

Scaled boundary cubature scheme for numerical integration over planar regions with affine and curved boundaries

Eric B. Chin^{a,*}, N. Sukumar^b

^aLawrence Livermore National Laboratory, 7000 East Avenue, Livermore, CA 94550, USA

^bDepartment of Civil and Environmental Engineering, University of California, Davis, CA 95616, USA

Abstract

This paper introduces the scaled boundary cubature (SBC) scheme for accurate and efficient integration of functions over polygons and two-dimensional regions bounded by parametric curves. Over two-dimensional domains, the SBC method reduces integration over a region bounded by m curves to integration over m regions (referred to as curved triangular regions), where each region is bounded by two line segments and a curve. With proper (counterclockwise) orientation of the boundary curves, the scheme is applicable to convex and nonconvex domains. Additionally, for star-convex domains, a tensor-product cubature rule with positive weights and integration points in the interior of the domain is obtained. If the integrand is homogeneous, we show that this new method reduces to the homogeneous numerical integration scheme; however, the SBC scheme is more versatile since it is equally applicable to both homogeneous and non-homogeneous functions. This paper also introduces several methods for smoothing integrands with point singularities and near-singularities. When these methods are used, highly efficient integration of weakly singular functions is realized. The SBC method is applied to a number of benchmark problems, which reveal its broad applicability and superior performance (in terms of time to generate a rule and accuracy per cubature point) when compared to existing methods for integration.

Keywords: scaled boundary parametrization, Bézier and NURBS curves, homogeneous functions, weakly singular functions, transfinite interpolation, isogeometric analysis

1. Introduction

In this paper, we propose a new method for the accurate and efficient numerical integration of functions over planar (two-dimensional) regions bounded by affine and/or curved edges, such as polygons or conics. The classes of functions include those that are either continuously differentiable in the interior of the domain, or are weakly singular or nearly singular at a point within or near the domain. Regions of integration may be convex or nonconvex and the boundary of the region is described by parametric curves. The boundary curves can be given in closed-form or represented using Bernstein polynomials, B-splines, or non-uniform rational B-splines (NURBS). For integration over bounded domains with curved boundaries, we show that the homogeneous numerical integration scheme [1–3] is a reduction of a tensor-product cubature scheme that is obtained on using a scaled boundary (SB) parametrization [4, 5] of the domain. We refer to this integration method as the *scaled boundary cubature* (SBC) scheme. While only two-dimensional domains are considered in this paper, we note the SBC scheme is also applicable to three-dimensional domains that are bounded by parametric surfaces.

1.1. Related work

Methods for numerical integration over polygons and domains bounded by parametric curves is currently an area of active research. An early, but still popular method of cubature is partitioning the integration domain, and then using well-known rules of integration on the resulting (potentially curved) triangular and quadrilateral partitions [6–9]. However, the process of generating a partition can be time consuming for complex geometries and cubature rules

*Corresponding author

Email address: chin23@llnl.gov (Eric B. Chin)

generated from this process are far from optimal. Optimized rules that retain polynomial precision can be formulated through the process of moment fitting [9, 10], which reduces the number of cubature points iteratively by the least-squares Newton method. However, this process is computationally expensive since it requires the solution of many linear systems of equations, and furthermore, moment fitting requires an initial cubature rule that is typically generated through partitioning. For the special case of a hyperrectangle intersecting an implicitly defined surface, high-order accurate integration schemes have also been devised. The need for this type of integration arises in isogeometric analysis (IGA) with trimmed patches and with unfitted finite element methods. One such integration method by Saye [11] introduces a height function that approximates the location of the interface to high degree. Another method by Scholz and Jüttler [12, 13] introduces higher order correction terms based on transport theorems to reduce error in integration on curved surfaces.

To avoid partitioning, some recent contributions have utilized generalized Stokes’s theorem to develop cubature rules that scale with the number of edges/curves in the polygonal/curved domains. For instance, in Sommariva and Vianello [14], Gauss-Legendre quadrature is used to compute the antiderivatives in Green’s formula (Stokes’s theorem in two dimensions), producing an exact cubature rule for integrating polynomials over a convex or nonconvex polygon. With this scheme, named Gauss-Green (G-G) cubature, cubature points can appear outside the polygonal domain of integration, though the accuracy of the rule is not necessarily affected. This approach is also viable on curved domains, as demonstrated by Sommariva and Vianello [15] and more recently by Gunderman et al. [16]. Another approach utilizing generalized Stokes’s theorem is the homogeneous numerical integration (HNI) method. Building on work by Lasserre [1, 17], the HNI method [2] combines generalized Stokes’s theorem with Euler’s homogeneous function theorem to produce a cubature rule for integrating polynomials and homogeneous functions over polytopes and regions bounded by homogeneous curves. More recently in Chin and Sukumar [3, 18], this idea is extended to domains bounded by arbitrary parametric curves such as polynomial and rational Bézier and B-spline curves. With the HNI method, a simple formula is realized for reducing the dimension of integration by one, resulting in a very efficient cubature rule. However, the restriction to homogeneous functions limits its broader appeal, and in addition, the implementation of the scheme is also more involved [3, 19].

Novel ideas in the computer simulation of physical phenomena and computer-aided manufacturing have motivated the need for improved numerical integration over polygons and regions bounded by curves. Galerkin methods based on the SB mapping, such as the ones explored by Chen et al. [20] and Klinkel and Reichel [21] would benefit from integration rules formulated from the scaled boundary parametrization. More broadly, techniques such as the extended finite element method (X-FEM) [6, 22], polygonal finite element methods [23], trimmed patches with IGA [24, 25], the NURBS enhanced finite element method [7], unfitted finite element methods such as embedded interface and fictitious domain methods [26–28], virtual element method [29–31], and discontinuous Galerkin methods [32, 33] need accurate, efficient cubature over polygons and regions bounded by curves. Additionally, in the field of contact mechanics and domain decomposition, accurately computing contributions to the force vector in a mortar method requires integration over polygons [34], and for high-order mortar methods, integration over planar regions bounded by curves [35] is needed. Furthermore, for applications in computer graphics and animation [36] and computer-aided design [37], computing areas, centroids, and moments of inertia require accurate integration over polygons and regions bounded by parametric curves.

1.2. Contributions and outline

The HNI method provides a simple formula for reducing integration over a d -dimensional domain to scaled integrals over the $(d - 1)$ -dimensional boundary facets of the domain. Therefore, cubature points lie entirely on the boundary and only $O(mp)$ integration points are needed to integrate a homogeneous polynomial of degree $(2p - 1)$ over an m -gon. The proof of the method utilizes generalized Stokes’s theorem to relate the domain integral to an equivalent integral over the domain’s boundary. However, instead of computing antiderivatives of the integrand, the HNI method utilizes Euler’s homogeneous function theorem to relate the integrand to its derivative. While not immediately apparent from its derivation, the HNI method implicitly and exactly computes a one-dimensional integral, which when combined with a parametrization for the boundary of the domain, parametrizes the entire domain of integration. In this paper, we characterize this parametrization, and show that it matches the domain mapping (SB parametrization) introduced in the scaled boundary finite element method [4, 5]. In this framework, the HNI method can be cast as a special case of the mapped integral, which is valid when the integrand is homogeneous.

Beyond the additional insights that the SB parametrization affords into the HNI method, we also demonstrate that integrating over the mapped space delivers an effective cubature rule for polytopes and curved solids in its own right.

The domain need not be convex or simply-connected if the integrand is continuously differentiable and computable outside the domain of integration. The method relies on tensor-product quadrature rules over the unit square, and requires $O(mp^2)$ integration points to exactly integrate a $(2p-1)$ -degree bivariate polynomial over an m -gon. Moreover, the limitation of homogeneous integrands is removed—non-homogeneous functions and functions that are not known explicitly can be integrated. In addition, we introduce a mapping of the domain that we refer to as the *generalized SB parametrization* and integral transformations that smooth singular integrands, enabling accurate cubature of weakly singular functions with relatively few cubature points. The name “generalized SB parametrization” reflects its similarity to the generalized Duffy transformation introduced by Mousavi and Sukumar [38]. These transformations can be applied to computational methods that require the integration of weakly singular functions, such as the X-FEM or the boundary element method (BEM).

The remainder of this paper is organized as follows. In Section 2, the HNI method is introduced, focusing specifically on how integrals over polygons and curved solids are decomposed using this method. In Section 3, we introduce the SBC method and develop several simplifications for polygonal domains. Transformations appropriate for integration of weakly singular functions are presented in Section 4, which increase cubature accuracy for a given number of cubature points. Next, in Section 5, we present numerical integration examples that arise in several distinct application areas to establish the accuracy and to showcase the capabilities of the integration method. These application areas include polygonal finite element methods, the X-FEM, and the BEM. The examples demonstrate integration of polynomials, smooth nonpolynomial functions such as the test functions in Franke [39], and weakly singular functions over convex and nonconvex polygons and domains bounded by parametric curves. In Section 5.4, we apply the SBC scheme to compute the L_2 error for transfinite mean value interpolation (TMVI) [40, 41] over a curved domain. Finally, we conclude with Section 6, where we recap the main developments in this paper and provide a few future directions of research.

2. Domain partitioning into curved triangles

In \mathbb{R}^2 , we define a closed region $\Omega \subset \mathbb{R}^2$ with piecewise smooth, orientable boundary $\partial\Omega$. We assume that $\partial\Omega$ is composed of m parametric curves, C_i for $i = 1, \dots, m$, where each curve occupies a unique subset of the boundary, such that

$$\partial\Omega = \bigcup_{i=1}^m C_i \quad \text{and} \quad C_j \setminus \partial C_j \cap C_k \setminus \partial C_k = \emptyset \text{ for } j, k = 1, \dots, m \text{ and } j \neq k. \quad (1)$$

An illustration of a region that conforms to this definition is provided in Fig. 1. All boundary curves are ordered sequentially as the boundary is traversed in a counterclockwise direction. We introduce a coordinate \mathbf{x}_0 that serves as the center of the partitioning, which leads to a decomposition of Ω into partitions bounded by three curves: two line segments and one curve. We will refer to these partitions as *curved triangles* or by the symbol \mathcal{T} .

If Ω is star-convex, we can select a point $\mathbf{x}_0 \in \Omega$ so that the boundaries of the curved triangles do not intersect any of the boundary curves, C_i ($i = 1, \dots, m$). Or, more generally, we can choose \mathbf{x}_0 such that any line segment that begins at \mathbf{x}_0 only intersects the boundary at one point. This choice for \mathbf{x}_0 with respect to the star-convexity of the domain is depicted in Fig. 2a. However, even if Ω is not star-convex, or if \mathbf{x}_0 is not selected with regard to the star-convexity of Ω , accurate integration is still possible provided a notion of positive and negative portions of a partition exists with the integration rule. A partitioning of Ω where positivity and negativity must be accounted for is illustrated in Fig. 2b. Specifically, for Fig. 2b, the integral of a function f over the region is

$$\int_{\Omega} f \, d\mathbf{x} = \int_{\Omega_1} f \, d\mathbf{x} + \int_{\Omega_2} f \, d\mathbf{x} + \int_{\Omega_3} f \, d\mathbf{x} - \int_{\Omega_4} f \, d\mathbf{x}, \quad (2)$$

where Ω_i is the curved triangle associated with C_i . Note the union of regions Ω_1 , Ω_2 , and Ω_3 contain Ω_4 , but Ω_4 is not in Ω . By subtracting the integral over Ω_4 , the overall effect is to ignore the contribution of the integral outside Ω . To pursue this approach for the integration, the function f must be computable and continuously differentiable in $\Omega_1 \cup \dots \cup \Omega_m$. While the example in Fig. 2b is applicable to partitions whose integrals are entirely negative or positive, a single partition could easily be defined that contains both positive and negative contributions to the integral. This would occur if C_i intersects a line segment starting at \mathbf{x}_0 more than once. Cubature rules with negative weights tend to be avoided for strictly positive integrands, since they can compromise accuracy due to round-off errors. For these cases, we recommend further subdivision of the integration domain until all subdivided integration domains are star-convex, coupled with careful selection of \mathbf{x}_0 .

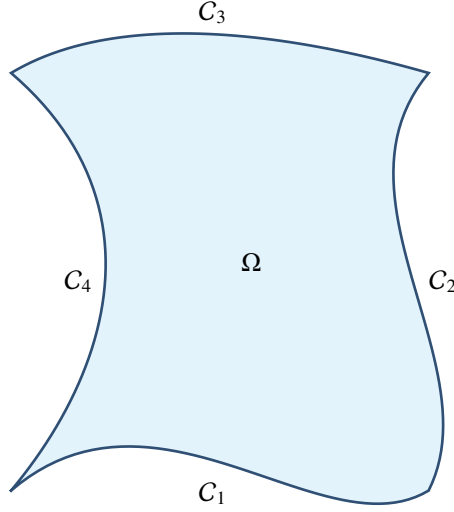


Figure 1: A region bounded by cubic Bézier curves.

2.1. Curved triangles in the HNI method

In this section, we introduce the HNI method and demonstrate it implicitly divides Ω , the domain of integration, into curved triangles, with the origin serving as \mathbf{x}_0 . Proof of the HNI method relies on Euler's homogeneous function theorem and generalized Stokes's theorem. A positively homogeneous function $h(\mathbf{x}) : V \rightarrow \mathbb{R}$ with degree of homogeneity $q \in \mathbb{R}$ is defined such that

$$h(\lambda \mathbf{x}) = \lambda^q h(\mathbf{x}) \quad \forall \mathbf{x} \in V \quad (3)$$

and for any real constant $\lambda > 0$. In (3), the domain $V = \mathbb{R}^2$ when $h(\mathbf{x})$ is a polynomial and $V = \mathbb{R}^2 \setminus \{\mathbf{0}\}$ when $h(\mathbf{x}) = \|\mathbf{x}\|^\beta$ for $\beta < 0$. For a continuously differentiable, positively homogeneous scalar-valued function $h := h(\mathbf{x})$ of degree q , Euler's homogeneous function theorem is:

$$\mathbf{x} \cdot \nabla h = qh. \quad (4)$$

For the vector field $\mathbf{x}h$, we can write $\nabla \cdot (\mathbf{x}h) = (\nabla \cdot \mathbf{x})h + \mathbf{x} \cdot \nabla h$, and on invoking Stokes's theorem (divergence theorem or Green's theorem in the plane), we have

$$\int_{\Omega} (\nabla \cdot \mathbf{x})h \, d\mathbf{x} + \int_{\Omega} \mathbf{x} \cdot \nabla h \, d\mathbf{x} = \int_{\Omega} \nabla \cdot (\mathbf{x}h) \, d\mathbf{x} = \int_{\partial\Omega} (\mathbf{x} \cdot \mathbf{n})h \, ds, \quad (5)$$

where \mathbf{n} is the unit outward normal to C_i and ds is the differential length on C_i . Substituting $\nabla \cdot \mathbf{x} = 2$ and Euler's homogeneous function theorem (4), (5) becomes

$$(2 + q) \int_{\Omega} h \, d\mathbf{x} = \int_{\Omega} \nabla \cdot (\mathbf{x}h) \, d\mathbf{x} = \int_{\partial\Omega} (\mathbf{x} \cdot \mathbf{n})h \, ds. \quad (6)$$

Given a curved triangular domain \mathcal{T} , the boundary is the union of two line segments: \mathcal{L}_1 and \mathcal{L}_2 and a curve \mathcal{C} (see Fig. 3). Thus, over \mathcal{T} , (6) is

$$(2 + q) \int_{\mathcal{T}} h \, d\mathbf{x} = \int_{\mathcal{C}} (\mathbf{x} \cdot \mathbf{n})h \, ds + \sum_{i=1}^2 \int_{\mathcal{L}_i} (\mathbf{x} \cdot \mathbf{n})h \, ds. \quad (7)$$

We note that line segments \mathcal{L}_1 and \mathcal{L}_2 pass through the point \mathbf{x}_0 . On choosing $\mathbf{x}_0 = \mathbf{0}$, any point on $\mathcal{L}_1 \setminus \{\mathbf{0}\}$ or $\mathcal{L}_2 \setminus \{\mathbf{0}\}$ is then perpendicular to \mathbf{n} , and therefore, $\mathbf{x} \cdot \mathbf{n} = 0$ for both line segments. As a result, (7) reduces to

$$(2 + q) \int_{\mathcal{T}} h \, d\mathbf{x} = \int_{\mathcal{C}} (\mathbf{x} \cdot \mathbf{n})h \, ds. \quad (8)$$

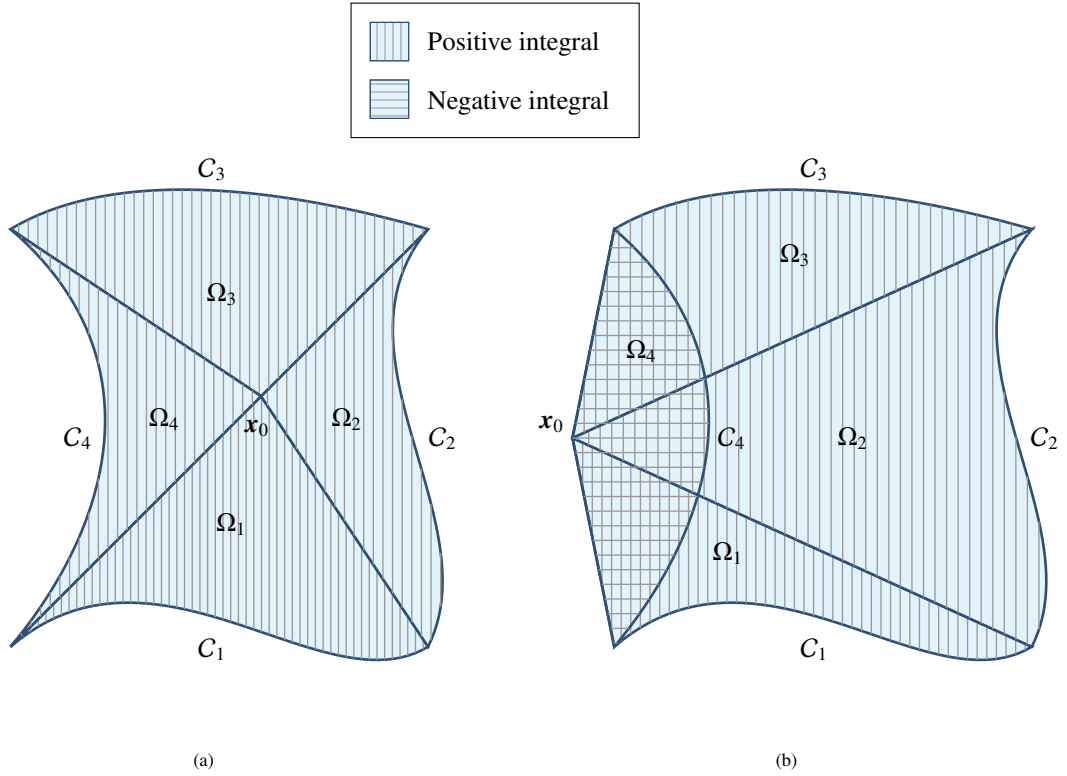


Figure 2: Two choices for decomposing the star-convex domain Ω in Fig. 1. (a) This choice of \mathbf{x}_0 results in positive regions of integration over all subdomains, (b) with this choice of \mathbf{x}_0 , the value of the integral over Ω_4 must be negated from the sum of the integrals over the other subregions.

Since a parametric description of C is presumed to be available, an explicit expression for \mathbf{n} is developed. We now define

$$C = \left\{ \mathbf{x} \in \mathbb{R}^2 : \mathbf{x} := \mathbf{c}(t) \text{ for } t \in [0, 1] \right\}, \quad (9)$$

where we have introduced $\mathbf{c}(t)$ as the equation of the curve parametrized by $t \in [0, 1]$. Bounds for t are limited between zero and one to simplify the expressions that follow. Curves that utilize different bounds of the parameter t can be scaled to match the definition used herein. We require the curve parameter t to follow a counterclockwise orientation. For example, on the curved triangle in Fig. 3, a counterclockwise orientation of t would result in $\mathbf{c}(0)$ and $\mathbf{c}(1)$ being points on \mathcal{L}_1 and \mathcal{L}_2 , respectively. Now, we observe that the unit normal vector is the unit tangent vector rotated through an angle of $-\pi/2$:

$$\mathbf{n} = \frac{\mathbf{c}'^\perp(t)}{\|\mathbf{c}'(t)\|}, \quad (10)$$

where $\mathbf{c}'^\perp(t) = \text{rot } \mathbf{c}'(t) := [c'_2(t) \ -c'_1(t)]^T$ is $\mathbf{c}'(t)$ rotated through $-\pi/2$ radians. Combining these results with (8) yields

$$\int_{\mathcal{T}} h d\mathbf{x} = \frac{1}{2+q} \int_C h \frac{\mathbf{c}(t) \cdot \mathbf{c}'^\perp(t)}{\|\mathbf{c}'(t)\|} ds, \quad (11)$$

where $\mathbf{c}(t) \cdot \mathbf{c}'^\perp(t) = c_1(t)c'_2(t) - c_2(t)c'_1(t)$. On using the HNI method, the integral of a homogeneous function h over a curved triangle \mathcal{T} with $\mathbf{x}_0 = \mathbf{0}$ is reduced to a line integral over the bounding curve C . We remark the quantity $(\mathbf{c}_i(t) \cdot \mathbf{c}'^\perp(t))/\|\mathbf{c}'(t)\|$ is the signed distance of the tangent line of $\mathbf{c}_i(t)$ to the point $\mathbf{x}_0 = \mathbf{0}$.

The boundary of Ω is defined in (1). Partitioning the boundary and using the parametric form of the curve given in (9), (6) becomes

$$\int_{\Omega} h d\mathbf{x} = \frac{1}{2+q} \sum_{i=1}^m \int_{C_i} h \frac{\mathbf{c}_i(t) \cdot \mathbf{c}'_i^\perp(t)}{\|\mathbf{c}'_i(t)\|} ds, \quad (12)$$

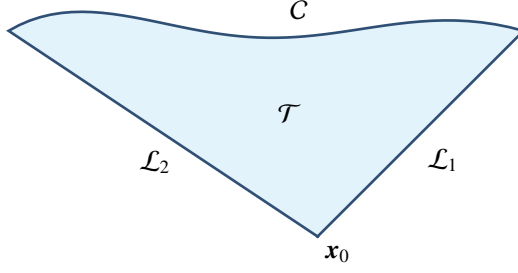


Figure 3: A curved triangle.

where $\mathbf{c}_i(t)$ is the parametric form of the curve C_i . Using (11), (12) can be equivalently expressed as

$$\int_{\Omega} h(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^m \int_{\Omega_i} h(\mathbf{x}) d\mathbf{x}, \quad (13)$$

which shows that the HNI method calculates integrals through a subdivision of Ω into curved triangles.

2.2. Scaled boundary parameter in the HNI method

As (11) makes clear, the HNI method converts integration from a two-dimensional curved triangle to a one-dimensional integral over the curve C . This implies the HNI method exactly computes the integral over another parameter, provided $h(\mathbf{x})$ is homogeneous. We introduce the map $\psi : [0, \rho(t)] \times [0, 1] \rightarrow \mathcal{T}^+ \supseteq \mathcal{T}$ that maps the radial distance from \mathbf{x}_0 , $r = \|\mathbf{x} - \mathbf{x}_0\|$, and the curve parameter to the domain \mathcal{T}^+ . If \mathcal{T} is star-convex with respect to \mathbf{x}_0 , then $\mathcal{T}^+ = \mathcal{T}$. With the HNI method, $\mathbf{x}_0 = \mathbf{0}$, so the radial distance reduces to $r = \|\mathbf{x}\|$. Define $\rho(t) := \|\mathbf{c}(t)\|$ to be the distance from the origin to C at parameter value t . On closer inspection of (11), the relationship between the HNI method and the radial distance r is revealed. In (11), the only part of the integrand not related to the geometry of the curve is $h(\mathbf{c}(t))$. So, $h(\mathbf{c}(t)) := (h \circ \mathbf{c})(t)$ must be the result of a one-dimensional integral times a function of t . Note that,

$$h(\mathbf{c}(t)) = \frac{\rho^2(t)}{\rho^2(t)} h(\mathbf{c}(t)) = \frac{1}{\rho^2(t)} \int_0^{\rho(t)} \frac{\partial}{\partial r} (r^2 (h \circ \psi)(r, t)) dr. \quad (14)$$

On using the product rule, we obtain

$$h(\mathbf{c}(t)) = \frac{1}{\rho^2(t)} \left(2 \int_0^{\rho(t)} r (h \circ \psi)(r, t) dr + \int_0^{\rho(t)} r^2 \frac{\partial (h \circ \psi)(r, t)}{\partial r} dr \right). \quad (15)$$

The map ψ is better understood through the lens of the polar transformation:

$$x = r \cos \theta(t), \quad y = r \sin \theta(t), \quad (16a)$$

and the mapping $\theta : (0, 1) \rightarrow (\theta_1, \theta_2)$,

$$\theta = \arctan \left(\frac{c_2(t)}{c_1(t)} \right), \quad (16b)$$

where $\mathbf{c}(t) = [c_1(t) \ c_2(t)]^T$. From (16), if $h(\mathbf{x})$ is a homogeneous function of degree q , then $(h \circ \psi)(r, t)$ is q -homogeneous in r when $\mathbf{x}_0 = \mathbf{0}$. As a result, on applying Euler's homogeneous function theorem to the second term on the right-hand side of (15), we obtain

$$\frac{1}{2+q} h(\mathbf{c}(t)) = \frac{1}{\rho^2(t)} \int_0^{\rho(t)} r (h \circ \psi)(r, t) dr. \quad (17)$$

Substituting (17) into (11) results in

$$\int_{\mathcal{T}} h(\mathbf{x}) d\mathbf{x} = \int_C \int_0^{\rho(t)} r (h \circ \psi)(r, t) dr \frac{\mathbf{c}_i(t) \cdot \mathbf{c}_i^{\perp}(t)}{\rho^2(t) \|\mathbf{c}'_i(t)\|} ds. \quad (18)$$

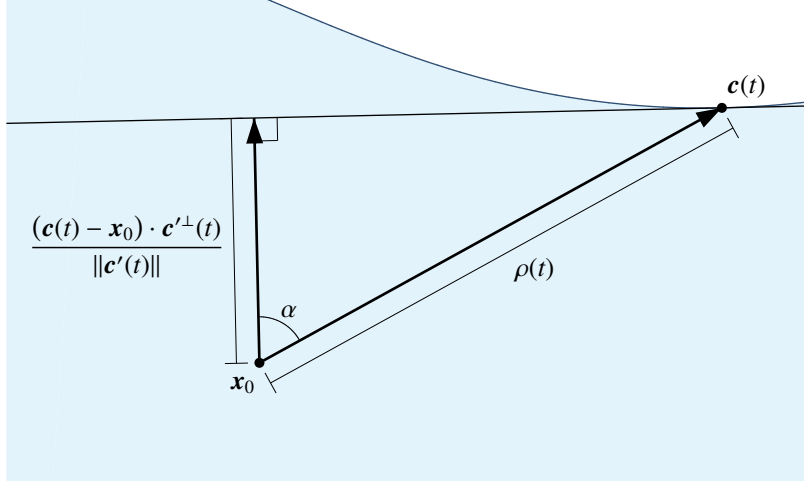


Figure 4: Interpreting quantities in (18) as the cosine of an angle α .

From (18), the HNI method implicitly (and exactly) integrates a parameter in the radial direction, provided $h(\mathbf{x})$ is a homogeneous function. A physical interpretation of $(\mathbf{c}_i(t) \cdot \mathbf{c}'^{\perp}(t)) / (\rho(t) \|\mathbf{c}'(t)\|)$ as the cosine of an angle $\alpha := \alpha(t)$ is illustrated in Fig. 4. Introducing the angle α allows (18) to be written as

$$\int_{\mathcal{T}} h(\mathbf{x}) d\mathbf{x} = \int_C \int_0^{\rho(t)} (h \circ \psi)(r, t) \frac{r \cos \alpha}{\rho(t)} dr ds. \quad (19)$$

The quantity $r \cos \alpha / \rho(t)$ can be interpreted as scaling of the differential length ds . Specifically, $r / \rho(t)$ provides a proportional scaling of the length ds based on the relative proximity of the radial coordinate to the curve $\mathbf{c}(t)$. Additionally, $\cos \alpha$ reduces ds as the vector $\mathbf{c}(t) - \mathbf{x}_0$ deviates in direction from the outward normal vector at $\mathbf{c}(t)$.

To simplify (18), we explicitly define

$$\xi = \frac{r}{\rho(t)}, \quad d\xi = \frac{dr}{\rho(t)}, \quad (20)$$

where $\xi \in [0, 1]$ is a scaled boundary (or a distorted polar) coordinate in the radial direction. The point \mathbf{x}_0 corresponds to $\xi = 0$ and along the boundary curve $\mathbf{c}(t)$ we have $\xi = 1$. Replacing (20) in (18) and integrating over the curve parameter t lead to

$$\int_{\mathcal{T}} h(\mathbf{x}) d\mathbf{x} = \int_0^1 \int_0^1 (h \circ \psi)(\xi \rho(t), t) \xi d\xi [\mathbf{c}_i(t) \cdot \mathbf{c}'^{\perp}(t)] dt. \quad (21)$$

We define the mapped domain of integration $\Omega_0 = [0, 1]^2$ as the parametric (reference) domain, where $(\xi, t) \in \Omega_0$ is a point in the unit square. Equation (21) is derived for a single curved triangle, though the (ξ, t) -parametrized integral can be applied to a more general curved domain Ω using (13), which relies on the (implicit) partitioning of Ω into curved triangles. Lastly, we mention that we can also derive (21) using the Poincaré Lemma for vector fields [42] (see Appendix A).

3. Scaled boundary transformation for integration

The developments in Section 2 provide some insight into how the HNI method works: by subdividing the domain into curved triangles and implicitly integrating in the radial direction. The integral in the radial direction is manifested through the scaled boundary coordinate, $\xi \in [0, 1]$. However, the derivation in Section 2 utilizes Euler's homogeneous function theorem, which limits application to homogeneous functions. Based on the insights of Section 2, there must exist a transformation from the (ξ, t) parametrization to the (x, y) parametrization that is not reliant on the properties of homogeneous functions. Herein, we show that this mapping is in fact the SB parametrization.

3.1. Integrating over curved planar domains

Mirroring the HNI method, we decompose the integral over a domain Ω into the summation of integrals over its curved triangular partitions, Ω_i for $i = 1, \dots, m$. Instead of requiring $\mathbf{x}_0 = \mathbf{0}$, we now permit arbitrary selection of \mathbf{x}_0 . Judicious selection of \mathbf{x}_0 on star-convex domains can result in positive cubature weights and/or improved distribution of cubature points throughout the domain. Recall the partition Ω_i is bounded by two line segments and one parametric boundary curve, C_i . The parametric equation for C_i is given by $\mathbf{c}_i(t)$, $t \in [0, 1]$. Next, we introduce the SB parametrization, which transforms the domain from $\Omega_i^+ \supseteq \Omega_i$ to $\Omega_0 = [0, 1]^2$. If Ω_i is star-convex with respect to \mathbf{x}_0 , then $\Omega_i^+ = \Omega_i$. The SB parametrization $\boldsymbol{\varphi} : \Omega_0 \rightarrow \Omega_i^+$ is:

$$\mathbf{x} = \boldsymbol{\varphi}(\xi, t) = \mathbf{x}_0 + \xi(\mathbf{c}_i(t) - \mathbf{x}_0), \quad (22)$$

which matches the domain mapping introduced in Song and Wolf [4], though we include the coordinate \mathbf{x}_0 to allow shifting the center of the parametrization away from the origin.

Let $J(\xi, t) = \det(\nabla \boldsymbol{\varphi}(\xi, t))$ be the Jacobian of the SB parametrization. Over a curved triangle \mathcal{T} , integration of a function $f : \mathcal{T} \rightarrow \mathbb{R}$ is then

$$\int_{\mathcal{T}} f(\mathbf{x}) d\mathbf{x} = \int_0^1 \int_0^1 f(\boldsymbol{\varphi}) J(\xi, t) d\xi dt, \quad (23)$$

where $\boldsymbol{\varphi} := \boldsymbol{\varphi}(\xi, t)$. Note that here the function f is not required to be homogeneous. The gradient of the parametrization is:

$$\nabla \boldsymbol{\varphi}(\xi, t) = \begin{bmatrix} \frac{\partial x_1}{\partial \xi} & \frac{\partial x_2}{\partial \xi} \\ \frac{\partial x_1}{\partial t} & \frac{\partial x_2}{\partial t} \end{bmatrix}, \quad (24a)$$

where

$$\frac{\partial \mathbf{x}}{\partial \xi} = \mathbf{c}(t) - \mathbf{x}_0, \quad \frac{\partial \mathbf{x}}{\partial t} = \xi \mathbf{c}'(t). \quad (24b)$$

The Jacobian is then

$$J(\xi, t) = \det(\nabla \boldsymbol{\varphi}(\xi, t)) = \xi \det \begin{bmatrix} \mathbf{c}(t) - \mathbf{x}_0 & \mathbf{c}'(t) \end{bmatrix} = \xi (\mathbf{c}(t) - \mathbf{x}_0) \cdot \mathbf{c}'^{\perp}(t), \quad (25)$$

where $\mathbf{c}'^{\perp}(t)$ is defined in (10). The vector $\mathbf{c}'^{\perp}(t)$ is in the outward normal direction from \mathcal{T} and $(\mathbf{c}(t) - \mathbf{x}_0) \cdot \mathbf{c}'^{\perp}(t)$ is the signed distance of the tangent to \mathbf{x}_0 , scaled by the inverse of the curve speed $\|\mathbf{c}'(t)\|$ (see Fig. 4). Equation (23) can also be derived by combining the SB parametrization with the Poincaré Lemma. See Appendix A for details.

An example of the SB parametrization over the region in Fig. 2a is illustrated in Fig. 5. We note the map introduced in (22) is not bijective since $J(0, t) = 0$, and $\xi = 0$ is a singular point. In Fig. 5, $\xi = 0$ corresponds to the point \mathbf{x}_0 , which is illustrated in Fig. 2a. Lack of bijectivity has ramifications when used within an isogeometric framework for the solution of boundary-value problems; for details and further connections, see Arioli et al. [43] and the references therein. However, when applied to numerical integration, the SB parametrization does not pose any issues since the edge in Ω_i that is associated with $\xi = 0$ has zero measure (area), and hence bijectivity for $\xi > 0$ suffices.

To close this section, we mention the integral over Ω is simply the summation of the integrals over each curved triangle:

$$\int_{\Omega} f(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^m \int_0^1 \int_0^1 f(\boldsymbol{\varphi}) \xi d\xi [(\mathbf{c}_i(t) - \mathbf{x}_0) \cdot \mathbf{c}'_i{}^{\perp}(t)] dt, \quad (26)$$

where the Jacobian of the transformation in (25) is explicitly introduced in (26). Note when $\mathbf{x}_0 = \mathbf{0}$, the transformed integral in (26) matches (21). Additionally, if the edges of Ω are properly ordered counterclockwise, then the sign of the integral is correctly accounted for even if \mathbf{x}_0 lies outside Ω or if Ω is nonconvex. If Ω_i is not star-convex, then the domain $\Omega_i^+ \setminus \Omega_i$ is integrated an even number of times, though multiplied by the scalar -1 half of those times. Finally, if $f(\mathbf{x})$ is a m -th degree polynomial and $\mathbf{c}_i(t)$ is a p -th degree polynomial curve, then $f(\boldsymbol{\varphi})$ is a polynomial, and furthermore, the integrand in (26) is a polynomial of degree $(m+1)$ in ξ and degree $[(m+2)p-1]$ in t that can be exactly integrated with an appropriate tensor-product Gauss rule.

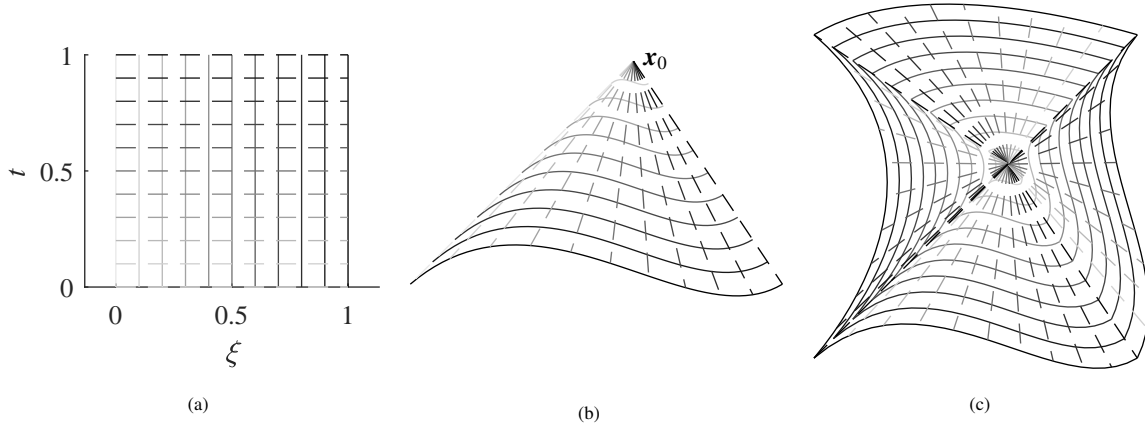


Figure 5: Isocontours of ξ (solid lines) and t (dashed lines) mapped from (a) the unit square to (b) a curved triangle. (c) The region in Fig. 2a with ξ and t isocontours on its curved triangles.

3.2. Integrating over polygons

When the region Ω is a polygon, boundary curves C_i ($i = 1, \dots, m$) are reduced to line segments. One choice for parametrizing the curve is

$$\tilde{c}_i(\tau) = \ell_i \mathbf{n}_i + \tau \boldsymbol{\tau}_i + \mathbf{x}_0, \quad (27)$$

where \mathbf{n}_i is the unit outward normal vector to C_i , ℓ_i is the signed distance from \mathbf{x}_0 to the line on which the line segment C_i lies, $\boldsymbol{\tau}_i = \mathbf{e}_3 \times \mathbf{n}_i$ is a unit vector tangent to C_i , and $(\tau_1)_i < \tau < (\tau_2)_i$ is a parameter. The interval of τ is given by $(\tau_1)_i = \sqrt{\|(\mathbf{x}_1)_i - \mathbf{x}_0\|^2 - \ell_i^2}$ and $(\tau_2)_i = \sqrt{\|(\mathbf{x}_2)_i - \mathbf{x}_0\|^2 - \ell_i^2}$, where $(\mathbf{x}_1)_i$ and $(\mathbf{x}_2)_i$ are the coordinates of the endpoints of C_i . Rather than $t \in [0, 1]$ as suggested in Section 2.1, a different domain of definition for t is used because it simplifies integral transforms for weakly singular functions that appear in Section 4.2. To map the domain to the unit square, the affine mapping

$$\tau = (\tau_1)_i (1 - t) + (\tau_2)_i t, \quad d\tau = ((\tau_2)_i - (\tau_1)_i) dt = \Delta\tau_i dt, \quad (28)$$

is applied. In (28), $\Delta\tau_i = (\tau_2)_i - (\tau_1)_i$ is the length of C_i . Given the boundary curve $\tilde{c}_i(\tau)$, (23) simplifies to

$$\int_{\Omega_i} f(\mathbf{x}) d\mathbf{x} = \ell_i \Delta\tau_i \int_0^1 \int_0^1 f(\boldsymbol{\varphi}) \xi d\xi dt. \quad (29)$$

Integration over the polygonal domain Ω is then given by

$$\int_{\Omega} f(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^m \ell_i \Delta\tau_i \int_0^1 \int_0^1 f(\boldsymbol{\varphi}) \xi d\xi dt. \quad (30)$$

For polygonal domains, selecting \mathbf{x}_0 as a vertex eliminates the need to perform cubature on the two edges that contain the vertex. This is apparent in (30) since $\ell_i = 0$ for the two edges that pass through \mathbf{x}_0 . However, it is worth noting that the influence of \mathbf{x}_0 being at a vertex on the number of cubature points reduces with increasing numbers of edges in the polygon. For domains that are bounded by both affine and non-affine curves, (29) and (23) can be utilized as needed to simplify integration over the domain.

When f is a p -th degree polynomial, it is a sum of monomials that have the form $x^\alpha y^\beta$, where $\alpha, \beta \in \mathbb{Z}^{\geq}$, $\alpha + \beta \leq p$, and \mathbb{Z}^{\geq} is the set of integers greater than or equal to zero. The monomial can be expressed in (ξ, t) -coordinates through the mapping of the domain introduced in (22). We assume $\mathbf{e}_1 = \mathbf{n}$, $\mathbf{e}_2 = \boldsymbol{\tau}$, and $\mathbf{x}_0 = \mathbf{0}$ in (27), where \mathbf{e}_1 and \mathbf{e}_2 are unit vectors in the x - and y -directions, respectively. These values for \mathbf{n} , $\boldsymbol{\tau}$, and \mathbf{x}_0 can be obtained through a translational and rotational mapping of the (x, y) -coordinates. When this mapping is applied to $f(\mathbf{x})$, it does not change the polynomial nature of $f(\mathbf{x})$ or polynomial degree of $f(\mathbf{x})$. The transformed variables are then

$$x = \xi \ell_i, \quad y = \xi \tau. \quad (31)$$

Inserting $f = x^\alpha y^\beta$ in (30) and substituting the transformed variables gives

$$\int_{\Omega} f(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^m \ell_i^{\alpha+1} \Delta\tau_i \int_0^1 \int_0^1 \xi^{\alpha+\beta+1} [(\tau_1)_i (1-t) + (\tau_2)_i t]^\beta d\xi dt. \quad (32)$$

Based on (32), a cubature rule capable of exactly integrating polynomials of degree $p+1$ in the ξ direction and degree p in the t direction over each triangle Ω_i is capable of exactly integrating a polynomial of degree p over a polygon using (30).

As will be explored in Section 5, cubature over polygons using this rule requires more integration points than a similar triangulation paired with an optimized, symmetric cubature rule; however, there are several advantages to the SBC approach. First, the tensor-product structure of the SBC rule allows it to be defined with reduced memory requirements, which may be beneficial in applications where memory is limited and data movement is computationally expensive (such as in GPU programming). Also, the triangulation of the domain is trivial with the SBC method and does not change with nonconvex polygons. Finally, Gauss rules of arbitrary order can be easily generated to compute (30), while high-order symmetric rules over triangles, if not available on-hand, are time consuming to generate.

Finally, we point out that if $h(\mathbf{x})$ is a homogeneous function of degree q , then on applying (17) and (20) to (30), we obtain

$$\int_{\Omega} h(\mathbf{x}) d\mathbf{x} = \frac{1}{2+q} \sum_{i=1}^m \ell_i \int_{\Omega_i} h(\mathbf{x}) ds. \quad (33)$$

Noting that $\ell_i = b_i/\|\mathbf{a}_i\|$, where $\mathbf{a}_i \cdot \mathbf{x} = b_i$ is the equation of the line on which C_i lies, (33) matches the result in Chin et al. [2]. In Chin et al. [2], integration over polynomials using the HNI method is presented. To integrate a p -th degree polynomial, the polynomial must first be decomposed into $(p+1)$ homogeneous polynomials. Then, (33) is applied to each homogeneous term. With the HNI method, $(p+1)^2/2$ evaluations per edge are required to exactly integrate p -th degree polynomials using a Gauss-Legendre rule. In comparison, integrating a p -th degree polynomial using (30) requires $(p+2)(p+1)/4$ Gauss-Legendre points per edge.

4. Integration of weakly singular and nearly singular functions

In this section, we introduce methods for handling integration of weakly singular functions. These methods modify the SB mapping from the unit square to the curved triangle to remove or smooth singularities in the integrand. Section 4.1 introduces an additional parameter in the SB mapping that eliminates rational singularities in the ξ direction. Section 4.2 introduces several mappings that smooth near singularities in the t direction of the SB map.

4.1. Cancelling weak singularities in the radial direction

Given a function $g(\mathbf{x}) \in C^0(\Omega)$, with a leading term that is a constant, the integral

$$I := \int_{\Omega} \frac{g(\mathbf{x})}{\|\mathbf{x} - \mathbf{x}_c\|^\beta} d\mathbf{x} \quad (34)$$

for $0 < \beta < 2$ is said to be *weakly singular* when $\Omega \subset \mathbb{R}^2$ and $\mathbf{x}_c \in \bar{\Omega}$. For a weakly singular integral, $|I| < \infty$. In (34), \mathbf{x}_c serves as the location of a point singularity. When $\mathbf{x}_c \in \bar{\Omega}$ or when \mathbf{x}_c is near $\partial\Omega$, standard polynomial-precision cubature rules perform poorly when used to integrate (34). To improve cubature convergence, special integration techniques are usually employed. One such technique is the Duffy transformation [44], which introduces the change of variables,

$$x = u, \quad y = xv = uv, \quad (35)$$

which maps the standard triangle $T_0 := [0, 1] \times [0, x]$ to the unit square $\Omega_0 = [0, 1]^2$. This mapping eliminates singularities when $\beta = 1$ in (34). On T_0 , the SB parametrization is the same as the Duffy transformation (see (31) and note $\ell_i = 1$), though the SB parametrization is not limited to this triangular domain. Its ability to eliminate singularities when $\beta = 1$ can be observed by applying the SB parametrization (22) with $\mathbf{x}_0 = \mathbf{x}_c$ to (34):

$$\int_{\Omega} \frac{g(\mathbf{x})}{\|\mathbf{x} - \mathbf{x}_c\|} d\mathbf{x} = \sum_{i=1}^m \int_0^1 \int_0^1 g(\boldsymbol{\varphi}) d\xi \frac{(\mathbf{c}_i(t) - \mathbf{x}_c) \cdot \mathbf{c}_i^{\prime\perp}(t)}{\|\mathbf{c}_i(t) - \mathbf{x}_c\|} dt. \quad (36)$$

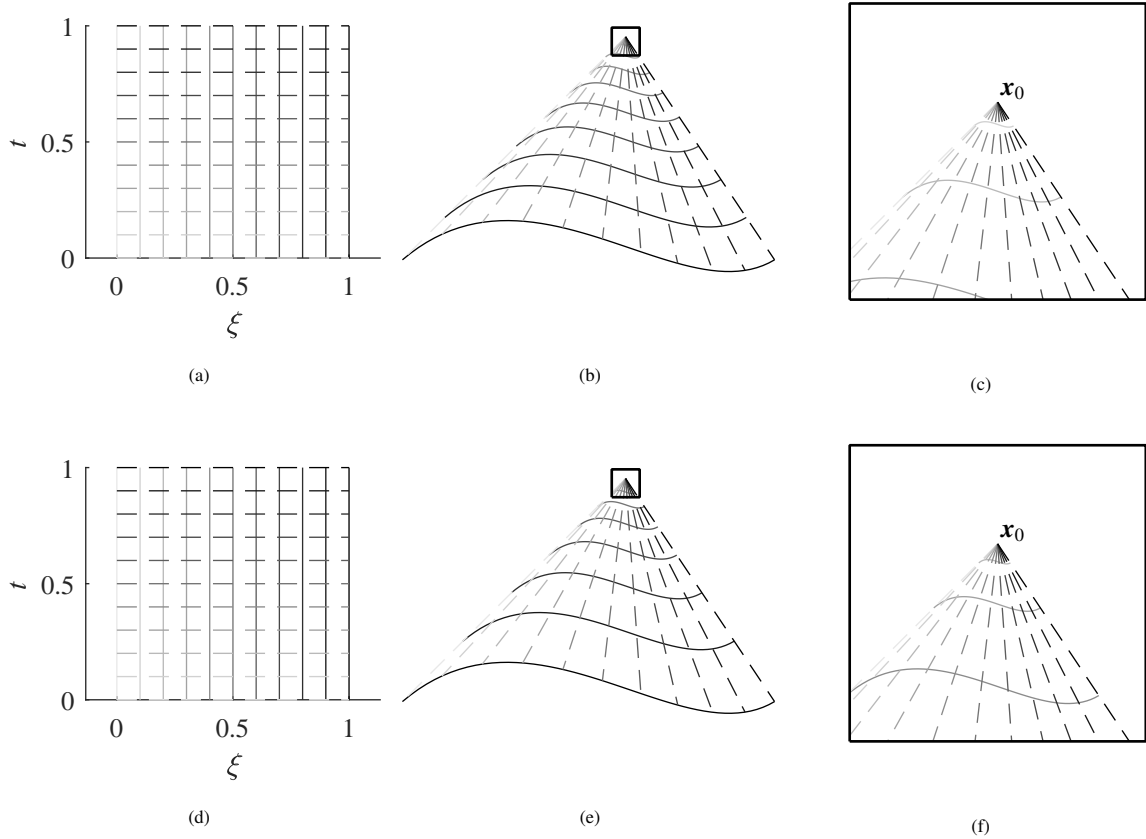


Figure 6: Isocontours of ξ (solid lines) and t (dashed lines) mapped from (a), (d) the unit square to (b), (e) a curved triangle with the generalized SB parametrization. (c), (f) Zoomed region illustrating more ξ isocontours near \mathbf{x}_0 as α increases. In (a)–(c), $\alpha = 2$ and in (d)–(f), $\alpha = 3$. As α increases, the isocontours of ξ shift toward \mathbf{x}_0 .

In (36), the singularity in the radial direction is eliminated, thereby leaving a smooth integrand.

More generally, when $0 < \beta < 2$, the generalized Duffy transformation [38] enables polynomial-precision integration to be recovered for $\Omega = T_0$ in (36). The transformation is:

$$x = u^\alpha, \quad y = xv = u^\alpha v. \quad (37)$$

In (37), α is selected such that $\alpha \in \mathbb{Z}_+$ and $\alpha(2 - \beta) \in \mathbb{Z}_+$. Mimicking the generalized Duffy transformation, a more general mapping on Ω_i is invoked:

$$\mathbf{x} = \varphi_\alpha(\xi, t) = \mathbf{x}_0 + \xi^\alpha(\mathbf{c}_i(t) - \mathbf{x}_0) \quad (38)$$

for $\alpha > 0$. We will refer to this as a generalized SB parametrization. The isocontours of ξ and t with the generalized SB parametrization for $\alpha = 2$ and $\alpha = 3$ are presented in Fig. 6. As α increases, the number of ξ -isocontours in the vicinity of \mathbf{x}_c increases. In the context of a cubature rule, this has the practical effect of placing more integration points closer to the singularity as α gets larger. For the triangle T_0 , the generalized SB transformation is identical to the generalized Duffy transformation. The map for $\mathbf{c}_i(t)$ in T_0 is:

$$\mathbf{c}_i(t) = [1 \quad t]^T, \quad (39a)$$

so the generalized SB map is

$$x = \xi^\alpha, \quad y = xv = \xi^\alpha t, \quad (39b)$$

which matches the generalized Duffy map in (37).

The gradient of the generalized SB transformation is given by (24a), where now

$$\frac{\partial \mathbf{x}}{\partial \xi} = \alpha \xi^{\alpha-1} (\mathbf{c}(t) - \mathbf{x}_0), \quad \frac{\partial \mathbf{x}}{\partial t} = \xi^\alpha \mathbf{c}'(t). \quad (40)$$

The Jacobian of the mapping is:

$$J(\xi, t) = \det(\nabla \boldsymbol{\varphi}_\alpha(\xi, t)) = \alpha \xi^{2\alpha-1} (\mathbf{c}_i(t) - \mathbf{x}_0) \cdot \mathbf{c}'_i(t). \quad (41)$$

After applying the generalized SB transformation (38) to (34) and selecting $\mathbf{x}_0 = \mathbf{x}_c$, we obtain

$$\int_{\Omega} \frac{g(\mathbf{x})}{\|\mathbf{x} - \mathbf{x}_c\|^\beta} d\mathbf{x} = \alpha \sum_{i=1}^m \int_0^1 \int_0^1 g(\boldsymbol{\varphi}_\alpha) \xi^{\alpha(2-\beta)-1} d\xi \frac{(\mathbf{c}_i(t) - \mathbf{x}_c) \cdot \mathbf{c}'_i(t)}{\|\mathbf{c}_i(t) - \mathbf{x}_c\|^\beta} dt, \quad (42)$$

where $\boldsymbol{\varphi}_\alpha := \boldsymbol{\varphi}_\alpha(\xi, t)$. On selecting α such that $\alpha(2-\beta) \in \mathbb{Z}_+$, the $r^{-\beta}$ -singularity in (34) is eliminated. However, depending on the choice for α and the form of $g(\boldsymbol{\varphi}_\alpha)$, integration convergence may still be poor. For example, if $g(\mathbf{x})$ is a polynomial function, selecting $\alpha \notin \mathbb{Z}_+$ results in $g(\boldsymbol{\varphi}_\alpha)$ being non-polynomial, which potentially increases cubature error and reduces convergence rates when using a Gauss rule in the ξ -direction. To ensure $g(\boldsymbol{\varphi}_\alpha)$ remains polynomial, we select the smallest $\alpha \in \mathbb{Z}_\beta$, where

$$\mathbb{Z}_\beta = \left\{ \alpha \in \mathbb{Z}_+ : \alpha(2-\beta) \in \mathbb{Z}_+ \right\}. \quad (43)$$

The same criterion for selecting α is used in the generalized Duffy transformation [38].

The generalized SB parametrization eliminates fractional singularities in an integrand and recovers polynomial-precision integration for the non-singular portion of an integrand. However, the degree of the transformed polynomials can be significantly higher than the untransformed integrand. For example, if $\beta = 1/\gamma$ for $\gamma \in \mathbb{Z}_+$, then $\alpha = \gamma$ is the best selection in \mathbb{Z}_β . If $g(\mathbf{x})$ is a p -th degree polynomial, the ξ integrand after applying the generalized SB mapping is a polynomial of degree $\gamma(p+2-\beta)-1 = O(\gamma p)$. While the increase in polynomial order may be a worthwhile tradeoff for improved handling of the singularity for small values of γ , we note the polynomial order of $g(\boldsymbol{\varphi}_\alpha)$ quickly increases as γ grows. Rather than resorting to a high-order Gauss-Legendre rule paired with the generalized SB transformation, we find that Gauss-Jacobi quadrature [45] in the ξ direction provides a much more efficient quadrature rule for integrands when γ is large.

One important distinction between the generalized Duffy transformation and the generalized SB parametrization is the applicability of the SB parametrization to arbitrary convex and nonconvex polygons and regions bounded by curves. This allows the generalized SB parametrization to be applied directly to these shapes, whereas the generalized Duffy transformation first requires a partitioning of Ω and a mapping from each subdomain to T_0 . This mapping, if not chosen appropriately, can distort the singularity. In addition, if the boundaries of the partitions are not affine, the mapping may be difficult to compute. Also, with the SB parametrization, integration over regions where $\mathbf{x}_c \notin \bar{\Omega}$ is handled intrinsically (see Section 2 for details), which allows the transformation to be easily applied to integrate over regions that contain nearby singularities. Integration of functions with singularities very close to the region of integration (*nearly singular* integrands) has poor cubature convergence when integrated with Gauss rules [19].

4.2. Cancelling near-singularities in the edge direction of polygons

When integrands contain singularities in the vicinity of $\partial\Omega$, integrands in the t direction can be nearly singular. While these integrands remain smooth and differentiable, they are poorly approximated by polynomials and standard Gauss quadrature rules do not lead to efficient evaluation of these integrals. When Ω is a polygon, the boundary curves C_i are line segments, which simplifies integration using the SB parametrization. Furthermore, this simplified representation allows the formulation of integral transformations that eliminate from the integrand near-singularities that are proximal to C_i . This idea was presented in Ma and Kamiya [46], where it was shown to improve integration in the BEM. More recently, Lv et al. [47] demonstrated the applicability of the transformation in combination with the generalized Duffy transformation. In this section, we show the transformation is also relevant here, and furthermore, we introduce two more transformations that cancel other types of near-singularities.

When the generalized SB parametrization is applied to the integral in (34), the result is (42). However, when Ω is a polygon, the curve parametrization simplifies to (27) with $\mathbf{x}_0 = \mathbf{x}_c$:

$$\tilde{\mathbf{c}}_i(\tau) = \ell_i \mathbf{n}_i + \tau \boldsymbol{\tau}_i + \mathbf{x}_c. \quad (44)$$

Given this parametrization, the distance to the singularity simplifies to

$$r(\tau) = \|\tilde{\mathbf{c}}_i(\tau) - \mathbf{x}_c\| = \sqrt{\ell_i^2 + \tau^2}. \quad (45)$$

Inserting (45) into (42) and parametrizing the boundary curves by τ results in

$$\int_{\Omega} \frac{g(\mathbf{x})}{\|\mathbf{x} - \mathbf{x}_c\|^\beta} d\mathbf{x} = \alpha \sum_{i=1}^m \ell_i \int_{(\tau_1)_i}^{(\tau_2)_i} \int_0^1 g(\boldsymbol{\varphi}_a) \xi^{\alpha(2-\beta)-1} d\xi \frac{1}{(\ell_i^2 + \tau^2)^{\beta/2}} d\tau. \quad (46)$$

When $\beta = 1$ (and $\alpha = 1$), the SB parametrization is recovered and $1/r$ singularities are removed in the ξ direction. To remove the near-singularity in the τ direction caused by the radial dependence of the integrand, we define the transformation

$$d\tilde{\tau} = \frac{1}{\sqrt{\ell_i^2 + \tau^2}} d\tau. \quad (47)$$

Integrating (47) gives

$$\tilde{\tau}(\tau) = \ln\left(\tau + \sqrt{\ell_i^2 + \tau^2}\right), \quad (48a)$$

and solving for t results in [46, 47]

$$\tau(\tilde{\tau}) = \frac{1}{2} e^{-\tilde{\tau}} \left(e^{2\tilde{\tau}} - \ell_i^2 \right). \quad (48b)$$

Applying the transform to (46) (with $\beta = \alpha = 1$) yields

$$\int_{\Omega} \frac{g(\mathbf{x})}{\|\mathbf{x} - \mathbf{x}_c\|^\beta} d\mathbf{x} = \sum_{i=1}^m \ell_i \int_{\ln((\tau_1)_i + \sqrt{\ell_i^2 + (\tau_1)_i^2})}^{\ln((\tau_2)_i + \sqrt{\ell_i^2 + (\tau_2)_i^2})} \int_0^1 g(\boldsymbol{\varphi}) d\xi d\tilde{\tau}, \quad (49)$$

where the near-singularity has been eliminated from the integral. Transforming integration to the unit square Ω_0 , we obtain:

$$\int_{\Omega} \frac{g(\mathbf{x})}{\|\mathbf{x} - \mathbf{x}_c\|^\beta} d\mathbf{x} = \sum_{i=1}^m \ell_i \ln\left(\frac{(\tau_2)_i + \sqrt{\ell_i^2 + (\tau_2)_i^2}}{(\tau_1)_i + \sqrt{\ell_i^2 + (\tau_1)_i^2}}\right) \int_0^1 \int_0^1 g(\boldsymbol{\varphi}) d\xi dt. \quad (50)$$

The SB parametrization paired with the near-singularity-cancelling transformation with $\beta = 1$ completely eliminates $1/r$ singularities from the integrand. However, even when the singularity is not of the type $1/r$, the $\beta = 1$ integral transformation can still reduce errors in integration by smoothing the near-singularity. This is explored further in Section 5.3.2.

Following a procedure similar to that above, integral transformations can be formulated for integrals with near-singularities that are of the form (34) with $\beta = 2$ and $\beta = 3$. While $|I| = \infty$ in (34) when $\beta \geq 2$ and $\mathbf{x}_c \in \bar{\Omega}$, the transformations are designed to eliminate the near-singularity from $(\ell_i^2 + \tau^2)^{-\beta/2}$ in (46), which remains bounded on all edges C_i where $|T_i| \neq 0$ (that is, $\ell_i \neq 0$). These two transformations, along with the transformation for $\beta = 1$, are listed in Table 1. The performance of all three transformations in integrating weak singularities is examined in Section 5.3.2. Further examples illustrating the performance of these transformations with the HNI method and singular integrands are available in Chin [48]. Chin et al. [2] present an alternative derivation for the integral transformation to cancel singularities of the type $1/r^2$ when $h(\mathbf{x})$ is a homogeneous function. This derivation relies on parametrization of (33) in terms of θ . The transformation in Chin et al. [2] is equivalent to the transformation in Table 1. This is apparent when noting the mapping to $\tilde{\tau}$ in Table 1 is equivalent to the mapping to θ , namely $\theta = \tan^{-1}(y/x)$, where without any loss of generality, we choose $\mathbf{n} = \mathbf{e}_1$ and $\mathbf{t} = \mathbf{e}_2$.

5. Numerical examples

This section evaluates the performance of the SBC method in numerically integrating various functions over different types of domains. Integration of polynomial and non-polynomial functions over polygons is presented in Section 5.1. We introduce comparisons to other methods of integration based on both accuracy per cubature point and time to generate an integration rule. Section 5.2 focuses on curved domains, demonstrating efficient integration of test

Table 1: Integral transformations in \mathbb{R}^2 to cancel near-singularities in (46).

$f(\mathbf{x})$	$d\tilde{\tau}$	$\tilde{\tau}(\tau)$	$t(\tilde{\tau})$
$\frac{g(\mathbf{x})}{r}$	$\frac{d\tau}{\sqrt{\ell_i^2 + \tau^2}}$	$\ln\left(\tau + \sqrt{\ell_i^2 + \tau^2}\right)$	$\frac{1}{2}e^{-\tilde{\tau}}(e^{2\tilde{\tau}} - \ell_i^2)$
$\frac{g(\mathbf{x})}{r^2}$	$\frac{\ell_i d\tau}{\ell_i^2 + \tau^2}$	$\tan^{-1} \frac{\tau}{\ell_i}$	$\ell_i \tan \tilde{\tau}$
$\frac{g(\mathbf{x})}{r^3}$	$\frac{\ell_i^2 d\tau}{(\ell_i^2 + \tau^2)^{3/2}}$	$\frac{\tau}{\sqrt{\ell_i^2 + \tau^2}}$	$\pm \frac{\ell_i \tilde{\tau}}{\sqrt{1 - \tilde{\tau}^2}}$

Table 2: Polynomials integrated over the polygons illustrated in Fig. 7.

p	$f(x, y)$
0	1
1	$x - 2y + 1$
2	$3x^2 + 4xy - 2y^2 - x + 2y - 3$
3	$4x^3 - 2x^2y - 3xy^2 + y^3 + 8x^2 - 4xy + 5y^2 - 6x - 4y + 7$
4	$-3x^4 - 5x^3y + 2x^2y^2 - 9xy^3 + y^4 + 3x^3 - 2x^2y - xy^2 + 5y^3 + 4x^2 - 7xy - 6y^2 - 4x + 6y - 8$
5	$10x^5 - 5x^4y - 7x^3y^2 + 6x^2y^3 + 3xy^4 + y^5 - x^4 + 2x^3y + 11x^2y^2 - 8xy^3 - 2y^4 - 3x^3 + 9x^2y + 8xy^2 - 10y^3 - 9x^2 - 6xy + 7y^2 + 5x - 4y + 4$

functions over regions bounded by both polynomial and rational curves. In Section 5.3, the examples center on the integration of weakly singular functions using the methods introduced in Section 4. These methods are compared and a test case that arises in the X-FEM is studied. Finally, Section 5.4 highlights how the SBC method can be used in conjunction with TMVI to compute L_2 error. We also point out a link between TMVI and homogeneous functions, which mirrors the connection between the SBC and HNI methods.

5.1. Polygonal domains

5.1.1. Polynomial integration

As discussed in Section 3.2, the SBC scheme is capable of exactly integrating polynomials of degree p over polygons when provided a polynomial-precise quadrature rule that is capable of integrating polynomials of degree $p + 1$ in the ξ direction and degree p in the t direction. This corresponds to a Gauss-Legendre rule of greater than or equal to $(p + 2)/2$ ξ -points and $(p + 1)/2$ t -points for a p -th degree polynomial integrand. Beyond requiring a closed polygonal region, no restrictions are placed on the shape of the polygon. Therefore, integration using the SBC method is possible over both convex and nonconvex polygons. We verify this approach on several convex and nonconvex polygons illustrated in Fig. 7. The coordinates of the vertices of these polygons are listed in Chin et al. [2]. Over each polygon, polynomials ranging from degree $p = 0$ to $p = 5$ are integrated. These polynomials are listed in Table 2. Exact values of the integrals are computed using region integration with `Integrate[]` in Mathematica 12.0.0.

This study is performed for two cases: first with \mathbf{x}_0 located at the vertex average of the polygon and secondly with $\mathbf{x}_0 = \mathbf{0}$. While selecting different locations for \mathbf{x}_0 can affect the numerical accuracy of the integral due to cancellation errors [18], integration with the SB parametrization is capable of computing integrals with an arbitrary location for \mathbf{x}_0 . The distribution of cubature points and weights using SBC on the polygons in Fig. 7a and Fig. 7c for \mathbf{x}_0 at the origin and at the average of the polygonal vertices is depicted in Fig. 8. The cubature points on the polygons correspond to three cubature points in the ξ direction and two cubature points in the t direction per triangle. When \mathbf{x}_0 is chosen as

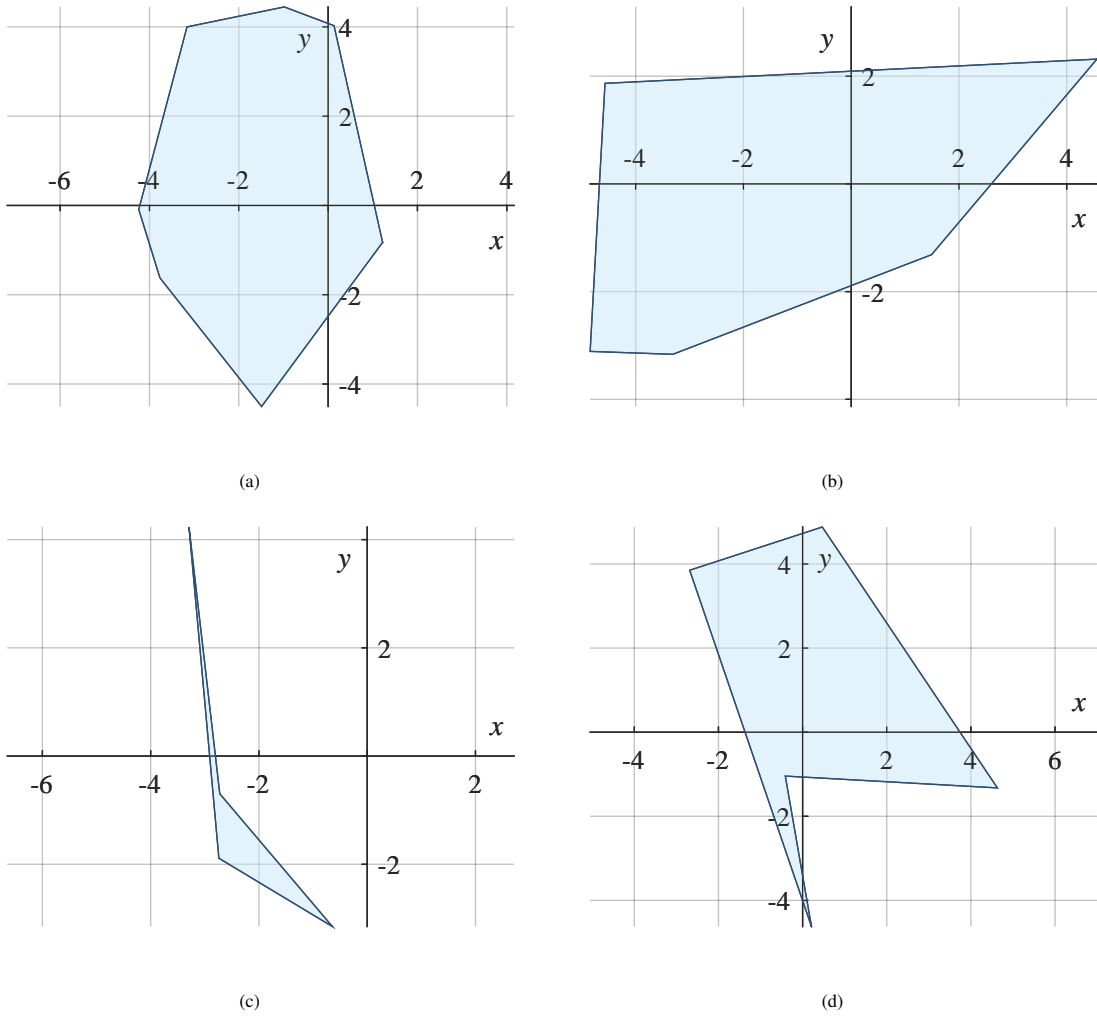


Figure 7: Convex and nonconvex polygons over which polynomials in Table 2 are integrated.

the mean of the vertex coordinates, cubature points are distributed more evenly across the convex polygon in Fig. 7a. For the nonconvex polygon in Fig. 7c, shifting \mathbf{x}_0 to the average of the vertex coordinates results in cubature points that are closer to the domain of integration and the range of cubature weights is reduced. Integration error with the minimal number of Gauss-Legendre points required for exact integration is presented in Table 3. The column “Relative integration error (shifted \mathbf{x}_0)” refers to the case where \mathbf{x}_0 is the average of the polygonal vertex coordinates while the column “Relative integration error ($\mathbf{x}_0 = \mathbf{0}$)” is for the case when \mathbf{x}_0 is the origin. As Table 3 reveals, the majority of integrals are computed with error that is close to machine precision (about $O(10^{-15})$), regardless of the location of \mathbf{x}_0 . For some integrals, error on the order of $O(10^{-14})$ is observed. Increased error in these integrals is due to cancellations caused by subtraction of numbers with similar magnitude. These results confirm the predicted accuracy of integrating polynomials over polygons discussed in Section 3.2.

5.1.2. Non-polynomial integration

Franke [39] used several non-polynomial functions to compare methods of interpolation for scattered data. In this section, we utilize these functions to validate the SBC method over polygons with non-polynomial integrands. The

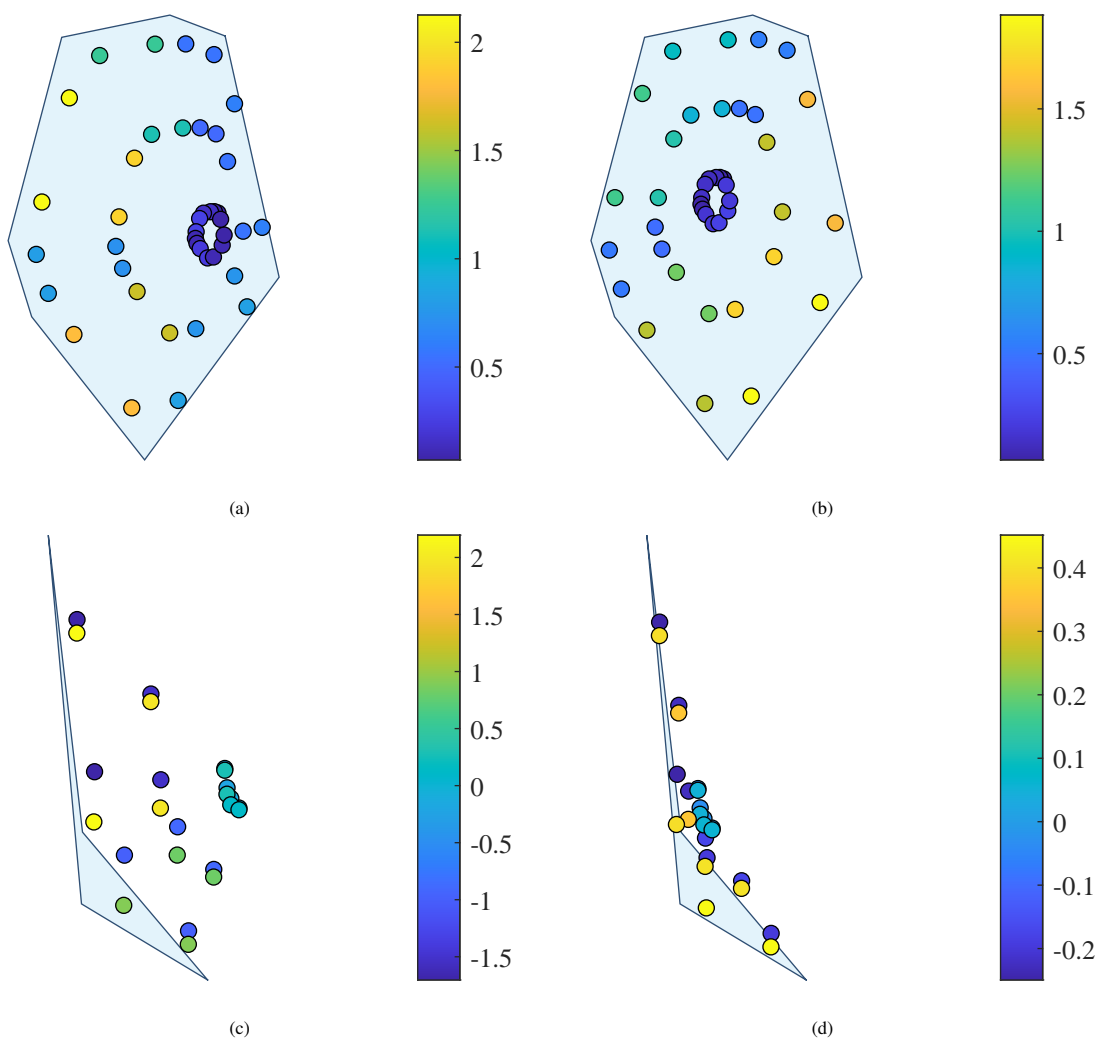


Figure 8: Distribution of cubature points (closed circles) over polygons in Fig. 7a in (a) and (b) and in Fig. 7c in (c) and (d). The point \mathbf{x}_0 is the origin in (a) and (c) and \mathbf{x}_0 is the average of the vertices in (b) and (d). The shading of each point (and the scale on the right of each subfigure) corresponds to the cubature weight.

functions we utilize are:

$$f_{F1}(\mathbf{x}) = \frac{3}{4} \exp\left(-\frac{(9x-2)^2 + (9y-2)^2}{4}\right) + \frac{3}{4} \exp\left(-\frac{(9x+1)^2}{49} - \frac{9y+1}{10}\right) + \frac{1}{2} \exp\left(-\frac{(9x-7)^2 + (9y-3)^2}{4}\right) + \frac{1}{5} \exp(-(9x-4)^2 - (9y-7)^2), \quad (51a)$$

$$f_{F2}(\mathbf{x}) = \frac{1}{9} [\tanh(9y - 9x) + 1], \quad (51b)$$

$$f_{F3}(\mathbf{x}) = \frac{\frac{5}{4} + \cos\left(\frac{27}{5}y\right)}{6[1 + (3x-1)^2]}. \quad (51c)$$

We integrate these functions over the polygons in Fig. 7. In Franke [39], the functions are designed to exhibit interesting features in the unit square, $[0, 1]^2$. However, the polygons in Fig. 7 are not contained in this domain. Accordingly, we apply an affine translation and scaling of each polygon such that the domain is the largest possible subset of the unit square. Contour maps of the three integrands are illustrated in Fig. 9 with the scaled polygons overlaid. Integration error versus the number of cubature points is revealed in Fig. 10. For all functions and polygons, integration error less

Table 3: Relative integration error for integrating polynomials over convex and nonconvex polygons using SBC.

p	ξ -points	t -points	Polygon in Fig. 7	Relative integration error (shifted \mathbf{x}_0)	Relative integration error ($\mathbf{x}_0 = \mathbf{0}$)
0	1	1	(a)	2.2×10^{-16}	0
			(b)	2.2×10^{-16}	0
			(c)	1.2×10^{-15}	2.4×10^{-15}
			(d)	0	1.3×10^{-16}
1	2	1	(a)	0	1.7×10^{-16}
			(b)	6.4×10^{-14}	4.1×10^{-15}
			(c)	1.5×10^{-15}	8.5×10^{-15}
			(d)	6.4×10^{-16}	1.3×10^{-16}
2	2	2	(a)	2.8×10^{-14}	1.8×10^{-14}
			(b)	0	0
			(c)	7.1×10^{-16}	1.4×10^{-16}
			(d)	5.2×10^{-16}	1.0×10^{-15}
3	3	2	(a)	2.8×10^{-16}	1.4×10^{-16}
			(b)	3.5×10^{-16}	8.8×10^{-16}
			(c)	1.1×10^{-15}	2.8×10^{-15}
			(d)	2.2×10^{-16}	4.4×10^{-16}
4	3	3	(a)	4.7×10^{-15}	1.6×10^{-15}
			(b)	0	1.4×10^{-16}
			(c)	2.8×10^{-16}	5.8×10^{-15}
			(d)	4.0×10^{-16}	0
5	4	3	(a)	0	2.7×10^{-16}
			(b)	2.1×10^{-16}	8.3×10^{-16}
			(c)	1.6×10^{-15}	1.4×10^{-15}
			(d)	6.3×10^{-16}	1.3×10^{-15}

than $\mathcal{O}(10^{-15})$ is obtained with fewer than 10,000 cubature points. For the results in Fig. 10, \mathbf{x}_0 is the average of the vertex coordinates of the polygons.

For the purpose of comparison, we repeat the above tests using the G-G method of Sommariva and Vianello [14]. These results are also presented in Fig. 10. Integration accuracy with the G-G scheme is on par with the SBC method. However, the SBC scheme enables cubature points to be placed entirely inside the domain of integration on more polygons as compared to G-G integration, and furthermore, by placing \mathbf{x}_0 at a vertex, the number of SBC integration points is reduced without compromising the polynomial precision of the cubature rule. To explore these effects, the same study in Fig. 10 is repeated with \mathbf{x}_0 selected with respect to star-convexity and at a vertex of the polygon. The vertex selected as \mathbf{x}_0 for each polygon is marked with an open circle in Fig. 9. The results of this exercise are presented in Fig. 11. Over the two convex polygons tested, the results are mixed. Since the integrands are non-polynomial, the accuracy of the rule is strongly dependent on the location of the cubature points. For the polygon in Fig. 7a, placing \mathbf{x}_0 at a vertex degrades the performance of the SBC rule, whereas for the polygon in Fig. 7b, integration error is reduced. We note the accuracy of the SBC rule is dependent on the vertex selected as \mathbf{x}_0 . For both nonconvex polygons, placing \mathbf{x}_0 at a vertex selected with respect to the star-convexity of the domain results in improved cubature convergence. For

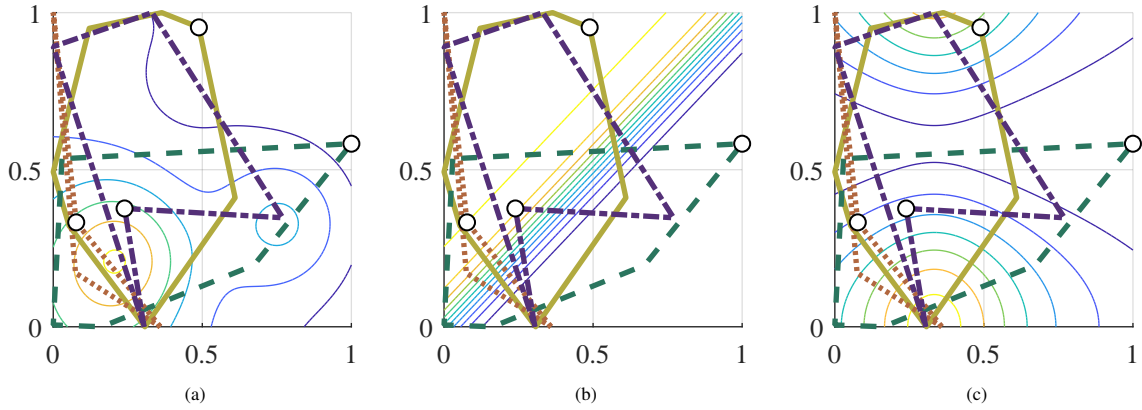


Figure 9: Contour maps of the three non-polynomial functions in (51) that are integrated over the overlaid polygonal domains. (a) $f_{F1}(x)$, (b) $f_{F2}(x)$, and (c) $f_{F3}(x)$.

the quadrilateral domain in Fig. 7c, the improvement is most pronounced. For this domain, choosing x_0 to be located at a vertex reduces the number of cubature points by a factor of one-half without reducing the polynomial precision of the rule. We point out that for both nonconvex polygons used, it is not possible to recover integration points that are entirely within the domain using the G-G scheme.

5.1.3. Timing to generate a cubature rule

The time required to integrate a function to a desired accuracy given a cubature rule is mainly dependent on two factors.

1. The overhead time required to generate the integration points and weights. This includes computing domain-specific quantities, such as triangulations, mappings, and distances, and mapping a base integration rule (over an interval or over a triangle) of a given order to the domain of integration.
2. The accuracy of the cubature method in integrating the given function. Less accurate methods require more cubature points to compensate, which can increase the time to generate the rule and require more function evaluations.

In this example, we explore the first factor (overhead time) with three benchmark cubature methods for polygons: SBC, G-G integration, and a symmetric Dunavant [49] rule applied to a constrained Delaunay triangulation (CDT) of the integration domain. For a given cubature rule, the computational costs are proportional to the number of cubature points. More cubature points require more function evaluations. Timing to perform a function evaluation is strongly dependent on the function, so we exclude this aspect from our study in this section. The other examples in this paper explore the economy of the SBC method in terms of accuracy per cubature point for various types of functions.

To ensure consistent timing computations, all examples are repeated 100 times, and the times reported here are the average of the 100 executions. All of the code is run in MATLAB 9.8.0 on a Linux computer with an Intel Xeon E5-2695 v4 CPU and 128 GB RAM. A MATLAB implementation of Gauss-Green cubature is provided by the `Polygauss` code from Sommariva.¹ The CDT is generated from the `deLaunayTriangulation` class in MATLAB. Implementations of the SBC method and the Dunavant rule are by the authors. The code for all methods (including `Polygauss`) is optimized through array preallocation and by eliminating hot spots using the MATLAB Profiler, where possible.

To examine the effects of overhead time on polygons of varying complexity, we compute the time to generate a rule versus the number of cubature points on polygons with 10 sides, 100 sides, and 1000 sides using the three benchmark methods. The polygons are cyclic polygons generated by joining evenly-spaced points on a circle. For the G-G and SBC methods, cubature rules with different numbers of points per edge are generated by increasing the number of Gauss points in each direction of the tensor-product rule. For the CDT, the number of cubature points per edge is increased using a higher order Dunavant rule. Fig. 12 presents the numerical results of this study. The SBC rule is the

¹Accessed from https://www.math.unipd.it/~alvise/SOFTWARE/GAUSSCUB_2013.zip.

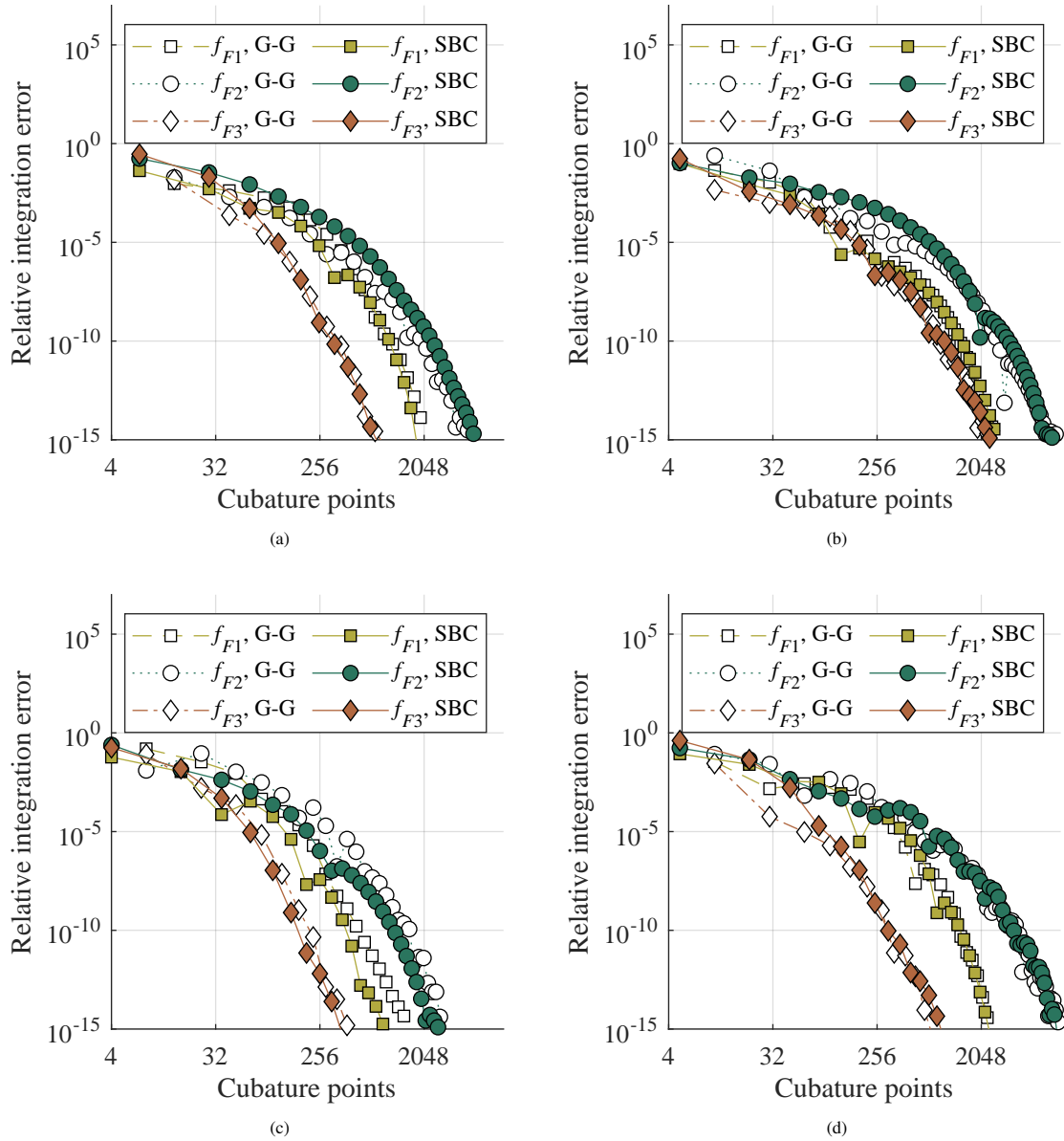


Figure 10: Relative integration error versus number of cubature points for the polygons in Fig. 7 mapped to the unit square. Integration is performed using the SBC scheme with \mathbf{x}_0 at the vertex average (filled markers) and with Gauss-Green (G-G) cubature (open markers). Polygon shape in (a) Fig. 7a, (b) Fig. 7b, (c) Fig. 7c, and (d) Fig. 7d.

fastest. Rules of order 1×1 to 20×20 per edge are generated in less than 1.7×10^{-5} seconds. This is approximately half the time required to generate the G-G rule and about ten percent of the time required to generate the CDT/Dunavant rule. For all three methods of integration, the time required to generate a rule per edge is approximately constant and increasing the number of cubature points per edge only slightly affects the time required to generate the integration rule. These findings suggest the majority of the wall-clock time is spent on polygon-specific tasks, such as generating a triangulation, mapping domains, and computing distances. Finally, we remark the CDT generation process is eased by the simplicity of the polygonal domain in this example. For complicated polygonal domains (e.g. nonconvex or multiply-connected domains), the time to generate the CDT will increase.

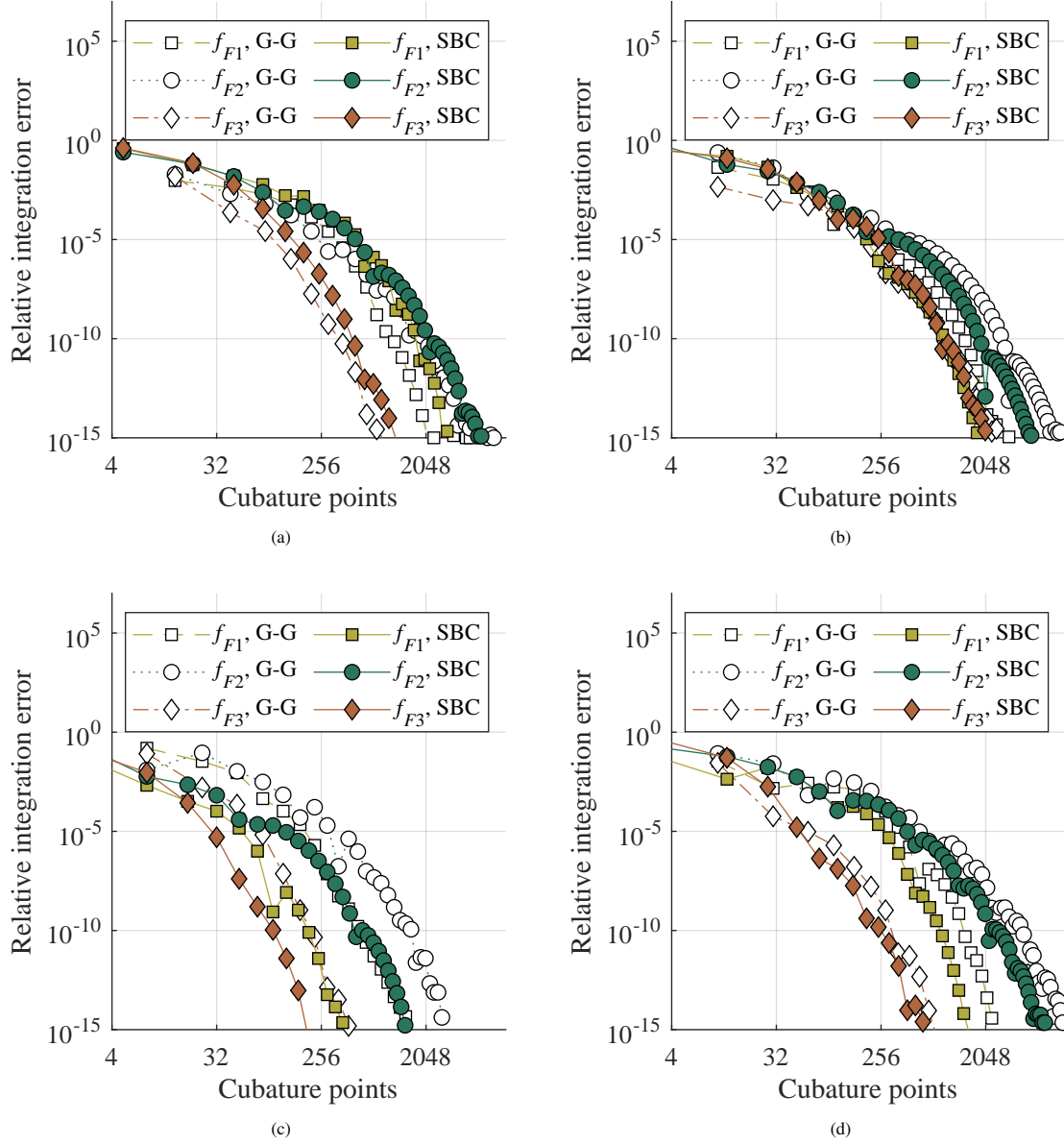


Figure 11: Relative integration error versus number of cubature points for the polygons in Fig. 7 mapped to the unit square. Integration is performed using SBC with x_0 at a vertex (filled markers) and with Gauss-Green (G-G) cubature (open markers). Polygon shape in (a) Fig. 7a, (b) Fig. 7b, (c) Fig. 7c, and (d) Fig. 7d.

5.1.4. Wachspress coordinates

One potential application of the SBC scheme is in generalized barycentric coordinates (GBCs) [23], which extends the application of barycentric coordinates to polytopes. GBCs have been successfully utilized to solve boundary-value problems under the umbrella of polytopal finite element methods. One well-known GBC is the Wachspress coordinates (basis functions) [50], which uses ideas from projective geometry to form a rational finite element basis over convex polygonal domains. Simplified equations to compute Wachspress basis functions are available in Floater et al. [51] and they are utilized here. In this example, a random polygonal mesh is formed over a square domain through the Voronoi tessellation of a random point sampling generated using a technique called maximal Poisson-disk sampling. The mesh used is illustrated in Fig. 13a. Then, the product of derivatives of Wachspress basis functions is integrated over the polygons using (i) SBC, (ii) triangulating from the vertex average and using a triangle rule from

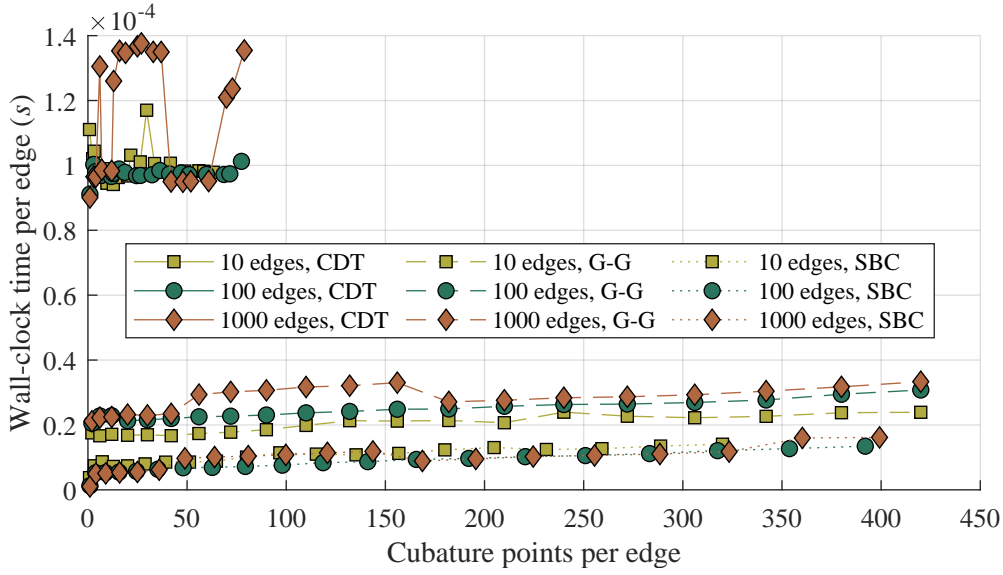


Figure 12: Time per edge to generate a cubature rule over polygons inscribed on a circle for the three benchmark methods.

Xiao and Gimbutas [52], and (iii) G-G cubature. The purpose of this example is to demonstrate the utility of SBC in a realistic scenario. The integrand is selected to mimic a typical finite element stiffness matrix computation. For SBC cubature, \mathbf{x}_0 is selected as the average of the vertex coordinates of the polygon. Similarly, triangulation is performed by connecting the edges of the polygon with the average of the vertex coordinates of each polygon.

To investigate convergence properties of the three cubature methods, we integrate over each polygon in the mesh and compute the relative error of integration as compared to a reference solution. Cubature rules of order 1 to 40 are used with each rule. The reference solution is computed using G-G cubature with 50 points in each direction per edge. Average integration accuracy versus number of cubature points is plotted in Fig. 13. Separate plots are presented for quadrilateral elements (Fig. 13b), pentagonal elements (Fig. 13c), and hexagonal elements (Fig. 13d). In these figures, the line denotes the average integration error over all of the basis functions integrated. The shaded regions give the range of integration error.

The results of Fig. 13 indicate integration using a triangle integration rule is, on average, slightly more accurate than using either SBC or G-G when integrating with the same number of integration points. This is expected since the Xiao-Gimbutas triangle rule is a symmetric rule devised specifically for triangles while both SBC and G-G cubature are tensor-product rules. When comparing G-G integration to SBC, both methods exhibit similar error per cubature point. While the triangle rule is more accurate on average, the ranges of error overlap in all plots, indicating the relative error between the three methods is small. While monotonic convergence does not generally occur with non-polynomial integrands, with SBC and G-G cubature, error reduces monotonically with additional cubature points. This does not occur with the triangle integration rule. Also, G-G cubature and SBC rules can be readily adapted to integrate polynomials of arbitrary order; on the other hand, symmetric triangle rules from moment fitting are time consuming to generate.

5.2. Planar domains bounded by parametric curves

In this section, we showcase the capabilities of SBC in computing integrals of polynomial and non-polynomial functions over curved domains bounded by polynomial and non-polynomial parametric curves. The functions inte-

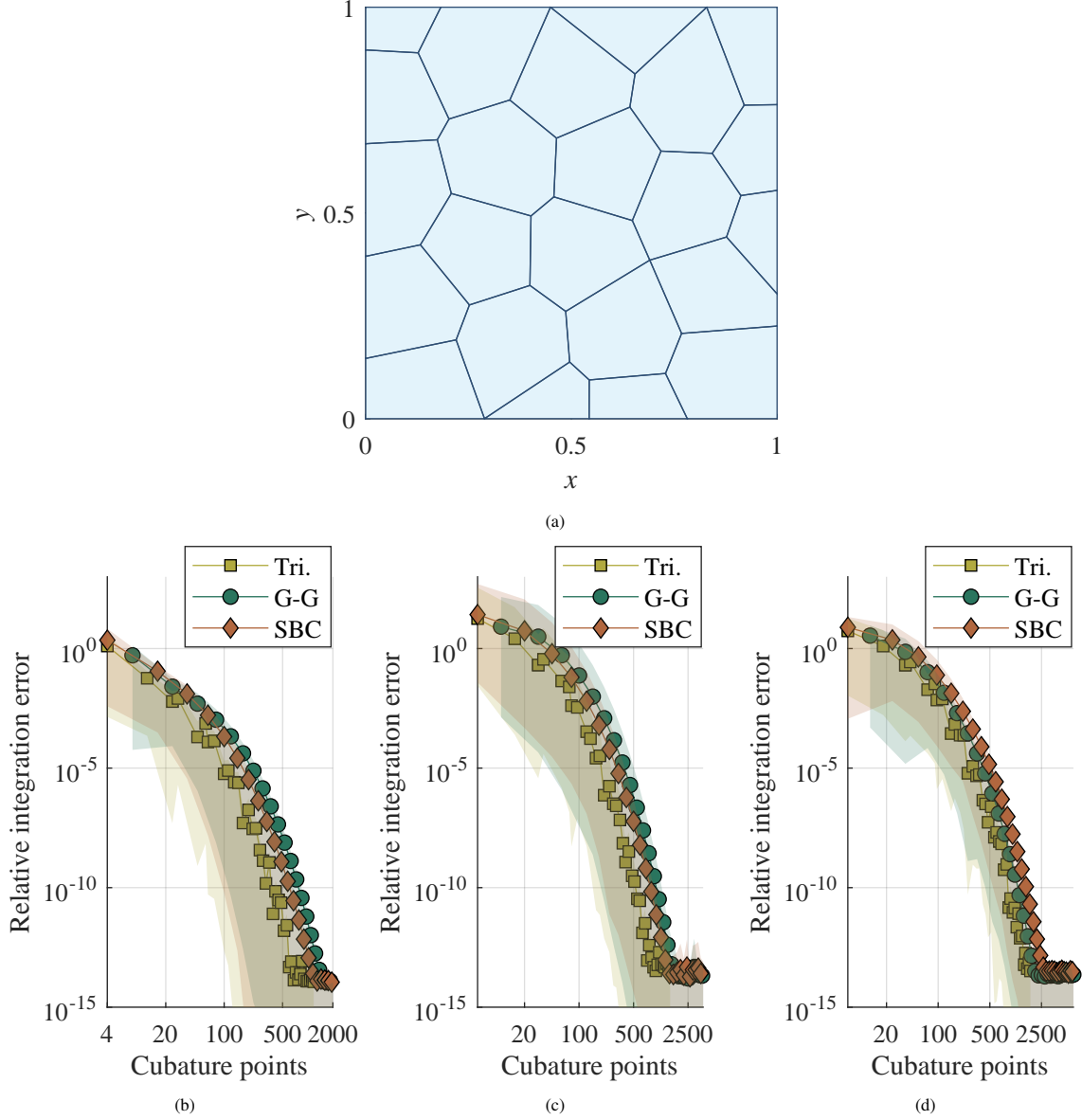


Figure 13: Integrating the products of derivatives of Wachspress basis functions using three cubature rules: triangulation (Tri.), G-G cubature, and SBC. (a) Polygonal mesh. Integration error versus cubature points over (b) quadrilateral elements, (c) pentagonal elements, and (d) hexagonal elements. The line denotes the average integration error while the shaded region bounds the minimum and maximum integration error.

grated are:

$$f_{C1}(\mathbf{x}) = 1, \quad (52a)$$

$$f_{C2}(\mathbf{x}) = 10x^5 - 5x^4y - 7x^3y^2 + 6x^2y^3 + 3xy^4 + y^5 - x^4 + 2x^3y + 11x^2y^2 - 8xy^3 - 2y^4 - 3x^3 + 9x^2y + 8xy^2 - 10y^3 - 9x^2 - 6xy + 7y^2 + 5x - 4y + 4, \quad (52b)$$

$$f_{C3}(\mathbf{x}) = \frac{3}{4} \exp\left(-\frac{(9x-2)^2 + (9y-2)^2}{4}\right) + \frac{3}{4} \exp\left(-\frac{(9x+1)^2}{49} - \frac{9y+1}{10}\right) + \frac{1}{2} \exp\left(-\frac{(9x-7)^2 + (9y-3)^2}{4}\right) + \frac{1}{5} \exp(-(9x-4)^2 - (9y-7)^2), \quad (52c)$$

$$f_{C4}(\mathbf{x}) = \exp\left(-\left[\left(\frac{x-0.4}{0.3}\right)^2 + \left(\frac{y-0.5}{0.4}\right)^2\right]^2\right) \cos_{22}^2(3x) \cos^2(8y). \quad (52d)$$

Table 4: Control points for the cubic Bézier curves that form the boundary of the domain pictured in Fig. 14a.

Curve	Point 1	Point 2	Point 3	Point 4
1	$\left(0, \frac{3}{26}\right)$	$\left(\frac{7}{26}, \frac{9}{26}\right)$	$\left(\frac{15}{26}, 0\right)$	$\left(\frac{10}{13}, \frac{3}{26}\right)$
2	$\left(\frac{10}{13}, \frac{3}{26}\right)$	$\left(\frac{23}{26}, \frac{9}{26}\right)$	$\left(\frac{15}{26}, \frac{17}{26}\right)$	$\left(\frac{10}{13}, \frac{23}{26}\right)$
3	$\left(\frac{10}{13}, \frac{23}{26}\right)$	$\left(\frac{1}{2}, \frac{25}{26}\right)$	$\left(\frac{5}{26}, 1\right)$	$\left(0, \frac{23}{26}\right)$
4	$\left(0, \frac{23}{26}\right)$	$\left(\frac{7}{26}, \frac{17}{26}\right)$	$\left(\frac{5}{26}, \frac{9}{26}\right)$	$\left(0, \frac{3}{26}\right)$

Table 5: Parametric equations for the boundary of the deltoid pictured in Fig. 14g.

i	\mathbf{c}_i
1	$\left[\frac{\left\{1 + 2 \cos\left(\frac{2\pi}{3}t\right)\right\}^2}{12} \quad \frac{1 + 2 \sin\left(\frac{2\pi}{3}t\right) - 4 \sin\left(\frac{4\pi}{3}t\right)}{6} \right]^T$
2	$\left[\frac{\left\{1 + 2 \cos\left(\frac{2\pi}{3}(t+1)\right)\right\}^2}{12} \quad \frac{1 + 2 \sin\left(\frac{2\pi}{3}(t+1)\right) - 4 \sin\left(\frac{4\pi}{3}(t+1)\right)}{6} \right]^T$
3	$\left[\frac{\left\{1 + 2 \cos\left(\frac{2\pi}{3}(t+2)\right)\right\}^2}{12} \quad \frac{1 + 2 \sin\left(\frac{2\pi}{3}(t+2)\right) - 4 \sin\left(\frac{4\pi}{3}(t+2)\right)}{6} \right]^T$

In (52), the first function is a constant (computes the area of each curved domain), the second function is a 5th degree polynomial, the third function is the Franke test function (introduced in (51a)), and the fourth function is the product of exponential and trigonometric functions. Each of these functions is integrated over three curved domains using the SBC method. The first domain (pictured in Fig. 14a) is bounded by four polynomial curves that are described using cubic Bézier functions. The control points of the Bézier curves are provided in Table 4. The second domain, from Sommariva and Vianello [15], is a lune that is illustrated in Fig. 14d. The boundary of the lune is described using four rational quadratic Bézier (non-polynomial) curves, which are capable of exactly representing conic sections. The final domain is a deltoid that is shown in Fig. 14g. While a single parametric equation can be used to describe the entire deltoid boundary, the boundary is subdivided into three curves to avoid cusps along each curve. These three trigonometric parametric equations are given in Table 5. For the three domains, the point \mathbf{x}_0 is selected as the average of the endpoints of each curve. With this selection, all cubature points are inside their respective domains and all weights are positive, despite all three shapes being nonconvex. Exact integrals are computed in Mathematica 12.0.0 using the functions `Integrate[]` and `NIntegrate[]`. Beyond applications in numerical integration, we note the SB parametrization can be used to make integration over complicated domains, such as the deltoid, tractable using traditional methods of integration.

To investigate cubature convergence in each direction of the tensor-product domain after applying the SB mapping, integration error versus the number of cubature points in each direction for the four test functions and three shapes is plotted in Fig. 14. Convergence properties in the ξ direction (resp. t direction) are interrogated by supplying a surplus of integration points in the t direction (resp. ξ direction), which ensures integration error is entirely in the ξ direction (resp. t direction). For the two non-polynomial integrands, approximately 20×20 cubature points are needed to realize error close to machine precision over the polynomial curve domain and the lune. About 40 t -points and 15 ξ -points are needed for the deltoid. In Figs. 14a, 14d, and 14g, dots are placed on the boundary at intervals of $\Delta t = 0.1$. Note the dots are not evenly spaced on the boundary of the deltoid, since the curve speed ($\|\mathbf{c}'(t)\|$) approaches zero around the

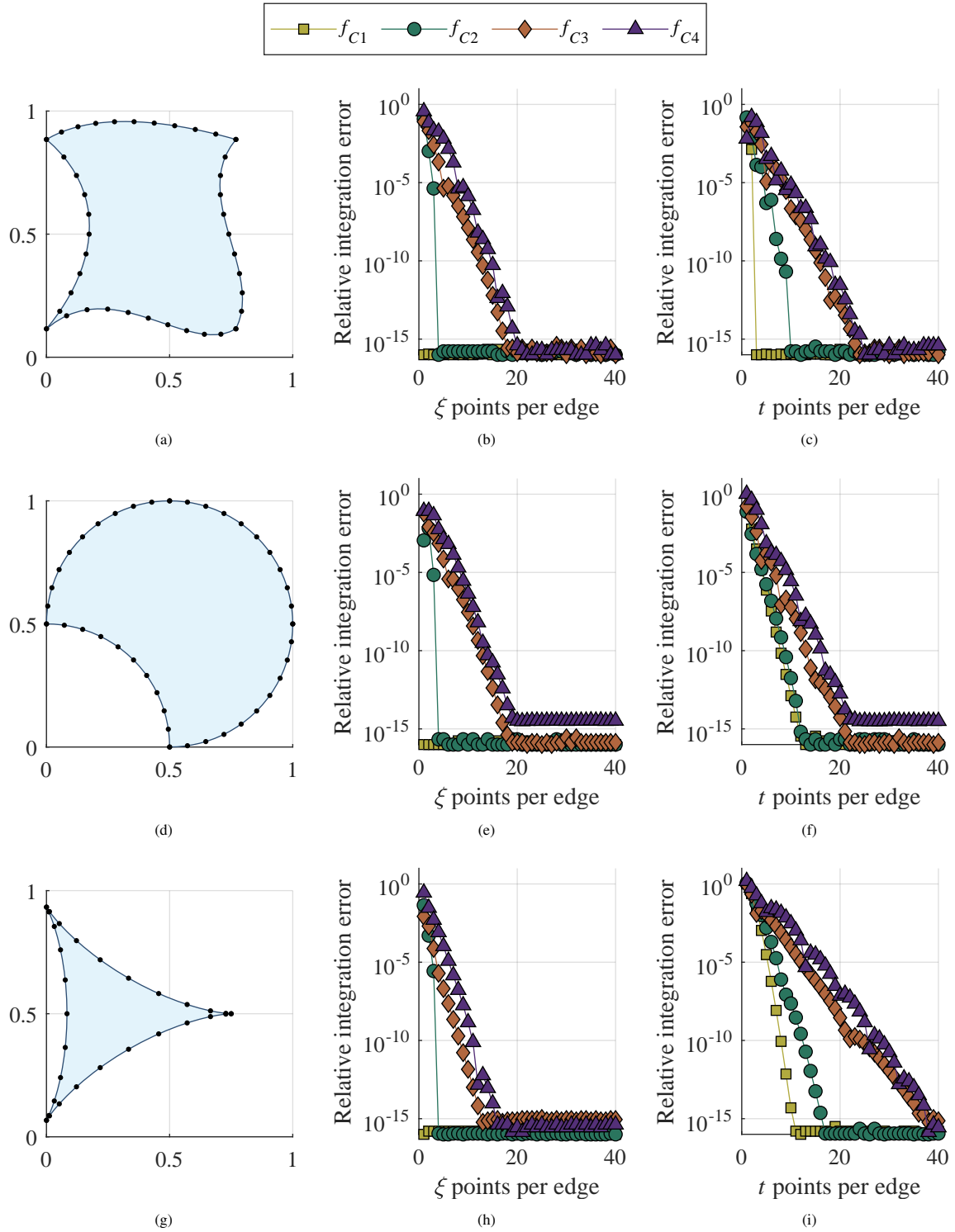


Figure 14: Number of quadrature points versus integration accuracy to integrate four functions over regions bounded by curves. (a) Domain bounded by cubic Bézier curves and integration accuracy versus number of (b) ξ quadrature points and (c) t quadrature points. (d) Lune (boundary represented by rational quadratic Bézier curves) and integration accuracy versus number of (e) ξ quadrature points and (f) t quadrature points. (g) Deltoid and integration accuracy versus number of (h) ξ quadrature points and (i) t quadrature points.

cusps. This results in extra weighting of the parameter space t toward the cusps, and ultimately, more cubature points near the cusps. The accuracy per cubature point for integrating non-polynomial functions over the deltoid would likely increase with a parametrization of the boundary with less variation in curve speed.

While integrating using the SBC method results in a tensor-product cubature rule over each curved triangle, the burden of integration is not shared equally in each direction. Besides integrating the transformed integrand, the Jacobian introduces a factor of ξ and the product of the curve $\mathbf{c}(t)$ with its derivative (hodograph). For polynomial integrands and polynomial boundary curves, the polynomial order of the integrand in the ξ - and t -directions can be exactly computed (see Section 3.1). For a degree 0 polynomial bounded by polynomial curves of degree 3, the Gauss rule must be capable of integrating a polynomial of degree 1 in the ξ direction and degree 5 in the t direction. This corresponds to 1 Gauss point in the ξ direction and 3 Gauss points in the t direction for exact integration. This matches the results presented for f_{C1} in Fig. 14b and Fig. 14c. When the polynomial integrand increases to degree 5 and the bounding curves remain degree 3, the Gauss rule must be capable of integrating a 6th degree polynomial in ξ and a polynomial of degree 20 in t . Using a Gauss Legendre rule, 4 points are needed in the ξ direction and 11 points are needed in the t direction. Fig. 14b and Fig. 14c reflect this for f_{C2} . Note the number of quadrature points in the ξ direction is not dependent on the curve $\mathbf{c}(t)$. Therefore, the number of ξ -points needed in the Gauss rule is constant for polynomial integrands. This holds for the lune and deltoid domains, which are bounded by non-polynomial curves (see Fig. 14e and Fig. 14h).

5.3. Integration of weakly singular functions

5.3.1. Comparison of ξ -transformation methods

Section 4.1 introduced the generalized SB transformation for cancelling radial singularities in the integrand. This section compares the generalized SB transformation and Gauss-Jacobi integration to the SB transformation for integrating different weakly singular functions. Gauss-Jacobi integration rules are constructed by solving for the roots of the Jacobi polynomial in Mathematica 12.0.0. Four different functions are utilized for this comparison:

$$f_{S1}(\mathbf{x}) = \frac{4 - 2x + y - x^2 + 2xy - 3y^2 + 3x^3 - 5x^2y + 5xy^2 - 4y^3}{(x^2 + y^2)^{1/4}}, \quad (53a)$$

$$f_{S2}(\mathbf{x}) = \frac{\exp\left(-\left[\left(\frac{x-0.25}{0.4}\right)^2 + \left(\frac{y-0.2}{0.7}\right)^2\right]^2\right) \cos^2(5x) \cos^2(5y)}{(x^2 + y^2)^{1/4}}, \quad (53b)$$

$$f_{S3}(\mathbf{x}) = \frac{4 - 2x + y - x^2 + 2xy - 3y^2 + 3x^3 - 5x^2y + 5xy^2 - 4y^3}{(x^2 + y^2)^{9/10}}, \quad (53c)$$

$$f_{S4}(\mathbf{x}) = \frac{\exp\left(-\left[\left(\frac{x-0.25}{0.4}\right)^2 + \left(\frac{y-0.2}{0.7}\right)^2\right]^2\right) \cos^2(5x) \cos^2(5y)}{(x^2 + y^2)^{9/10}}. \quad (53d)$$

The first two functions ($f_{S1}(\mathbf{x})$ and $f_{S2}(\mathbf{x})$) contain $r^{-1/2}$ singularities and the last two functions ($f_{S3}(\mathbf{x})$ and $f_{S4}(\mathbf{x})$) contain $r^{9/5}$ in the denominator. Two of the functions ($f_{S1}(\mathbf{x})$ and $f_{S3}(\mathbf{x})$) contain the same cubic polynomial numerator and the remaining two functions ($f_{S2}(\mathbf{x})$ and $f_{S4}(\mathbf{x})$) contain the same exponential numerator. We note all of these integrands are weakly singular, and contain a fractional power for the singularity. For r^{-1} singularities, the SB transformation is capable of accurate integration in the ξ direction without further manipulation.

Each of these functions is integrated over two triangles (T_2 and T_3) and one curved triangle (T_4) whose boundary is defined by a cubic Bézier curve. The vertex locations of T_2 and T_3 are provided in Table 6. T_2 represents a triangle with a high base to height ratio and T_3 is a triangle where the base to height ratio is close to unity. The control points for the Bézier curve in T_4 are listed in Table 7. The point $\mathbf{x}_c = \mathbf{0}$ and the endpoints of the curve define the two line segments that bound T_4 . These three shapes are illustrated in Fig. 15 and Fig. 16. To analyze the effect of the integration techniques in the ξ direction, a surplus of cubature points is provided in the t direction ensuring integration error above machine precision is only present in the ξ direction. Exact integrals are computed using Mathematica 12.0.0.

Results for the two functions with the $r^{-1/2}$ singularity are presented in Fig. 15. For both the function with polynomial numerator ($f_{S1}(\mathbf{x})$) and the one with non-polynomial numerator ($f_{S2}(\mathbf{x})$), a Gauss-Jacobi rule with $\alpha = 0$ and $\beta = 1/2$ provides the most accurate integration per quadrature point. To integrate $f_{S1}(\mathbf{x})$ to about 15 digits of precision,

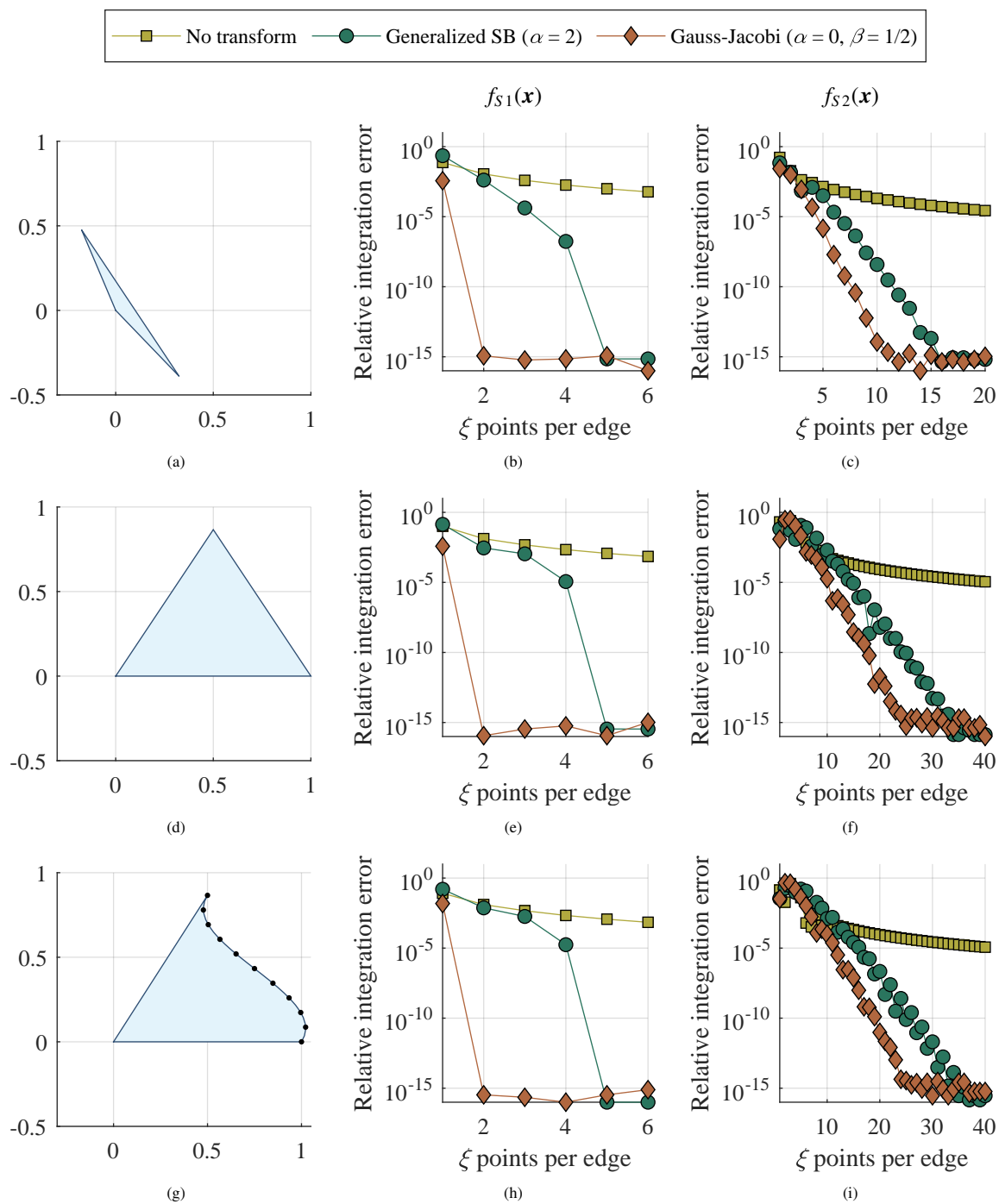


Figure 15: Number of ξ quadrature points versus integration accuracy to integrate $f_{S1}(x)$ over (b) T_2 , (e) T_3 , and (h) T_4 and to integrate $f_{S2}(x)$ over (c) T_2 , (f) T_3 , and (i) T_4 . The domain of (a) T_2 , (d) T_3 , and (g) T_4 .

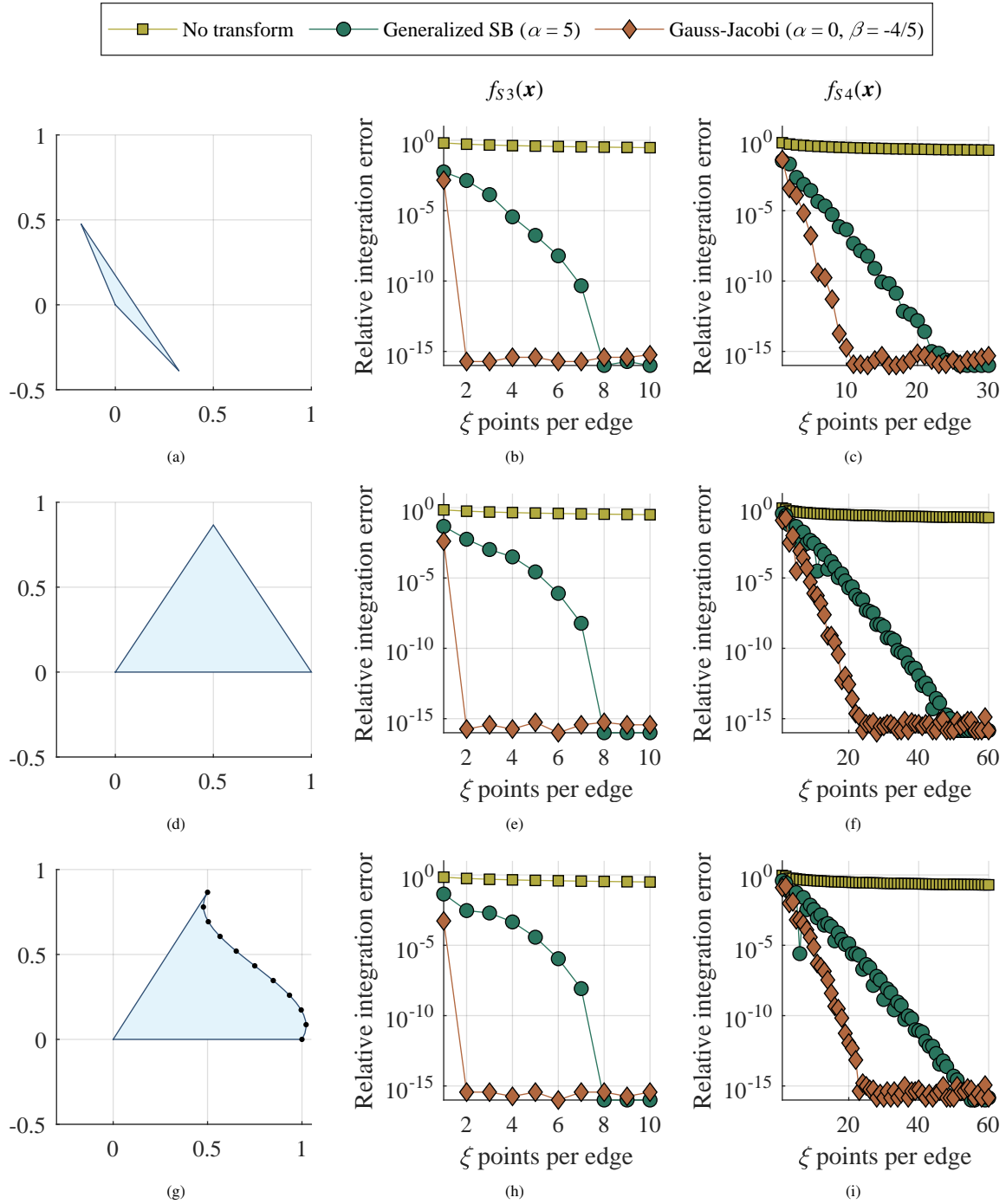


Figure 16: Number of ξ quadrature points versus integration accuracy to integrate $f_{S3}(x)$ over (b) T_2 , (e) T_3 , and (h) T_4 and to integrate $f_{S4}(x)$ over (c) T_2 , (f) T_3 , and (i) T_4 . The domain of (a) T_2 , (d) T_3 , and (g) T_4 .

Table 6: Vertices of the triangles over which weakly singular integrals are computed.

Triangle	Vertex 1	Vertex 2	Vertex 3
T_1	(0, 0)	$\left(\frac{103}{400}, -\frac{99\sqrt{3}}{400}\right)$	$\left(-\frac{97}{400}, \frac{101\sqrt{3}}{400}\right)$
T_2	(0, 0)	$\left(\frac{13}{40}, -\frac{9\sqrt{3}}{40}\right)$	$\left(-\frac{7}{40}, \frac{11\sqrt{3}}{40}\right)$
T_3	(0, 0)	(1, 0)	$\left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$

Table 7: Control points of the curved triangle over which weakly singular integrals are computed.

Curved Triangle	Point 1	Point 2	Point 3	Point 4
T_4	(1, 0)	$\left(\frac{7}{6}, \frac{\sqrt{3}}{6}\right)$	$\left(\frac{1}{3}, \frac{\sqrt{3}}{3}\right)$	$\left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$

two quadrature points are needed in the ξ direction over all three shapes. For $\mathcal{O}(10^{-15})$ accurate integration of $f_{S2}(\mathbf{x})$, 12 quadrature points are needed in the ξ direction for T_2 and approximately 25 quadrature points are needed for T_3 and T_4 . The generalized SB transformation with $\alpha = 2$ also enables accurate cubature of $f_{S1}(\mathbf{x})$ and $f_{S2}(\mathbf{x})$, though not as efficiently as the Gauss-Jacobi rule. Machine precision integration (about 15 digits of precision) of $f_{S1}(\mathbf{x})$ over the three domains requires 5 points in the ξ direction. For machine precision integration of the non-polynomial integrand ($f_{S2}(\mathbf{x})$), 16 quadrature points are needed over T_2 and about 35 quadrature points are needed over T_3 and T_4 . When SBC is used with no transformation in the ξ direction, no more than six digits of precision can be obtained with 64 ξ -quadrature points for either $f_{S1}(\mathbf{x})$ or $f_{S2}(\mathbf{x})$ over any of the three domains.

Integration error versus number of quadrature points in the ξ direction for integrating $f_{S3}(\mathbf{x})$ and $f_{S4}(\mathbf{x})$ are illustrated in Fig. 16. Qualitatively, the results of integrating the two functions with a $r^{-9/5}$ singularity mirror those for the $r^{-1/2}$ singularity. For the Gauss-Jacobi rule with $\alpha = 0$ and $\beta = -4/5$, accuracy per quadrature point is almost identical to the results presented in Fig. 15, though slightly fewer quadrature points are needed to integrate $f_{S4}(\mathbf{x})$ to machine precision as compared to $f_{S2}(\mathbf{x})$. For these integrands, the generalized SB transformation with $\alpha = 5$ is less efficient than the $\alpha = 2$ transformation used to integrate $f_{S1}(\mathbf{x})$ and $f_{S2}(\mathbf{x})$. Eight quadrature points are needed in the ξ direction to integrate $f_{S3}(\mathbf{x})$ to machine precision and 24 (resp. 50) quadrature points are needed to integrate $f_{S4}(\mathbf{x})$ to 15 digits of accuracy on T_2 (resp. on T_3 and T_4).

5.3.2. Comparison of t -transformation methods

This section compares the three integral transformations introduced in Section 4.2 that are designed to smooth near-singularities in the t direction. Functions with $r^{-1/2}$, r^{-1} , and $r^{-9/5}$ singularities are integrated. The functions with the $r^{-1/2}$ and $r^{-9/5}$ singularities are given in (53) and the functions with the r^{-1} singularity are

$$f_{S5}(\mathbf{x}) = \frac{2x^2 + 2y(y + \sqrt{x^2 + y^2}) + x(y + 2\sqrt{x^2 + y^2})}{(x^2 + y^2)^{3/2}}, \quad (54a)$$

$$f_{S6}(\mathbf{x}) = \frac{\exp\left(-\left[\left(\frac{x-0.25}{0.4}\right)^2 + \left(\frac{y-0.2}{0.7}\right)^2\right]\right) \cos^2(5x) \cos^2(5y)}{\sqrt{x^2 + y^2}}. \quad (54b)$$

Equation (54a) is the sum of the terms that appear in the kernel for linear elasticity and elasto-plasticity in the BEM [53]. Equation (54b) is analogous to (53b) and (53d), though the singularity is of the form r^{-1} . The accuracy of each integral transformation is tested by integrating in the ξ direction to $\mathcal{O}(10^{-15})$ accuracy, then increasing the number of cubature points in the t direction. The exact integral values are computed using Mathematica 12.0.0.

These six functions are integrated over three different triangles: T_1 , T_2 , and T_3 , which correspond to triangles with very high, high, and near unitary base to height ratios, respectively. The vertices of these triangles are listed in Table 6. No triangles bounded by non-affine curves are included since the integral transformations in the t direction are formulated for domains bounded by line segments. Since polygons are decomposed into triangles by the SBC method, the results in this section (integration over triangles) can be generalized to polygonal domains of integration.

Integration accuracy versus the number of t -direction points is plotted in Figs. 17, 18, and 19 for functions containing $r^{-1/2}$, r^{-1} , and $r^{-9/5}$ singularities, respectively. Only the r^{-1} -cancelling transformation is able to consistently provide about 15 digits of precision in integration with fewer than 60 quadrature points in the t direction. As the strength of the singularity increases, the performance of the r^{-2} -cancelling transformation improves. This mirrors results presented in Chin [48], where the r^{-2} -cancelling transformation outperforms the r^{-1} -cancelling transformation as the power of the singularity becomes more negative. For integration of the weakly singular integrands presented in this section, the lowest accuracy per quadrature point is consistently obtained using either no transformation or the r^{-3} -cancelling transformation. Stronger singularities are likely needed for the r^{-3} transformation to be competitive versus the r^{-1} and r^{-2} transformations.

As the base to height ratio of the triangular integration domain approaches unity, the efficacy of the r^{-1} - and r^{-2} -cancelling transformations wanes. When integrating the non-polynomial integrands ($f_{S2}(\mathbf{x})$, $f_{S6}(\mathbf{x})$, and $f_{S4}(\mathbf{x})$) over T_3 , the benefit of the transformation is minimal—as compared to SBC with no t -transformation, only two fewer quadrature points are needed to integrate the non-polynomial integrands to 15 digits of precision when the r^{-1} -cancelling transformation is applied. As the base to height ratio of the triangular integration domain increases, the near-singularity in the integrand is less pronounced, and consequently, the number of quadrature points needed for accurate integration in the t direction decreases. This suggests the number of quadrature points in the t direction can be scaled based on the base to height ratio of a triangle for a more efficient cubature rule.

5.3.3. Integrals in the extended finite element method

Modeling cracks in linear elastic continua using the X-FEM requires integration of weakly singular functions. In the X-FEM, one component of the stiffness matrix calculations is of the form:

$$K_{IJ} = \int_{\Omega} \left[\frac{\partial(F(\mathbf{x})N_I(\boldsymbol{\xi}(\mathbf{x})))}{\partial x} \frac{\partial N_J(\boldsymbol{\xi}(\mathbf{x}))}{\partial x} + \frac{\partial(F(\mathbf{x})N_I(\boldsymbol{\xi}(\mathbf{x})))}{\partial y} \frac{\partial N_J(\boldsymbol{\xi}(\mathbf{x}))}{\partial y} \right] dx, \quad (55)$$

where $F(\mathbf{x})$ is an enrichment function used to capture singular behavior in the vicinity of a crack tip and $N_I(\boldsymbol{\xi})$ is a finite element basis function. Typically, four crack-tip enrichment functions are used, but in this study, only the function

$$F(\mathbf{x}) = \sqrt{r} \sin\left(\frac{1}{2}\theta\right), \quad (56a)$$

where

$$r = \sqrt{x^2 + y^2} \quad \text{and} \quad \theta = \arctan \frac{y}{x} \quad \forall \theta \in [-\pi, \pi], \quad (56b)$$

is considered. Note the derivative of $F(\mathbf{x})$ is a weakly singular function with a $r^{-1/2}$ singularity, and furthermore, $F(\mathbf{x})$ contains a discontinuity at $\theta = \pm\pi$. In the X-FEM, the location of the discontinuity coincides with the location of a crack in the domain. Bilinear quadrilaterals with four nodes are used to form the finite element basis. In the isoparametric domain $\Xi = [-1, 1]^2$, these shape functions are

$$N_1(\boldsymbol{\xi}) = \frac{(1-\xi)(1-\eta)}{4}, \quad N_2(\boldsymbol{\xi}) = \frac{(1+\xi)(1-\eta)}{4}, \quad N_3(\boldsymbol{\xi}) = \frac{(1+\xi)(1+\eta)}{4}, \quad N_4(\boldsymbol{\xi}) = \frac{(1-\xi)(1+\eta)}{4}. \quad (57)$$

Mapping between an arbitrary quadrilateral and Ξ is performed using the isoparametric mapping

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{I=1}^4 N_I(\boldsymbol{\xi}) \mathbf{x}_I, \quad (58)$$

where \mathbf{x}_I is the nodal coordinate associated with $N_I(\boldsymbol{\xi})$.

For the bilinear quadrilateral element, (55) defines 16 different integrals. In this example, these 16 integrals are computed over two element domains: Ω_1 and Ω_2 pictured in Fig. 20a. In the element domains, the nodal locations are

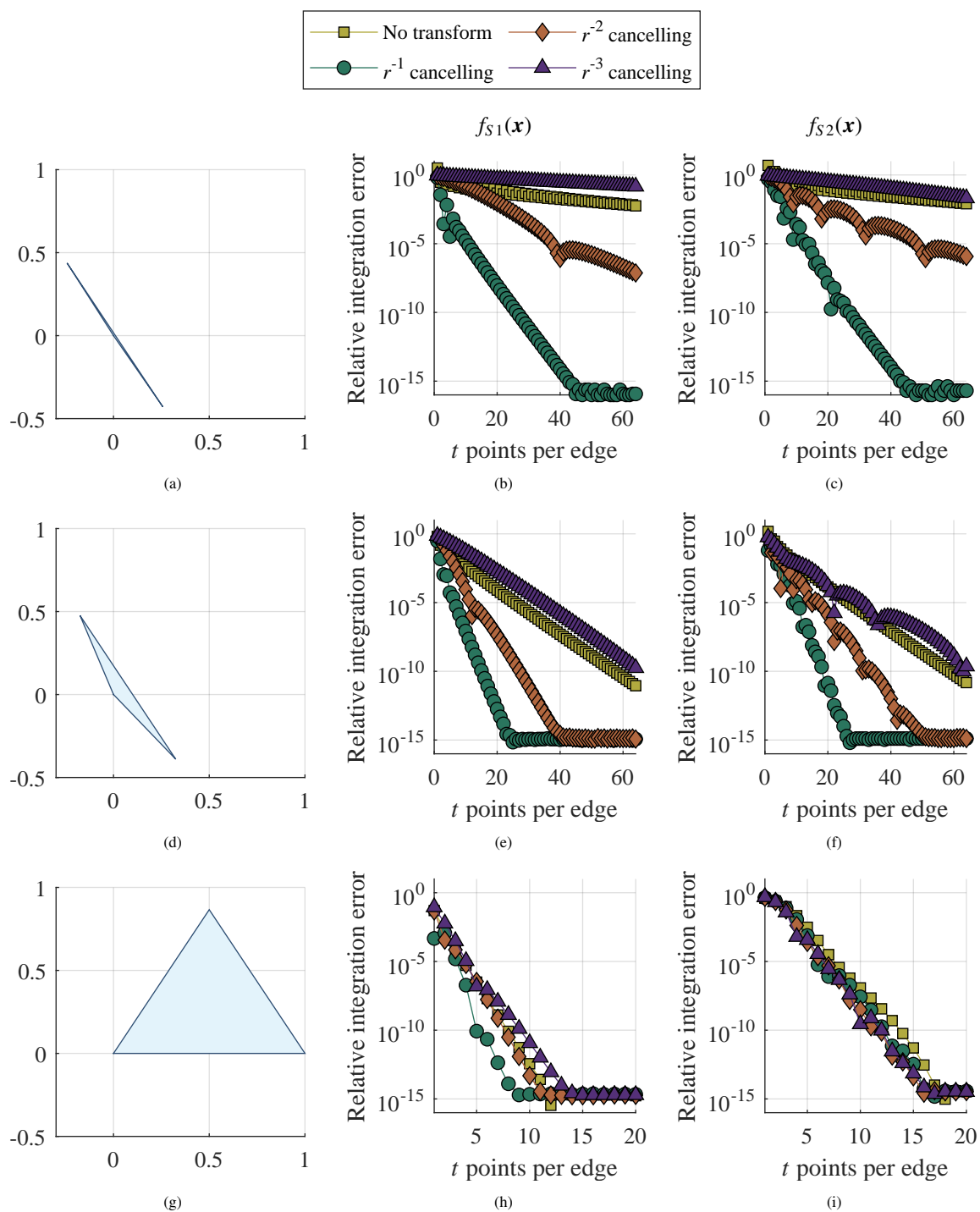


Figure 17: Number of t quadrature points versus integration accuracy to integrate $f_{S1}(x)$ over (b) T_1 , (e) T_2 , and (h) T_3 and to integrate $f_{S2}(x)$ over (c) T_1 , (f) T_2 , and (i) T_3 . The domain of (a) T_1 , (d) T_2 , and (g) T_3 .

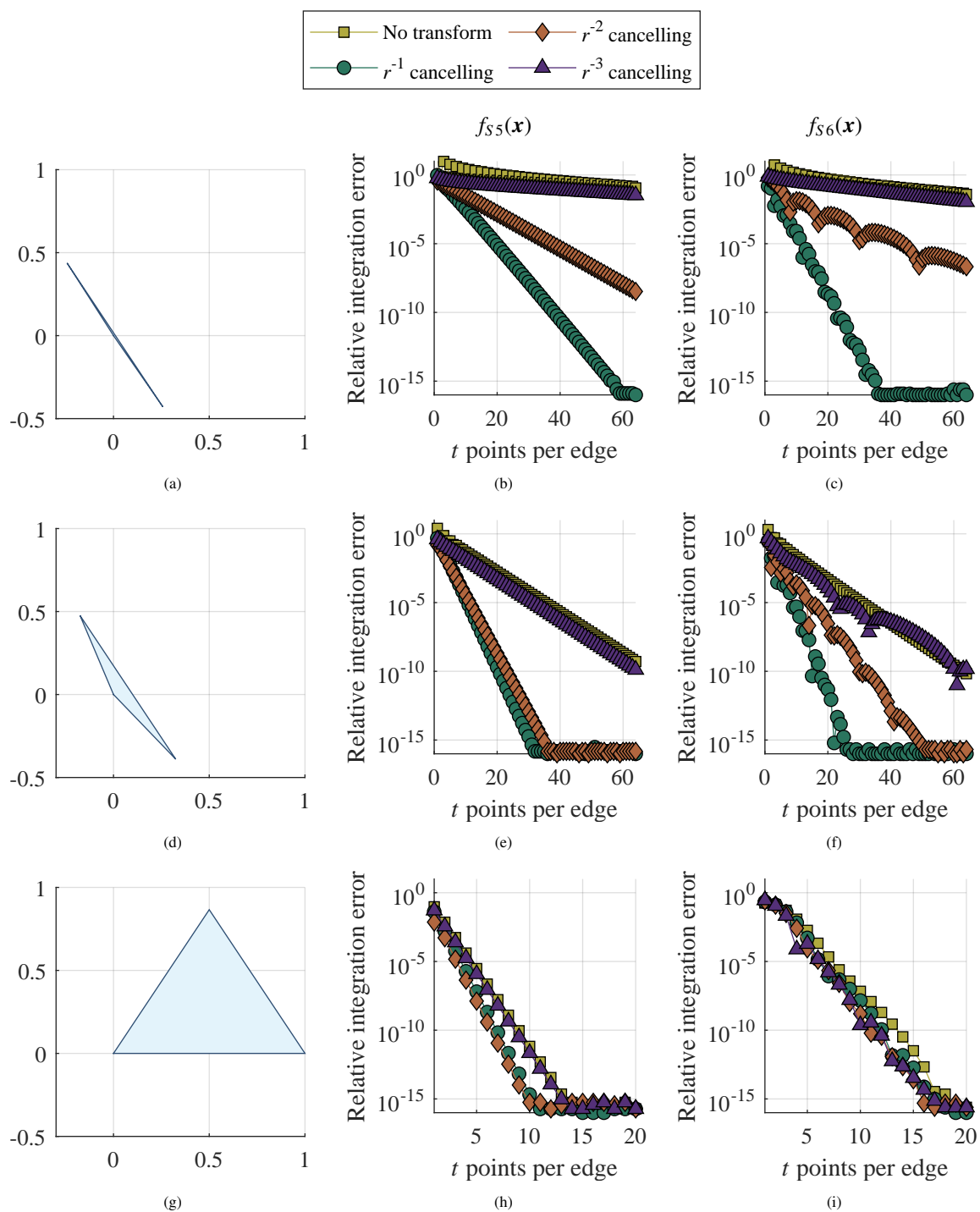


Figure 18: Number of t quadrature points versus integration accuracy to integrate $f_{S5}(x)$ over (b) T_1 , (e) T_2 , and (h) T_3 and to integrate $f_{S6}(x)$ over (c) T_1 , (f) T_2 , and (i) T_3 . The domain of (a) T_1 , (d) T_2 , and (g) T_3 .

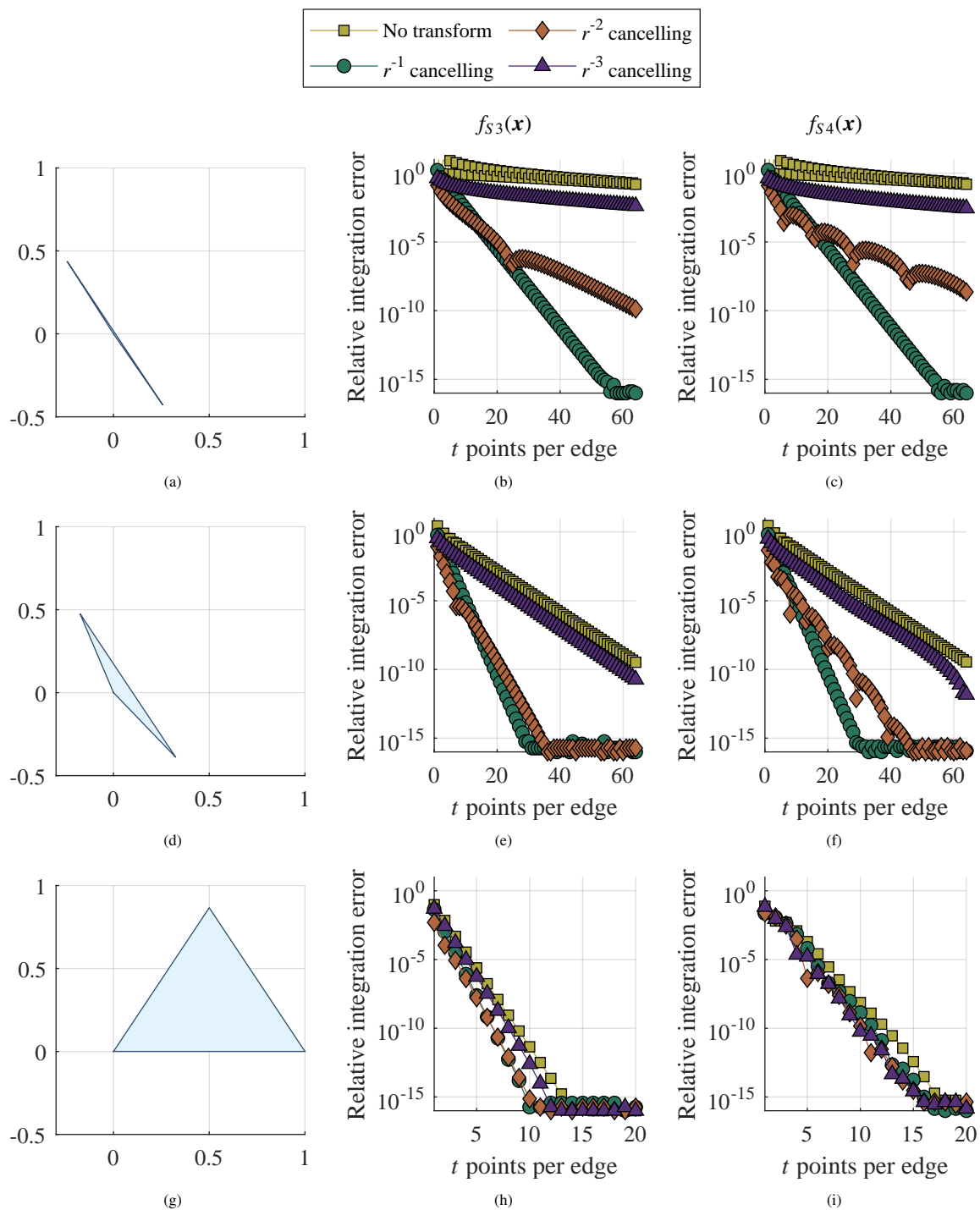


Figure 19: Number of t quadrature points versus integration accuracy to integrate $f_{S3}(x)$ over (b) T_1 , (e) T_2 , and (h) T_3 and to integrate $f_{S4}(x)$ over (c) T_1 , (f) T_2 , and (i) T_3 . The domain of (a) T_1 , (d) T_2 , and (g) T_3 .

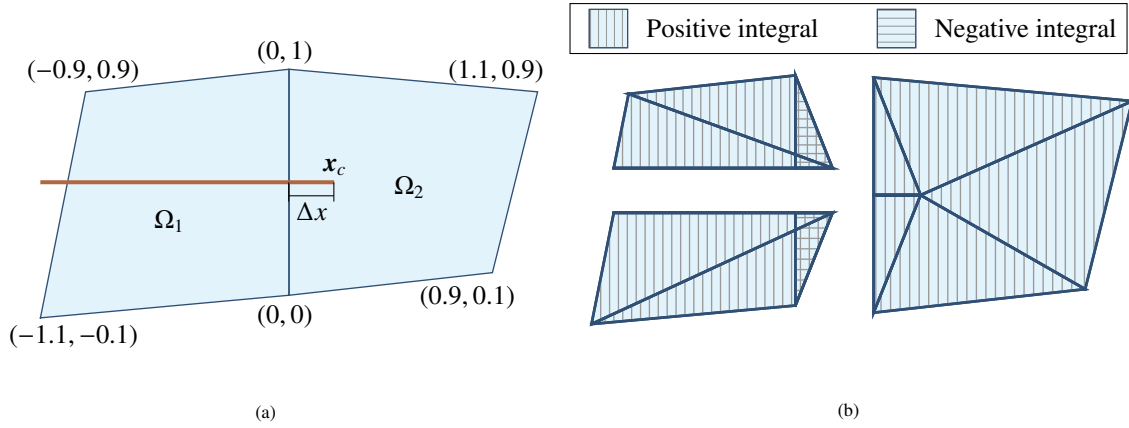


Figure 20: Problem setup for numerically integrating functions that appear in the X-FEM. (a) Problem domain with an integrand discontinuity (thick line) and (b) domain partitioning after applying SBC to regions with continuous integrands.

perturbed such that the integrands are non-polynomial functions. This example is presented to demonstrate a realistic application of singularity-cancelling methods applicable to the SBC method described in Section 4. Three different crack-tip locations are tested with varying proximity to the element interface. The distances to the interface in the three different crack-tip locations are $\Delta x = 0.001$, $\Delta x = 0.01$, and $\Delta x = 0.1$. The length Δx is defined in Fig. 20a. The relative distance of a point singularity to the boundary of the integration domain is shown to affect integration accuracy in the examples in Section 5.3. Selecting different values of Δx for this problem is designed to highlight this effect. Reference integrals for this example are computed using the SBC method with many cubature points.

To enable singularity cancelling transformations with the SBC method, the location of the crack tip ($r = 0$ in $F(\mathbf{x})$) is selected as $\mathbf{x}_0 = \mathbf{x}_c$. Since $F(\mathbf{x})$ is discontinuous, each integration domain must be split across the discontinuity to ensure smooth, continuous integrands over each triangular partition. The location of the discontinuity for this example is denoted by a thick line in Fig. 20a. Selecting the point $r = 0$ as \mathbf{x}_0 and splitting the domains across the discontinuity results in the triangular decomposition of the element domains pictured in Fig. 20b. Since $\mathbf{x}_0 = \mathbf{x}_c \notin \bar{\Omega}_1$, some cubature points will lie outside the domain of integration and some cubature weights will be negative. These points and weights coincide with the horizontally hatched triangles in Fig. 20b.

Integration accuracy versus the number of cubature points is illustrated in Fig. 21. In the t direction, the r^{-1} singularity cancelling transformation is applied to all results. Based on the example presented in Section 5.3.2, this is the transformation that provides the most accurate integration for an integrand with a $r^{-1/2}$ singularity. In the ξ direction, three integration methods are tested: no transformation in the ξ direction, the generalized SB transformation with $\alpha = 2$, and Gauss-Jacobi quadrature with $\alpha = 0$ and $\beta = 1/2$. An equal number of cubature points is used in the t and ξ directions on each triangular partition. In the plots presented in Fig. 21, the average error over the 16 integrals is presented as a solid line. Shaded regions give the range of error over the 16 integrals. Compared to the results in Section 5.3.1, the generalized SB transformation and the Gauss-Jacobi rule are closely aligned. The only case where the Gauss-Jacobi rule provided noticeably better integration is over Ω_2 with $\Delta x = 0.001$. Regardless, either method is able to provide approximately 14 digits of integration accuracy with a 16×16 rule over each triangular partition. Without any ξ -transformation, less than five digits of integration accuracy are obtained on average with a 20×20 rule over each triangle. For this problem, the accuracy of integration did not seem to be affected by the presence of points outside the element domain and negative cubature weights, since the results for Ω_1 and Ω_2 are comparable. The ability to easily generate rules with points outside the domain with the SBC scheme enables economical integration of singular integrands when the singularity is not present within the domain. Computing the same integral using, for example, very high-order tensor-product Gauss cubature provides unsatisfactory results [19].

5.4. Transfinite mean value interpolation

Transfinite barycentric interpolation over domains bounded by curves is the continuous counterpart of generalized barycentric coordinates over polygons [54]. Consider an open, bounded convex domain Ω with boundary $\Gamma = \partial\Omega$. Generalized barycentric coordinates [55, 56], $\phi : \Omega \rightarrow \mathbb{R}_+^m$, on an m -gon are nonnegative and satisfy the partition of

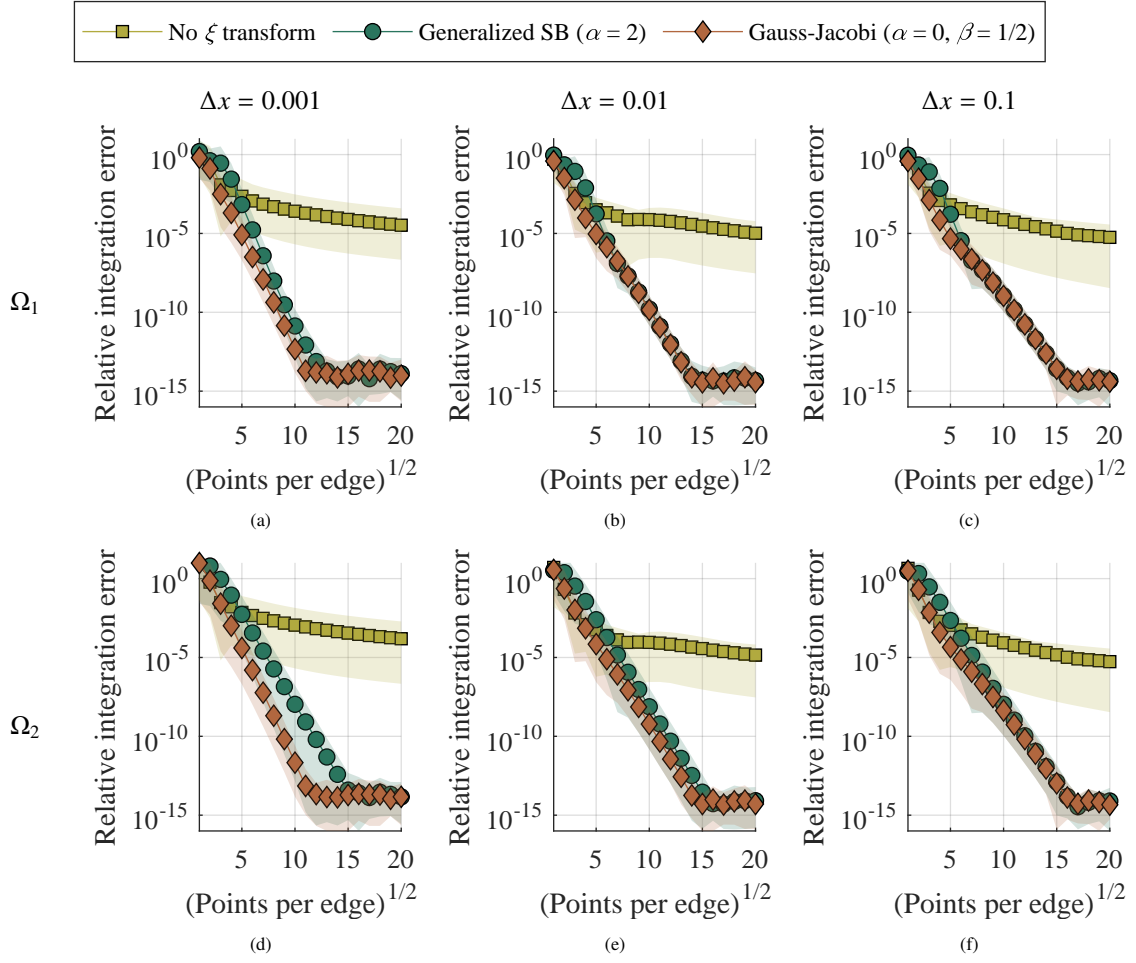


Figure 21: Cubature points per edge versus integration error for a weakly singular extended finite element enriched basis function that is integrated over Ω_1 with (a) $\Delta x = 0.001$, (b) $\Delta x = 0.01$, and (c) $\Delta x = 0.1$ and over Ω_2 with (d) $\Delta x = 0.001$, (e) $\Delta x = 0.01$, and (f) $\Delta x = 0.1$.

unity and linear precision properties:

$$\mathbf{x} = \sum_{i=1}^m \phi_i(\mathbf{x}) \mathbf{x}_i, \quad \phi_i(\mathbf{x}) = \frac{w_i(\mathbf{x})}{W(\mathbf{x})}, \quad W(\mathbf{x}) = \sum_{j=1}^m w_j(\mathbf{x}),$$

where the weight function $w_i : \bar{\Omega} \rightarrow \mathbb{R}_+$ is nonnegative. The weights $\mathbf{w} = (w_1, w_2, \dots, w_m)$ are also known as homogeneous coordinates. Similarly, given a function $g : \Gamma \rightarrow \mathbb{R}$ that is prescribed on a curved boundary, the transfinite mean value interpolant $u : \Omega \rightarrow \mathbb{R}$ is defined as [40, 41]

$$u(\mathbf{x}) = \frac{\int_{S_v} g(\mathbf{y}(\mathbf{x}, \mathbf{v})) K(\mathbf{x}, \mathbf{y}) dS_v}{W(\mathbf{x})}, \quad W(\mathbf{x}) = \int_{S_v} K(\mathbf{x}, \mathbf{v}) dS_v, \quad K(\mathbf{x}, \mathbf{v}) = \frac{1}{\|\mathbf{v} - \mathbf{x}\|}, \quad (59)$$

where $\mathbf{x} \in \Omega \setminus S_v$ and $\mathbf{v} \in S_v$, S_v is the unit circle that is centered at \mathbf{x} , the ray from \mathbf{x} that passes through \mathbf{v} intersects the boundary Γ at \mathbf{y} , and $K(\mathbf{x}, \mathbf{v})$ is a singular kernel function [40, 41]. See Fig. 22 for an illustration. For this study, we let $g(\mathbf{x})$ be a function that is defined over $\bar{\Omega}$ whose mean value interpolant is sought. We have $\lim_{\mathbf{x} \rightarrow \mathbf{y}} u(\mathbf{x}) = g(\mathbf{y})$, and in addition, if g is an affine function over $\bar{\Omega}$, then the transfinite interpolant $u = g$. So similar to GBCs on polygons, the TMVI can exactly reproduce any linear polynomial. The function $\psi(\mathbf{x}) = 1/W(\mathbf{x})$ behaves like an approximate distance function to the boundary and its normal derivative on the boundary Γ is $1/2$ [41]. Dyken and Floater [41] use the circumferential mean value theorem to derive the transfinite mean value interpolant in (59), and its representation is analogous in the discrete case to the Shepard interpolant where inverse-distance singular nodal weight functions are

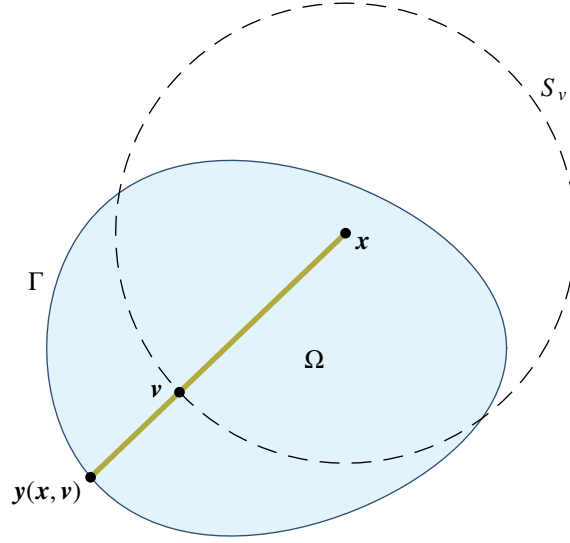


Figure 22: Egg-shaped convex domain, with illustration of variables that appear in (59).

used for scattered data interpolation [57]. For a nonconvex domain, a ray from \mathbf{x} can intersect the boundary C at one or more (odd number) points; see Dyken and Floater [41] for the general form of (59).

On projecting the boundary curve Γ onto the unit circle, the integrals in (59) can be written for a nonconvex domain in terms of the curve parameter $t \in [0, 1]$ as [40]

$$u(\mathbf{x}) = \frac{\int_0^1 g(\mathbf{c}(t))K(\mathbf{x}, t) dt}{W(\mathbf{x})}, \quad W(\mathbf{x}) = \int_0^1 K(\mathbf{x}, t) dt, \quad K(\mathbf{x}, t) = \frac{(\mathbf{c}(t) - \mathbf{x}) \cdot \mathbf{c}'^\perp(t)}{\|\mathbf{c}(t) - \mathbf{x}\|^3}, \quad (60)$$

which is convenient for numerical computations. In (60), $\mathbf{c}'^\perp(t) := \text{rot}(\mathbf{c}'(t))$ is obtained by rotating $\mathbf{c}'(t)$ through 90° in the clockwise direction.

As noted in [41], the TMVI is derived by Lee [58] using flux integrals and Euler's homogeneous function theorem, with generalizations of mean value coordinates on polygons and polyhedra appearing in Lee [59]. Here we present the connections. Consider a convex domain Ω with boundary Γ . Since the Laplace operator is rotationally invariant, the kernel $K(\mathbf{x}, \mathbf{y}) := K(\mathbf{y} - \mathbf{x})$ is only dependent on the radial coordinate. Let $\boldsymbol{\rho} = \mathbf{y} - \mathbf{x}$ and $\rho := \|\boldsymbol{\rho}\|$, where $\mathbf{y} \in C$. On noting that $\mathbf{n}_\rho = \frac{\mathbf{c}'^\perp(t)}{\|\mathbf{c}'(t)\|}$ is the unit outward normal vector to C and $ds_\rho = \|\mathbf{c}'(t)\| dt$ is the arc-length differential on C , we can rewrite (59) as

$$u(\mathbf{x}) = \frac{\int_\Gamma g(\mathbf{x} + \boldsymbol{\rho})K(\mathbf{x}, \boldsymbol{\rho}) ds_\rho}{\int_\Gamma K(\mathbf{x}, \boldsymbol{\rho}) ds_\rho}, \quad K(\mathbf{x}, \boldsymbol{\rho}) = \frac{\boldsymbol{\rho}}{\rho^3} \cdot \mathbf{n}_\rho. \quad (61)$$

Now, (61) is in the form of a flux integral. So, on noting the correspondence, we consider $K(\mathbf{x}, \boldsymbol{\rho}) := \mathbf{F}(\boldsymbol{\rho}) \cdot \mathbf{n}_\rho$ to be the ansatz for the kernel function, where $\mathbf{F}(\boldsymbol{\rho}) := h(\rho)\boldsymbol{\rho}$ is a vector field with $h(\rho)$ being a scalar-valued function. If g is an affine function, then the TMVI interpolant $u = g$. So, let $g(\mathbf{x}) = \mathbf{a} \cdot \mathbf{x}$, where $\mathbf{a} \in \mathbb{R}^2$ is an arbitrary vector. Then, substituting $u(\mathbf{x}) = \mathbf{a} \cdot \mathbf{x}$ on the left-hand side and $g(\mathbf{x} + \boldsymbol{\rho}) = \mathbf{a} \cdot (\mathbf{x} + \boldsymbol{\rho})$ on the right-hand side, we obtain

$$\mathbf{a} \cdot \mathbf{x} = \mathbf{a} \cdot \frac{\int_\Gamma (\mathbf{x} + \boldsymbol{\rho}) [\mathbf{F}(\boldsymbol{\rho}) \cdot \mathbf{n}_\rho] ds_\rho}{\int_\Gamma [\mathbf{F}(\boldsymbol{\rho}) \cdot \mathbf{n}_\rho] ds_\rho} = \mathbf{a} \cdot \mathbf{x} + \mathbf{a} \cdot \frac{\int_\Gamma \boldsymbol{\rho} [\mathbf{F}(\boldsymbol{\rho}) \cdot \mathbf{n}_\rho] ds_\rho}{\int_\Gamma [\mathbf{F}(\boldsymbol{\rho}) \cdot \mathbf{n}_\rho] ds_\rho}, \quad (62)$$

and therefore, since \mathbf{a} is arbitrary,

$$\mathbf{I} := \frac{\int_\Gamma \boldsymbol{\rho} [\mathbf{F}(\boldsymbol{\rho}) \cdot \mathbf{n}_\rho] ds_\rho}{\int_\Gamma [\mathbf{F}(\boldsymbol{\rho}) \cdot \mathbf{n}_\rho] ds_\rho} = \mathbf{0} \quad (63)$$

must hold in $\mathbb{R}^2 \setminus \{\mathbf{0}\}$. For the above to be true, the numerator must be identically equal to $\mathbf{0}$ and the denominator must not vanish. On substituting $\mathbf{F}(\boldsymbol{\rho}) = h(\boldsymbol{\rho})\boldsymbol{\rho}$ in the above equation, we obtain

$$\mathbf{I} = \frac{\int_{\Gamma} [h(\boldsymbol{\rho})\boldsymbol{\rho} \otimes \boldsymbol{\rho}] \cdot \mathbf{n}_{\boldsymbol{\rho}} ds_{\boldsymbol{\rho}}}{\int_{\Gamma} [h(\boldsymbol{\rho})\boldsymbol{\rho} \cdot \mathbf{n}_{\boldsymbol{\rho}}] ds_{\boldsymbol{\rho}}}, \quad (64)$$

and on invoking the divergence theorem, we have

$$\mathbf{I} = \lim_{\epsilon \rightarrow 0^+} \frac{\int_{\Omega \setminus S_{\epsilon}} [h(\boldsymbol{\rho})(\mathbf{I} \cdot \boldsymbol{\rho}) + h(\boldsymbol{\rho})(\nabla \cdot \boldsymbol{\rho})\boldsymbol{\rho} + \boldsymbol{\rho}(\nabla h(\boldsymbol{\rho}) \cdot \boldsymbol{\rho})] dA}{\int_{\Omega \setminus S_{\epsilon}} [\nabla h(\boldsymbol{\rho}) \cdot \boldsymbol{\rho} + h(\boldsymbol{\rho})(\nabla \cdot \boldsymbol{\rho})] dA}, \quad (65)$$

where S_{ϵ} is a circle of radius ϵ that is centered at \mathbf{x} . Now, let $h(\boldsymbol{\rho}) := \rho^q$ be a q -homogeneous function. We have $\nabla \cdot \boldsymbol{\rho} = 2$ and by Euler's homogeneous function theorem $\nabla h \cdot \boldsymbol{\rho} = qh$. Then, the above equation becomes

$$\mathbf{I} = \frac{\int_{\Omega \setminus S_{\epsilon}} (q+3)h(\boldsymbol{\rho})\boldsymbol{\rho} dA}{\int_{\Omega \setminus S_{\epsilon}} (q+2)h(\boldsymbol{\rho}) dA}. \quad (66)$$

So if $q = -3$, then the numerator vanishes for $\boldsymbol{\rho} \in \mathbb{R}^2 \setminus \{\mathbf{0}\}$ and $\mathbf{I} = \mathbf{0}$ a.e., since the integrand in the denominator, $-h(\boldsymbol{\rho})$, is strictly negative. Therefore, by ensuring linear precision, we find that the vector field $\mathbf{F} = \frac{\boldsymbol{\rho}}{\rho^3}$ matches that used in the transfinite mean value interpolant in (59). This provides the flux integral representation to construct transfinite interpolants [58, 59].

We now apply the TMVI to two functions over a curved domain: a linear polynomial and a trigonometric function. Since linear polynomials are exactly reproduced by the TMVI, the linear polynomial should be reproduced without any error. Both functions are interpolated over an open, bounded convex domain Ω with boundary $\Gamma = \partial\Omega$ (see Fig. 22). The curved boundary $\Gamma : [0, 1] \rightarrow \mathbb{R}^2$ shown in Fig. 22 has the parametric representation:

$$\mathbf{c}(\theta(t)) = r \cos \theta \mathbf{e}_1 + \frac{ar \sin \theta}{b + r \cos \theta} \mathbf{e}_2, \quad \mathbf{c}'(\theta(t)) = -2\pi r \sin \theta \mathbf{e}_1 + \frac{2\pi ar(r + b \cos \theta)}{(b + r \cos \theta)^2} \mathbf{e}_2, \quad (67)$$

where $\theta := \theta(t) = 2\pi t$ and $(\cdot)'$ denotes differentiation with respect to t . In these numerical computations, we choose $a = 4$, $b = 5$, and $r = 1$. The functions we choose to interpolate over $\bar{\Omega}$ are:

$$g_1(\mathbf{x}) = 1 - 2x + 3y, \quad g_2(\mathbf{x}) = \sin x \sin y. \quad (68)$$

These functions are plotted in Figs. 23a and 23b. The TMVI is computed using (60) and the resulting interpolants are illustrated in Figs. 23c and 23d. The relative L_2 error over Ω is defined as:

$$\mathcal{E} = \frac{\|u - g\|_{2,\Omega}}{\|u\|_{2,\Omega}}, \quad (69)$$

which is computed for $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ using the SBC scheme. As expected, interpolation of $g_1(\mathbf{x})$ yields machine precision relative L_2 error of 2.2×10^{-16} . Interpolation of the trigonometric function using TMVI is not exact away from Γ . Though the surfaces displayed in Fig. 23b and Fig. 23d appear nearly identical, the relative L_2 error is 1.24×10^{-2} .

As a second, related example, we use a modified inverse of the weight function in the TMVI to provide a smooth approximation of the distance function. Belyaev et al. [60] introduced L_p -distance fields ($p \geq 1$):

$$\psi(\mathbf{x}) = \left(\frac{1}{W_p(\mathbf{x})} \right)^{1/p}, \quad W_p(\mathbf{x}) = \int_0^1 \frac{(\mathbf{c}(t) - \mathbf{x}) \cdot \mathbf{c}'^{\perp}(t)}{\|\mathbf{c}(t) - \mathbf{x}\|^{2+p}} dt, \quad (70)$$

which approximates the exact distance function. In (70), as $p \rightarrow \infty$, $\psi(\mathbf{x})$ approaches the exact distance function. Selecting $p = 1$ recovers the weight function in the TMVI. Consider the egg-shaped domain Ω (Fig. 22) with boundary Γ whose parametric representation is given in (67). In Fig. 24a, we plot the relative L_2 error for the distance function versus p ($p \in [1, 100]$). With $p = 1$, the relative L_2 error is 76 percent and with $p = 100$, relative L_2 error reduces to 1.1 percent. The approximate distance functions for $p = 1, 10$, and 100 are plotted in Figs. 24b, 24c, and 24d, respectively. The exact distance function,

$$d(\mathbf{x}) = \min_{t \in [0,1]} \|\mathbf{x} - \mathbf{c}(t)\| \quad \forall \mathbf{x} \in \Omega, \quad (71)$$

is calculated by minimizing the square of the distance function using Newton's method.

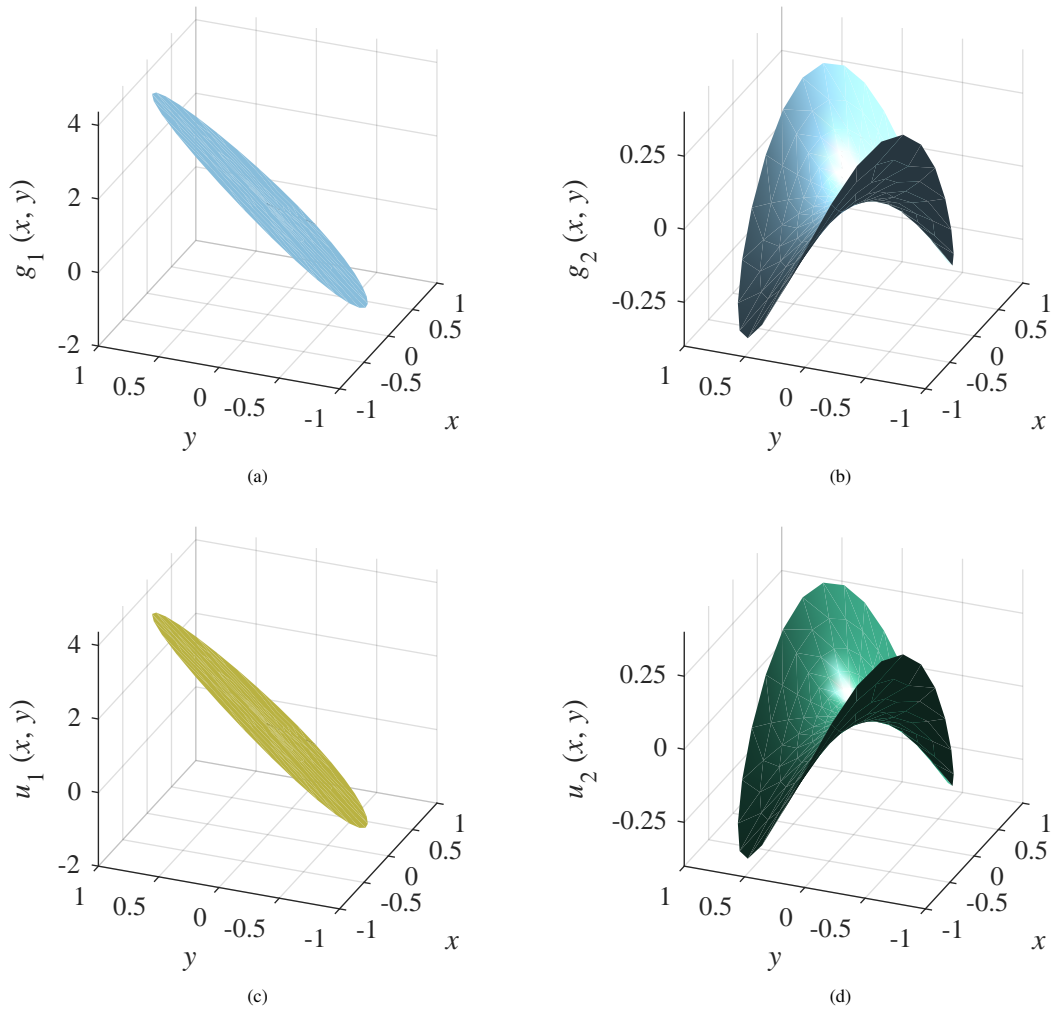


Figure 23: Interpolating functions over the domain pictured in Fig. 22 using the TMVI. (a) $g_1(x)$; (b) $g_2(x)$; (c) $u_1(x)$, the TMVI of $g_1(x)$; and (d) $u_2(x)$, the TMVI of $g_2(x)$.

6. Conclusions

In this paper, we proposed the scaled boundary cubature (SBC) method for numerical integration over polygons and domains that are bounded by parametric curves. This method of integration applies the SB parametrization introduced in the context of the SB-FEM [4, 5] to transform integration from a region bounded by a curve and two line segments to the unit square. As explained in Section 2, the method is very closely related to the homogeneous numerical integration scheme (HNI) [2, 3], but is not solely limited to the integration of homogeneous functions. A broad suite of numerical examples were considered in Section 5 that revealed the promise of using the SBC scheme in the polygonal finite element method, extended finite element method, and transfinite interpolation of functions over curved domains using the transfinite mean value interpolant (TMVI). In these examples, integration was performed over a number of convex and nonconvex shapes, demonstrating the flexibility of the method. Furthermore, in Section 5.4 the close connection between the TMVI and homogeneous functions was noted, paralleling the link between the SBC and HNI schemes. The simplicity of the SBC scheme enabled fast cubature rule generation when compared to other methods of integration, such as Gauss-Green cubature [14, 15] and triangulation. In the examples presented in Section 5, accuracy per cubature point for the SBC scheme was comparable to triangulation and the Gauss-Green scheme. For star-convex domains, it is possible to place all cubature points inside the domain of integration and to have all positive cubature weights with the SBC scheme.

The SBC method was also suitable for integrating functions with point singularities. On subdividing the domain

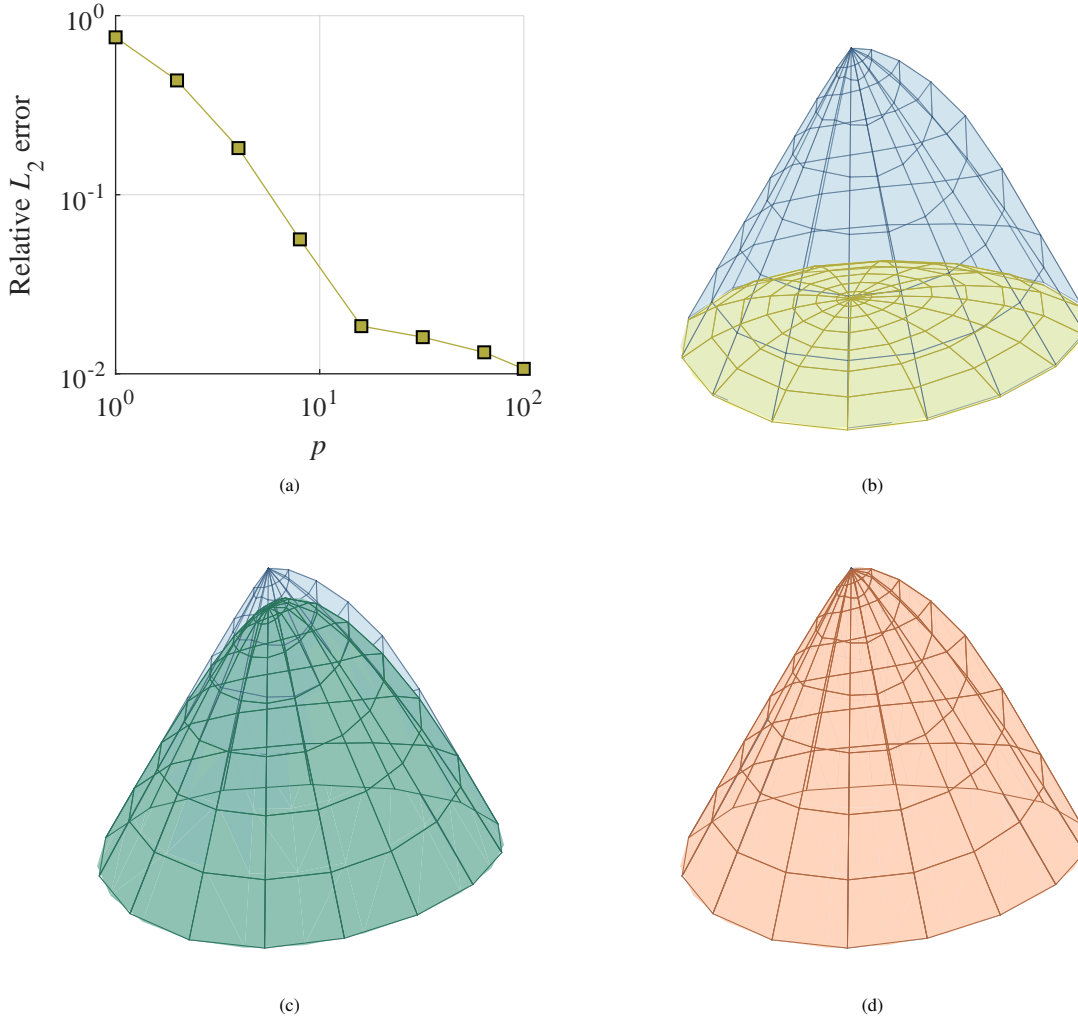


Figure 24: Approximating the distance function over the domain pictured in Fig. 22 using L_p -distance fields. (a) L_2 error versus p , L_p -distance field with (b) $p = 1$, (c) $p = 10$, and (d) $p = 100$. In (b)–(d), the distance function is plotted as a semi-transparent overlay.

of integration into curved triangles that share a point, the SBC scheme simplifies the formulation of integral transformations that smooth singularities at that point. In Section 4.1, we demonstrated the SBC scheme is equivalent to the Duffy transformation [44] when the domain of integration is the standard triangle. Therefore, the SBC scheme provides efficient integration of functions containing a r^{-1} radial singularity without modification. We also introduced a modified mapping in Section 4.1 that we named the generalized SB transformation due to its similarity to the generalized Duffy transformation [38] when applied to the standard triangle. This led to efficient integration of functions with fractional radial singularities. Additionally, we pointed out that Gauss-Jacobi quadrature also provides efficient integration of functions with fractional power point singularities. To smooth near-singularities along the edges of polygons, we described several t -direction integral transformations in Section 4.2. When the transformations in Section 4.2 are combined with the transformations in Section 4.1, accurate integration of weakly singular functions with minimal cubature points is realized. These capabilities were explored in examples that appeared in Section 5.3. As these examples revealed, the SBC scheme with singularity cancelling transformations can be readily applied to the X-FEM and the BEM.

The fundamental need to integrate functions over polytopes and curved solids appears in numerical methods and also within many higher-order embedded domain methods, where *exact* representation of geometry is desirable. In this paper, we have proposed a new method for numerical integration of smooth functions and weakly singular functions over elements with curved boundaries. We showed that the method is accuracy and efficient, while underscoring

the simplicity in its implementation. The performance of SBC reveals that it can find use in polygonal finite element methods, boundary element methods, enriched partition-of-unity methods, embedded domain methods, and isogeometric analysis. Since the SBC scheme enables tensor-product Gauss integration rules to be utilized over complex element shapes, there is greater flexibility in the choice of approximation spaces that one can adopt in the underlying computational method. The main ideas and contributions in this work readily extend to three-dimensional regions that are bounded by affine and curved parametric surfaces, which is the subject of ongoing research.

Appendix A. Connection to the Poincaré Lemma

In this appendix, we use the Poincaré Lemma to provide an alternate derivation for the SBC scheme. The Poincaré Lemma for vector fields is [42]

$$h(\mathbf{x}) = \nabla \cdot \left[\int_0^1 h(\xi \mathbf{x}) \xi \mathbf{x} d\xi \right]. \quad (\text{A.1})$$

We introduce a curved triangle \mathcal{T} that is bounded by two affine lines, \mathcal{L}_1 and \mathcal{L}_2 , and a curved boundary $C : [0, 1] \rightarrow \mathbb{R}^2$ that is parametrized by $\mathbf{c}(t)$. Let the vertex at the intersection of \mathcal{L}_1 and \mathcal{L}_2 lie at the origin. We integrate $h(\mathbf{x})$ over \mathcal{T} and apply the divergence theorem:

$$\int_{\mathcal{T}} h(\mathbf{x}) d\mathbf{x} = \int_{\mathcal{T}} \nabla \cdot \left[\int_0^1 h(\xi \mathbf{x}) \xi \mathbf{x} d\xi \right] d\mathbf{x} = \int_{\partial \mathcal{T}} \left[\int_0^1 h(\xi \mathbf{x}) \xi \mathbf{x} d\xi \right] \cdot \mathbf{n} ds. \quad (\text{A.2})$$

On noting that coordinates on the segments $\mathcal{L}_1 \setminus \{\mathbf{0}\}$ and $\mathcal{L}_2 \setminus \{\mathbf{0}\}$ are perpendicular to the outward normal (see (8)), we can write

$$\int_{\mathcal{T}} h(\mathbf{x}) d\mathbf{x} = \int_C \left[\int_0^1 h(\xi \mathbf{x}) \xi d\xi \right] (\mathbf{x} \cdot \mathbf{n}) ds = \int_0^1 \int_0^1 h(\xi \mathbf{c}(t)) \xi d\xi (\mathbf{c}(t) \cdot \mathbf{n}(t)) \|\mathbf{c}'(t)\| dt. \quad (\text{A.3})$$

Upon substituting the expression for the unit normal given in (10) into (A.3) and noting $h(\xi \mathbf{c}(t)) = (h \circ \psi)(\xi \rho(t), t)$ (see (16)), (21) is recovered.

If (A.1) is modified to incorporate a constant shift \mathbf{x}_0 :

$$h(\mathbf{x}) = \nabla \cdot \left[\int_0^1 h(\boldsymbol{\varphi}) \xi (\mathbf{x} - \mathbf{x}_0) d\xi \right], \quad (\text{A.4})$$

where $\boldsymbol{\varphi} := \boldsymbol{\varphi}(\xi, t) = \mathbf{x}_0 + \xi(\mathbf{x} - \mathbf{x}_0)$ is the SB parametrization introduced in (22), then (23) is the result after integrating over \mathcal{T} . This reveals that the SBC method can be derived using the Poincaré Lemma.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

References

- [1] J. B. Lasserre, Integration on a convex polytope, *Proceedings of the American Mathematical Society* 126 (8) (1998) 2433–2441.
- [2] E. B. Chin, J. B. Lasserre, N. Sukumar, Numerical integration of homogeneous functions on convex and nonconvex polygons and polyhedra, *Computational Mechanics* 56 (6) (2015) 967–981.
- [3] E. B. Chin, N. Sukumar, Modeling curved interfaces without element-partitioning in the extended finite element method, *International Journal for Numerical Methods in Engineering* 120 (5) (2019) 607–649.
- [4] C. Song, J. P. Wolf, The scaled boundary finite-element method—alias consistent infinitesimal finite-element cell method—for elastodynamics, *Computer Methods in Applied Mechanics and Engineering* 147 (3–4) (1997) 329–355.
- [5] J. P. Wolf, C. Song, The scaled boundary finite-element method—a fundamental solution-less boundary-element method, *Computer Methods in Applied Mechanics and Engineering* 190 (2001) 5551–5568.
- [6] N. Moës, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, *International Journal for Numerical Methods in Engineering* 46 (1) (1999) 131–150.
- [7] R. Sevilla, S. Fernández-Méndez, A. Huerta, NURBS-enhanced finite element method (NEFEM), *International Journal for Numerical Methods in Engineering* 76 (2008) 56–83.
- [8] T. P. Fries, S. Omerović, D. Schöllhammer, J. Steidl, Higher-order meshing of implicit geometries—Part I: Integration and interpolation in cut elements, *Computer Methods in Applied Mechanics and Engineering* 313 (2017) 759–784.
- [9] E. Artioli, A. Sommariva, M. Vianello, Algebraic cubature on polygonal elements with a circular edge, *Computers & Mathematics with Applications* 79 (7) (2020) 2057–2066.
- [10] S. E. Mousavi, H. Xiao, N. Sukumar, Generalized Gaussian quadrature rules on arbitrary polygons, *International Journal for Numerical Methods in Engineering* 82 (1) (2010) 99–113.
- [11] R. I. Saye, High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles, *SIAM Journal on Scientific Computing* 37 (2) (2015) A993–A1019.
- [12] F. Scholz, B. Jüttler, Numerical integration on trimmed three-dimensional domains with implicitly defined trimming surfaces, *Computer Methods in Applied Mechanics and Engineering* 357 (2019) 112577.
- [13] F. Scholz, B. Jüttler, High-order quadrature on planar domains based on transport theorems for implicitly defined moving curves, *Tech. Rep. 89, Nationales Forschungsnetzwerk, Geometry + Simulation* (May 2020).
- [14] A. Sommariva, M. Vianello, Product Gauss cubature over polygons based on Green’s integration formula, *BIT Numerical Mathematics* 47 (2007) 441–453.
- [15] A. Sommariva, M. Vianello, Gauss-Green cubature and moment computation over arbitrary geometries, *Journal of Computational and Applied Mathematics* 231 (2009) 886–896.
- [16] D. Gunderman, K. Weiss, J. A. Evans, Spectral mesh-free quadrature for planar regions bounded by rational parametric curves, *Computer-Aided Design* 130 (2021) 102944.
- [17] J. B. Lasserre, Integration and homogeneous functions, *Proceedings of the American Mathematical Society* 127 (3) (1999) 813–818.
- [18] E. B. Chin, N. Sukumar, An efficient method to integrate polynomials over polytopes and curved solids, *Computer Aided Geometric Design* 82 (2020) 101914.
- [19] E. B. Chin, J. B. Lasserre, N. Sukumar, Modeling crack discontinuities without element-partitioning in the extended finite element method, *International Journal for Numerical Methods in Engineering* 110 (11) (2017) 1021–1048.
- [20] L. Chen, B. Simeon, S. Klinkel, A NURBS based Galerkin approach for the analysis of solids in boundary representation, *Computer Methods in Applied Mechanics and Engineering* 305 (2016) 777–805.
- [21] S. Klinkel, R. Reichel, A finite element formulation in boundary representation for the analysis of nonlinear problems in solid mechanics, *Computer Methods in Applied Mechanics and Engineering* 347 (2019) 295–315.
- [22] N. Sukumar, J. E. Dolbow, N. Moës, Extended finite element method in computational fracture mechanics: a retrospective examination, *International Journal of Fracture* 196 (2015) 189–206.
- [23] K. Hormann, N. Sukumar (Eds.), *Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics*, CRC Press, New York, NY, 2017.
- [24] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Computer Methods in Applied Mechanics and Engineering* 194 (39–41) (2005) 4135–4195.
- [25] B. Marussig, T. J. R. Hughes, A review of trimming in isogeometric analysis: challenges, data exchange, and simulation aspects, *Archives of Computational Methods in Engineering* 25 (2018) 1059–1127.
- [26] J. E. Bishop, Rapid stress analysis of geometrically complex domains using implicit meshing, *Computational Mechanics* 30 (5) (2003) 460–478.
- [27] E. Burman, S. Claus, P. Hansbo, M. G. Larson, A. Massing, CutFEM: Discretizing geometry and partial differential equations, *International Journal for Numerical Methods in Engineering* 104 (7) (2014) 472–501.
- [28] D. Schillinger, M. Ruess, The finite cell method: A review in the context of higher-order structural analysis of CAD and image-based geometric models, *Archives of Computational Methods in Engineering* 22 (2015) 391–455.
- [29] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, A. Russo, Basic principles of virtual element methods, *Mathematical Models and Methods in Applied Sciences* 23 (1) (2013) 199–214.
- [30] L. Beirão da Veiga, A. Russo, G. Vacca, The virtual element method with curved edges, *ESAIM: Mathematical Modelling and Numerical Analysis* 53 (2019) 375–404.
- [31] E. Benvenuti, A. Chiozzi, G. Manzini, N. Sukumar, Extended virtual element method for the Laplace problem with singularities and discontinuities, *Computer Methods in Applied Mechanics and Engineering* 256 (2019) 571–597.
- [32] A. J. Lew, G. C. Buscaglia, A discontinuous-Galerkin-based immersed boundary method, *International Journal for Numerical Methods in Engineering* 76 (2008) 427–454.
- [33] A. Cangiani, E. H. Georgoulis, P. Houston, *hp*-version discontinuous Galerkin methods on polygonal and polyhedral meshes, *Mathematical Models and Methods in Applied Sciences* 24 (10) (2014) 2009–2041.
- [34] M. A. Puso, T. A. Laursen, A mortar segment-to-segment contact method for large deformation solid mechanics, *Computer Methods in Applied Mechanics and Engineering* 193 (6–8) (2004) 601–629.

- [35] C. Hesck, P. Betsch, Transient three-dimensional contact problems: mortar method. Mixed methods and conserving integration, *Computational Mechanics* 48 (2011) 461–475.
- [36] E. Guendelman, R. Bridson, R. Fedkiw, Nonconvex rigid bodies with stacking, *ACM Transactions on Graphics* 22 (3) (2003) 871–878.
- [37] A. Krishnamurthy, S. McMains, Accurate GPU-accelerated surface integrals for moment computation, *Computer-Aided Design* 43 (2011) 1284–1295.
- [38] S. E. Mousavi, N. Sukumar, Generalized Duffy transformation for integrating vertex singularities, *Computational Mechanics* 45 (2010) 127–140.
- [39] R. Franke, A critical comparison of some methods for interpolation of scattered data, Tech. Rep. NPS-53-79-003, Naval Postgraduate School (1979).
- [40] T. Ju, S. Schaefer, J. Warren, Mean value coordinates for closed triangular meshes, *ACM Transactions on Graphics* 24 (3) (2005) 561–566.
- [41] C. Dyken, M. S. Floater, Transfinite mean value interpolation, *Computer Aided Geometric Design* 26 (1) (2009) 117–134.
- [42] R. E. Terrell, *The fundamental theorem of calculus and the Poincaré lemma*, Cornell University, Ithaca, NY 14853 (2009).
URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.159.3286&rep=rep1&type=pdf>
- [43] C. Arioli, A. Shamanskiy, S. Klinkel, B. Simeon, Scaled boundary parametrizations in isogeometric analysis, *Computer Methods in Applied Mechanics and Engineering* 349 (2019) 576–594.
- [44] M. G. Duffy, Quadrature over a pyramid or cube of integrands with a singularity at a vertex, *SIAM Journal on Numerical Analysis* 19 (6) (1982) 1260–1262.
- [45] A. Chernov, C. Schwab, Exponential convergence of Gauss-Jacobi quadratures for singular integrals over simplices in arbitrary dimension, *SIAM Journal on Numerical Analysis* 50 (3) (2012) 1433–1455.
- [46] H. Ma, N. Kamiya, Distance transformation for the numerical evaluation of near singular boundary integrals with various kernels in boundary element method, *Engineering Analysis with Boundary Elements* 26 (2002) 329–339.
- [47] J.-H. Lv, Y.-Y. Jiao, X.-T. Feng, P. Wriggers, X.-Y. Zhuang, T. Rabczuk, A series of Duffy-distance transformation for integrating 2D and 3D vertex singularities, *International Journal for Numerical Methods in Engineering* 118 (1) (2019) 38–60.
- [48] E. B. Chin, Numerical integration of homogeneous functions with applications in the extended finite element method, Ph.D. thesis, University of California, Davis (2019).
- [49] D. A. Dunavant, High degree efficient symmetrical Gaussian quadrature rules for the triangle, *International Journal for Numerical Methods in Engineering* 21 (1985) 1129–1148.
- [50] E. L. Wachspress, *A Rational Finite Element Basis*, Academic Press, New York, 1975.
- [51] M. Floater, A. Gillette, N. Sukumar, Gradient bounds for Wachspress coordinates on polytopes, *SIAM Journal on Numerical Analysis* 52 (1) (2014) 515–532.
- [52] H. Xiao, Z. Gimbutas, A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions, *Computers & Mathematics with Applications* 59 (2009) 663–676.
- [53] A. Nagarajan, S. Mukherjee, A mapping method for numerical evaluation of two-dimensional integrals with $1/r$ singularity, *Computational Mechanics* 12 (1993) 19–26.
- [54] A. G. Belyaev, P.-A. Fayolle, Transfinite barycentric coordinates, in: Hormann and Sukumar [23], pp. 43–62.
- [55] M. S. Floater, Generalized barycentric coordinates and applications, *Acta Numerica* 24 (2015) 161–214.
- [56] D. Anisimov, Barycentric coordinates and their properties, in: Hormann and Sukumar [23], pp. 3–22.
- [57] D. Shepard, A two-dimensional interpolation function for irregularly-spaced data, in: *Proceedings of the 23rd ACM national conference, Association for Computing Machinery*, New York, New York, 1968, pp. 517–524.
- [58] S. L. Lee, *Mean value representations and curvatures of compact convex hypersurfaces*, Preprint (2007).
URL <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.389.951&rep=rep1&type=pdf>
- [59] S. L. Lee, Vector fields for mean value coordinates, *SIAM Journal on Mathematical Analysis* 40 (6) (2009) 2437–2450.
- [60] A. Belyaev, P.-A. Fayolle, A. Pasko, Signed L_p -distance fields, *Computer-Aided Design* 45 (2) (2013) 523–528.