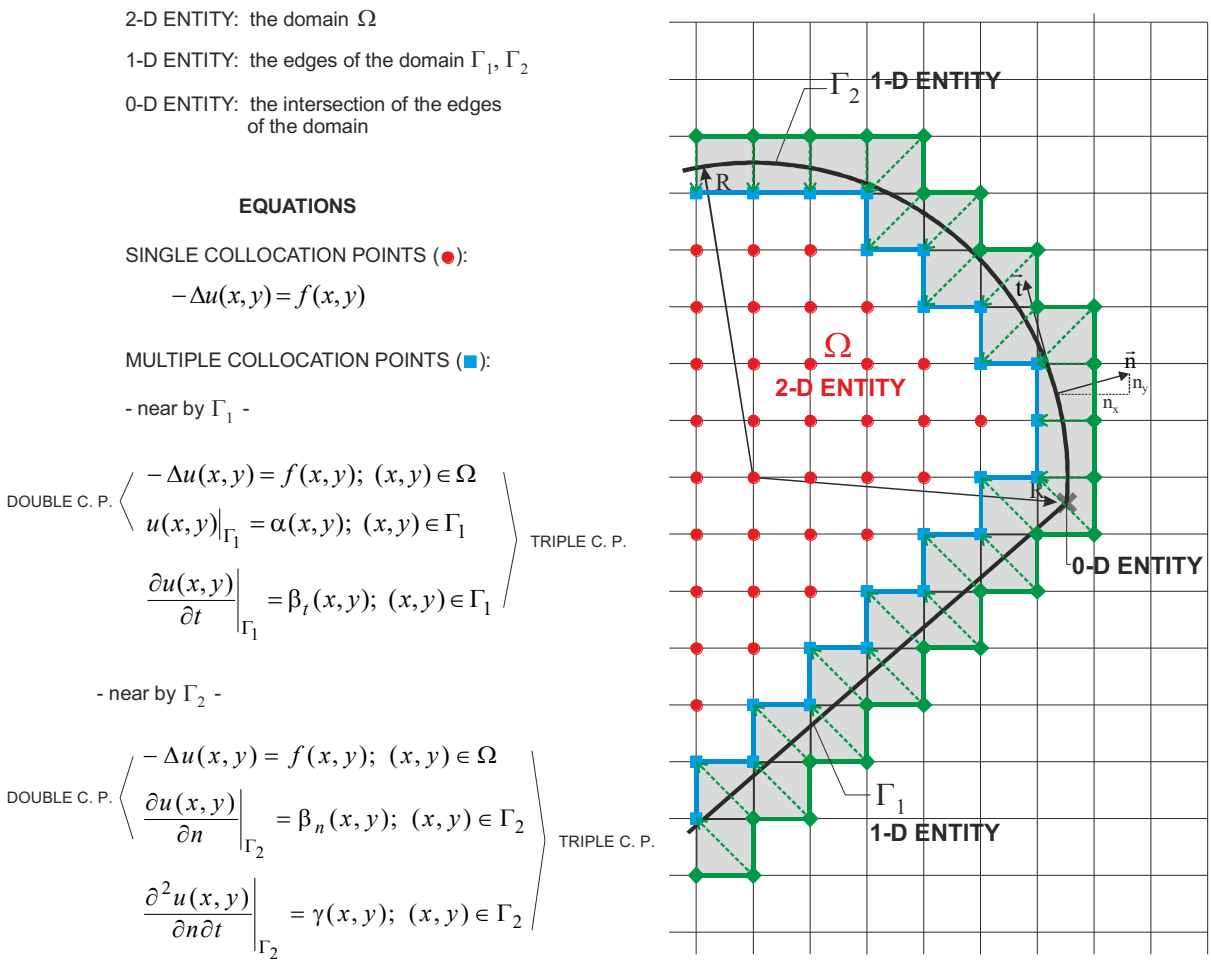


Figure 6. The boundary in the zone of the edge point (0-D entity) between the line and circle 1-D boundaries is magnified.

Figure 6 shows the "zoom out" of the boundary in the zone of the edge point (0-D entity) between the line and circle 1-D boundaries. The linking between the points of the internal blue polygon and the external green polygon is presented. Depending on the number of missed boundary conditions (types of edge points, convex or concave), all the needed remaining conditions should be satisfied; first, different Dirichlet and Neumann boundary conditions are used, and then the boundary conditions for both 1-D curves and finally known tangent changes of boundary conditions. These boundary conditions, as well as all other known derivations of the Dirichlet and Neumann boundary conditions, will also be satisfied from corresponding internal blue collocation points (Figure 6) according to expressions (16) and (17) as well as (A.2.4)-(A.2.11) given in Appendix A.2. If there are more missed external basis functions in certain special cases, their coefficients will be presented in the form of linear

combinations of used internal and external basis functions, as will be shown in the sequel for all other external basis functions that do not cut the internal blue polygon. Linear systems (13), (16), and (17) with additional conditions from Figure 6 create a solution approximation (12), obtaining Fup coefficients for all the internal basis functions and external basis functions that belong to the green external polygon. The other external basis functions cut the boundary and contribute only to the solution approximation (12) between the blue internal polygon and immersed boundary.

Since these external basis functions have no values in the abovementioned linear solution system, their Fup coefficients should be obtained in the final postprocessing step to avoid the changing of the boundary conditions. Therefore, the 2-D polynomial (15) consists of 9 Fup coefficients (locally denoted from 1 to 9 in Figure 5) and 9 second-order polynomials, representing the solution approximation of the cut element. The other remaining 7 Fup coefficients (locally denoted from 10 to 16 in Figure 5) can be described as linear combinations of 9 known Fup coefficients. Seven additional conditions can be placed in the center of the cut element, and third-order derivatives are neglected because expression (15) is a second-order polynomial:

$$\begin{aligned}
&\left.\frac{\partial^3 u_N}{\partial x^3}\right|_{(\Delta x/2,\Delta y/2)}=0; \quad \left.\frac{\partial^3 u_N}{\partial y^3}\right|_{(\Delta x/2,\Delta y/2)}=0; \quad \left.\frac{\partial^4 u_N}{\partial x^3\partial y}\right|_{(\Delta x/2,\Delta y/2)}=0;\\
&\left.\frac{\partial^4 u_N}{\partial x\partial y^3}\right|_{(\Delta x/2,\Delta y/2)}=0; \quad \left.\frac{\partial^5 u_N}{\partial x^3\partial y^2}\right|_{(\Delta x/2,\Delta y/2)}=0; \quad \left.\frac{\partial^5 u_N}{\partial x^2\partial y^3}\right|_{(\Delta x/2,\Delta y/2)}=0; \qquad (18)\\
&\left.\frac{\partial^6 u_N}{\partial x^3\partial y^3}\right|_{(\Delta x/2,\Delta y/2)}=0.
\end{aligned}$$

Substituting (12) in (18) gives a linear system of seven equations with seven external unknown Fup coefficients described with respect to the known internal and external collocation basis functions:

$$
\begin{aligned}
C_{10} &= 3\,C_7 - 3\,C_4 + C_1\,;\\
C_{11} &= 3\,C_8 - 3\,C_5 + C_2\,;\\
C_{12} &= 3\,C_9 - 3\,C_6 + C_3\,;\\
C_{13} &= 3\,C_3 - 3\,C_2 + C_1\,;\\
C_{14} &= 3\,C_6 - 3\,C_5 + C_4\,;\\
C_{15} &= 3\,C_9 - 3\,C_8 + C_7\,;\\
C_{16} &= C_1 - 3\,C_2 + 3\,C_3 - 3\,C_4 + 9\,C_5 - 9\,C_6 + 3\,C_7 - 9\,C_8 + 9\,C_9\,.
\end{aligned}
\tag{19}
$$

Each external basis function from (19) belongs to more cut elements. Therefore, only one cut element closest to the particular external basis function is chosen for its calculation. This choice is not crucial because condition (19) ensures that only 2-D polynomial higher-order derivatives vanish on each cut element. In this way, all the internal and external Fup coefficients in (12) are calculated.

3. NUMERICAL EXAMPLES

Selected 2-D numerical Poisson examples defined on irregular geometry demonstrate the efficiency of the proposed collocation algorithm. The first two examples consider smooth problems in which the methodology can achieve the maximum order of convergence, as in IGA collocation. Another two nonsmooth examples in the L-shaped domain show that the convergence is reduced, as expected, but the method is still stable and robust.

3.1. A circular ring with a smooth solution

In the first example, a 2-D Poisson problem with homogeneous Dirichlet boundary conditions is considered[43]:

$$
\begin{aligned}
&-\Delta u + u = f \quad \text{in } \Omega,\\
&u = 0 \quad \text{on } \Gamma.
\end{aligned}
\tag{20}
$$

The domain is defined on the quarter part of the circular ring with an inner radius of $R_{in} = 1.0$ and an outer radius of $R_{out} = 4.0$; see Figure 7a. The considered domain is located in the positive quadrant of the coordinate system. The right-side *f* in (20) is obtained with respect to the chosen exact solution:

$$
u = (x^2 + y^2 - 1)\,(x^2 + y^2 - 16)\sin(x)\sin(y) \tag{21}
$$

B-Rep is therefore defined by two circle arcs and two lines. An approximation solution is sought with respect to (12) in the range of the collocation uniform grids ($\Delta x = \Delta y$), starting from an 8x8 grid, while each next level is two times denser in each direction. The numerical solution with 4017 *Fup₂* basis functions is illustrated in Figure 7b.

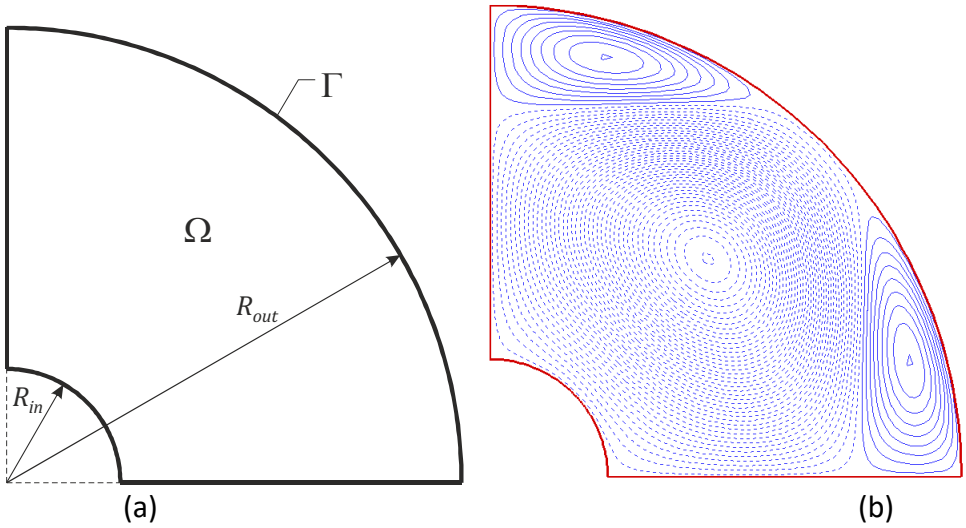


Figure 7. (a) Geometry of the circular ring; (b) Numerical solution with 4017 *Fup₂* basis functions.

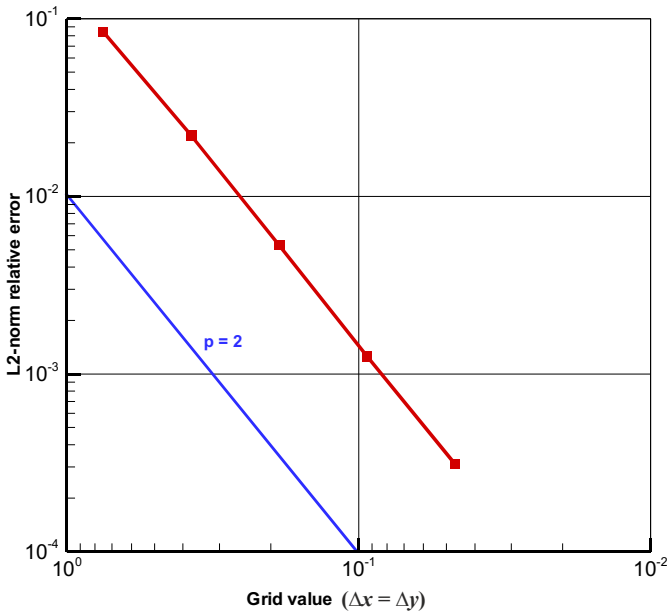


Figure 8. Relative L_2 error norm for problem (20-21).

The relative L_2 error norm is presented in Figure 8. The convergence plot confirms that the convergence rate $p=2$ for the $Fup_2(x,y)$ basis functions used is suboptimal but is the same as that for other collocation solutions on regular geometry[32] or in IGA collocation[39,41]. This means that the presented shifted boundary collocation procedure does not reduce the accuracy due to the shifted imposition of boundary conditions.

3.2 The torsion problem of prismatic rod with triangular cross sections

In this example, the proposed collocation method is applied to a 2-D torsion problem. The geometry is given with a triangle cross section, as presented in Figure 9a. The torsion problem is defined by the Poisson equation with Dirchlet homogenous boundary conditions:

$$\frac{\partial^2 u(x,y)}{\partial x^2}+\frac{\partial^2 u(x,y)}{\partial y^2}=-2G\vartheta\,;\quad u(x,y)\big|_{\Gamma}=0 \tag{22}$$

where u is the stress function, G is the shear modulus, and ϑ is the twist angle along the rod axis. The shear stress components are the first derivatives of the stress function: $\tau_{xz}=\partial u/\partial y$, $\tau_{yz}=-\partial u/\partial x$. From the solution of (22), the torsional rigidity is given by:

$$C_t=\frac{2}{\vartheta}\int_{\Omega}u(x,y)\,dx\,dy\,. \tag{23}$$

For the triangle cross section in 9a, the stress function u and torsional rigidity are given in [19]:

$$u(x,y)=\frac{G\vartheta a^2}{2}\cdot\left[\left(\frac{x}{a}\right)^3-\left(\frac{x}{a}\right)^2-3\left(\frac{x}{a}\right)\left(\frac{y}{a}\right)^2-\left(\frac{y}{a}\right)^2+\frac{4}{27}\right]\;;\quad C_t=\frac{Ga^4}{15\sqrt{3}} \tag{24}$$

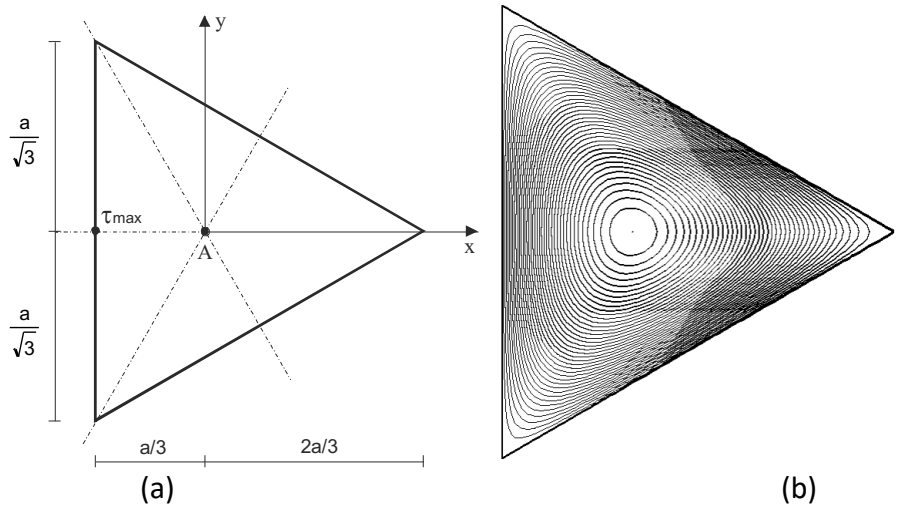


Figure 9. (a) Geometry; (b) stress function *u(x,y)*.

The torsion problem for a=12 (G=1, ϑ=1) in Figure 9a is solved in a series of uniform collocation grids, as in Example 3.1. Figure 9b shows the stress solution obtained by the proposed method, while Figure 10 presents the shear stress components.

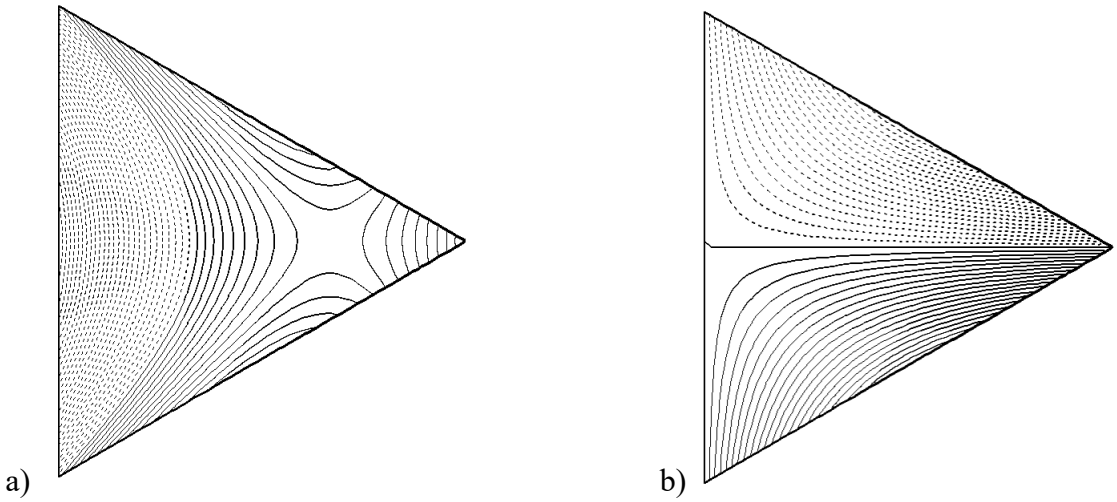


Figure 10. Shear stress components: a) τ_{yz}; b) τ_{xz}.

Figure 11 shows the maximum obtained convergence rate of p = 2 for torsional rigidity (11a) and the L_2 error solution norm (11b), as obtained in the first example.

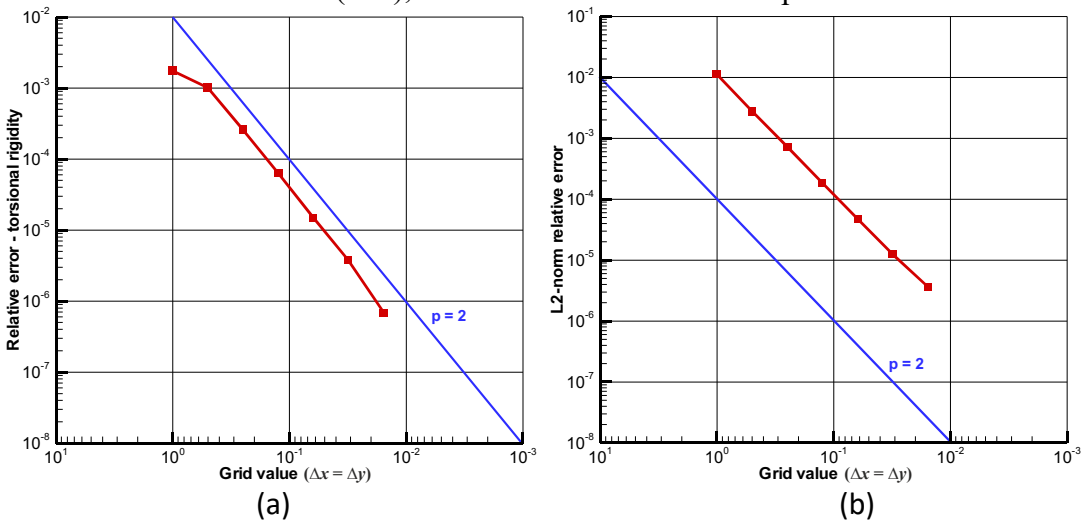


Figure 11. Convergence plot for the (a) torsional rigidity and (b) L_2 error solution norm.

3.3 L-shaped problem

In the next example, the geometry is defined on the L-shaped domain $\Omega = [-1,1]^2 \setminus [0,1]^2$, as shown in Figure 12a. An example is selected from[41] as a singular test with an exact solution. The stationary thermal problem is defined by the Laplace operator:

$$\Delta u = 0 \quad in \ \Omega \tag{25}$$

and the prescribed Dirichlet and Neumann boundary conditions presented in Figure 12a:

$$\begin{aligned} u &= 0 \quad on \ \Gamma_D, \\ \frac{\partial u}{\partial n} &= q_N \quad on \ \Gamma_N. \end{aligned} \tag{26}$$

The Neumann flux function q_N in (26) is defined such that the exact solution is the following in polar coordinates (r,θ):

$$u = r^{2/3} \sin\left(\frac{2\theta - \pi}{3}\right) \tag{27}$$

where $r^2 = x^2 + y^2$ and $\theta = \arctan(y/x)$. A corner point (0,0) causes a sharp gradient of solution derivatives representing singularities.

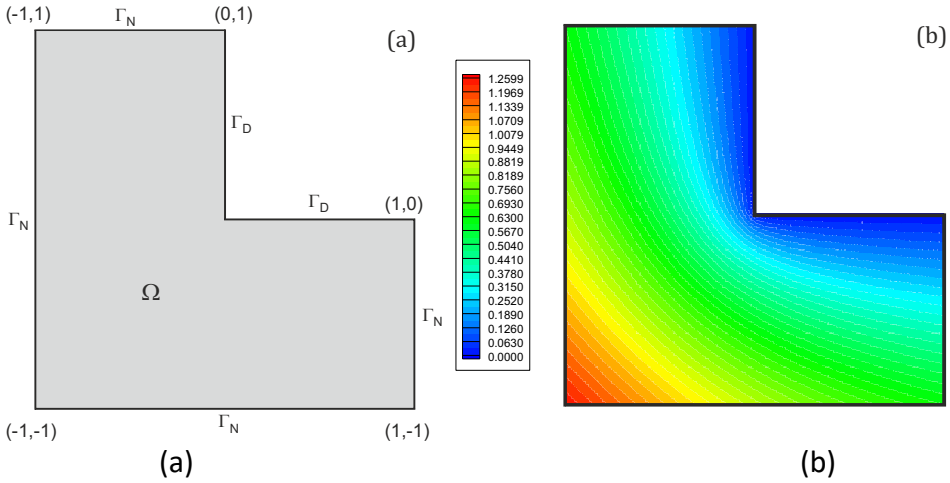


Figure 12. L-shaped problem: (a) geometry with denoted boundary conditions; (b) exact solution.

The starting collocation grid is defined by $\Delta x = \Delta y = 0.25$. Moreover, to obtain a convergence plot, eight refined grids are used so that each next grid has two times more collocation points in each direction.

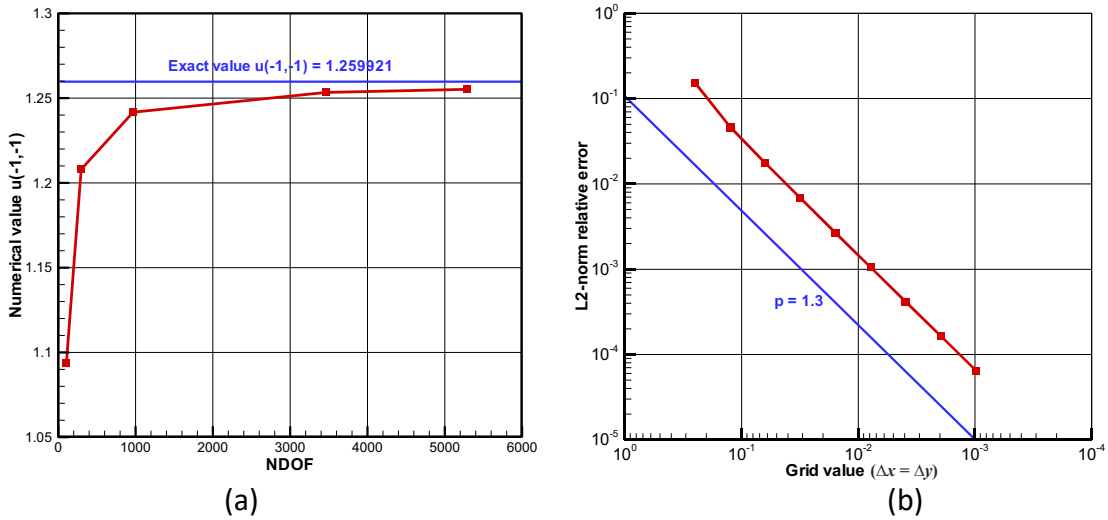


Figure 13. Convergence plots for the L-shaped problem 3.3: (a) solution in corner (-1, -1); (b) L_2 error solution norm.

Figure 13a shows convergence to the corner (-1, -1) of the approximation to the maximum solution value on the considered domain (27). A singular concave corner decreases the convergence rate, as shown in Figure 13b, which has also been reported many times in the literature[41]. However, despite reduced convergence, the solution is still stable and robust, analogous to our earlier IGA solutions (see, for instance[38]).

3.4 Filtration through the porous medium

Filtration through the porous medium in the L-shaped domain is described by the Laplace operator and prescribed Dirichlet and no-flux Neumann boundary conditions, as shown in Figure 14a.

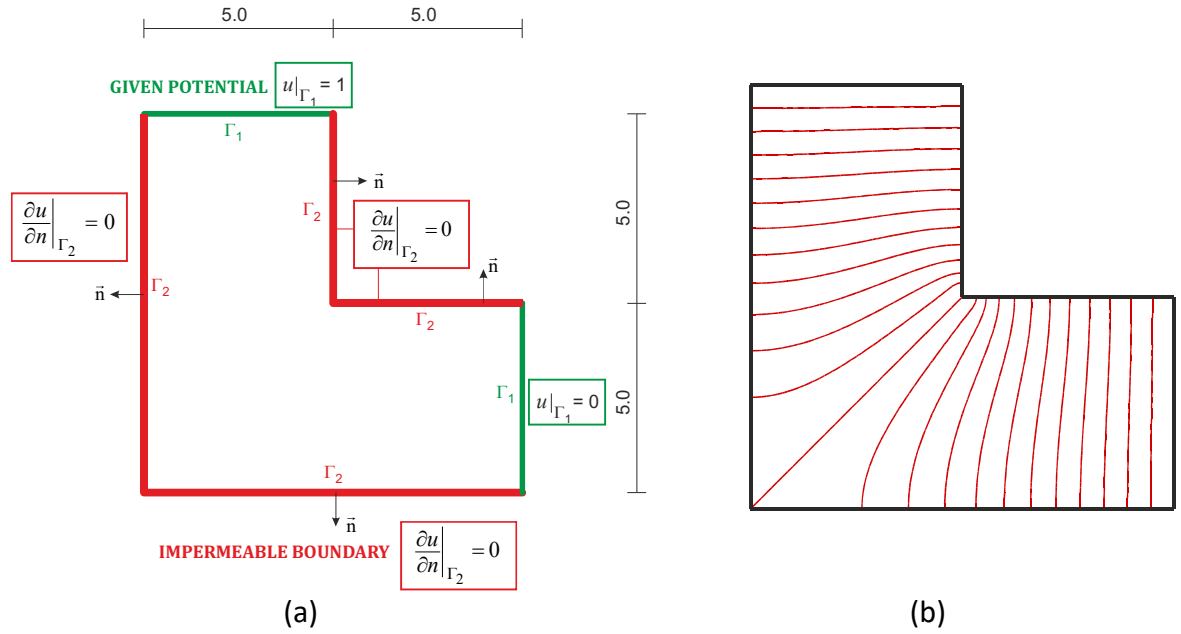


Figure 14. Filtration through the porous medium in the L-shaped domain: (a) geometry with prescribed boundary conditions; (b) numerical solution.

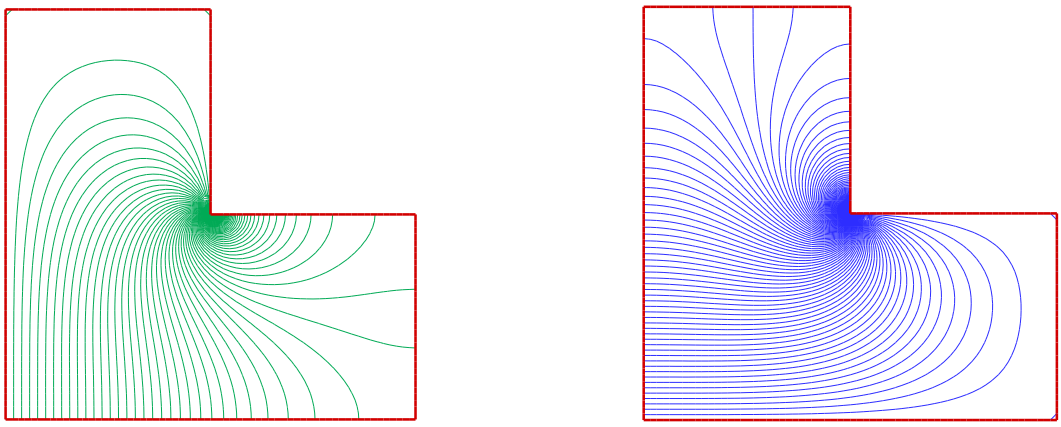


Figure 15. Isolines of partial derivatives (velocities): (a) $\partial u/\partial x$; (b) $\partial u/\partial y$.

Figure 14b presents a numerical solution with 833 collocation points. The obtained partial derivatives $\partial u/\partial x$ and $\partial u/\partial y$ can be described as velocities in the x and y directions, as shown in Figure 15. The density of the isolines in a singular corner clearly represents why this solution also has no maximum convergence rate. Figure 16 shows that the total solution integral on the L-shaped domain has convergence p=1, which is less than the two obtained in the first two smooth problems.

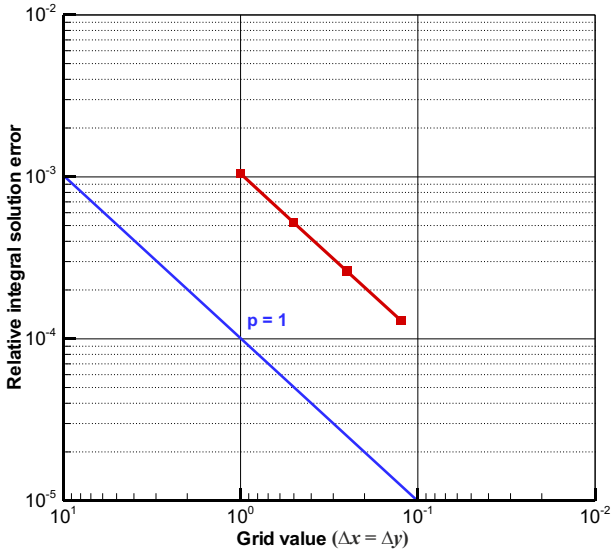


Figure 16. Convergence plot for the total integral of the solution in example 3.4.

4. CONCLUSIONS

Classical mesh-based numerical methods, such as the finite element methods via the Galerkin weak formulation, suffer from complex and computationally expensive meshing procedures, especially for irregular geometries with holes, trimmed boundary surfaces and singularities. In contrast, immersed boundary methods use a background regular mesh that does not fit the boundary representation, completely eliminating classical conforming meshing. A problem arises for cut finite elements due to the demand for numerical integration, imposition of essential Dirichlet boundary conditions, and stabilization of the small cut elements or support of the used basis functions. These drawbacks are removed with the application of the shifted boundary finite element method, where the cut elements are discarded from the analysis, but imposition of boundary conditions are replaced with internal surrogate boundaries.

In this paper, we propose an alternative shifted collocation boundary method, which is a truly meshless approach without any numerical integration. However, the differential equation is satisfied in internal Greville collocation points on regular background grid, and all the boundary conditions are satisfied only in the internal collocation points of the internal surrogate boundaries via a Taylor series expansion or equivalent 2D cut element polynomials up to the order of the spline basis functions used. Although Fup_2 basis functions are used, the procedure is analogous for B_3 splines. Application to higher-order basis functions and/or adaptive procedures will be left for future publications.

The methodology is demonstrated on 2-D Poisson-like examples with the same convergence rate as that of IGA collocation, which proves that a shifted imposition of boundary conditions does not reduce the accuracy or efficiency of the proposed collocation procedure.

Appendix A. Functional connection between 2-D polynomial and linear combination of *Fup2* basis functions on "cut elements"

A.1. Illustration on 1-D domain

Algebraic polynomial of the second order $P_2(x)$ and its two non-zero derivatives can be written as follows:

$$P_2(x) = a_0 + a_1 x + a_2 x^2 ; \quad P_2'(x) = a_1 + 2a_2 x ; \quad P_2''(x) = 2a_2 . \tag{A.1.1}$$

At the origin of the coordinate system, at x=0, the value and the first two derivatives of $P_2(x)$ are:

$$P_2(0) = a_0 ; \quad P_2'(0) = a_1 ; \quad P_2''(0) = 2a_2 . \tag{A.1.2}$$

Polynomial coefficients are described with respect to the value and the first two derivatives of $P_2(0)$:

$$a_0 = P_2(0) ; \quad a_1 = P_2'(0) ; \quad a_2 = \frac{1}{2} P_2''(0) . \tag{A.1.3}$$

Figure A.1 considers uniform collocation grid and distribution of *Fup2* basis functions. At the origin of the coordinate system, at x=0, we consider internal collocation point which is the closest to the boundary point. The interval between them creates so called "cut element". According to the idea of shifted boundary collocation method proposed in this paper, derivatives of $P_2(x)$ in the internal collocation point at x=0 can be presented as a linear combination of three $Fup_2(x)$ basis functions whose coefficients are denoted as C_{-1}, C_0 and C_1:

$$\begin{aligned} &\frac{5}{9} C_{-1} + \frac{26}{9} C_0 + \frac{5}{9} C_1 = a_0 ; \\ &-\frac{2}{\Delta x} C_{-1} + \frac{2}{\Delta x} C_1 = a_1 ; \\ &\frac{1}{2}\left(\frac{4}{\Delta x^2} C_{-1} - \frac{8}{\Delta x^2} C_0 + \frac{4}{\Delta x^2} C_1 \right) = a_2 . \end{aligned} \tag{A.1.4}$$

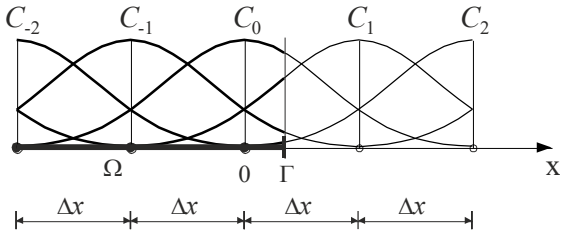


Figure A.1 Distribution of $Fup_2(x)$ basis functions on 1-D uniform collocation grid close to the boundary.

Substituting (A.1.4) into (A.1.1) yields:

$$\left(\frac{5}{9} - \frac{2x}{\Delta x} + \frac{2x^2}{\Delta x^2} \right) C_{-1} + \left(\frac{26}{9} - \frac{4x^2}{\Delta x^2} \right) C_0 + \left(\frac{5}{9} + \frac{2x}{\Delta x} + \frac{2x^2}{\Delta x^2} \right) C_1 = P_2(x) \tag{A.1.5}$$

Therefore, $P_2(x)$ can be described as a linear combination of three polynomials and Fup coefficients. Furthermore, the polynomial $P_2(x)$ in (A.1.5) is equal to the Taylor truncated series up to the second order from internal collocation point at x=0 to the boundary, i.e. on the cut element (see and A.1.3 that $P_2(x)$ coefficients are the same as in the Taylor series at x=0). The polynomial $P_2(x)$ can be used for imposition of boundary conditions from internal collocation points. Since this imposition is shifted from the real boundary, similarities between the proposed collocation procedure and shifted boundary finite element method are obvious.

Due to shifted imposition of boundary conditions, Fup coefficient C_2 is unknown. Its value can be obtained as a linear combination of other three Fup coefficients:

$$C_2 = 3\,C_1 - 3\,C_0 + C_{-1}\,. \tag{A.1.6}$$

The expression (A.1.6) is defined from the condition that the third derivative of the solution should vanish on the cut element. Therefore, the Fup solution at the center of the cut element should be equal to zero, i.e.

$$\left.\frac{\partial^3 u_N}{\partial x^3}\right|_{(\Delta x/2)} = 0\,. \tag{A.1.7}$$

A.2. Illustration on 2-D domain

The algebraic polynomial $P_2(x,y)$ and all mixed partial derivatives can be written as follows:

$$\begin{aligned}
P_2^{(0,0)}(x,y) &= a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 + a_6 x^2 y + a_7 xy^2 + a_8 x^2 y^2 \\
P_2^{(1,0)}(x,y) &= a_1 + 2a_3 x + a_4 y + 2a_6 xy + a_7 y^2 + 2a_8 xy^2 \\
P_2^{(0,1)}(x,y) &= a_2 + a_4 x + 2a_5 y + a_6 x^2 + 2a_7 xy + 2a_8 x^2 y \\
P_2^{(2,0)}(x,y) &= 2a_3 + 2a_6 y + 2a_8 y^2 \\
P_2^{(1,1)}(x,y) &= a_4 + 2a_6 x + 2a_7 y + 4a_8 xy \\
P_2^{(0,2)}(x,y) &= 2a_5 + 2a_7 x + 2a_8 x^2 \\
P_2^{(2,1)}(x,y) &= 2a_6 + 4a_8 y \\
P_2^{(1,2)}(x,y) &= 2a_7 + 4a_8 x \\
P_2^{(2,2)}(x,y) &= 4a_8
\end{aligned} \tag{A.2.1}$$

Substituting x=0 and y=0 into (A.2.1), coefficients a_i, i = 0, ..., 8, can be written with respect to the mixed derivatives of $P_2(0,0)$ at the origin of the cut element (Figure 5):

$$\begin{aligned}
&a_0 = P_2^{(0,0)}(0,0)\,; \quad a_1 = P_2^{(1,0)}(0,0)\,; \quad a_2 = P_2^{(0,1)}(0,0)\,; \\
&a_3 = \frac{1}{2}P_2^{(2,0)}(0,0)\,; \quad a_4 = P_2^{(1,1)}(0,0)\,; \quad a_5 = \frac{1}{2}P_2^{(0,2)}(0,0)\,; \\
&a_6 = \frac{1}{2}P_2^{(2,1)}(0,0)\,; \quad a_7 = \frac{1}{2}P_2^{(1,2)}(0,0)\,; \quad a_8 = \frac{1}{4}P_2^{(2,2)}(0,0)\,.
\end{aligned} \tag{A.2.2}$$

Analogously as in 1-D domain, $P_2(0,0)$ derivatives can be described with respect to nine $Fup_2(x,y)$ whose coefficients are denoted by C_1, …, C_9 in Figure 5. Once again, the origin of the local cut element is located at the internal collocation point of blue internal polygon (see Figure 5). From (A.2.2) coefficients a_i, i = 0, ..., 8 are:

$$
\begin{aligned}
a_0 &= \tfrac{1}{81}(25C_1+130C_2+25C_3+130C_4+676C_5+130C_6+25C_7+130C_8+25C_9);\\
a_1 &= \tfrac{10}{9\Delta x}(-C_1-5.2C_2-C_3+0\cdot C_4+0\cdot C_5+0\cdot C_6+C_7+5.2C_8+C_9);\\
a_2 &= \tfrac{10}{9\Delta y}(-C_1+0\cdot C_2+C_3-5.2C_4+0\cdot C_5+5.2C_6-C_7+0\cdot C_8+C_9);\\
a_3 &= \tfrac{10}{9\Delta x^2}(C_1+5.2C_2+C_3-2C_4-10.4C_5-2C_6+C_7+5.2C_8+C_9);\\
a_4 &= \tfrac{4}{\Delta x\cdot\Delta y}(C_1+0\cdot C_2-C_3+0\cdot C_4+0\cdot C_5+0\cdot C_6-C_7+0\cdot C_8+C_9);\\
a_5 &= \tfrac{10}{9\Delta y^2}(C_1-2C_2+C_3+5.2C_4-10.4C_5+5.2C_6+C_7-2C_8+C_9);\\
a_6 &= \tfrac{4}{\Delta x^2\Delta y}(-C_1+0\cdot C_2+C_3+2C_4+0\cdot C_5-2C_6-C_7+0\cdot C_8+C_9);\\
a_7 &= \tfrac{4}{\Delta x\cdot\Delta y^2}(-C_1+2C_2-C_3+0\cdot C_4+0\cdot C_5+0\cdot C_6+C_7-2C_8+C_9);\\
a_8 &= \tfrac{4}{\Delta x^2\Delta y^2}(C_1-2C_2+C_3-2C_4+4C_5-2C_6+C_7-2C_8+C_9).
\end{aligned}
\tag{A.2.3}
$$

Substituting (A.2.3) into the first equation of (A.2.1) yields 2-D polynomial $P_2(x,y)$ as a linear combination of Fup coefficients C_i; i = 1, ..., 9, and the second order polynomials as given in the expression (15) in the paper. After substituting (A.2.3) into remaining equations of (A.2.1), all mixed derivatives of $P_2(x,y)$ are obtained as follows:

$$
\begin{aligned}
P_2^{(1,0)}(x,y) = \frac{2}{9\Delta x^2\Delta y^2}\Big\{&(18y^2-18y\Delta y+5\Delta y^2)(2x-\Delta x)\cdot C_1+\\
&2(13\Delta y^2-18y^2)(2x-\Delta x)\cdot C_2+(18y^2+18y\Delta y+5\Delta y^2)(2x-\Delta x)\cdot C_3-\\
&4x(18y^2-18y\Delta y+5\Delta y^2)\cdot C_4+8x(18y^2-13\Delta y^2)\cdot C_5-\\
&4x(18y^2+18y\Delta y+5\Delta y^2)\cdot C_6+(18y^2-18y\Delta y+5\Delta y^2)(2x+\Delta x)\cdot C_7+\\
&2(13\Delta y^2-18y^2)(2x+\Delta x)\cdot C_8+(18y^2+18y\Delta y+5\Delta y^2)(2x+\Delta x)\cdot C_9\Big\}
\end{aligned}
\tag{A.2.4}
$$

$$
\begin{aligned}
P_2^{(0,1)}(x,y) = \frac{2}{9\Delta x^2\Delta y^2}\Big\{&(2y-\Delta y)(18x^2-18x\Delta x+5\Delta x^2)\cdot C_1-\\
&4y\,(18x^2-18x\Delta x+5\Delta x^2)\cdot C_2+(2y+\Delta y)(18x^2-18x\Delta x+5\Delta x^2)\cdot C_3+\\
&2(\Delta y-2y)(18x^2-13\Delta x^2)\cdot C_4+8y(18x^2-13\Delta x^2)\cdot C_5+\\
&2(\Delta y+2y)(13\Delta x^2-18x^2)\cdot C_6+(2y-\Delta y)(18x^2+18x\Delta x+5\Delta x^2)\cdot C_7-\\
&4y\,(18x^2+18x\Delta x+5\Delta x^2)\cdot C_8+(2y+\Delta y)(18x^2+18x\Delta x+5\Delta x^2)\cdot C_9\Big\}
\end{aligned}
\tag{A.2.5}
$$

$$P_2^{(2,0)}(x,y)=\frac{4}{9\Delta x^2\Delta y^2}\Big\{(18y^2-18y\Delta y+5\Delta y^2)\cdot C_1+$$
$$2(13\Delta y^2-18y^2)\cdot C_2+(18y^2+18y\Delta y+5\Delta y^2)\cdot C_3-$$
$$2(18y^2-18y\Delta y+5\Delta y^2)\cdot C_4+4(18y^2-13\Delta y^2)\cdot C_5- \quad \text{(A.2.6)}$$
$$2(18y^2+18y\Delta y+5\Delta y^2)\cdot C_6+(18y^2-18y\Delta y+5\Delta y^2)\cdot C_7+$$
$$2\,(13\Delta y^2-18y^2)\cdot C_8+(18y^2+18y\Delta y+5\Delta y^2)\cdot C_9\Big\}$$

$$P_2^{(1,1)}(x,y)=\frac{4}{9\Delta x^2\Delta y^2}\Big\{(2y-\Delta y)(2x-\Delta x)\cdot C_1+4y\,(\Delta x-2x)\cdot C_2+$$
$$(2y+\Delta y)(2x-\Delta x)\cdot C_3+4x(\Delta y-2y)\cdot C_4+16xy\cdot C_5- \quad \text{(A.2.7)}$$
$$4x(2y+\Delta y)\cdot C_6+(2y-\Delta y)(2x+\Delta x)\cdot C_7-$$
$$4y(2x+\Delta x)\cdot C_8+(2y+\Delta y)(2x+\Delta x)\cdot C_9\Big\}$$

$$P_2^{(0,2)}(x,y)=\frac{4}{9\Delta x^2\Delta y^2}\Big\{\,(18x^2-18x\Delta x+5\Delta x^2)\cdot C_1-$$
$$2(18x^2-18x\Delta x+5\Delta x^2)\cdot C_2+(18x^2-18x\Delta x+5\Delta x^2)\cdot C_3+$$
$$2(13\Delta x^2-18x^2)\cdot C_4+4(18x^2-13\Delta x^2)\cdot C_5+ \quad \text{(A.2.8)}$$
$$2(13\Delta x^2-18x^2)\cdot C_6+(18x^2+18x\Delta x+5\Delta x^2)\cdot C_7-$$
$$2(18x^2+18x\Delta x+5\Delta x^2)\cdot C_8+(18x^2+18x\Delta x+5\Delta x^2)\cdot C_9\Big\}$$

$$P_2^{(2,1)}(x,y)=\frac{8}{\Delta x^2\Delta y^2}\Big\{(2y-\Delta y)\cdot C_1-4y\cdot C_2+(2y+\Delta y)\cdot C_3+2(\Delta y-2y)\cdot C_4+ \quad \text{(A.2.9)}$$
$$8y\cdot C_5-2(2y+\Delta y)\cdot C_6+(2y-\Delta y)\cdot C_7-4y\cdot C_8+(2y+\Delta y)\cdot C_9\Big\}$$

$$P_2^{(1,2)}(x,y)=\frac{8}{\Delta x^2\Delta y^2}\Big\{(2x-\Delta x)\cdot C_1+2(\Delta x-2x)\cdot C_2+(2x-\Delta x)\cdot C_3-4x\cdot C_4+ \quad \text{(A.2.10)}$$
$$8x\cdot C_5-4x\cdot C_6+(2x+\Delta x)\cdot C_7-2(2x+\Delta x)\cdot C_8+(2x+\Delta x)\cdot C_9\Big\}$$

$$P_2^{(2,2)}(x,y)=\frac{16}{\Delta x^2\Delta y^2}\Big\{C_1-2\cdot C_2+C_3-2\cdot C_4+4\cdot C_5-2\cdot C_6+C_7-2\cdot C_8+C_9\Big\} \quad \text{(A.2.11)}$$

According to the Figure 5, other seven remaining Fup coefficients, C_{10}, …, C_{16}, on the cut element defined by rectangle $\Delta x\times\Delta y$ are obtained using seven conditions (18) that all mixed third derivatives of the Fup solution (12) in the center of the cut element are equal to zero:

$$
\begin{aligned}
C_{10} &= 3\,C_7 - 3\,C_4 + C_1\,; \\
C_{11} &= 3\,C_8 - 3\,C_5 + C_2\,; \\
C_{12} &= 3\,C_9 - 3\,C_6 + C_3\,; \\
C_{13} &= 3\,C_3 - 3\,C_2 + C_1\,; \\
C_{14} &= 3\,C_6 - 3\,C_5 + C_4\,; \\
C_{15} &= 3\,C_9 - 3\,C_8 + C_7\,; \\
C_{16} &= C_1 - 3\,C_2 + 3\,C_3 - 3\,C_4 + 9\,C_5 - 9\,C_6 + 3\,C_7 - 9\,C_8 + 9\,C_9\,.
\end{aligned}
\qquad \text{(A.2.12)}
$$

In conclusion, the 2-D polynomial $P_2(x,y)$ is used for imposition of boundary conditions according to the expressions (16) and (17). Remaining external Fup coefficients in (A.2.12) are calculated in the post processing step in order to define complete solution on the entire domain.

Acknowledgments

This research was funded by the Croatian Science Foundation (in Croatian: Hrvatska zaklada za znanost -HRZZ) through the scientific project "Multiphysics modelling of surface-subsurface water systems", grant number: IP-2020-02-2298.

This research is partially supported through project KK.01.1.1.02.0027, a project co-financed by the Croatian Government and the European Union through the European Regional Development Fund - the Competitiveness and Cohesion Operational Program.

References:

[1] Zienkiewicz, Olgierd Cecil, and P. B. Morice. *The finite element method in engineering science*. 1971, Vol. 1977. London: McGraw-hill.

[2] E. Hinton and D. R. Owen, AN INTRODUCTION TO FINITE ELEMENT COMPUTATIONS, 1979, Pineridge Press, Swansea,. 385 pp.

[3] Hughes, Thomas JR. *The finite element method: linear static and dynamic finite element analysis*. 2003, Courier Corporation.

[4] T.J.Hughes, J.A.Cottrell and Y.Bazilevs,"Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement", *Computer Methods in Applied Mechanics and Engineering*, 2005, vol.194,no.39-41,pp.4135 4195, ISSN:00457825.DOI:10.1016/j.cma.2004.10.008.

[5] J.A.Cottrell, T.J.R.Hughes and Y.Bazilevs, "Isogeometric Analysis Toward Intergration of CAD and FEA", 2009, p.335.

[6] Peskin CS. Flow patterns around heart valves: a numerical method. J Comput Phys. 1972. 10(2): 252–271.

[7] Höllig K, Reif U, Wipper J. Weighted extended B-spline approximation of Dirichlet problems. SIAM J Numer Anal., 2001, 39(2):442–462.

[8] de Prenter F, Verhoosel CV, van Brummelen EH, Larson MG, Badia S. Stability and conditioning of immersed finite element methods: analysis and remedies. Archives of Computational Methods in Engineering. 2023 Jul;30(6):3617-56.

[9] Rvachev, V.L.; Theory of R-functions and Some Applications; Naukova Dumka; Kiev; 1982.

[10] Rvachev, V. L., and Sheiko, T. I. *R*-Functions in Boundary Value Problems in Mechanics. ASME. *Appl. Mech. Rev*. 1995. 48(4): 151–188. https://doi.org/10.1115/1.3005099

[11] Freytag M, Shapiro V, Tsukanov I. Field modeling with sampled distances. Computer-Aided Design. 2006 Feb 1;38(2):87-100.

[12] Tsukanov, I., and Posireddy, S. R. (July 22, 2011). "Hybrid Method of Engineering Analysis: Combining Meshfree Method with Distance Fields and Collocation Technique." ASME. *J. Comput. Inf. Sci. Eng*. September 2011; 11(3): 031001. https://doi.org/10.1115/1.3572035
[13] Kozulić, V. and Gotovac, B. Application of the Solution Structure Method in Numericall Solving Poisson's Equation on the Basis of Atomic Functions. Int. J. Comput. Methods; 2018; 15(5); 1-25.
[14] Höllig K. Finite element methods with B-splines. Society for Industrial and Applied Mathematics. 2003, Jan 1.
[15] Höllig K, Apprich C, Streit A. Introduction to the Web-method and its applications. Advances in Computational Mathematics. 2005. 215-37.
[16] Sanches RAK, Bornemann PB, Cirak F. Immersed b-spline (i-spline) finite element method for geometrically complex domains. Comput Methods Appl Mech Eng. 2011. 200(13–16):1432–1445.
[17] Parvizian J, Düster A, Rank E. Finite cell method. Comput Mech. 2007. 41:121–133.
[18] Schillinger D, Ruess M. The finite cell method: a review in the context of higher-order structural analysis of CAD and image-based geometric models. Arch Comput Methods Eng. 2015. 22:391–455.
[19] Hansbo A, Hansbo P. An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. Comput Methods Appl Mech Eng. 2002. 191(47–48):5537–5552.
[20] Burman E, Hansbo P. Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. Appl Numer Math. 2012. 62(4):328–341.
[21] Burman E, Claus S, Hansbo P, Larson MG, Massing A. CutFEM: discretizing geometry and partial differential equations. Int J Numer Meth Eng. 2015. 104(7):472–501.
[22] Kamensky D, Hsu M-C, Schillinger D, Evans JA, Aggarwal A, Bazilevs Y, Sacks MS, Hughes TJR. An immersogeometric variational framework for fluid-structure interaction: application to bioprosthetic heart valves. Comput Methods Appl Mech Eng. 2015. 284:1005–1053.
[23] Main A, Scovazzi G. The shifted boundary method for embedded domain computations. Part II: linear advection-diffusion and incompressible Navier-Stokes equations. J Comput Phys. 2018. 372:996–1026.
[24] Main A, Scovazzi G. The shifted boundary method for embedded domain computations. Part I: Poisson and Stokes problems. J Comput Phys. 2018. 372:972–995.
[25] Atallah NM, Canuto C, Scovazzi G. Analysis of the shifted boundary method for the Poisson problem in domains with corners. Math Comput. 2021. 90:2041–2069.
[26] Zorrilla R, Rossi R, Scovazzi G, Canuto C, Rodríguez-Ferran A. A shifted boundary method based on extension operators. Computer Methods in Applied Mechanics and Engineering. 2024. Mar 1;421:116782.
[27] Kansa. E. J. Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics—II solutions to parabolic, hyperbolic and elliptic partial differential equations. Computers & Mathematics with Applications. 1990. vol. 19, pp. 147–161.
[43] Gunter, F.C. and Liu, W.K.; Implementation of boundary conditions for meshless methods; Comput. Methods Appl. Mech. Eng.; 1998; 163; 205–230.

[2] F. Auricchio, L. Beirão da Veiga, T. J. R. Hughes, A. Reali, and G. Sangalli. Isogeometric collocation methods. Math. Models Methods Appl. Sci., 20(11):2075-2107, 2010.
[5] D. Schillinger, J. A. Evans, A. Reali, M. A. Scott, and T. J.R. Hughes. Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations. Comput. Methods Appl. Mech. Engrg., 267:170-232, 2013.
[6] Höllig, K., Reif, U. i Wipper, J.; Weighted extended B–spline approximation of Dirichlet problems; SIAM J. Numer. Anal.; 2001; 39(2); 442–462.
[7] Apprich, C., *et al*. (2016). Collocation with WEB-Splines. *Adv. Comput. Math.*, **42**: 823–842.
[8] Kozulić, V. i Gotovac, B.; Application of the Solution Structure Method in Numerically Solving Poisson's Equation on the Basis of Atomic Functions; Int. J. Comput. Methods; 2018; 15(5); 1-25.

[9] Gotovac, B. i Kozulić, V.; On a selection of basis functions in numerical analyses of engineering problems; Int. J. Eng. Model.; 1999; 12(1–4); 25–41.
[10] Kozulić, V. i Gotovac, B.; Elasto-Plastic Analysis of Structural Problems Using Atomic Basis Functions; Computer modeling in engineering & sciences; 2011; 80(3-4); 251-274.
[11] Rvachev, V.L.; Theory of R-functions and Some Applications; Naukova Dumka; Kiev; 1982.
[12] Rvachev, V. L. and Rvachev, V. A. (1971). On a finite function. *Dokl. Akad. Nauk Ukrainian SSR*, ser. A, pp. 705.
[13] Rvachev, V. L. and Rvachev, V. A. (1979). *Non-classical Methods for Approximate Solution of Boundary-Value Problems*, Naukova dumka, Kiev.
[18] H. Gomez and L. De Lorenzis. The variational collocation method. Comput. Methods Appl. Mech. Engrg., 309:152-181, 2016.