



Márton Pető · Fabian Duvigneau · Daniel Juhre ·
Sascha Eisenträger

Enhanced numerical integration scheme based on image compression techniques: Application to rational polygonal interpolants

Received: 15 May 2020 / Accepted: 28 August 2020 / Published online: 12 September 2020
© The Author(s) 2020

Abstract Polygonal finite elements offer an increased freedom in terms of mesh generation at the price of more complex, often rational, shape functions. Thus, the numerical integration of rational interpolants over polygonal domains is one of the challenges that needs to be solved. If, additionally, strong discontinuities are present in the integrand, e.g., when employing fictitious domain methods, special integration procedures must be developed. Therefore, we propose to extend the conventional quadtree-decomposition-based integration approach by image compression techniques. In this context, our focus is on unfitted polygonal elements using Wachspress shape functions. In order to assess the performance of the novel integration scheme, we investigate the integration error and the compression rate being related to the reduction in integration points. To this end, the area and the stiffness matrix of a single element are computed using different formulations of the shape functions, i.e., global and local, and partitioning schemes. Finally, the performance of the proposed integration scheme is evaluated by investigating two problems of linear elasticity.

Keywords Fictitious domain methods · Polygonal finite element method · Polygonal finite cell method · Discontinuous integrands · Compressed quadtree-decomposition

1 Introduction

The traditional finite element method (FEM) offers a robust and well-studied approach for simulating a large variety of physical phenomena governed by partial differential equations [1, 2]. Nonetheless, throughout the years, several extensions have been developed in order to increase the accuracy, widen the field of application and decrease the computation time. In this contribution, we focus on the combination of two such extensions, namely the fictitious domain approach and polygonal elements employing shape functions based on generalized barycentric coordinates. In this context, we propose an efficient solution for computing piece-wise rational integrals arising in the expressions for the element matrices.

1.1 Fictitious domain methods

In the conventional FEM, the computational mesh has to conform to the boundary of the domain of interest. As this is one particular bottleneck in the simulation pipeline, extensive research has been devoted to extending

M. Pető (✉) · F. Duvigneau · D. Juhre
Otto von Guericke University Magdeburg, Magdeburg, Germany
E-mail: marton.petoe@ovgu.de

S. Eisenträger
University of New South Wales, Sydney, Australia

the FEM to unfitted discretizations that do not necessarily conform to geometric features of the given problem. One possibility to exploit this idea is utilized in the extended finite element (XFEM) [3,4] and the generalized finite element methods (GFEM) [4,5], where, e.g., an accurate modeling of crack propagation can be achieved. On the other hand, in the context of fictitious domain methods, geometrically highly complex domains are embedded into a larger computational domain of simple shape to achieve a straightforward mesh generation process [6–8]. Representatives of this class of methods are the cut finite element method (cutFEM) [9–12], the Cartesian grid finite element method (cgFEM) [13], the fixed grid finite element method (Fg-FEM) [14,15] and the finite cell method (FCM) [16–19]. All of these approaches generally lead to elements intersected by geometric features, and therefore, the application of unfitted meshes inherently involves the computation of discontinuous integrals. Thus, an accurate computation of the element matrices of intersected elements is one of the key factors determining the success of this class of methods.

1.2 Polytopal finite element method

The conventional FEM is based on finite elements of simple shapes, such as triangles and quadrilaterals in 2D and tetrahedrons and hexagons (less often pyramids and wedges) in 3D. The usage of polygonal and polyhedral finite elements offers additional advantages when it comes to enhanced mesh generation procedures, such as transition elements for avoiding hanging nodes [20,21] and the derivation of conformal polygonal elements in an originally non-conformal mesh [22–24]. Furthermore, polygonal elements enable a straightforward discretization and modeling of polycrystalline [25] and rock materials [26]. In the remainder, we follow the theory of conformal polygonal finite element methods based on generalized barycentric coordinates [27,28]. However, we have to keep in mind that other polygonal/polyhedral element formulations based on, e.g., the Voronoi cell finite element method (VCFEM) [25,29], the virtual element method (VEM) [30,31] and the scaled boundary finite element method (SBFEM) [32–34], are also possible. An extensive description of these and other related approaches can be found in the review article by Perumal [35]. The main challenges of the polytopal FEM can be seen in the lack of commercial and free mesh generation software, construction of appropriate (high-order) interpolants and in the need for accurate numerical integration schemes.

1.3 Motivation

The investigation of polygonal elements in the context of fictitious domain concept is an exotic combination which further widens the application fields of these methods. A useful feature of such a combination is, e.g., the flexible insertion and modification of voids and inclusions in already existing FE-meshes, consisting completely or partially of polygonal elements. Non-conformal polygonal meshes were already investigated by Duczek and Gabbert [36] in the context of the polygonal finite cell method (poly-FCM). In this case, multiple challenges have to be tackled regarding the numerical integration due to (i) the polygonal shape of the integration domain, (ii) the strong discontinuity within the element and (iii) the generally non-polynomial integrand over the element domain, computed by the linear combination of polygonal basis functions. In the remainder of this article, we especially consider rational shape functions, such as the Wachspress interpolant [37]. Consequently, the computation of the element matrices in unfitted polygonal meshes involves integrals of the following form

$$\int_{\Omega^{\text{poly}}} \alpha \mathcal{R} \, d\Omega, \quad (1)$$

where Ω^{poly} is a polygonal region, \mathcal{R} a rational function and α a step function making the integrand discontinuous. In this contribution, we follow the poly-FCM approach by Duczek and Gabbert [36] which uses a quadtree-decomposition-based integration scheme for computing Eq. (1) over the intersected elements. As an extension to their approach, we implement and investigate the novel numerical integration scheme proposed by Petö et al. [38] that compresses the integration sub-cells resulting from the quadtree-decomposition (QTD) to achieve reduced computational times. Since this latter approach leads to a lower integration point density in the polygonal elements, a special focus is placed on the integration accuracy when computing Eq. (1). Furthermore, note that similar discontinuous rational integrands arise also in case of traditional finite elements when they are (i) distorted and (ii) intersected by domain interfaces. Therefore, the findings regarding the effect of the compressed QTD-based integration scheme have ramifications beyond the scope of polygonal elements.

The remainder of this article is structured in the following way: First, in Sect. 2, the fictitious domain approach and the discretization of the weak form via an unfitted polygonal mesh is outlined in the context of linear elasticity. Then, in Sect. 3, the polygonal shape functions are discussed briefly, while in Sect. 4 the different integration schemes over polygonal elements are elaborated with special attention to discontinuous integrands (see Ref. [38]). Finally, in Sects. 5, 6 and 7, the influence of the proposed integration scheme on (i) the resulting number of integration points, (ii) the reduction in computational time and (iii) the integration accuracy is investigated by means of various examples.

2 Fictitious domain approach for polygonal elements

Consider the weak form of a second-order boundary value problem over the domain Ω with its corresponding boundary $\partial\Omega$ [2,39]

$$\mathcal{B}(\mathbf{u}, \mathbf{v}) = \mathcal{F}(\mathbf{v}), \quad (2)$$

where in case of linear elasticity, \mathbf{u} is the displacement field and \mathbf{v} is a test function. Denoting the elasticity matrix in Voigt notation by \mathbf{C} , the standard strain-displacement operator by \mathbf{L} , the body forces in Ω by \mathbf{b} and the surface tractions on the Neumann boundary $\Gamma_N \subset \partial\Omega$ by $\hat{\mathbf{t}}$, the bilinear \mathcal{B} and linear \mathcal{F} functionals are expressed as

$$\mathcal{B}(\mathbf{u}, \mathbf{v}) = \int_{\Omega} (\mathbf{L}\mathbf{v})^T \cdot \mathbf{C} \cdot \mathbf{L}\mathbf{u} \, d\Omega \quad \text{and} \quad (3)$$

$$\mathcal{F}(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{v}^T \cdot \mathbf{b} \, d\Omega + \int_{\Gamma_N} \mathbf{v}^T \cdot \hat{\mathbf{t}} \, d\Gamma. \quad (4)$$

Note that the Neumann boundary conditions are already incorporated in Eq. (4), while the Dirichlet boundary conditions, i.e., the prescribed displacements $\hat{\mathbf{u}}$, are realized by an additional condition $\mathbf{u}(\mathbf{x}) = \hat{\mathbf{u}}(\mathbf{x}) \, \forall \mathbf{x} \in \Gamma_D$.

2.1 Fundamentals

In fictitious domain methods, the original problem stated in Eq. (2) is not solved over the domain Ω , but over an extended domain $\Omega_e \supset \Omega$. In the following, we refer to the original domain as the *physical domain* Ω_{phys} , while the domain extending Ω_{phys} to Ω_e we call *fictitious domain* $\Omega_{\text{fict}} = \Omega_e \setminus \Omega_{\text{phys}}$ (see Fig. 1). There are numerous methods that are based on this approach. Here, we follow the general description used in the finite cell method (FCM) [16–19], for which a first extension to polygonal meshes (poly-FCM) was already outlined by Duzcek and Gabbert [36]. Assuming that Ω_{fict} provides zero stiffness and that it is not subjected to any body forces and surface tractions, the weak form over the extended domain reads [16,40]

$$\mathcal{B}_e(\mathbf{u}, \mathbf{v}) = \mathcal{F}_e(\mathbf{v}), \quad (5)$$

where the bilinear and linear functionals

$$\mathcal{B}_e(\mathbf{u}, \mathbf{v}) = \int_{\Omega_e} (\mathbf{L}\mathbf{v})^T \cdot \alpha \mathbf{C} \cdot \mathbf{L}\mathbf{u} \, d\Omega \quad \text{and} \quad (6)$$

$$\mathcal{F}_e(\mathbf{v}) = \int_{\Omega_e} \mathbf{v}^T \cdot \alpha \mathbf{b} \, d\Omega + \int_{\Gamma_N} \mathbf{v}^T \cdot \hat{\mathbf{t}} \, d\Gamma \quad (7)$$

are formulated with the help of the indicator function α given in Eq. (8). In order to avoid severe ill-conditioning of the system matrices, a small value $\alpha \ll 1$ is assigned to the integration points in Ω_{fict} [41]. According to Fries and Belytschko [4], such a possible ill-conditioning is expected only for cut elements that exhibit a large difference between the areas of Ω_{phys} and Ω_{fict} .

$$\alpha(\mathbf{x}) = \begin{cases} 1.0, & \forall \mathbf{x} \in \Omega_{\text{phys}} \\ 0.0 \text{ (in practice : } \alpha \ll 1), & \forall \mathbf{x} \in \Omega_{\text{fict}} \end{cases} \quad (8)$$

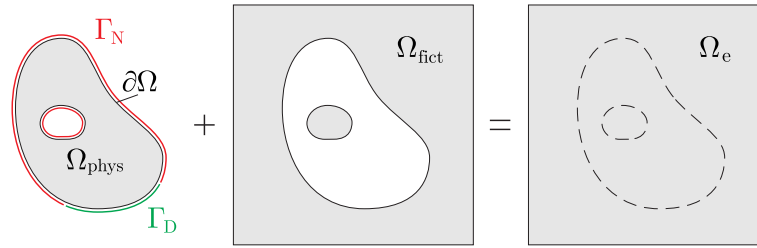


Fig. 1 Fictitious domain concept

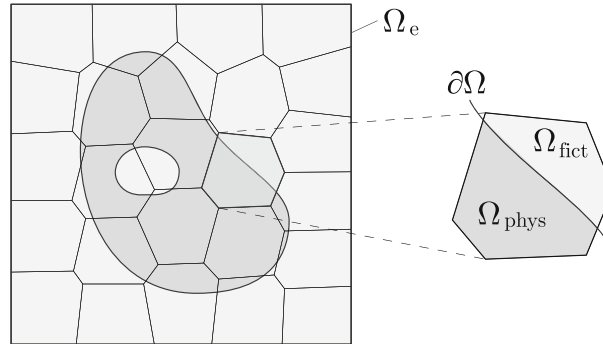


Fig. 2 Unfitted polygonal mesh over the extended domain Ω_e

2.2 Discretization of the weak form

The discretization of the weak form over the extended domain Ω_e generally results in an unfitted mesh where the elements are located either completely inside the physical or fictitious domains, or they are intersected by the boundary $\partial\Omega$. While fictitious elements can be discarded, physical elements are treated as standard polygonal finite elements. The bottleneck of the unfitted discretization is seen in the cut elements (Fig. 2), where (i) the implementation of the boundary conditions and (ii) the computation of element matrices require special care. For more information regarding the first issue, we refer to Refs. [7, 8, 42, 43]. The computation of the cut element matrices—constituting the main focus of this article—will be discussed in greater detail in Sect. 4.2.

Similar to the standard FEM [1, 2], the fields $\mathbf{u}(\mathbf{x})$ and $\mathbf{v}(\mathbf{x})$ are approximated by a linear combination of the shape functions $\{N_i^{\text{poly}}(\boldsymbol{\xi})\}_{i=1}^{n_v}$ (see Sect. 3) and nodal values $\{\mathbf{u}_i\}_{i=1}^{n_v}$ and $\{\mathbf{v}_i\}_{i=1}^{n_v}$ over each polygonal element

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{n_v} N_i^{\text{poly}}(\boldsymbol{\xi}) \mathbf{u}_i \quad \text{and} \quad \mathbf{v}(\mathbf{x}) = \sum_{i=1}^{n_v} N_i^{\text{poly}}(\boldsymbol{\xi}) \mathbf{v}_i. \quad (9)$$

Then, following the Bubnov–Galerkin approach, the global system of equations is assembled based on the individual element contributions. The system of equations given in Eq. (10), where \mathbf{K} , \mathbf{U} and \mathbf{F} are the global stiffness matrix, nodal displacement vector and nodal force vector, respectively, is solved by standard approaches

$$\mathbf{K}\mathbf{U} = \mathbf{F}. \quad (10)$$

3 Polygonal shape functions

Let us consider a two-dimensional polygonal element with n_v vertices $\{\mathbf{x}_i\}_{i=1}^{n_v}$ given in the global Ω_x^{poly} space. Each global polygonal element Ω_x^{poly} has its corresponding reference space, the so-called *canonical* polygonal element Ω_ξ^{poly} defined in the local coordinate system $\boldsymbol{\xi} = [\xi_1, \xi_2]^T$, as depicted in Fig. 3. The domain Ω_ξ^{poly}

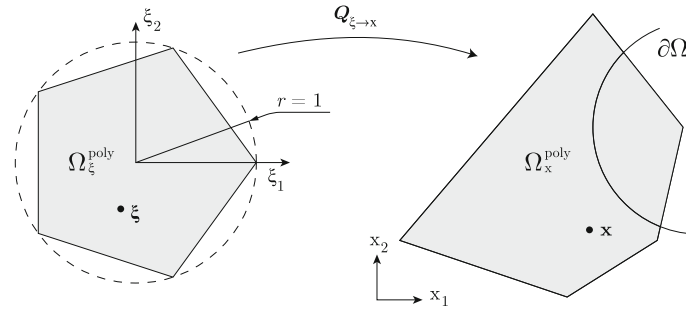


Fig. 3 A polygonal element in its local $\Omega_{\xi}^{\text{poly}}$ and global Ω_x^{poly} configurations transformed by the geometry mapping $Q_{\xi \rightarrow x}$ given in Eq. (17)

is a regular polygon, where the coordinates of the i^{th} vertex are defined on a unit circle by

$$\xi_{1i} = \cos(2\pi \cdot i/n_v) \quad \text{and} \quad \xi_{2i} = \sin(2\pi \cdot i/n_v). \tag{11}$$

Numerous efforts have been made in order to derive interpolants that can be used as shape functions for polygonal elements. These approaches differ in the dimension (polygon or polyhedron) and in the assumptions on their shapes (regular, irregular-convex or irregular-concave) as well as in their computational complexities. In this paper, only a short introduction is given to the available shape functions; for detailed information see Refs. [36,44]. In order to apply the shape functions in a finite element computation, they must fulfil the requirements for a Galerkin approximant, i.e.,

$$\text{non-negativeness : } 0 \leq \phi_i(\mathbf{x}) \leq 1 \tag{12}$$

$$\text{interpolation nodal data : } \phi_i(\mathbf{x}_j) = \delta_{ij} \tag{13}$$

$$\text{partition of unity : } \sum_{i=1}^{n_v} \phi_i(\mathbf{x}) = 1 \tag{14}$$

$$\text{linear completeness : } \sum_{i=1}^{n_v} \phi_i(\xi) x_i = \mathbf{x} \tag{15}$$

where ϕ_i denotes the interpolant corresponding to node i . Most approaches use generalized barycentric coordinates

$$\phi_i(\mathbf{x}) = \frac{w_i(\mathbf{x})}{\sum_{j=1}^{n_v} w_j(\mathbf{x})}, \tag{16}$$

as the basis for deriving suitable interpolants, where w_i is a weighting function associated with node i that can be computed in numerous ways. As a first approach, Wachspress proposed a rational basis for convex polygons [45], where w_i is computed based on area computations in the polygonal domain. A MATLAB implementation for this approach can be found in Ref. [46]. The original formulation of the Wachspress shape functions was later simplified by Meyer et al. [47] proposing a method that relies on vector operations only, thus reducing possible round-off errors. Finally, the extension of the Wachspress shape functions to convex polytopes was established by Warren [48]. Computing w_i based on the *mean value coordinates* (MVC), introduced by Floater [49] enables a formulation for both convex and concave polygonal domains. The MVC was extended to three dimensions as well [50,51]. Another approach for computing w_i is based on *natural element* or *Laplace coordinates* [27,52], both of which are widely used in the *natural element method*. Tabarraei and Sukumar [28] provided symbolically pre-computed Laplacian shape functions for regular polygons up to $n_v = 8$, over which they are identical to the Wachspress shape function [27]. A polygonal interpolant that is not based on Eq. (16) can be constructed by using the *maximum entropy* approach proposed by Sukumar [53,54], which has been also extended to quadratic Serendipity-type shape functions by Rand et al. [55].

In the remainder, we use the rational basis functions proposed by Wachspress as polygonal shape functions $N^{\text{poly}}(\xi) = \phi(\xi)$. These functions are defined in the reference space over the canonical element. Based on the

isoparametric concept, the geometry mapping from Ω_ξ^{poly} to Ω_x^{poly} (Fig. 3) is achieved by a linear combination of basis functions $\{N_i^{\text{poly}}(\xi)\}_{i=1}^{n_v}$ and nodal positions $\{\mathbf{x}_i = [x_{1i}, x_{2i}]^T\}_{i=1}^{n_v}$ in the global space

$$\mathbf{x} = \mathcal{Q}_{\xi \rightarrow x}(\xi) = \sum_{i=1}^{n_v} N_i^{\text{poly}}(\xi) \begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix}. \quad (17)$$

4 Numerical integration techniques for polygonal elements

Considering standard finite elements, quadrature rules are readily available both in two and three dimensions, while the integration over polytopal elements is more challenging. Note that the need for an accurate integration over polytopal domains does not only arise in FEMs using polygonal/polyhedral elements, but also in immersed boundary methods, where the intersected FEs often contain polytopal sub-domains due to the discontinuity. Numerous contributions have been made for developing accurate integration schemes for polynomial functions over such domains. One may transform the integral over the complicated polytopal domain into surface or line integrals using the divergence theorem [56–59], or derive special quadrature rules for the polytopal domains via solving the moment fitting equations [60–64,64]. A third approach, especially used in the polygonal FEM [27,28,36,65,66], is based on partitioning the polygonal elements into a set of triangular or quadrilateral sub-domains with readily available quadrature rules.¹ This approach is discussed in the next sub-section in greater detail.

4.1 Partitioning scheme

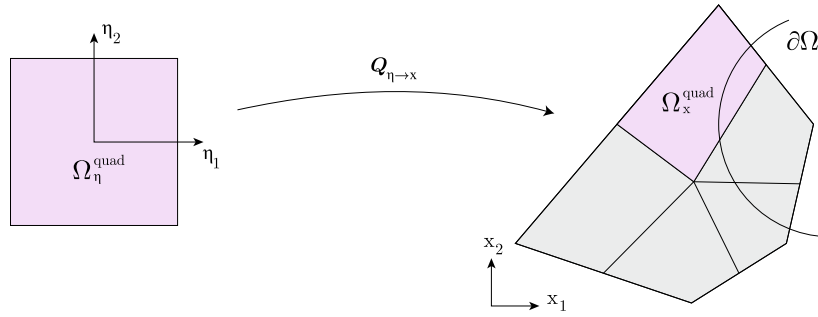
The most common approach regarding the sub-division for integration purposes is to split the polygonal element with n_v vertices into n_v triangular or quadrilateral sub-domains [36]. A third option is to partition the polygonal element into special triangular and quadrilateral regions that are distorted in such a way that only one or two adjacent edges are skewed and use the quadrature formulae derived by Sarada and Nagaraja [67]. In the following, we focus on the first two approaches due to their popularity and the possibility of a straightforward extension by using tetrahedra in 3D. The reference space for the triangular and quadrilateral sub-domains is denoted by Ω_η^{tri} and $\Omega_\eta^{\text{quad}}$, respectively. In this case, standard quadrature rules can be applied where $\boldsymbol{\eta} = [\eta_1, \eta_2]^T$ refers to the local coordinates. In general, the sub-division of a polygonal element can be executed either in the global or in the local space. If shape functions are easily constructed for the global polygon, e.g., when using Wachspress,² MVC or maximum entropy shape functions, it is a natural choice to perform the integration in the global element. On the other hand, if it is more advantageous to define the shape functions over Ω_ξ^{poly} , which is the case for the Laplace shape functions [27,66], the partitioning for integration purposes should be performed in the canonical element [28]. In the following, both the partitioning based on the global Ω_x^{poly} and local polygonal (canonical) elements Ω_ξ^{poly} are discussed.

4.1.1 Sub-division of the global polygonal element

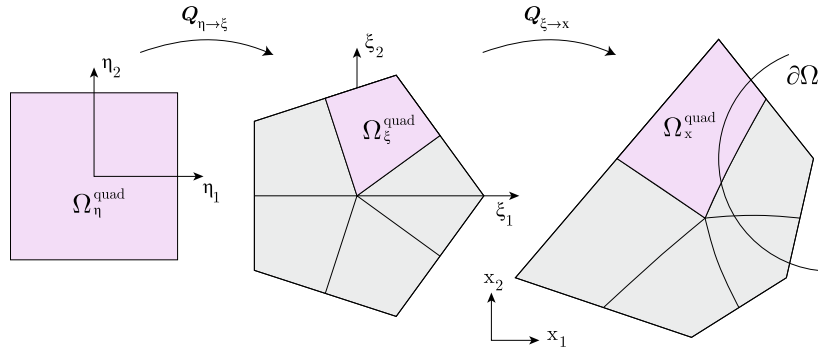
One approach is to directly sub-divide the global polygonal element Ω_x^{poly} into triangular Ω_x^{tri} or quadrilateral Ω_x^{quad} sub-domains [27,66]. The relation of these domains to their local spaces, Ω_η^{tri} and $\Omega_\eta^{\text{quad}}$, is established by means of a simple linear/bilinear geometry mapping $\mathbf{x} = \mathcal{Q}_{\eta \rightarrow x}(\boldsymbol{\eta})$, as depicted in Fig. 4a. Consequently, the integral of an arbitrary function \mathcal{F} over Ω_x^{poly} is computed for the triangular and quadrilateral decompositions by

¹ Note that this sub-division serves integration purposes only and, therefore, does not add any degrees of freedom (DOFs) to the system.

² Note that the Wachspress shape function requires Ω_x^{poly} to be a convex polygon.



(a) Sub-division of the polygonal element in the global space. The integration over the sub-element requires one mapping.



(b) Sub-division of the polygonal element in its reference space. The integration over the sub-element requires two mappings.

Fig. 4 Example for the global Ω_x^{poly} and local Ω_ξ^{poly} quadrilateral partitioning schemes. The domain with purple colour highlights a single quadrilateral sub-domain in different spaces using the geometry mappings indicated by the arrows. The decomposition into triangular sub-domains follows analogously. **a** Sub-division of the polygonal element in the global space. The integration over the sub-element requires one mapping. **b** Sub-division of the polygonal element in its reference space. The integration over the sub-element requires two mappings. (Color figure online)

$$\int_{\Omega_x^{\text{poly}}} \mathcal{F} d\Omega = \sum_{i=1}^{n_v} \int_0^1 \int_0^{1-\eta_1} \mathcal{F} |J_{\eta \rightarrow x}^{(i)}| d\eta_1 d\eta_2 \quad \text{and} \quad (18)$$

$$\int_{\Omega_x^{\text{poly}}} \mathcal{F} d\Omega = \sum_{i=1}^{n_v} \int_{-1}^1 \int_{-1}^1 \mathcal{F} |J_{\eta \rightarrow x}^{(i)}| d\eta_1 d\eta_2, \quad (19)$$

respectively, where $|J_{\eta \rightarrow x}^{(i)}|$ is the determinant of the Jacobian matrix $J_{\eta \rightarrow x}^{(i)}$ of the geometry mapping $\mathcal{Q}_{\eta \rightarrow x}$. Since $\mathcal{Q}_{\eta \rightarrow x}$ can be realized via standard linear/bilinear FE shape functions, the determinant of its Jacobian matrix is

$$|J_{\eta \rightarrow x}^{(i)}| = \begin{cases} \text{const.} & \text{for triangular sub-domain} \\ a_0 + a_1 \eta_1 + a_2 \eta_2 & \text{for quadrilateral sub-domain} \end{cases}, \quad (20)$$

where a_0 , a_1 and a_2 are constants depending on the coordinates of the corner vertices of the sub-domain $\Omega_x^{\text{quad}(i)}$.

4.1.2 Sub-division of the local polygonal element

Another approach is to perform the partitioning in the local space of the polygonal element Ω_ξ^{poly} , resulting in Ω_ξ^{tri} and Ω_ξ^{quad} [26–28,36,66]. In this case, the relation between Ω_ξ^{poly} and Ω_x^{poly} is established based on a geometry mapping $x = \mathcal{Q}_{\xi \rightarrow x}(\xi)$ using the rational polygonal shape functions according to Eq. (17).

Compared to the first approach, an additional second mapping $\xi = \mathcal{Q}_{\eta \rightarrow \xi}(\eta)$ from $\Omega_\eta^{\text{tri/quad}}$ to $\Omega_\xi^{\text{tri/quad}}$ is needed. Here, standard linear/bilinear shape functions are sufficient. An example for subdividing a polygonal element into quadrilateral sub-elements is depicted in Fig. 4b. Taking both geometry mappings into account, the integral of the function \mathcal{F} over Ω_x^{poly} can be transformed into

$$\int_{\Omega_x^{\text{poly}}} \mathcal{F} d\Omega = \int_{\Omega_\xi^{\text{poly}}} \mathcal{F} |\mathbf{J}_{\xi \rightarrow x}| d\Omega = \sum_{i=1}^{n_v} \int_0^1 \int_0^{1-\eta_1} \mathcal{F} |\mathbf{J}_{\eta \rightarrow \xi}^{(i)}| |\mathbf{J}_{\xi \rightarrow x}| d\eta_1 d\eta_2 \quad (21)$$

$$\int_{\Omega_x^{\text{poly}}} \mathcal{F} d\Omega = \int_{\Omega_\xi^{\text{poly}}} \mathcal{F} |\mathbf{J}_{\xi \rightarrow x}| d\Omega = \sum_{i=1}^{n_v} \int_{-1}^1 \int_{-1}^1 \mathcal{F} |\mathbf{J}_{\eta \rightarrow \xi}^{(i)}| |\mathbf{J}_{\xi \rightarrow x}| d\eta_1 d\eta_2 \quad (22)$$

for the triangular and quadrilateral decompositions, respectively. Considering the geometry mapping $\mathcal{Q}_{\eta \rightarrow \xi}^{(i)}$ for the sub-domain $i \in [1; n_v]$ with standard linear/bilinear FE shape functions, the determinant of its Jacobian matrix is the form

$$|\mathbf{J}_{\eta \rightarrow \xi}^{(i)}| = \begin{cases} \text{const.} & \text{for triangular sub-domain} \\ b_0 + b_1\eta_1 + b_2\eta_2 & \text{for quadrilateral sub-domain} \end{cases}, \quad (23)$$

where b_0, b_1 and b_2 are constants depending on the coordinates of the corner vertices of the sub-domain $\Omega_\xi^{\text{quad}(i)}$. Furthermore, due to the use of Wachspress shape functions for the geometry mapping, the determinant $|\mathbf{J}_{\xi \rightarrow x}|$ is a rational function regardless of the shape of the sub-domains.

Comparing Eqs. (18) and (19) to Eqs. (22) and (21), it is clear that the sub-division of the global element leads to integrands of lower complexity. Consequently, using the same quadrature rule in $\Omega_\eta^{\text{tri/quad}}$, Eqs. (18) and (19) can be computed with higher accuracy than Eqs. (22) and (21). On the other side, since Ω_ξ^{poly} is the same for all global polygonal elements, the shape functions can be written and computed more conveniently in the local space of the polygonal element. A comparison of the integration quality for the two partitioning schemes using different polygonal shape functions and number of integration points can be found in the work by Sukumar and Tabarraei [27]. In the context of linear elasticity, Tabarraei and Sukumar [28] use the triangular partitioning scheme in the local coordinate system, where in each triangle 25 integration points are used for an accurate computation of the integrals similar to Eq. (21).

4.2 Considering discontinuous integrands

The integration techniques discussed in the previous section allow for a simple and robust computation of continuous integrals over polygonal elements. However, if the integrand is discontinuous, more sophisticated integration techniques are required to guarantee highly accurate results. Discontinuous integrands generally arise in the *extended finite element method* (XFEM) [4, 68], the *generalized finite element method* (GFEM) [4, 5, 69], and in the *fictitious domains methods* [7, 8, 70], such as the *cut finite element method* (cutFEM) [9–12], *Cartesian grid finite element method* (cgFEM) [13], *fixed grid finite element method* (Fg-FEM) [14, 15] and *finite cell method* (FCM) [16–19]. There are numerous techniques that are commonly used in these methods for computing the discontinuous integrals. These techniques are, e.g., based on (i) the derivation of unique quadrature rules using *moment fitting* techniques [64, 71, 72], (ii) finding an *equivalent polynomial* that has the same integral value as the discontinuous integral [73, 74], (iii) reducing the dimensionality of the integral by applying the *divergence theorem* [57], or (iv) the construction of a *local integration mesh* (LIM) for better capturing the domains Ω_{phys} and Ω_{fict} in the cut element. The construction of an LIM consisting of boundary-conforming sub-regions has obvious benefits in terms of integration accuracy [12, 75–79]; however, we will concentrate on an LIM based on the *quadtree-decomposition* (QTD) of the intersected domains that does not conform to the boundary [17, 80]. Although this technique results in a poorer convergence, it does not require finding the discontinuity within the elemental domain often involving the solution of a nonlinear system of equations [77]. In our opinion, this approach fits better into the general philosophy of fictitious domain methods. Furthermore, it works robustly regardless of the geometry description that is provided and the dimensionality of the problem.

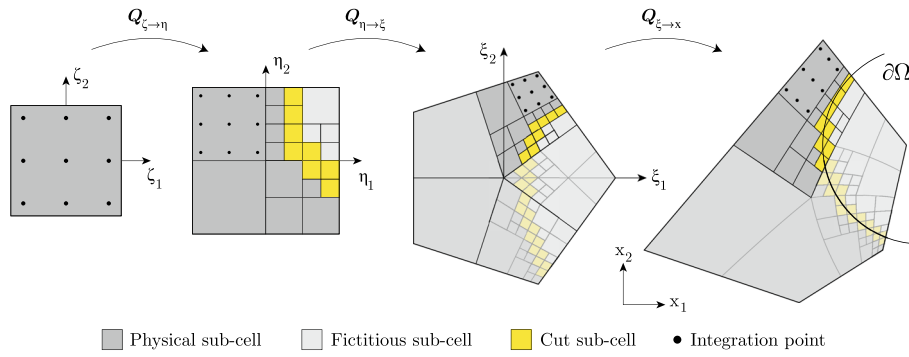


Fig. 5 Numerical integration in the cut polygonal element based on the quadtree-decomposition ($k = 3$) of the quadrilateral sub-domains in the canonical element

4.2.1 Quadtree-decomposition

Using the QTD in a two-dimensional setting, each triangular/quadrilateral sub-domain of the polygonal element undergoes a successive decomposition into a set of triangular/quadrilateral sub-domains called *sub-cells* with decreased size and increased density in the vicinity of the discontinuity $\partial\Omega$. Similar to the classification of the global elements, the resulting unfitted sub-cells can also be classified as physical, fictitious and cut as depicted in Fig. 5. The QTD algorithm with triangular sub-cells is outlined in Refs. [36, 81]. In the remainder, the focus is kept on the partitioning of the polygonal elements into quadrilateral sub-domains for which the compression algorithms are readily available (Sect. 4.2.2) [38]. The QTD is performed in the local-space of the cut quadrilateral sub-domains $\Omega_\eta^{\text{quad}}$ by repeating the following three steps for each cut sub-cell until the refinement level k of the QTD is reached:

1. Split the given sub-cell into four equal-sized quadrants representing four new sub-cells.
2. Map the four new sub-cells to the global polygonal element Ω_x^{poly} using:
 - (a) $\mathcal{Q}_{\eta \rightarrow x}$ (polygon partitioning was performed in Ω_x^{poly} ; Sect. 4.1.1),
 - (b) a combination of $\mathcal{Q}_{\eta \rightarrow \xi}$ and $\mathcal{Q}_{\xi \rightarrow x}$ (polygon partitioning was performed in Ω_ξ^{poly} ; Sect. 4.1.2).
3. Determine for each sub-cell whether it is cut by the boundary or not and mark the cut sub-cells for further decomposition. This procedure is executed in the global space.

Each created sub-cell is defined in its own local space Ω_ζ^{sc} with the coordinate system $\zeta = [\zeta_1, \zeta_2]^T$. The relation of the reference sub-cell to the corresponding sub-cell in the η -space Ω_η^{sc} is achieved by the geometry mapping $\eta = \mathcal{Q}_{\zeta \rightarrow \eta}(\zeta)$, as depicted in Fig. 5. Considering n_v quadrilateral domains in the polygonal element, each of which containing n_{sc} sub-cells resulting from the QTD, the integral of the piece-wise rational function $\alpha\mathcal{R}$ over the cut polygonal element Ω_x^{poly} is computed as the sum of integrals over the individual quadrilateral sub-cells by Eqs. (24) and (25) for the local and global partitions of an element, respectively

$$\int_{\Omega_x^{\text{poly}}} \alpha\mathcal{R} \, d\Omega = \sum_{i=1}^{n_v} \sum_{j=1}^{n_{\text{sc}}} \int_{-1}^1 \int_{-1}^1 \alpha\mathcal{R} \underbrace{|\mathbf{J}_{\zeta \rightarrow \eta}^{(j)}|}_{\text{const.}} \underbrace{|\mathbf{J}_{\eta \rightarrow \xi}^{(i)}|}_{\text{linear}} \underbrace{|\mathbf{J}_{\xi \rightarrow x}|}_{\text{rational}} \, d\zeta_1 d\zeta_2, \quad (24)$$

$$\int_{\Omega_x^{\text{poly}}} \alpha\mathcal{R} \, d\Omega = \sum_{i=1}^{n_v} \sum_{j=1}^{n_{\text{sc}}} \int_{-1}^1 \int_{-1}^1 \alpha\mathcal{R} \underbrace{|\mathbf{J}_{\zeta \rightarrow \eta}^{(j)}|}_{\text{const.}} \underbrace{|\mathbf{J}_{\eta \rightarrow x}^{(i)}|}_{\text{linear}} \, d\zeta_1 d\zeta_2. \quad (25)$$

While for each sub-cell $|\mathbf{J}_{\zeta \rightarrow \eta}^{(j)}|$ has a constant value due to the simple geometry mapping $\mathcal{Q}_{\zeta \rightarrow \eta}$ [17], the non-constant terms are functions of the mapped coordinates. During the numerical integration in each sub-cell, a Gaussian quadrature rule is applied using $n \times n$ integration points as depicted for a single sub-cell in Fig. 5 with $n = 3$.

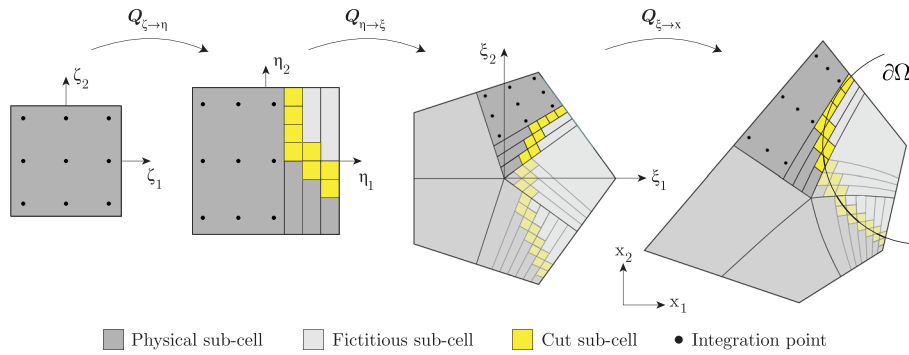


Fig. 6 Numerical integration in the cut polygonal element based on the IBR compression technique (same example as depicted in Fig. 5)

4.2.2 Sub-cell compression schemes

Although the QTD-based integration scheme allows for a robust and fully automatic computation of discontinuous integrals, its main drawback is seen in the exponentially increasing number of sub-cells as the refinement level k increases. This leads to high numerical costs both for performing the QTD itself and for computing the integrals over the sub-cells [82], especially for nonlinear problems [83]. As a solution to this problem, Petö et al. [38] introduced a novel approach based on image compression techniques for reducing the number of integration points and consequently the computational time. In the following, the main idea of this method is briefly discussed; for further information, we refer to the original article [38]. The compression algorithm is inserted directly between the QTD and the numerical integration stages as an intermediate step and requires only minor modification in an already existing code. There are several algorithms that can be used for the compression of two-dimensional sub-cells, such as the *run-length encoding* (RLE), *image block representation* (IBR) and *minimal rectangular partition* (MRP) [84–86]. Among these methods, the IBR (see Fig. 6) is the most suitable choice combining high compression rates, a low computational overhead and low implementational effort. If the QTD-based integration is used, the discontinuity in the integrals is captured by agglomerating integration points in the vicinity of the boundary. For the sake of consistency, the cut sub-cells are generally not subjected to the compression step.

Let the compression rate $\lambda = n_{\text{IP}}^{\text{after}}/n_{\text{IP}}^{\text{before}}$ define the ratio of the number of integration points before $n_{\text{IP}}^{\text{before}}$ and after $n_{\text{IP}}^{\text{after}}$ the compression, where a successful compression is indicated by λ -values below one. Distributing $n \times n$ integration points in the physical, fictitious and cut sub-cells, the compression ratio λ for the IBR algorithm is depicted by the black dashed curve in Fig. 7 based on the example shown in Fig. 6. Note that already for smaller refinement levels k , a significant compression can be achieved ($\lambda \approx 0.7$ for $k = 3$) and that λ is independent from n . However, according to Abedian et al. [83, 87], the fictitious integration points, whose only purpose is to prevent a possible ill-conditioning of the system matrices, can be replaced by a significantly smaller set of integration points. Following this *reduction in fictitious integration points* (RFIP) approach and only considering the physical integration points after the compression, even better results can be achieved, requiring only 30–40% of the original integration points, as depicted by the solid lines in Fig. 7.

In Ref. [38], a piece-wise polynomial integrand is assumed for which it was shown, that compressing only the physical sub-cells but not the cut ones, the same accuracy can be obtained as with the standard QTD-based integration scheme; however, with significantly less computational time. If the discontinuous integrand is not a piece-wise polynomial but a piece-wise rational function, the integration accuracy of the Gaussian quadrature depends on the density of the integration points in all integration sub-domains. That is to say, it is not enough to monitor the integration regions where the integrand is discontinuous (cut sub-cells), but it is equally important to guarantee an accurate integration in the physical sub-cells, where the integrand is continuous. Therefore, when compressing the physical sub-cells, a deterioration of the integration accuracy is expected due to the rational nature of the integrand function. The actual effect due to the compression is examined in detail in Sects. 5, 6, and 7.

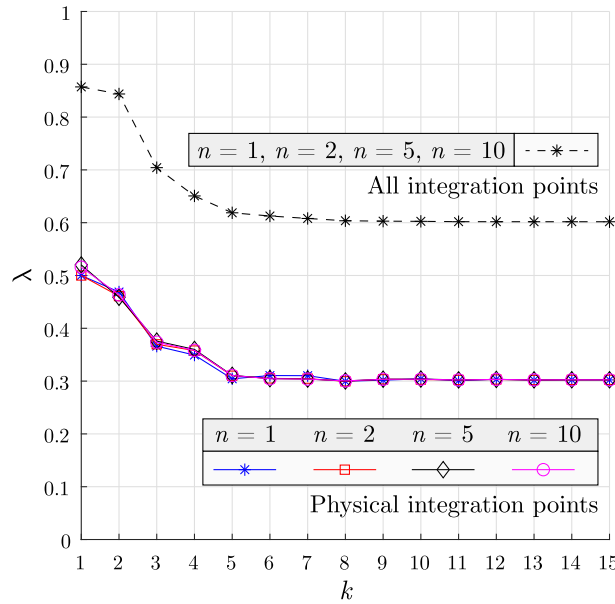


Fig. 7 Compression rates λ for the IBR algorithm when considering either all integration points in the polygonal element after the compression (black dashed line) or only the physical ones (solid lines)

5 Integration error

The integration error introduced when computing the integrals in Eqs. (24) and (25) is caused by the fact, that neither rational nor discontinuous functions can be integrated exactly via standard Gaussian quadrature rules. In the following, the errors in integration accuracy caused by the rational and discontinuous nature of the integrands are denoted by $e_{\mathcal{R}}$ and $e_{\mathcal{D}}$, respectively. Using a QTD-based integration scheme, for the integral over all sub-cells

$$I(e_{\mathcal{R}}, e_{\mathcal{D}}) = I_{\text{phys}}(e_{\mathcal{R}}) + I_{\text{cut}}(e_{\mathcal{R}}, e_{\mathcal{D}}) \tag{26}$$

holds, where I_{phys} and I_{cut} denote the integrals over the physical and cut sub-cells, respectively. Note that while I_{phys} only depends on $e_{\mathcal{R}}$, for I_{cut} both $e_{\mathcal{R}}$ and $e_{\mathcal{D}}$ are relevant. Since the compressed sub-cells generally have a decreased integration point density compared to the uncompressed ones (see Figs. 5 and 6), using the same integration order results in an increased error $e_{\mathcal{R}}$ when compression is used. However, following the approach by Petö et al. [38] and only compressing the physical sub-cells, this affects I_{phys} only while I_{cut} is unchanged. In order to investigate the influence of the compression (particularly based on the IBR algorithm) on the integration accuracy, in Sects. 5.1 and 5.2 the area and stiffness matrix of the cut polygonal element depicted in Figs. 5 and 6 are computed.³ The vertices V of the distorted polygonal element in the global space as well as the center \mathbf{x}_c of the unit circle ($r = 1$) introducing the discontinuity are given as:

$$V = \left\{ \begin{bmatrix} 2.7 \\ 2.0 \end{bmatrix}, \begin{bmatrix} 2.0 \\ 3.0 \end{bmatrix}, \begin{bmatrix} 0.5 \\ 1.0 \end{bmatrix}, \begin{bmatrix} 1.8 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 2.5 \\ 1.0 \end{bmatrix} \right\}, \mathbf{x}_c = \begin{bmatrix} 3 \\ 2 \end{bmatrix}. \tag{27}$$

Furthermore, since partitioning the polygonal element in the global and local spaces (see Sects. 4.1.1 and 4.1.2) generally leads to different integral formulations, the effect of the IBR compression is examined for both settings using quadrilateral sub-domains. Throughout the simulations in this article rational Wachspress shape functions [27] are used, which can be also formulated directly over the convex global element enabling a simpler and less error-prone numerical integration.

³ The quality of the results depends on the degree of distortion of the polygonal element and not its number of vertices. Nonetheless, the qualitative characteristics of the curves presented in this section are generally valid.

5.1 Numerical example: Error in the area approximation

The approximate solution for the area A is obtained via Eqs. (24) and (25), where for the current example $\mathcal{R} = 1$ applies. Based on the model of the single polygonal element given in Eq. (27), a reference solution for the area can be computed analytically resulting in $A_{\text{ref}} = 2.286616571808802$ which can be used for computing the error e_A by Eq. (28), where in case of a QTD-based integration $A = A_{\text{phys}} + A_{\text{cut}}$ holds

$$e_A = \left| \frac{A_{\text{ref}} - A}{A_{\text{ref}}} \right| \times 100\% = \left| 1 - \frac{A_{\text{phys}}}{A_{\text{ref}}} - \frac{A_{\text{cut}}}{A_{\text{ref}}} \right| \times 100\%. \quad (28)$$

5.1.1 Local sub-division scheme

The approximation quality when using a local partitioning scheme is depicted in Fig. 8 for various refinement levels $k = 1, 2, \dots, 15$ and integration orders with $n = 1, 2, 5$ and 10 integration points per direction. Based on Eq. (28), the ratios

$$\rho_{\text{phys}} = (1 - A_{\text{phys}}/A_{\text{ref}}) \times 100\% \quad (29)$$

$$\rho_{\text{cut}} = A_{\text{cut}}/A_{\text{ref}} \times 100\%, \quad (30)$$

depicted in Fig. 8a and b are introduced. Both values show an exponential convergence as the number of refinement levels k is increased. In case of ρ_{cut} , the number of integration points per direction n plays a role for small values of k only, while for ρ_{phys} its effect is only significant for rather large values of k . Using $n = 1$ results in such a poor integration over the large physical sub-cells that the ratio ρ_{phys} cannot be further improved after a certain refinement level ($k = 4$), resulting in a poor convergence of the entire error e_A as depicted in Fig. 8c. Since ρ_{cut} is not affected by the compression, the increased error in e_A when a compression is used is directly correlated to ρ_{phys} . Nonetheless, not only that increasing n results in an improved convergence of ρ_{phys} and consequently of e_A in general, but also the small differences between the results obtained with and without compression vanish, even for slightly increased n , as depicted in Fig. 8d.

5.1.2 Global sub-division scheme

As a comparison, we perform the same computations for the second case, where the polygonal element Ω_x^{poly} is subdivided in the global space; see a Eq. (25). In this case, no rational mapping from Ω_ξ^{poly} to Ω_x^{poly} has to be considered. Therefore, when computing the area ($\mathcal{R} = 1$), the chosen rational polygonal shape function does not play any role. The accuracy of the computation of the area depends solely on how well $|\mathbf{J}_{\eta \rightarrow x}|$ in Eq. (19) can be integrated. Since $|\mathbf{J}_{\eta \rightarrow x}|$ is a linear function as given in Eq. 20, the accuracy of the Gaussian quadrature over the physical sub-cells is recovered already for a single integration point ($n = 1$) per sub-cell. Consequently, the same accuracy is obtained regardless of the number of integration points and whether a compression was performed or not. This can be seen in Fig. 9a where all curves for $n = 1, 2, 5$ and 10 are overlapping for the uncompressed and compressed sub-cells. Nonetheless, due to the discontinuous integrals over the cut sub-cells, the influence of n is still present as observed in Fig. 9b. Here, the relative error in the area approximation e_A improves for increased values of n .

5.2 Numerical example: Error in the stiffness matrix

In this sub-section, the integration accuracy for computing the stiffness matrix is investigated using the same cut polygonal element and integration approaches as in the previous section. For this computation, a homogeneous material with a Young's modulus of $E = 209.9 \text{ MPa}$ and a Poisson's ratio of $\nu = 0.29$ is chosen and a plane-stress state is assumed. In the following, we denote the polygonal shape functions defined over Ω_x^{poly} by $\hat{N}_i^{(\text{poly})}(\mathbf{x})$ while for the shape functions over Ω_ξ^{poly} the notation $N_i^{(\text{poly})}(\boldsymbol{\xi})$ is used. Applying the same notation to the stiffness matrices in the global and local formulations, the variables $\hat{\mathbf{K}}$ and \mathbf{K} are introduced, respectively. The components of $\hat{\mathbf{K}}$ typically contain integrals of the form

$$\hat{k} = \int_{\Omega_x^{\text{poly}}} \frac{\partial \hat{N}_i^{(\text{poly})}(\mathbf{x})}{\partial x_m} \frac{\partial \hat{N}_j^{(\text{poly})}(\mathbf{x})}{\partial x_n} d\Omega, \quad (31)$$

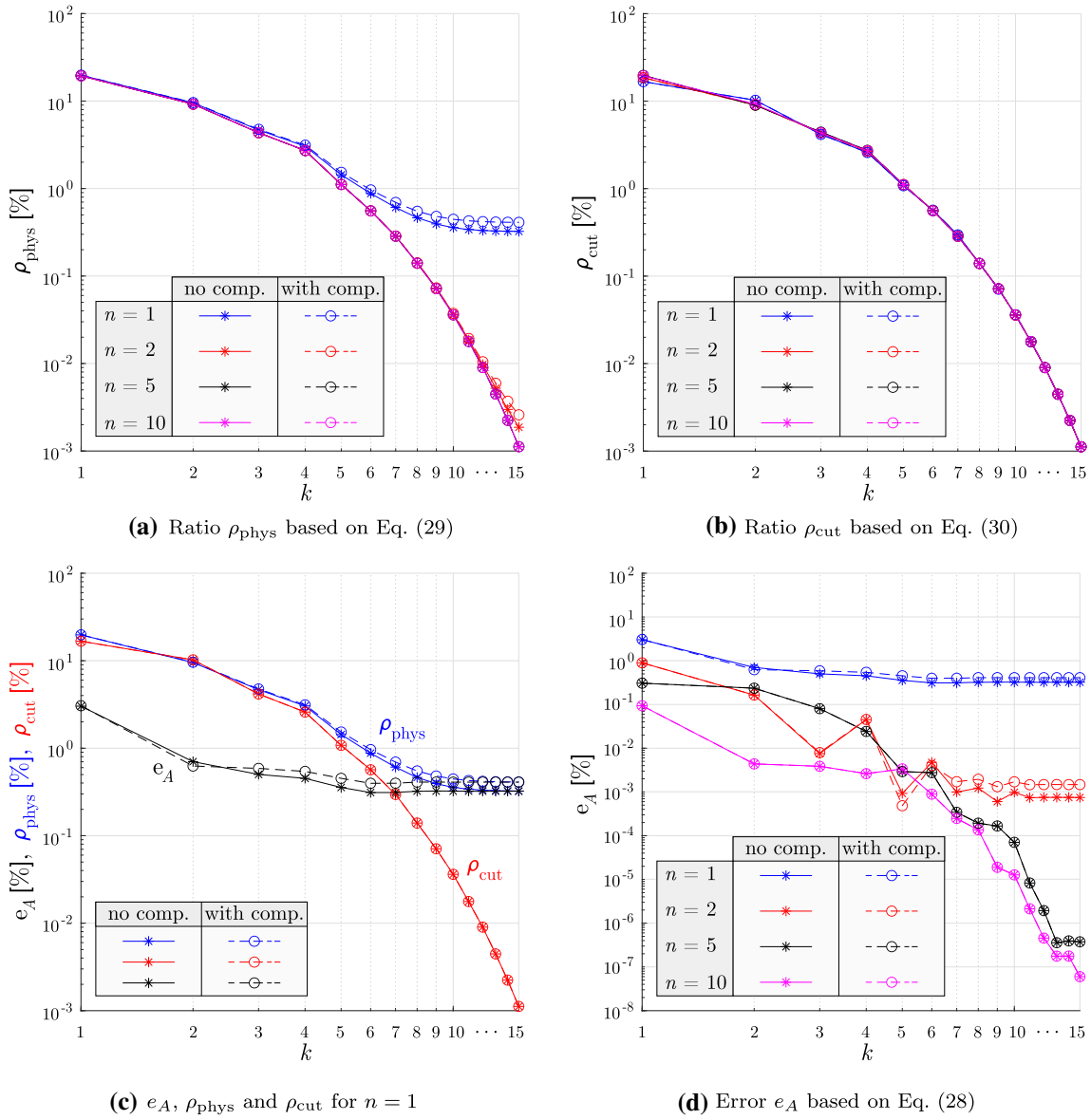


Fig. 8 Effect of the compression on the integration accuracy when computing the area of the physical domain in a cut polygonal element. The sub-division is performed in the local space. **a** Ratio ρ_{phys} based on Eq. (29). **b** Ratio ρ_{cut} based on Eq. (30). **c** e_A , ρ_{phys} and ρ_{cut} for $n = 1$. **d** Error e_A based on Eq. (28)

while the components of \mathbf{K} are expressed by more complex integrals due to the change in variables:

$$\begin{aligned}
 k &= \int_{\Omega_{\xi}^{\text{poly}}} \left(\frac{\partial \xi_1}{\partial x_m} \frac{\partial N_i^{(\text{poly})}(\xi)}{\partial \xi_1} + \frac{\partial \xi_2}{\partial x_m} \frac{\partial N_i^{(\text{poly})}(\xi)}{\partial \xi_2} \right) \\
 &\quad \times \left(\frac{\partial \xi_1}{\partial x_m} \frac{\partial N_j^{(\text{poly})}(\xi)}{\partial \xi_1} + \frac{\partial \xi_2}{\partial x_m} \frac{\partial N_j^{(\text{poly})}(\xi)}{\partial \xi_2} \right) |J_{\xi \rightarrow x}| d\Omega. \quad (32)
 \end{aligned}$$

In both cases, $i, j = 1, 2, \dots, n_v$ and $m, n = 1, 2$. The rational derivatives of the Wachspress shape functions can be found in Ref. [46], where the given formulae also apply to the global shape functions. In the following, for computing $\hat{\mathbf{K}}$ the global partition-based approach given in Eq. (25) is used, while for \mathbf{K} the integrals are solved via the local approach in Eq. (24). For evaluating the convergence in the stiffness matrices, we use the formula [2]

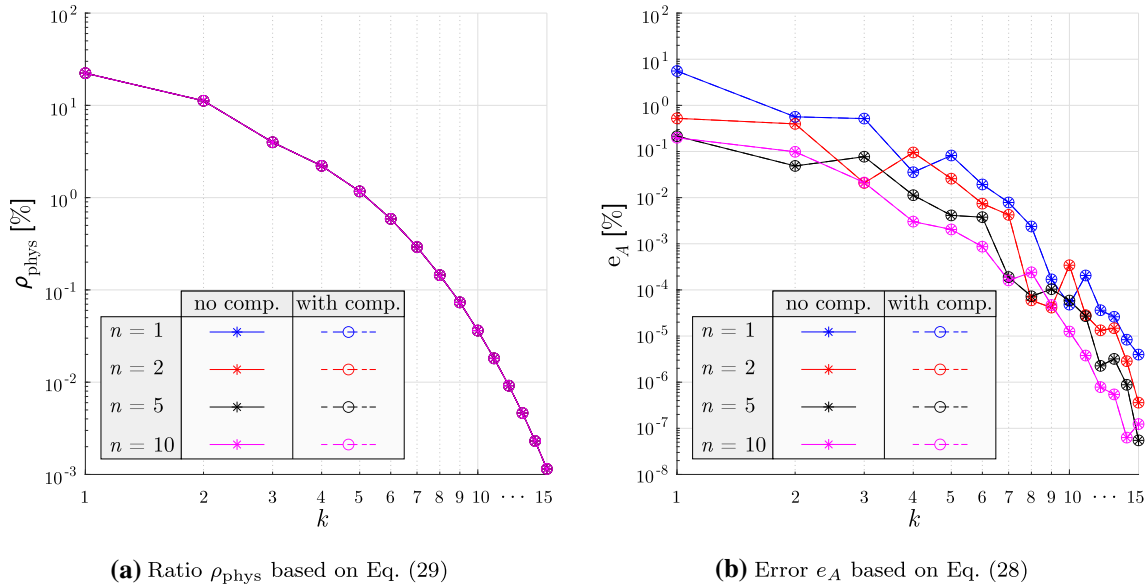


Fig. 9 Ratio ρ_{phys} and error e_A for the global sub-division scheme. **a** Ratio ρ_{phys} based on Eq. (29). **b** Error e_A based on Eq. (28)

$$e_K = \frac{\|\mathbf{K}_{\text{ref}} - \mathbf{K}\|_{L_2}}{\|\mathbf{K}_{\text{ref}}\|_{L_2}} \times 100\%, \quad (33)$$

where for the $2n_v \times 2n_v$ -dimensional stiffness matrix of a two-dimensional polygonal element with n_v vertices, the standard L_2 -norm $\|\cdot\|_{L_2}$ is defined as

$$\|\mathbf{K}\|_{L_2} = \left(\sum_{i=1}^{2n_v} \sum_{j=1}^{2n_v} |K_{ij}|^2 \right)^{1/2}. \quad (34)$$

The reference values for the global and local approaches are obtained by using a highly accurate integration with $k = 15$ and $n = 50$ resulting in the norms

$$\|\hat{\mathbf{K}}_{\text{ref}}\|_{L_2} = 2.587287053379724 \times 10^5 \quad \text{and} \quad (35)$$

$$\|\mathbf{K}_{\text{ref}}\|_{L_2} = 2.582566542956506 \times 10^5. \quad (36)$$

The results obtained with and without compression are depicted for the global and local formulations in Fig. 10a and b, respectively. Note that since for the two plots two different reference values were used, $\hat{\mathbf{K}}_{\text{ref}}$ and \mathbf{K}_{ref} , the errors cannot be directly compared. However, it can be observed that due to the significantly simpler integral formulations of the global approach, the integration accuracy is affected only marginally by the compression. In contrast, the integrand in case of the local formulation contains extra mappings and inverse Jacobian terms, resulting in a more complex rational expression. Consequently, the deterioration of the integration accuracy can be clearly observed when the compression is used.

6 Numerical example: Perforated plate

So far, the integration error with and without the compression of sub-cells was investigated only on the element level. In this section, the effect of the integration error is investigated in the context of finite element analysis by solving a linear elastic problem (see Fig. 11a), using unfitted polygonal meshes with different element sizes, such as the one depicted in Fig. 11b. The assumed material properties are the same as the ones given in Sect. 5.2. For the decomposition, the QTD-approach is used without and with the IBR compression. In both cases, a possible ill-conditioning of the system matrices is avoided by using the RFIP approach [38, 83, 87], requiring only a negligible number of fictitious integration points. For the approximation of the displacement field,

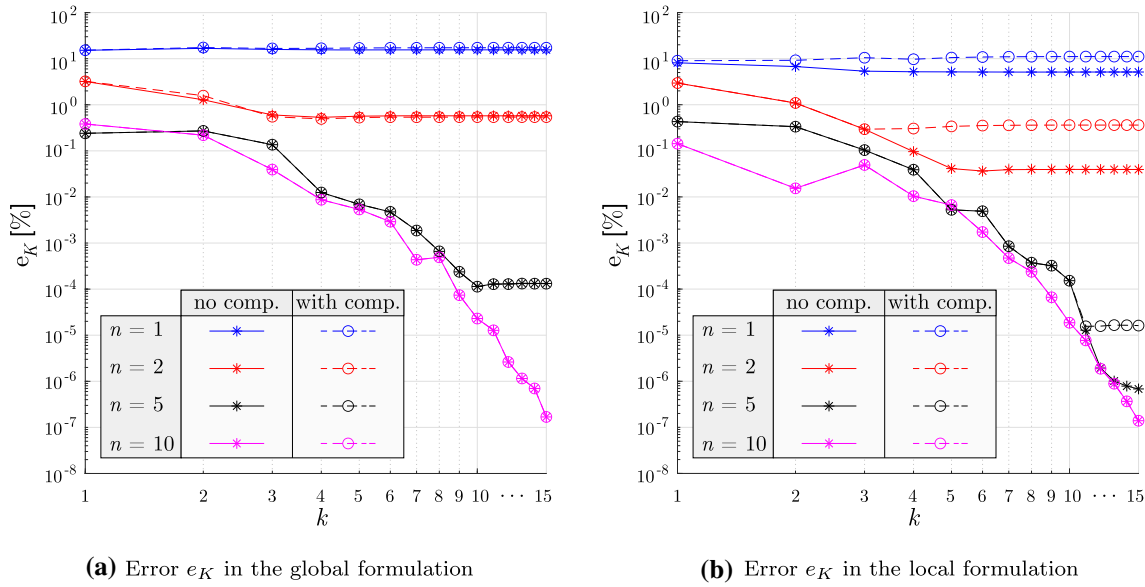


Fig. 10 Error in the L_2 -norm of the stiffness matrix for various refinement levels k and integration points n per direction and sub-cell. **a** Error e_K in the global formulation. **b** Error e_K in the local formulation

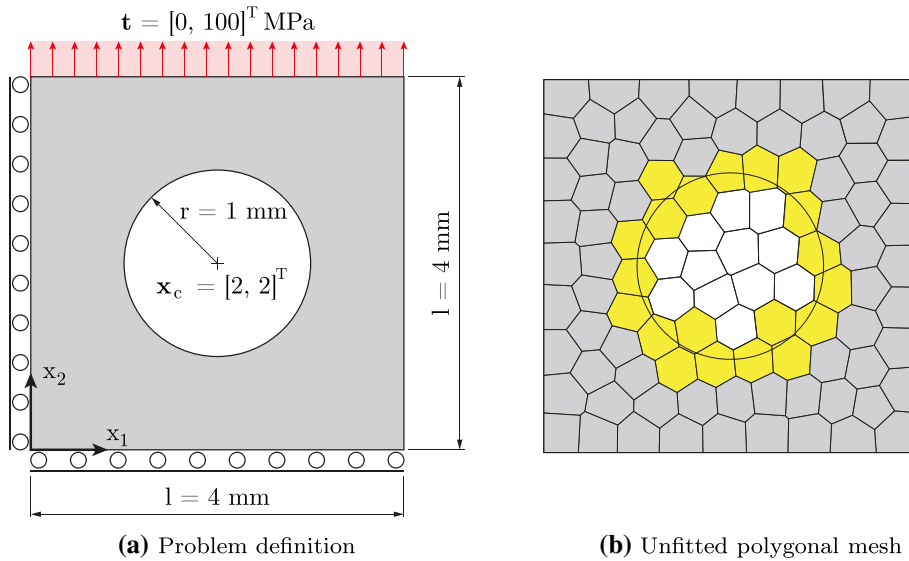


Fig. 11 Problem of a statically loaded perforated plate and its exemplary discretization by an unfitted mesh consisting of quadrilateral, pentagonal, hexagonal and heptagonal polygonal elements. **a** Problem definition. **b** Unfitted polygonal mesh

rational Wachspress shape functions are used. In the remainder of the article, the polygonal shape functions are defined in the local space of the elements Ω_ξ^{poly} due to the following reasons:

1. According to Sect. 5, the integration quality in polygonal elements is visibly affected by the compression of sub-cells, while this influence is not observed in polygonal elements employing global shape functions. Since the effect of reducing the number of integration points is the main focus of this article, locally defined shape functions allow for a more stringent investigation of the compressed QTD.
2. Not all polygonal shape function can be defined over arbitrarily distorted polygonal elements in the global space. The rational Wachspress interpolant is a suitable representative, which can be defined in the global space; however, it requires the element to be convex.
3. In most FE softwares, the shape functions are commonly generated in the local space of the implemented elements.

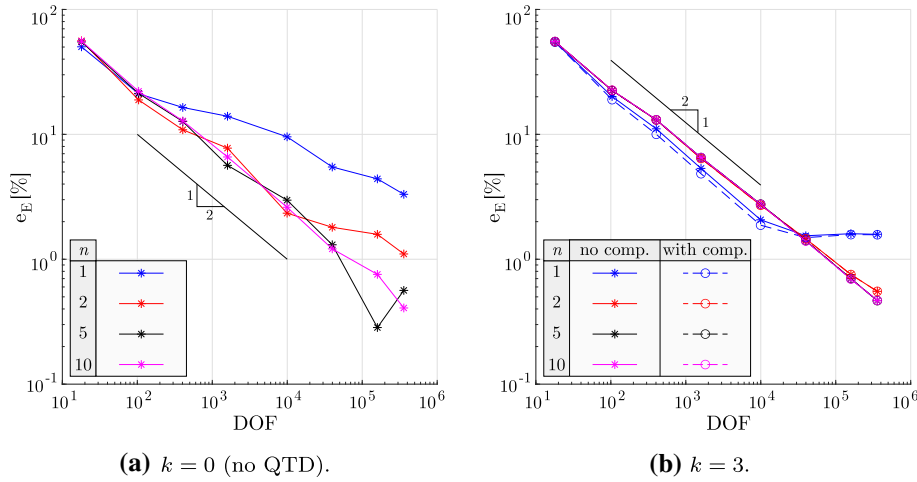


Fig. 12 Error in the energy norm e_E during h -refinement using different refinement levels k of the QTD and number of integration points per direction n . In each plot, the theoretical convergence rate for linear shape functions is depicted. For $n = 5$ and 10 in sub-figure **(b)**, the theoretical convergence rate is already obtained in the depicted DOF-range, and therefore, these curves are overlapping. **a** $k = 0$ (no QTD). **b** $k = 3$. (Color figure online)

Considering the reasons mentioned above, we employ a local partitioning scheme, which is aligned with the definition of the shape functions in Ω_ξ^{poly} . Consequently, integrals of the form given in Eq. (24) have to be solved.

6.1 Convergence in the energy norm

The quality of the results is measured by the relative error e_E in the energy norm $\|\cdot\|_E$ according to [2]

$$e_E = \frac{\|\mathbf{u}_{\text{ref}} - \mathbf{u}\|_E}{\|\mathbf{u}_{\text{ref}}\|_E} = \sqrt{\left| \frac{\mathcal{B}_e(\mathbf{u}_{\text{ref}}, \mathbf{u}_{\text{ref}}) - \mathcal{B}_e(\mathbf{u}, \mathbf{u})}{\mathcal{B}_e(\mathbf{u}_{\text{ref}}, \mathbf{u}_{\text{ref}})} \right|} \times 100[\%], \quad (37)$$

for which the reference value $1/2 \mathcal{B}_e(\mathbf{u}_{\text{ref}}, \mathbf{u}_{\text{ref}}) = 0.7021812127$ from Ref. [16] is used. The corresponding results are depicted in Fig. 12 for various settings, also including the case $k = 0$ where no integration sub-cells are used (Fig. 12a). Considering a uniform h -refinement of the given problem with linear elements, the theoretical optimal convergence rate in the energy norm is algebraic with the factor $\beta = 1/2$ [2, 16]. In the following, the effect of the integration accuracy of the piece-wise integrands on the convergence rate is discussed, while focusing on the terms *piece-wise* (Sect. 6.1.1) and *rational* (Sect. 6.1.2), separately.

6.1.1 Effect of the discontinuity

Using $k = 0$ with low-order quadrature rules results in a poor convergence; however, increasing the value of n quickly leads to an improved solution quality as both the rational and discontinuous parts of the piece-wise rational integrands are integrated more accurately, leading to decreased errors e_D and e_R (cf. Sect. 5). For $n = 10$ the theoretical convergence rate is basically obtained (magenta curve in Fig. 12a). Although in case of $k = 0$, no QTD is performed both the physical and cut polygonal elements are partitioned once for integration purposes (to facilitate the use of standard Gaussian quadrature rules) as depicted in Fig. 4b. Despite the optimal convergence rate, using $k = 0$ with a globally increased quadrature order is extremely inefficient. As an example, the discretization by 40,000 elements is investigated for the settings $k = 0$, $n = 10$ (magenta curve Fig. 12a) and $k = 3$, $n = 2$ (red curve in Fig. 12b). While in both cases an optimal convergence and an error $e_E < 1\%$ is reached, the first settings results in 230,700 integration points in total and 9 times more computational time for generating the stiffness matrices than the second setting.

Performing a QTD with $k = 3$ (Fig. 12b) does not only lead to better results due to the more accurate resolution of the boundary, but the results are obtained much more efficiently as the number of integration points is not increased everywhere, but only in the vicinity of the discontinuity. With the reduced error e_D in

case of $k = 3$, the optimal convergence rate is already obtained for $n = 2$ basically in the entire investigated DOF-range. The fast recovery of the theoretical convergence rate with increasing n is explained by the fact that only low-order elements are used. That is to say, due to the relatively poor approximation of the displacement field by low-order elements the integration error caused by the discontinuous integrand in cut elements is balanced with the discretization error. Since the optimal convergence rate is already obtained for $k = 3$, using higher refinement levels $k > 3$ enabling an even better resolution of the boundary is not meaningful and results in the same curves as depicted in Fig. 12b.

6.1.2 Effect of the rational integrand

The stationary value of the curve $n = 1$ (blue) in Fig. 12b is caused by the inaccurate integration of the rational functions present in the whole computational domain, i.e., both in the physical and cut elements, which leads to an error $e_{\mathcal{R}}$ dominating the entire solution quality. This phenomenon has been already discussed with respect to Figs. 8 and 10, revealing the higher importance of $e_{\mathcal{R}}$ over $e_{\mathcal{D}}$ for small values of n . Although the stationary state is not yet reached, but essentially the start of this effect can be observed also for the curve $n = 2$ (red) in Fig. 12b due to the same reason. Furthermore, since the depicted curves in Fig. 12b are identical to the ones obtained with higher refinement levels k , the conjecture that the deterioration of the convergence rates for $n = 1$ and 2 are not caused by $e_{\mathcal{D}}$ is confirmed. On the contrary, for the curves when $k = 0$ (Fig. 12a), where the discontinuity is approximated very poorly, the sub-optimal convergence is caused not only by $e_{\mathcal{R}}$, but also by $e_{\mathcal{D}}$, resulting in the difference between Fig. 12a and b.

6.2 Efficiency of the compression

Although on the element level in Sect. 5 the effect of the compression was clearly detected for the local definition of the shape functions and partitioning of the element, for the current example, basically no loss in accuracy resulting from the compression can be observed in the wide range of investigated discretizations. What is even more, for achieving the same accuracy, only 55–38% of the original integration points are required by the compressed QTD, depending on the chosen integration settings (see Fig. 13). Note that while in case of Fig. 7 the compression was compared to a QTD where all sub-cells and integration points are considered, in the current case, the comparison is made to a QTD where the fictitious integration points are already considerably reduced by the RFIP approach [38, 83, 87]. Nonetheless, still a remarkable compression is achieved, even when $k = 3$ and $n = 2$ is used.

Moreover, the overall reduction in the computational time is not only influenced by the number of resulting integration points after the compression, but also by the time invested for performing the compression itself. The spent time depends on the complexity of the algorithm for detecting cut sub-cells and on the implementation quality of the QTD procedure. In the ideal case, the additional time should be always smaller than the time saved during the integration over the compressed QTD [38]. The time savings for computing the element matrices are depicted in Fig. 14 for three different settings. In all three cases, a discretization by 40,000 elements is used, representing the first stage where an error $e_E \leq 1\%$ is obtained (Fig. 12). Already for the simplest meaningful setting a reduction in computational time is possible, which can be even further increased once a higher refinement level k and quadrature rule by $n \times n$ integration points are used (Fig. 14).

Remark 1 In this example, Wachspress' shape function defined in the canonical element was investigated in particular, for which a clear deterioration in the integration accuracy due to the IBR compression was shown on the element level (see Sect. 5). Nonetheless, when embedded into an FE analysis, the compression had marginal effects on the solution quality and already when $n \geq 2$ was used, optimal convergence rates were obtained (see Fig. 12b). As such results cannot be further improved, the same performance of the globally defined shape functions is very likely, which were only tested on the element level in Sect. 5 and no negative effects resulting from the compression were shown.

Remark 2 Note that regarding the quality of the results and the success of the compression, the fact that only low-order polygonal shape functions are used plays a significant role. If high-order shape functions are utilized, the optimal convergence rates would not be achieved already for $k = 3$ and especially not for $k = 0$. Also, from Sect. 5 it is known that the effect on the integration accuracy of the compression is present on the element level. In our opinion, the fact that it is hardly detectable in the results of the simulation is due to the reason that the introduced integration error is dominated by the discretization error due to the low-order

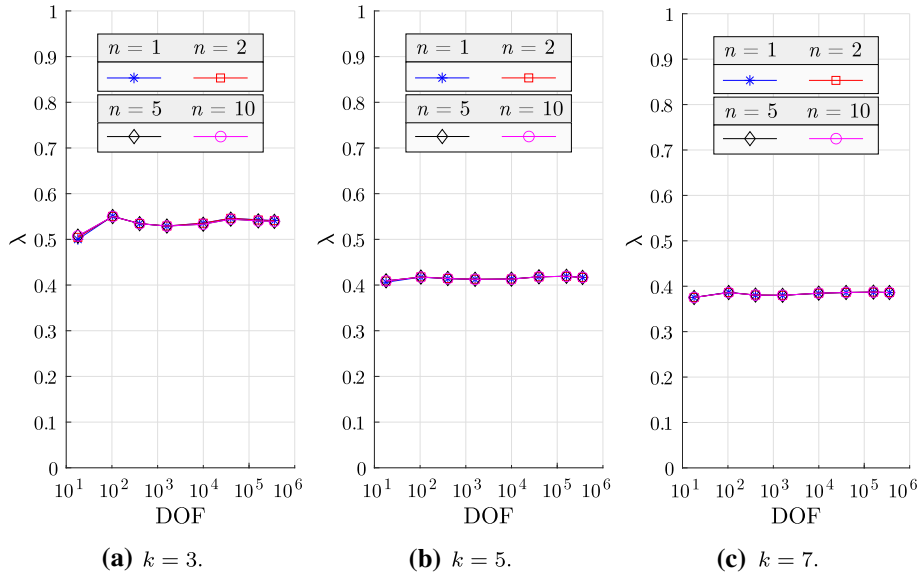


Fig. 13 Compression ratio λ for various mesh sizes, refinement levels k and integration points per direction n . **a** $k = 3$. **b** $k = 5$. **c** $k = 7$

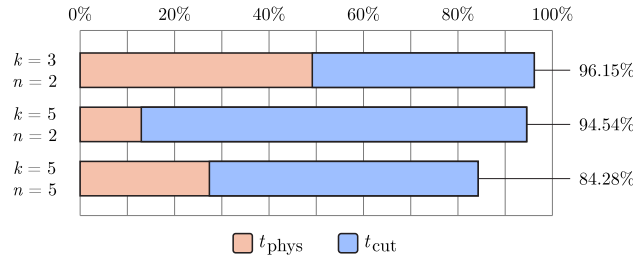
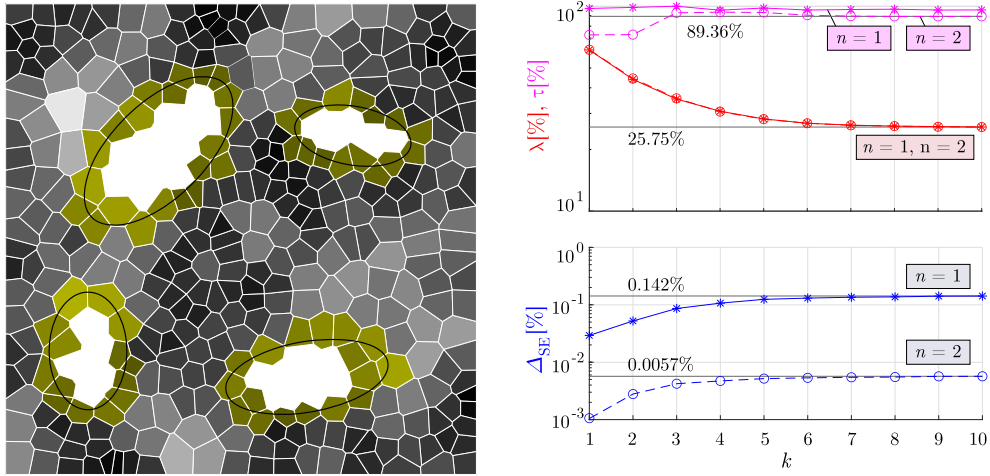


Fig. 14 Time requirements for the computation of the element matrices (physical and cut) for a compressed compared to an uncompressed QTD. For all three cases, a discretization by 40,000 elements is used. The colours in the bars indicate the time proportions spent on the computation of the physical and cut element matrices. (Color figure online)

shape functions; however, this statement requires further research. Furthermore, it has already been shown in Ref. [38] that the computational time can be significantly reduced for higher refinement levels k and polynomial degrees p . Regarding high-order elements also high-order quadrature rules have to be used which correlates with an increase in n (investigated in this contribution). Due to the low-order polygonal shape functions, lower values of k and n already suffice for obtaining an optimal algebraic convergence rate. While a reduction in computational time is possible even for $k = 3$ with $n = 2$, as it is shown in Fig. 14, the full potential of the compression scheme is expected to emerge for unfitted polygonal meshes once high-order polygonal interpolants are available, where an accurate numerical integration is of utmost importance.

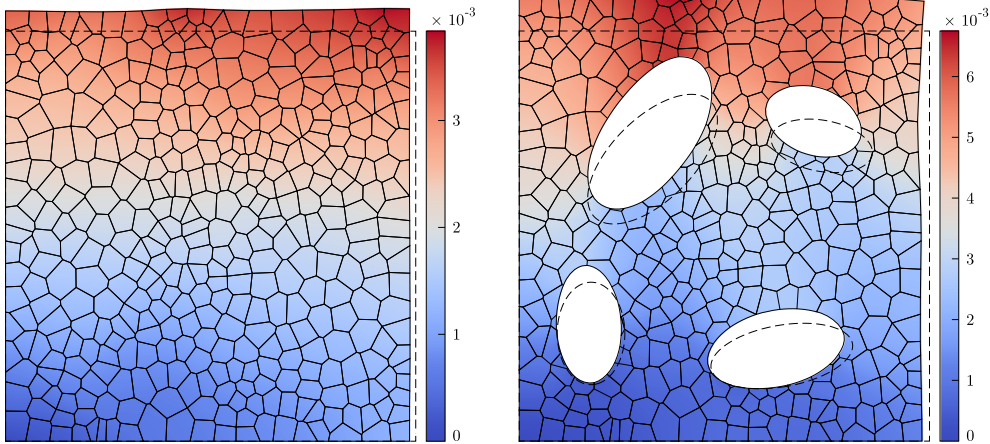
7 Numerical example: Porous multicrystalline domain

In this section, a porous multicrystalline domain depicted in Fig. 15a is investigated, where the Young’s modulus varies linearly for each grain depending on the grain size from 50.000 (smallest) to 250.000 MPa (largest), as indicated by the greyscale coloring. While setting up an FE-mesh for such domain is cumbersome, a Cartesian FC-mesh would lead to oscillatory stress fields and reduced convergence properties due to the inaccurate capturing of the material interfaces [88, 89]. Hence, the computational mesh is obtained by assigning for each grain a single polygonal element. The magnitude of the displacement field of such a domain is depicted in Fig. 15c. The void regions are introduced through the fictitious domain concept, leading to cut polygonal elements indicated by yellow color in Fig. 15a. The corresponding displacement field of the porous material is shown in Fig. 15d.



(a) Multicrystalline domain with void regions. The same boundary conditions apply as in the problem depicted in Fig. 11.

(b) Error in the strain energy according to Eq. (38) (blue), reduction in integration points (red) and computational time (purple).



(c) Displacements without void regions ($u_{\min} = 0$ mm, $u_{\max} = 3.83 \cdot 10^{-3}$ mm)

(d) Displacements with void regions ($u_{\min} = 0$ mm, $u_{\max} = 6.75 \cdot 10^{-3}$ mm)

Fig. 15 Numerical example of a porous multicrystalline domain with **a** computational mesh and **b** solution characteristics when the IBR compression scheme is used. The magnitude of the displacement fields is depicted for a computational domain **c** without and **d** with the ellipsoidal cavities. In both cases, a scaling factor of 59.48 is used, while the dashed lines indicate the contours of the undeformed bodies. **a** Multicrystalline domain with void regions. The same boundary conditions apply as in the problem depicted in Fig. 11. **b** Error in the strain energy according to Eq. (38) (blue), reduction in integration points (red) and computational time (purple). **c** Displacements without void regions ($u_{\min} = 0$ mm, $u_{\max} = 3.83 \cdot 10^{-3}$ mm). **d** Displacements with void regions ($u_{\min} = 0$ mm, $u_{\max} = 6.75 \cdot 10^{-3}$ mm). (Color figure online)

For studying the effect of the IBR compression, a constant discretization is used in conjunction with various refinement levels k and $n \times n$ integration points per sub-cell. The corresponding results are depicted in Fig. 15b, where the IBR compression using only the physical integration points is compared to the standard QTD utilizing all the integration points (both physical and fictitious, cf. Fig. 7). A possible ill-conditioning during the simulation with compression is avoided by using the approach introduced by Abedian et al. [83,87]. Note that for the current example no reference solution exists. Therefore, instead of using Eq. (37), the relative deviation in the strain energy due the reduction in integration points is measured by

$$\Delta_{SE} = \left| \frac{\mathcal{B}_e(\mathbf{u}_{QTD}, \mathbf{u}_{QTD}) - \mathcal{B}_e(\mathbf{u}_{IBR}, \mathbf{u}_{IBR})}{\mathcal{B}_e(\mathbf{u}_{QTD}, \mathbf{u}_{QTD})} \right| \times 100[\%], \quad (38)$$

where \mathbf{u}_{QTD} and \mathbf{u}_{IBR} are the approximated displacement fields (for a given k and n) without compression and with the IBR compression scheme, respectively. There are two important facts to be noted regarding Δ_{SE} in Fig. 15b: First, the differences are growing with increasing k until they reach a stationary value. This is in perfect agreement with the general behavior of the compression which has a higher compression rate λ for increasing values of k (see Fig. 7 and red curves in Fig. 15b). It is clear that a better compression leads to less integration points and therefore, to higher differences in the integral values. Once the compression ratio λ reaches a stationary value, the deviations Δ_{SE} also converge to a finite maximum value. Second, despite the growing nature of Δ_{SE} in Fig. 15b, the maximum deviation is marginal already for $n = 2$ ($\Delta_{SE} = 0.0057\%$). It is crucial to emphasize that due to the lack of a reference solution, Δ_{SE} in Eq. (38) expresses only a relative difference between the two approximate solutions \mathbf{u}_{QTD} and \mathbf{u}_{IBR} . Although Δ_{SE} may increase for higher values of k , the error compared to the real solution (see e_E in Sect. 6) is not increasing and generally, both \mathbf{u}_{QTD} and \mathbf{u}_{IBR} are more accurate when a higher refinement level k is used. In terms of efficiency of the compression, a significant compression ratio λ could be obtained requiring only 25.75% of the integration points in its converged state (red curves in Fig. 15b). However, the time savings during the numerical integration are reduced by the time investment for performing the compression, and therefore, the saving in computation time τ (purple curves in Fig. 15b) cannot reach the same rates as λ . In the current example for $n = 2$, this means about 10% time saving during the computation of the element matrices. While τ can be even more significant, i.e., even smaller when more integration points are used [38], such settings are not required for low-order polygonal shape functions, as also concluded at the end of the previous numerical example.

8 Conclusion

Following the poly-FCM approach proposed by Duczek and Gabbert [36] and the fundamental concept of polygonal elements based on generalized barycentric coordinates [27,28], the enhanced numerical integration scheme based on compressed quadtree-decompositions introduced by Petö et al. [38] was investigated in the context of unfitted polygonal meshes. The straightforward implementation of this novel approach for polygonal meshes was clearly demonstrated and the resulting integral formulations over the compressed local integration mesh were discussed both for local and global sub-divisions of polygonal elements. Since the compression procedure leads to a decreased integration point density for integrating discontinuous non-polynomial functions, the effect of the compression on the integration quality was investigated, with a particular focus on the rational shape functions proposed by Wachspress. The influence of the compression on the integration quality was clearly detectable on the element level based on a local definition of the shape functions and a local decomposition of the polygonal element. On the contrary, for globally defined shape functions and partitioning schemes, the area approximation and the computation of the stiffness matrix are written in a much simpler form, for which the compression resulted basically in no errors. When embedded into an entire simulation of a linear elasticity problem, the compression had a negligible effect on the quality of the results, while requiring 26–55% of the original integration points. In conclusion, the compression is compatible with unfitted polygonal meshes with piece-wise rational integrands and it is able to achieve similar results as described in Ref. [38] in the context of the standard FCM with Cartesian meshes and polynomial shape functions. However, the lack of proper high-order polygonal shape functions constitutes a significant limitation, both with respect to the achievable convergence rates and the exploitation of the full potential of the compressed QTD-based integration scheme.

Acknowledgements The authors thank the German Research Foundation (DFG) for its financial support under Grant DU 1904/2-1. The authors also acknowledge the fruitful discussions with Resam Makvandi.

Funding Open Access funding provided by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted

by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

- Zienkiewicz, O.C., Taylor, R.L., Zhu, J.Z.: *The Finite Element Method: Its Basis and Fundamentals*. Butterworth-Heinemann, Oxford (2005)
- Szabó, B., Babuška, I.: *Introduction to Finite Element Analysis*. Wiley, Hoboken (2011)
- Sukumar, N., Moës, N., Moran, B., Belytschko, T.: Extended finite element method for three-dimensional crack modelling. *Int. J. Numer. Methods Eng.* **48**(11), 1549–1570 (2000)
- Fries, T.-P., Belytschko, T.: The extended/generalized finite element method: An overview of the method and its applications. *Int. J. Numer. Methods Eng.* **84**, 253–304 (2010)
- Strouboulis, T., Copps, K., Babuška, I.: The generalized finite element method. *Comput. Methods Appl. Mech. Eng.* **190**(32–33), 4081–4193 (2001)
- Roma, A.M., Peskin, C.S., Berger, M.J.: An adaptive version of the immersed boundary method. *J. Comput. Phys.* **153**(2), 509–534 (1999)
- Hansbo, A., Hansbo, P.: An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Comput. Methods Appl. Mech. Eng.* **191**(47–48), 5537–5552 (2002)
- Ramière, I., Angot, P., Belliard, M.: A fictitious domain approach with spread interface for elliptic problems with general boundary conditions. *Comput. Methods Appl. Mech. Eng.* **196**(4–6), 766–781 (2007)
- Burman, E., Hansbo, P.: Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method. *Comput. Methods Appl. Mech. Eng.* **199**(41–44), 2680–2686 (2010)
- Burman, E., Hansbo, P.: Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. *Appl. Numer. Math.* **62**(4), 328–341 (2012)
- Burman, E., Hansbo, P.: Fictitious domain methods using cut elements: III. A stabilized Nitsche method for Stokes' problem. *ESAIM: Math. Model. Numer. Anal.* **48**(3), 859–874 (2014)
- Burman, E., Claus, S., Hansbo, P., Larson, M.G., Massing, A.: CutFEM: discretizing geometry and partial differential equations. *Int. J. Numer. Methods Eng.* **104**(7), 472–501 (2014)
- Nadal, E., Ródenas, J.J., Albelda, J., Tur, M., Tarancón, J.E., Fuenmayor, F.J.: Efficient finite element methodology based on cartesian grids: application to structural shape optimization. *Abstr. Appl. Anal.* **1–19**, 2013 (2013)
- García-Ruiz, M.J., Steven, G.P.: Fixed grid finite elements in elasticity problems. *Eng. Comput.* **16**(2), 145–164 (1999)
- García, M.J., Henao, M.-A., Ruiz, O.E.: Fixed grid finite element analysis for 3d structural problems. *Int. J. Comput. Methods* **02**(04), 569–586 (2005)
- Parvizian, J., Düster, A., Rank, E.: Finite cell method: h - and p -extension for embedded domain problems in solid mechanics. *Comput. Mech.* **41**(1), 121–133 (2007)
- Düster, A., Parvizian, J., Yang, Z., Rank, E.: The finite cell method for three-dimensional problems of solid mechanics. *Comput. Methods Appl. Mech. Eng.* **197**(45–48), 3768–3782 (2008)
- 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.* **22**(3), 391–455 (2014)
- Düster, A., Rank, E., Szabó, B.: The p -Version of the Finite Element and Finite Cell Methods, pp. 1–35. *Encyclopedia of Computational Mechanics* (2017)
- Dohrmann, C.R., Key, S.W., Heinstein, M.W.: A method for connecting dissimilar finite element meshes in two dimensions. *Int. J. Numer. Methods Eng.* **48**(5), 655–678 (2000)
- Peters, J.F., Heymsfield, E.: Application of the 2-d constant strain assumption to FEM elements consisting of an arbitrary number of nodes. *Int. J. Solids Struct.* **40**(1), 143–159 (2003)
- Biabanaki, S.O.R., Khoei, A.R.: A polygonal finite element method for modeling arbitrary interfaces in large deformation problems. *Comput. Mech.* **50**(1), 19–33 (2011)
- Biabanaki, S.O.R., Khoei, A.R., Wriggers, P.: Polygonal finite element methods for contact-impact problems on non-conformal meshes. *Comput. Methods Appl. Mech. Eng.* **269**, 198–221 (2014)
- Khoei, A.R., Yasbolaghi, R., Biabanaki, S.O.R.: A polygonal finite element method for modeling crack propagation with minimum remeshing. *Int. J. Fract.* **194**(2), 123–148 (2015)
- Ghosh, S., Moorthy, S.: Elastic-plastic analysis of arbitrary heterogeneous materials with the Voronoi cell finite element method. *Comput. Methods Appl. Mech. Eng.* **121**(1–4), 373–409 (1995)
- Saksala, T.: Numerical modelling of rock materials with polygonal finite elements. *Rakenteiden Mekaniikka* **50**(3), 216–219 (2017)
- Sukumar, N., Tabarraei, A.: Conforming polygonal finite elements. *Int. J. Numer. Methods Eng.* **61**(12), 2045–2066 (2004)
- Tabarraei, A., Sukumar, N.: Application of polygonal finite elements in linear elasticity. *Int. J. Comput. Methods* **03**(04), 503–520 (2006)
- Zhang, H.W., Wang, H., Chen, B.S., Xie, Z.Q.: Analysis of Cosserat materials with Voronoi cell finite element method and parametric variational principle. *Comput. Methods Appl. Mech. Eng.* **197**(6–8), 741–755 (2008)
- Beirão da Veiga, L., Brezzi, F., Cangiani, A., Manzini, G., Marini, L.D., Russo, A.: Basic principles of virtual element methods. *Math. Models Methods Appl. Sci.* **23**(01), 199–214 (2012)

31. Beirão da Veiga, L., Brezzi, F., Marini, L.D., Russo, A.: Virtual element method for general second-order elliptic problems on polygonal meshes. *Math. Models Methods Appl. Sci.* **26**(04), 729–750 (2016)
32. Song, C., Wolf, J.P.: The scaled boundary finite-element method—alias consistent infinitesimal finite-element cell method—for elastodynamics. *Comput. Methods Appl. Mech. Eng.* **147**(3–4), 329–355 (1997)
33. Wolf, J.P., Song, C.: The scaled boundary finite-element method—a primer: derivations. *Comput. Struct.* **78**(1–3), 191–210 (2000)
34. Song, C.: The scaled boundary finite element method in structural dynamics. *Int. J. Numer. Methods Eng.* **77**(8), 1139–1171 (2009)
35. Perumal, L.: A brief review on polygonal/polyhedral finite element methods. *Math. Probl. Eng.* **2018**, 1–22 (2018)
36. Duczek, S., Gabbert, U.: The finite cell method for polygonal meshes: poly-FCM. *Comput. Mech.* **58**(4), 587–618 (2016)
37. Wachspress, Eugene: *A Rational Finite Element Basis*. Academic Press, New York (1975)
38. Petö, M., Duvigneau, F., Eisenträger, S.: Enhanced numerical integration scheme based on image-compression techniques: application to fictitious domain methods. *Adv. Model. Simul. Eng. Sci.* **7**(1), 1–42 (2020)
39. Brenner, S.C., Scott, L.R.: *The Mathematical Theory of Finite Element Methods*. Springer, New York (2008)
40. Dauge, M., Düster, A., Rank, E.: Theoretical and numerical investigation of the finite cell method. *J. Sci. Comput.* **65**(3), 1039–1064 (2015)
41. Parvizian, J., Düster, A., Rank, E.: Topology optimization using the finite cell method. *Optim. Eng.* **13**(1), 57–78 (2011)
42. Del Pino, S., Pironneau, O.: A fictitious domain based general PDE solver. *Numer. Methods Sci. Comput. Var. Probl. Appl.* (2003)
43. Ruess, M., Schillinger, D., Bazilevs, Y., Varduhn, V., Rank, E.: Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell method. *Int. J. Numer. Methods Eng.* **95**(10), 811–846 (2013)
44. Riem, M.: *Entwicklung und Untersuchung polytooper Finiter Elemente für die nichtlineare Kontinuumsmechanik*. PhD thesis, University of Erlangen-Nürnberg (2013)
45. Wachspress, E.: *Rational Bases and Generalized Barycentrics*. Springer International Publishing, Berlin (2016)
46. Talischi, C., Paulino, G.H., Pereira, A., Menezes, I.F.M.: PolyTop: a Matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes. *Struct. Multidiscip. Optim.* **45**(3), 329–357 (2012)
47. Meyer, M., Barr, A., Lee, H., Desbrun, M.: Generalized barycentric coordinates on irregular polygons. *J. Graph. Tools* **7**(1), 13–22 (2002)
48. Warren, J.: Barycentric coordinates for convex polytopes. *Adv. Comput. Math.* **6**(1), 97–108 (1996)
49. Floater, M.S.: Mean value coordinates. *Comput. Aided Geom. Des.* **20**(1), 19–27 (2003)
50. Floater, M.S., Kós, G., Reimers, M.: Mean value coordinates in 3d. *Comput. Aided Geom. Des.* **22**(7), 623–631 (2005)
51. Ju, T., Schaefer, S., Warren, J.: Mean value coordinates for closed triangular meshes. In: *ACM SIGGRAPH 2005 Papers on—SIGGRAPH 05*. ACM Press, New York (2005)
52. Sukumar, N., Moran, B., Semenov, A.Y., Belikov, V.V.: Natural neighbour Galerkin methods. *Int. J. Numer. Methods Eng.* **50**(1), 1–27 (2000)
53. Sukumar, N.: Construction of polygonal interpolants: a maximum entropy approach. *Int. J. Numer. Methods Eng.* **61**(12), 2159–2181 (2004)
54. Sukumar, N.: Quadratic maximum-entropy serendipity shape functions for arbitrary planar polygons. *Comput. Methods Appl. Mech. Eng.* **263**, 27–41 (2013)
55. Rand, A., Gillette, A., Bajaj, C.: Quadratic serendipity finite elements on polygons using generalized barycentric coordinates. *Math. Comput.* **83**(290), 2691–2716 (2014)
56. Dasgupta, G.: Integration within polygonal finite elements. *J. Aerosp. Eng.* **16**(1), 9–18 (2003)
57. Duczek, S., Gabbert, U.: Efficient integration method for fictitious domain approaches. *Comput. Mech.* **56**(4), 725–738 (2015)
58. Cattani, C., Paoluzzi, A.: Boundary integration over linear polyhedra. *Comput. Aided Des.* **22**(2), 130–135 (1990)
59. Sudhakar, Y., Moitinho de Almeida, J.P., Wall, W.A.: An accurate, robust, and easy-to-implement method for integration over arbitrary polyhedra: application to embedded interface methods. *J. Comput. Phys.* **273**, 393–415 (2014)
60. Mousavi, S.E., Xiao, H., Sukumar, N.: Generalized Gaussian quadrature rules on arbitrary polygons. *Int. J. Numer. Methods Eng.* **82**, 99–113 (2010)
61. Xiao, H., Gimbutas, Z.: A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Comput. Math. Appl.* **59**(2), 663–676 (2010)
62. Mousavi, S.E., Sukumar, N.: Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons. *Comput. Mech.* **47**(5), 535–554 (2010)
63. Sudhakar, Y., Wall, W.A.: Quadrature schemes for arbitrary convex/concave volumes and integration of weak form in enriched partition of unity methods. *Comput. Methods Appl. Mech. Eng.* **258**, 39–54 (2013)
64. Müller, B., Kummer, F., Oberlack, M.: Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *Int. J. Numer. Methods Eng.* **96**(8), 512–528 (2013)
65. Milbradt, P., Pick, T.: Polytope finite elements. *Int. J. Numer. Methods Eng.* **73**(12), 1811–1835 (2008)
66. Rajagopal, A., Kraus, M., Steinmann, P.: Hyperelastic analysis based on a polygonal finite element method. *Mech. Adv. Mater. Struct.* **25**(11), 930–942 (2017)
67. Sarada, J., Nagaraja, K.V.: Generalized gaussian quadrature rules over two-dimensional regions with linear sides. *Appl. Math. Comput.* **217**(12), 5612–5621 (2011)
68. Sukumar, N., Chopp, D.L., Moës, N., Belytschko, T.: Modeling holes and inclusions by level sets in the extended finite-element method. *Comput. Methods Appl. Mech. Eng.* **190**(46–47), 6183–6200 (2001)
69. Strouboulis, T., Copps, K., Babuška, I.: The generalized finite element method: an example of its implementation and illustration of its performance. *Int. J. Numer. Methods Eng.* **47**(8), 1401–1417 (2000)
70. Glowinski, R., Kuznetsov, Yu.: Distributed Lagrange multipliers based on fictitious domain method for second order elliptic problems. *Comput. Methods Appl. Mech. Eng.* **196**(8), 1498–1506 (2007)

71. Joulaian, M., Hubrich, S., Düster, A.: Numerical integration of discontinuities on arbitrary domains based on moment fitting. *Comput. Mech.* **57**(6), 979–999 (2016)
72. Zhang, Z., Jiang, W., Dolbow, J.E., Spencer, B.W.: A modified moment-fitted integration scheme for x-FEM applications with history-dependent material data. *Comput. Mech.* **62**(2), 233–252 (2018)
73. Ventura, G., Benvenuti, E.: Equivalent polynomials for quadrature in Heaviside function enriched elements. *Int. J. Numer. Methods Eng.* **102**(3–4), 688–710 (2014)
74. Abedian, A., Düster, A.: Equivalent Legendre polynomials: numerical integration of discontinuous functions in the finite element methods. *Comput. Methods Appl. Mech. Eng.* **343**, 690–720 (2019)
75. Hughes, T.J.R.: *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publ Inc., New York (2000)
76. Cheng, K.W., Fries, T.-P.: Higher-order XFEM for curved strong and weak discontinuities. *Int. J. Numer. Methods Eng.* **82**, 564–590 (2010)
77. Fries, T.-P., Omerović, S.: Higher-order accurate integration of implicit geometries. *Int. J. Numer. Methods Eng.* **106**(5), 323–371 (2015)
78. Kudela, L., Zander, N., Bog, T., Kollmannsberger, S., Rank, E.: Efficient and accurate numerical quadrature for immersed boundary methods. *Adv. Model. Simul. Eng. Sci.* **2**(1), 10 (2015)
79. Kudela, L., Zander, N., Kollmannsberger, S., Rank, E.: Smart octrees: accurately integrating discontinuous functions in 3D. *Comput. Methods Appl. Mech. Eng.* **306**, 406–426 (2016)
80. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: *Computational Geometry*. Springer, Berlin (2008)
81. Duczek, S., Duvigneau, F., Gabbert, U.: The finite cell method for tetrahedral meshes. *Finite Elem. Anal. Des.* **121**, 18–32 (2016)
82. Abedian, A., Parvizian, J., Düster, A., Khademyzadeh, H., Rank, E.: Performance of different integration schemes in facing discontinuities in the finite cell method. *Int. J. Comput. Methods* **10**(03), 1350002 (2013)
83. Abedian, A., Parvizian, J., Düster, A., Rank, E.: The finite cell method for the J_2 flow theory of plasticity. *Finite Elem. Anal. Des.* **69**, 37–47 (2013)
84. Suk, T., Höschl, C., Flusser, J.: Rectangular decomposition of binary images. In: *Advanced Concepts for Intelligent Vision Systems*, pp. 213–224. Springer, Berlin (2012)
85. Eppstein, D.: Graph-theoretic solutions to computational geometry problems. In: *Graph-Theoretic Concepts in Computer Science*, pp. 1–16. Springer, Berlin (2010)
86. Salomon, D., Motta, G.: *Handbook of Data Compression*. Springer, London (2010)
87. Abedian, A., Parvizian, J., Düster, A., Rank, E.: Finite cell method compared to h -version finite element method for elasto-plastic problems. *Appl. Math. Mech.* **35**(10), 1239–1248 (2014)
88. Joulaian, M., Düster, A.: Local enrichment of the finite cell method for problems with material interfaces. *Comput. Mech.* **52**(4), 741–762 (2013)
89. Joulaian, M.: *The hierarchical finite cell method for problems in structural mechanics*. PhD thesis, Hamburg Technical University (2017)