

Accepted Manuscript

Shape Aware Quadratures

Vaidyanathan Thiagarajan, Vadim Shapiro

PII: S0021-9991(18)30324-3
DOI: <https://doi.org/10.1016/j.jcp.2018.05.024>
Reference: YJCPH 8023

To appear in: *Journal of Computational Physics*

Received date: 12 December 2017
Revised date: 12 May 2018
Accepted date: 13 May 2018

Please cite this article in press as: V. Thiagarajan, V. Shapiro, Shape Aware Quadratures, *J. Comput. Phys.* (2018), <https://doi.org/10.1016/j.jcp.2018.05.024>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



Highlights

- Non-trivial generalization of our earlier work “Adaptively Weighted Numerical Integration (AW)” to efficiently account for topological features in addition to geometric features using higher-order shape/topological sensitivities.
- Formulation of a new integration scheme to efficiently account for small features in the context of domain integration.
- Generalization of the notion of moment approximations and correction factors in the context of moment fitting equations using a variety of sensitivity analysis techniques.
- Detailed derivation of first/second-order shape and topological sensitivity of moment integrals.
- Demonstration of SAQ in the context of polynomial integration over 2D/3D domains with hundreds/thousands of small features using non-conforming mesh (octree/quadtree) based integration.

excessive fragmentation near the boundary, resulting in significant computational cost and/or loss of accuracy.

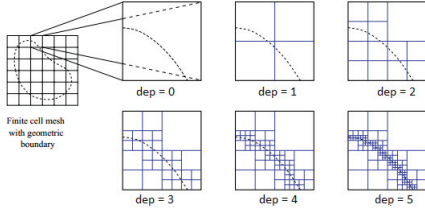


Figure 2: Illustration of excessive boundary cell fragmentation in quadtree based integration [5]

Fragmentation of boundary cells become even more pronounced in the perforated plate problem of Fig. 1. The presence of small features such as voids or inclusions smaller than an integration cell (in both 2D and 3D) poses a serious issue. Fig. 3 illustrates this situation in which the void is completely situated inside the cell. Since all vertices of a cell are inside the geometric domain, this void is missed completely by conventional integration techniques. There are several ways to detect such small features. If the user knows that small features exist in the domain geometry, an initial grid with tighter spacing (Fig. 4(a)) can be provided to ensure a cell corner intersects the feature. This is a simple way to globally ensure that features down to a given size are detected and accounted for. However, because the increased grid resolution applies everywhere in the domain, including regions where it is not needed, it can unnecessarily drive up the computational cost. This approach can be optimized if an additional information about location of small geometric features is available. In this case, as it is illustrated by Fig. 4 (b) and Fig. 4 (c), a denser grid is imposed within a cell or a non-uniform adaptive subdivision can be applied depending on the location of small features.

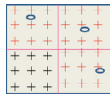


Figure 3: Features smaller than the integration grid resolution are missed

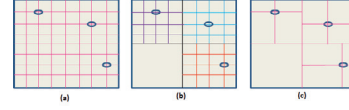


Figure 4: Capturing the effect of small features by (a) uniform subdivision (b) overlaying of fine grid and (c) non-uniform adaptive subdivision

However, these techniques become prohibitively expensive for a model with large number of small features such as the perforated plate problem of Fig. 1. This explains the need for accurate volumetric integration scheme with minimal subdivision to accelerate the whole solution process. Such integration requires the integration scheme adapts to the geometry and topology (i.e. shape) of the domain in addition to the integrand. Issues related to integrand adaptivity are well understood [6]. On the other hand, efficient shape adaptivity is far from complete and will be the main focus of this paper. Specifically, the problem this paper addresses can be stated formally as follows:

Numerically evaluate the integral

$$\int_{\Omega} f(\mathbf{x}) d\Omega \quad (1)$$

where $f : \Omega \rightarrow \mathbb{R}$ is an arbitrary integrable function defined over an arbitrary^a domain $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3) that is typically represented by a solid model [7].

^aBy arbitrary domain we mean any 2D or 3D closed regular set [7] whose boundary is an orientable manifold

We further assume that f can be suitably approximated by a function $\hat{f} \in \text{span}(\{b_i\}_{i=1}^m)$ such that the basis functions $b_i : \mathbb{R}^d \rightarrow \mathbb{R}$ are symbolically integrable functions that are sufficiently smooth. In other words, \hat{f} can be written as a linear combination of sufficiently smooth basis functions b_i i.e. $\hat{f}(\mathbf{x}) = \sum_{i=1}^m c_i b_i(\mathbf{x})$. The coefficients c_i need not necessarily be known ahead of time.

In this paper, we propose a new integration technique called *Shape Aware Quadratures* (SAQ) that is a non-trivial generalization of our earlier work *Adaptively Weighted Numerical Integration Method* (AW) [1, 2, 3]. AW computes quadrature rule for an arbitrary domain by solving a system of suitable linear moment fitting equations. Further, in AW, the

moments are approximated over a simplified homeomorphic domain with corrections based on first-order SSA [1, 2, 3]. AW works well for domains without small features (such as small holes and inclusions). However, the presence of several thousand small features can substantially deteriorate the performance of AW as it involves careful construction of an approximate domain (Ω_0) that is homeomorphic to the original domain (Ω). This homeomorphic domain (Ω_0) construction is often difficult and computationally expensive due to the inherent complexity of the original domain in the presence of small features. Moreover, the method involves computing several boundary integrals over this complex approximate boundary thereby further affecting its performance. Hence, in this paper, we overcome this limitation by generalizing AW to account for small geometric and topological features efficiently using higher-order shape sensitivity [8, 9], topological sensitivity [10, 11], and other types of sensitivities [12, 13]. The introduction of these sensitivities eliminate the need for an homeomorphic approximate domain in addition to substantially reducing the number of boundary integral computations. Thus, unlike AW, SAQ can efficiently integrate arbitrary integrable functions over arbitrary domains even in the presence of thousands of small features.

In SAQ, given an arbitrary domain Ω with a set of appropriately chosen quadrature nodes and order of integration, we compute the quadrature weights by solving a system of linear moment fitting equations [14, 15, 16] for an appropriate set of basis functions in the least square sense. Setting up the moment-fitting equations involves computing integrals of the basis functions (known as moments) over an arbitrary geometric domain. This task itself is non-trivial, because it either entails some kind of domain decomposition, or can be reduced to repeated boundary integration as was proposed in [17] for bivariate domains. Adaptation of the latter approach avoids excessive domain fragmentation, but leads to a significant computational overhead, particularly in 3D. We overcome this challenge by first computing the moments over a simpler domain Ω_0 , usually a polygon/polyhedron, that is a reasonable approximation of the original domain Ω . The computed moments are then corrected for the deviation of the shape of Ω_0 from Ω by means of shape correction factors. The shape correction factors can be computed via various sensitivity analysis techniques such as the shape sensitivity [8, 9], topological sensitivity [10, 11], feature sensitivity [12],

and modification sensitivity [13]. The moments over approximate domain are further reduced to boundary integrals by the application of divergence theorem [18, 19, 20, 21, 22]. The approximate moment fitting equations, thus obtained, can then be easily solved for quadrature weights. In other words, the shape correction factor ensures that the quadrature rule determined by the moment fitting equations is “aware” of the shape of integration domain – hence the name *Shape Aware Quadratures (SAQ)*. The resulting quadrature weights are not exact, but are accurate enough to integrate functions over any arbitrary domain when they are well approximated by the chosen set of basis functions. In addition, small features such as holes, notches, and thin features can be accounted for automatically by employing the correct order/type of shape correction factor(s) in the moment approximations.

Outline. A brief survey of relevant background material in volumetric integration and sensitivity analysis is presented in section 2. The formulation of *Shape Aware Quadrature (SAQ)* is developed in section 3 as a means to efficiently integrate arbitrary integrable functions over arbitrary domains in the presence of small features. In section 4, we explain the derivation of shape correction factors via shape and topological sensitivity analysis. The algorithmic details of SAQ are described in section 5. We present the experimental validation of SAQ over 2D and 3D domains in the presence of numerous small features in section 6. Section 7 presents the conclusion and open issues.

2. Background

In this section we will briefly review the relevant background material necessary to understand Shape Aware Quadratures (SAQ). For a detailed survey of numerical integration we refer the reader to [1, 2, 3, 23, 24, 25, 26].

2.1. Divergence Theorem

Several authors [18, 20, 19, 21, 22] have advocated the use of divergence theorem to integrate symbolically integrable functions over polygonal/polyhedral domains ($\Omega_0 \subset \mathbb{R}^d$, $d = 2$ or 3). Here, application of divergence theorem (once for polygonal domain and twice for polyhedral domain) converts the 2D/3D integral into an 1D integral over the edges

of the polygon/polyhedron. The resulting integral is then efficiently evaluated using 1D Gauss quadrature rules. The integral of a symbolically integrable function $b_i(\mathbf{X})$ over a polygonal/polyhedral domain with n^* edges/faces (i.e. $\Gamma_0 = \cup_{k=1}^{n^*} \Gamma_0^k$) can be reduced to the following boundary integral by the application of divergence theorem (see [1, 2] for details)

$$\int_{\Omega_0} b_i(\mathbf{X}) d\Omega_0 = \sum_{k=1}^{n^*} \int_{\Gamma_0^k} \beta_X^i(\mathbf{X}) N_X^k d\Gamma_0^k \quad (2)$$

N_X^k is the X -component of the normal to the k^{th} edge/face (Γ_0^k) of the polygon/polyhedron and in general β_X^i is non-unique as there are many ways to define it. One possible way to compute β_X^i is to use the following definition

$$\beta_X^i(\mathbf{X}) = \begin{cases} \int b_i(X, Y) dX & \text{for polygons} \\ \int b_i(X, Y, Z) dX & \text{for polyhedra} \end{cases}$$

In the case of polyhedra, one more application of divergence theorem to Eq.(2) would reduce the original integral into a 1D integral over the edges of the polyhedra. It is important to note that this technique cannot be applied to integrands that are not symbolically integrable.

2.2. Moment Fitting Equations

For functions that are not symbolically integrable, moment fitting equations are a systematic way to derive quadrature rule for arbitrary domains. In general, a quadrature in \mathbb{R}^d is a formula of the form :

$$\sum_{i=1}^n w_i f(\mathbf{x}_i) \approx \int_{\Omega} W(\mathbf{x}) f(\mathbf{x}) d\Omega \quad (3)$$

where $\Omega \subset \mathbb{R}^d$ is the integration region, f is an integrand defined on Ω , and $W(\mathbf{x})$ is in general a non-negative weight function but is assumed to be unity (i.e. $W(\mathbf{x}) = 1$) in this paper. A quadrature rule is determined by points $\mathbf{x}_i \in \mathbb{R}^d$ that are usually called quadrature nodes, and the quadrature weights w_i [16]. A standard technique for constructing quadrature rules is to solve the following system of moment

fitting equations [14, 15, 16]

$$\begin{Bmatrix} \int_{\Omega} b_1(\mathbf{x}) d\Omega \\ \int_{\Omega} b_2(\mathbf{x}) d\Omega \\ \vdots \\ \int_{\Omega} b_m(\mathbf{x}) d\Omega \end{Bmatrix} = \begin{bmatrix} b_1(\mathbf{x}_1) & b_1(\mathbf{x}_2) & \cdots & b_1(\mathbf{x}_n) \\ b_2(\mathbf{x}_1) & b_2(\mathbf{x}_2) & \cdots & b_2(\mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ b_m(\mathbf{x}_1) & b_m(\mathbf{x}_2) & \cdots & b_m(\mathbf{x}_n) \end{bmatrix} \begin{Bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{Bmatrix} \quad (4)$$

The same equations may be written in a more compact form as

$$\{\mathbf{M}\}_{m \times 1} = [\mathbf{A}]_{m \times n} \{\mathbf{w}\}_{n \times 1} \quad (5)$$

with

$$\{\mathbf{M}\} = \begin{Bmatrix} \int_{\Omega} b_1(\mathbf{x}) d\Omega \\ \int_{\Omega} b_2(\mathbf{x}) d\Omega \\ \vdots \\ \int_{\Omega} b_m(\mathbf{x}) d\Omega \end{Bmatrix} \quad (6)$$

where $\{b_i\}_{i=1}^m$ is the set of basis functions¹ and $\{\mathbf{M}\}$ is the vector of moments defined over the domain of integration Ω .

The quadrature nodes and the corresponding weights $\{\mathbf{x}_i, w_i\}_{i=1}^n$ can be determined numerically by solving the above set of non-linear moment fitting equations. The resulting quadrature rule can be used to integrate any function that is in the function space spanned by these basis functions, assuming that the integral of basis functions and solution to moment fitting equations were computed exactly. Each integration point in d -dimensions contribute $d + 1$ unknowns: d coordinate components and one weight. Thus, $\lceil \frac{m}{d+1} \rceil$ could serve as an estimate for the number of points required to integrate m basis functions in d dimensions [28]. We refer the reader to [16, 29, 30, 31, 32] for the application of this method in generating quadrature rules for convex/non-convex polygon/polyhedron.

2.3. Shape Sensitivity Analysis

We could also try to evaluate $\int_{\Omega} f(\mathbf{x}) d\Omega$ in two steps: first integrate function $f(x)$ over another domain Ω_0 that approximates Ω , and then correct the result by the difference between the two integrals.

One popular approximation is based on the application of first-order Shape Sensitivity Analysis (SSA) [8] from the shape optimization [8] literature. As

¹Popular choices of basis functions include the bivariate polynomials $x^p y^r$ ($p + r \leq o$) and trivariate polynomials $x^p y^r z^s$ ($p + r + s \leq o$) in 2D and 3D respectively (for the given integration order o). Other recommended choices of basis functions include Legendre and Chebyshev polynomials [27].

a first-order approximation, consider a reference domain Ω_0 that is homeomorphic to and in the neighborhood of the original domain Ω . In other words, there exists a continuous bijective mapping $\mathbf{T}(\mathbf{X}, t)$ (with continuous inverse) that deforms the reference domain into the original domain. Let \mathbf{T} be a homeomorphic mapping that transforms the reference domain in to the deformed domain i.e. $\mathbf{x} = \mathbf{T}(\mathbf{X}, t)$ (see Fig. 5). Thus, $\mathbf{x} \in \Omega_t$ is a function of the shape parameter t . The parameter t denotes the amount of shape change in the design variable direction such that $t = 0$ represents the reference domain Ω_0 [8]. If $\mathbf{T}(\mathbf{X}, t)$ is assumed to be regular enough in the neighborhood of $t = 0$, then it can be expanded using the Taylor series around the initial mapping point $\mathbf{T}(\mathbf{X}, 0)$ as [8]

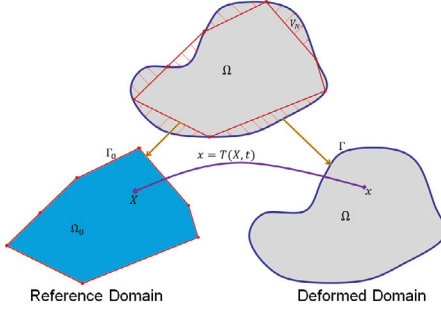


Figure 5: Reference and Deformed domains for SSA

$$\begin{aligned} \mathbf{x} &= \mathbf{T}(\mathbf{X}, t) \\ &= \mathbf{T}(\mathbf{X}, 0) + t \frac{\partial \mathbf{T}}{\partial t}(\mathbf{X}, 0) + \dots \\ &= \mathbf{X} + t \mathbf{V}(\mathbf{X}) + \dots \end{aligned} \quad (7)$$

where $\mathbf{V}(\mathbf{X}) = \frac{\partial \mathbf{T}}{\partial t}(\mathbf{X}, 0) = \left. \frac{d\mathbf{x}}{dt} \right|_{t=0}$ is the design velocity vector that is defined over the reference domain Ω_0 . First-order shape sensitivity is developed by considering only the first two terms of the above Taylor series. Thus, ignoring the higher order terms, \mathbf{T} is assumed to be a homeomorphic linear mapping in t given by

$$\mathbf{x} = \mathbf{T}(\mathbf{X}, t) = \mathbf{X} + t \mathbf{V}(\mathbf{X}), \mathbf{X} \in \Omega. \quad (8)$$

In this context, the original domain Ω could also be referred to as the deformed domain. On applying first-order SSA [8], and assuming that f can be suitably extended outside of Ω , we get the following ap-

proximation to the integral [8]

$$I = \int_{\Omega} f d\Omega \approx \int_{\Omega_0} f d\Omega_0 + \int_{\Gamma_0} f V_N d\Gamma_0, \quad (9)$$

where V_N is the normal component of the design velocity. Thus the original integral over Ω was approximated using SSA over a simpler domain Ω_0 . Furthermore, at least in principle, the integrals over the polygonal/polyhedral domain (Ω_0) can be computed using the quadrature rules via the moment fitting equations as explained in [16, 29]. Although this method would give a reasonable approximation to the original integral, it suffers from a serious drawback. In order to evaluate Eq.(9), function $f(\mathbf{x})$ must be sampled over the entire polygonal/polyhedral domain Ω_0 and/or boundary Γ_0 . This either assumes that Ω_0 lies completely within the original domain Ω , or requires extension of the function to outside the domain (Ω), usually based on application specific heuristic arguments as in [33]. To circumvent this rather severe restriction, the boundary term in Eq.(9) could be written in domain form using divergence theorem:

$$I \approx \int_{\Omega_0} [f + \nabla \cdot (f \mathbf{V})] d\Omega_0, \quad (10)$$

reducing the original integration over Ω to that over approximate simplified domain Ω_0 . Although this eliminates the need for function extension, it does require the domain velocity computation which is computationally expensive, ambiguous, and/or not easily implemented [9].

These difficulties were overcome by us in our earlier work on AW integration [1, 2, 3] where first-order SSA was used to approximate moments arising in moment fitting equations. This was possible because the basis functions arising in moment fitting equations are defined everywhere in \mathbb{R}^d and not just in the domain of integration (Ω_0). For further details of AW, we refer the reader to section 3 and our earlier work [1, 2, 3].

2.4. Topological Sensitivity Analysis

The first-order topological derivative is the sensitivity of a quantity of interest (QOI) (Q) defined over a region of interest (in the domain) when an infinitesimal hole is introduced in the domain [10]. In this paper, we will always assume that the QOI is a very simple integral of the form $\int_{\Omega} b_i(\mathbf{x})$ such that the function b_i is only a function of the coordinates (and is independent of any underlying field as is the case in a bound-

any value setting). In this case, topological derivative may be used to estimate sensitivity of such an integral with respect to introduction of small holes. Formally, topological sensitivity is defined as

$$D_T^1(\mathbf{x}) = \lim_{\epsilon \rightarrow 0} \frac{Q(\Omega_\epsilon) - Q(\Omega)}{f_1(\epsilon)} \quad (11)$$

where $\Omega_\epsilon = \Omega - B_\epsilon$ and $f_1(\epsilon)$ is a monotonically decreasing function that depends on the problem under consideration. The ball B_ϵ has a radius of ϵ with its center at the point $\mathbf{x} \in \Omega$. There are closed form expressions for topological sensitivity of simple quantity of interest such as compliance when studying the effect of perturbation of the domain in the context of boundary value problems [34]. For more general QOIs, the computation of topological derivative requires solving adjoint problems. For details we refer the reader to [35, 11, 36, 37].

For a finite spherical hole of radius r with center at $\hat{\mathbf{x}} \in \Omega$, one could approximate its effect on the QOI as

$$Q_1 \approx Q_0 + \mu(B_r) * D_T^1(\hat{\mathbf{x}}) \quad (12)$$

where Q_0 is the QOI computed in the domain without the hole and $\mu(B_r)$ is the measure (area or volume) of the ball of radius r . Analogous to Eq.(12), for a domain with an arbitrarily shaped small feature, one could define the following estimate for any given QOI

$$Q_1 \approx Q_0 + \mu(B_\epsilon) * D_T^1(\hat{\mathbf{x}}) \quad (13)$$

where ϵ is taken to be the radius of the smallest ball that contains the given small feature ω . For relatively larger features, the estimate given in Eq.(13) could be erroneous. Hence, for relatively larger features one could often use the generalization of topological sensitivity such as the feature sensitivity [12, 38, 39] and modification sensitivity [13, 33, 40].

3. SAQ

Shape Aware Quadratures (SAQ) are quadratures that adapt to the geometry and topology (i.e. shape) of the integration domain automatically/efficiently.

One of the main applications of SAQ is in the integration of arbitrary integrable functions over arbitrary domains in the presence of small features such as holes, notches, and fillets. In order to define small features we will use the standard morphological opening $(A \circ B)^2$ and closing $((A \bullet B))^3$ operators [41]

Definition

The union of small features of a domain A relative to an user defined size δ is given by one of the following operations

1. $A - (A \circ B)$
2. $(A \bullet B) - A$

where the structuring set B is a ball of diameter δ . All *small positive features* of a domain A are given by $A - (A \circ B)$ while all *small negative features* are filled by $(A \bullet B) - A$.

3.1. Formulation

We will begin by noting that the moment fitting equations (Eq. 5) is a system of non-linear equations that can be solved for the position (\mathbf{x}_i) and weights (w_i) using an iterative solution scheme such as the Newton-Raphson [42]. However, such solution schemes are computationally very expensive and often do not scale for larger problems. But, if we fix the position of the integration points as in [28, 1, 2, 3], the equations become linear (in weights) and can be solved easily. Knowing $\{\mathbf{M}\}$, one could solve the moment equations for weights $\{\mathbf{w}\}$ using linear least squares as

$$\{\mathbf{w}\}_{n \times 1} = [\mathbf{A}^\dagger]_{n \times m} \{\mathbf{M}\}_{m \times 1} \quad (14)$$

where $[\mathbf{A}^\dagger]$ is the Moore-Penrose pseudo inverse [43]. One popular stable algorithm to numerically compute $[\mathbf{A}^\dagger]$ is based on the QR factorization [44].

Thus, given an arbitrary domain Ω with a set of appropriately chosen quadrature nodes and order of integration, quadrature weights are obtained by solving a system of linear moment fitting equations for an appropriate set of basis functions in the least

² $A \circ B$ is the dilation of the erosion of a set A by B i.e. $A \circ B = (A \ominus B) \oplus B$

³ $A \bullet B$ is the erosion of the dilation of a set A by B i.e. $A \bullet B = (A \oplus B) \ominus B$

Note:

$$A \ominus B = \{a \in A \mid B_a \subseteq A\} \text{ where } B_a = \{b + a \mid b \in B\}, \forall a \in A$$

$$A \oplus B = \{a + b \mid a \in A, b \in B\}$$

square sense. Setting up the moment-fitting equations involves computing integrals of the basis functions (known as moments) over an arbitrary geometric domain. This task itself is non-trivial, because it either entails some kind of domain decomposition, or can be reduced to repeated boundary integration as was proposed in [17] for bivariate domains. Adaption of the latter approach avoids excessive domain fragmentation, but leads to a significant computational overhead, particularly in 3D (especially in the presence of small features).

This challenge is overcome by first computing the moments (i.e. integral of basis functions b_i) over a simpler domain Ω_0 , usually a polygon/polyhedron, that is a reasonable approximation (in a Hausdorff distance sense) of the original domain Ω (see Fig. 6). The computed moments are then corrected for the deviation of the shape of Ω_0 from Ω based on sensitivity analysis. Thus,

$$M_i \approx M_i^0 + C_i^S \quad (15)$$

M_i^0 is the i^{th} moment computed over the simplified domain Ω_0 i.e. $M_i^0 = \int_{\Omega_0} b_i(\mathbf{X}) d\Omega_0$ and C_i^S is the shape correction factor corresponding to M_i . In general, the application of these sensitivities results in correction factors that are boundary integrals over some domain. In other words, the shape correction factor (C_i^S) is of the following form

$$C_i^S = \sum_{f=1}^{N_f} \int_{\Gamma_f} g_i^f(b_i, \mathbf{X}) d\Gamma_f \quad (16)$$

$\Gamma_f \subset \mathbb{R}^{d-1}$ is the boundary of a domain that depends on the type of sensitivity employed in computing the corrections. For example, if the correction factor was computed based on shape sensitivity analysis (SSA), then Γ_f is usually a polygonal/polyhedral boundary that approximates Γ as shown in Fig. 5. Here, g_i^f is a scalar function depending only on the basis function b_i and Γ_f .

The moment approximations can entirely be written as a boundary integral by the application of divergence theorem [18, 20, 19, 21, 22] i.e. M_i^0 in Eq.(15)

can be replaced by Eq.(2) as

$$M_i \Big|_{\Omega} \approx \int_{\Gamma_0} \beta_X^i(\mathbf{X}) N_X d\Gamma_0 + \sum_{f=1}^{N_f} \int_{\Gamma_f} g_i^f(b_i, \mathbf{X}) d\Gamma_f \quad (17)$$

g_i^f depends on the type/order of sensitivity used. Its exact form can be deduced from Table.1 and Table.2 for SSA and TSA respectively. We note that that Eq.(17) is a significant generalization of our earlier formulation used to derive the Adaptively Weighted Numerical Integration method [1, 2, 3].

In other words, this equation accounts for topological features (such as holes and inclusions) in addition to geometric features (such as notches and fillets) in the given domain. Thus, for the sake of convenience and understanding, let us write the shape correction factor as the sum of geometric (C_i^G) and topological correction factors (C_i^T) i.e.

$$M_i \approx M_i^0 + C_i^G + C_i^T \quad (18)$$

C_i^G and C_i^T respectively account for the geometric and topological deviations of Ω_0 from Ω in computing the i^{th} moment approximation. If the simplified domain Ω_0 is chosen to be homeomorphic⁴ to Ω then $C_i^T = 0$ (see Fig. 7) and then the above reduces to the following :

$$M_i \approx M_i^0 + C_i^G \quad (19)$$

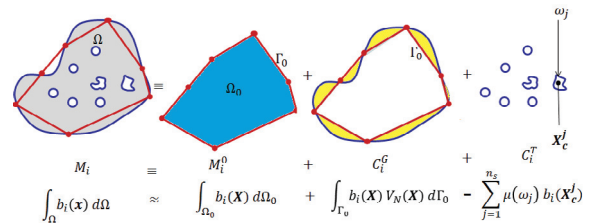


Figure 6: SAQ - Simplified domain Ω_0 is not homeomorphic to Ω

⁴by homeomorphic we mean the mapping $\mathbf{T}(\mathbf{x}, t)$ that we assume in Eq.(8) is continuous and bijective with a continuous inverse

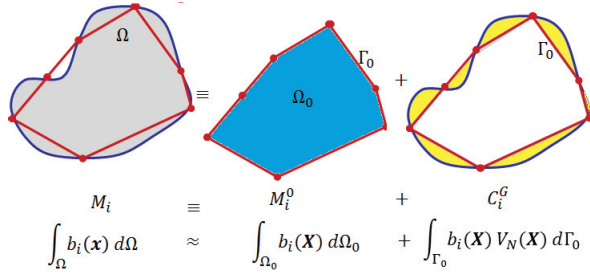


Figure 7: When simplified domain Ω_0 is homeomorphic to Ω , SAQ recovers AW [1, 2, 3] as a special case when first-order SSA is used for geometric corrections (C_i^G)

C_i^G and C_i^T can be obtained using a variety of sensitivity analysis techniques. For instance, if we consider the domain Ω and its approximation Ω_0 as shown in Fig. 6, then one could employ first-order SSA (section 2.3) for C_i^G and TSA (section 2.4) for C_i^T to obtain the following moment approximations

$$M_i \approx \int_{\Gamma_0} \beta_X^i(\mathbf{X}) N_X d\Gamma_0 + \int_{\Gamma_0} b_i(\mathbf{X}) V_N(\mathbf{X}) d\Gamma_0 - \sum_{j=1}^{n_s} \mu(\omega_j) b_i(\mathbf{X}_c^j) \quad (20)$$

$V_N(\mathbf{X}) = \mathbf{V}(\mathbf{X}) \cdot \mathbf{N}(\mathbf{X})$ is the normal component of the design velocity at a point \mathbf{X} on the boundary Γ_0 of the reference (piecewise linear) domain Ω_0 . $\mu(\omega_j)$ and \mathbf{X}_c^j are respectively the measure and centroid of the j^{th} negative small feature ω_j . Here, C_i^G corrects for the geometric deviation of Ω_0 (without holes) from Ω and C_i^T accounts for the presence of small holes (or topology). The correction C_i^G can be easily derived by the straightforward application of first-order SSA [8] to the moments (for details refer to Appendix A). Likewise, the correction term C_i^T can be obtained by the straightforward application of first-order TSA [11] (for details refer to Appendix B). It is to be noted that μ in Eq.(20) can in general be evaluated as a boundary integral by the application of divergence theorem or SSA.

Likewise, employing first-order SSA for both C_i^G and C_i^T we get

$$M_i \approx \int_{\Gamma_0} \beta_X^i(\mathbf{X}) N_X d\Gamma_0 + \int_{\Gamma_0} b_i(\mathbf{X}) V_N(\mathbf{X}) d\Gamma_0 - \sum_{j=1}^{n_s} \int_{\Gamma_j^0} [\beta_X^i(\mathbf{X}) N_X + b_i(\mathbf{X}) V_N(\mathbf{X})] d\Gamma_j^0 \quad (21)$$

where Γ_j^0 is the approximate polygonal/polyhedral domain chosen homeomorphic to the boundary of the small feature ω_j . Alternatively, one could employ first-order SSA for C_i^G and divergence theorem (or zeroth-order SSA) for C_i^T resulting in the following

$$M_i \approx \int_{\Gamma_0} \beta_X^i(\mathbf{X}) N_X d\Gamma_0 + \int_{\Gamma_0} b_i(\mathbf{X}) V_N(\mathbf{X}) d\Gamma_0 - \sum_{j=1}^{n_s} \int_{\Gamma_j^0} [\beta_X^i(\mathbf{X}) N_X] d\Gamma_j^0 \quad (22)$$

If there are no topological features (as in Fig. 7), and if we employ first-order SSA for C_i^G , then Eq.(21) reduces to the following

$$M_i \approx \int_{\Gamma_0} \beta_X^i(\mathbf{X}) N_X d\Gamma_0 + \int_{\Gamma_0} b_i(\mathbf{X}) V_N(\mathbf{X}) d\Gamma_0 \quad (23)$$

It is important to note that in this scenario we recover AW method [1, 2, 3] as a special case of SAQ. In other words, SAQ is same as AW when Ω_0 is chosen homeomorphic to Ω and first-order SSA is employed to compute the geometric correction factors C_i^G .

3.2. High-level Algorithm

For 2D domains, the boundary integral arising in moment approximations could be obtained simply by employing the usual Gaussian quadrature over the edges of the polygon. Similarly, boundary integration over 3D domains bounded by triangles or convex quadrilaterals is straightforward. For more general 3D domains, we proceed in two steps. First, we compute the appropriate weights for a chosen set of quadrature nodes for each of the polygonal faces of the approximating polyhedron by solving the moment fitting equations (Eq.(14)) with exact $\{\mathbf{M}\}$ given by Eq.(2). Then, using the determined weights for the polygonal faces, we obtain approximate $\{\mathbf{M}\}$ for the arbitrary 3D domain by numerically evaluating Eq.(16) over the faces of the polyhedron. Using this approximate $\{\mathbf{M}\}$, the approximate weights for the arbitrary domain are obtained by solving the moment fitting equations i.e. Eq.(14).

The approximate moment fitting equations, thus obtained, can then be easily solved for quadrature weights that adapt to the shape of the original domain. The resulting weights are not exact, but are accurate enough to integrate functions over any arbitrary domain when they are well approximated by the chosen set of basis functions. This integration scheme

could be advantageously employed in cell decomposition methods, for example using quadrees [45] or octrees [46], to reduce fragmentation.

Algorithm 1: High-level SAQ Algorithm

Input :

- Order of integration (o)
 - Quadrature nodes ($\{\mathbf{x}\}$)
 - Integration domain (Ω)
 - Integrand (f)
1. Construct the domain Ω_0 approximate to Ω
 2. Choose a complete set of basis functions (b_i) depending on the order of integration (o)
 3. Compute the moment approximations:
 $\{\mathbf{M}\} \approx \{\mathbf{M}^0\} + \{\mathbf{C}^G\} + \{\mathbf{C}^T\}$
 4. Evaluate $[A]$ at the given set of quadrature nodes ($\{\mathbf{x}\}$)
 5. Solve for weights : $\{\mathbf{w}\} = [\mathbf{A}^\dagger]\{\mathbf{M}\}$

Output: Approximation to the integral :

$$\int_{\Omega} f(\mathbf{x})d\Omega \approx \sum_{i=1}^n w_i f(\mathbf{x}_i)$$

4. SAQ - Sensitivity Analysis of Moments

The correction factors C_i^G and C_i^T can in general be obtained using any order SSA [8], TSA [37], feature sensitivity (FSA) [12] or modification sensitivity [13]. However, in this paper, we will limit ourselves to SSA and TSA only.

One could write Eq.(18) in the following form

$$M_i \approx \int_{\Gamma_0} \beta_X^i(\mathbf{X}) d\Gamma_0 + \sum_{k=1}^{n_g} C_i^{G^k} + \sum_{l=1}^{n_t} C_i^{T^l} \quad (24)$$

$C_i^{G^k}$ is the k^{th} order geometric correction factor while $C_i^{T^l}$ is the l^{th} order topological correction factor. For the example domain shown in Fig. 6, if we consider up to second-order SSA for geometric corrections and first-order TSA for topological corrections then we have the following moment approximations

$$\begin{aligned} M_i &\approx \int_{\Gamma_0} \beta_X^i N_X d\Gamma_0 + \sum_{k=1}^2 C_i^{G^k} + \sum_{l=1}^1 C_i^{T^l} \\ &= \int_{\Gamma_0} \beta_X^i N_X d\Gamma_0 + C_i^{G^1} + C_i^{G^2} + C_i^{T^1} \\ &= \int_{\Gamma_0} \beta_X^i N_X d\Gamma_0 + t \int_{\Gamma_0} b_i V_N d\Gamma_0 \\ &\quad + \frac{t^2}{2} \int_{\Gamma_0} [\nabla b_i]^T \mathbf{N} V_N^2 d\Gamma_0 + \sum_{j=1}^{n_s} \mu(\omega_j) [-b_i(\mathbf{X}_c^j)] \\ &= \int_{\Gamma_0} \beta_X^i N_X d\Gamma_0 + \int_{\Gamma_0} b_i V_N d\Gamma_0 \\ &\quad + \frac{1}{2} \int_{\Gamma_0} [\nabla b_i]^T \mathbf{N} V_N^2 d\Gamma_0 \\ &\quad - \sum_{j=1}^{n_s} \mu(\omega_j) [b_i(\mathbf{X}_c^j)] \end{aligned} \quad (25)$$

Although, in principle, both SSA and TSA could be applied to compute C^G and C^T , it is more efficient to consider SSA for geometric corrections (C^G) and TSA for topological corrections. Hence, we will limit our discussion in this section to SSA for geometric corrections and TSA for topological corrections.

4.1. SSA of Moments

In general, the n_g^{th} -order geometric correction factor using SSA can be written as

$$\begin{aligned} C_i^G &\approx t D_{S_i}^1(\Gamma_0, V_N) + \frac{t^2}{2!} D_{S_i}^2(\Gamma_0, V_N) \\ &\quad + \frac{t^3}{3!} D_{S_i}^3(\Gamma_0, V_N) + \dots \frac{t^{n_g}}{n_g!} D_{S_i}^{n_g}(\Gamma_0, V_N) \end{aligned} \quad (26)$$

where $t \in [0, 1]$ is the shape parameter and D_S^k is the k^{th} -order shape derivative of the i^{th} moment for the given approximate polygonal/polyhedral boundary Γ_0 and normal velocity $V_N(\mathbf{X})$. Table 1 lists the SSA of moments up to 2^{nd} order. In the table, $b_i(\mathbf{X})$ is the basis function corresponding to the i^{th} moment (M_i) and $V_N(\mathbf{X}) = \hat{\mathbf{V}}(\mathbf{X}) \cdot \mathbf{N}(\mathbf{X})$ is the normal component of the design velocity. It is important to note that the second-order shape sensitivity given in Table 1 is only valid as long as the vertices in 2D/edges in 3D of the approximating polygon/polyhedron Ω_0 lie on Ω as shown in Fig. 8. If this condition is not satisfied, then the design velocity would become discontinuous and thereby violating the fundamental

assumptions of SSA [8]. This condition can almost always be met in 2D by suitably constructing Γ_0 via point sampling or quadtree decomposition of Γ [45]. However, the same cannot be said for arbitrary domains in 3D. Nevertheless, this error can be minimized by choosing a finer polyhedral approximation Γ_0 in 3D.

Further, for our purposes, we will assume that the design velocity vector $\hat{\mathbf{V}}(\mathbf{X})$ is piecewise continuous over every edge/face of Ω_0 . Then, the perturbation of the domain by distance γ in the direction normal to the boundary is $\mathbf{x} - \mathbf{X} = \gamma \mathbf{N}(\mathbf{X})$, which implies that

$$\begin{aligned} V_N(\mathbf{X}) &= V(\mathbf{X}) \cdot \mathbf{N}(\mathbf{X}) \\ &= t \hat{\mathbf{V}}(\mathbf{X}) \cdot \mathbf{N}(\mathbf{X}) \\ &= t \frac{\mathbf{x} - \mathbf{X}}{t} \cdot \mathbf{N}(\mathbf{X}) \\ &= \gamma \mathbf{N}(\mathbf{X}) \cdot \mathbf{N}(\mathbf{X}) = \gamma \end{aligned} \quad (27)$$

Thus, $V_N(\mathbf{X})$ is simply the shortest distance (γ) from \mathbf{X} to the original boundary (Γ) as measured in the normal direction. For further details, we refer the reader to Appendix A. Even for a coarse approximation Ω_0 of Ω one can get accurate results from SSA based correction factors, as long as we include enough higher order shape sensitivities. For instance, for the set of trivariate basis functions up to order 1 (i.e. $\{1, X, Y, Z\}$), the following second-order expansion can be proved to be exact even for a very coarse Ω_0 approximating Ω (as long as the boundary integrals are computed exactly)

$$\begin{aligned} C_i^G &= t D_S^1(\mathbf{X}_j) + \frac{t^2}{2!} D_S^2(\mathbf{X}_j) \\ &= \int_{\Gamma_0} b_i V_N d\Gamma_0 + \frac{1}{2} \int_{\Gamma_0} \nabla b_i^T \mathbf{N} V_N^2 d\Gamma_0 \end{aligned}$$

Thus, the second-order approximation of the moments over an arbitrary domain (using SSA for geometric corrections and assuming $C^T = 0$) can be written as a boundary integral over the approximate polygonal/polyhedral boundary Γ_0 as

$$M_i \approx \int_{\Gamma_0} \left[\beta_X^i N_X + b_i V_N + \frac{1}{2} \nabla b_i^T \mathbf{N} V_N^2 \right] d\Gamma_0 \quad (28)$$

where β_X^i is obtained by applying divergence theorem to $\int_{\Omega_0} b_i d\Omega_0$ as explained in section 2.1. The first-order approximation (Eq. 23) is obtained by simply omitting the last term in the above equation. For area/volume computations (i.e. $b_i(\mathbf{X}) = 1$), it is easy to verify geometrically (see Fig. 8) that the following equality holds

$$\begin{aligned} M_1 &= M_1^0 + C_G \\ &= M_1^0 + t D_S^1(\mathbf{X}_j) \\ &= \int_{\Omega_0} (1) d\Omega_0 + \int_{\Gamma_0} (1) V_N d\Gamma_0 \\ &= \int_{\Gamma_0} \left[X N_X + V_N \right] d\Gamma_0 \end{aligned} \quad (29)$$

In the above equation (and in Eq.(28)), the equality holds as long as the mapping \mathbf{T} that we assume in Eq.(8) is a homeomorphism and the vertices (or edges in 3D) of Γ_0 lie exactly on Γ (for more details see Appendix A).

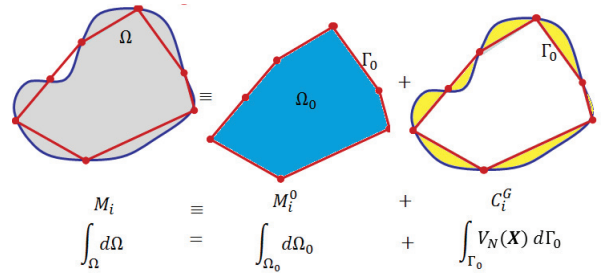


Figure 8: Geometric corrections based on first-order SSA is exact for computation of area (or volume) provided Γ_0 is homeomorphic to Γ and all the vertices (or edges) of Γ_0 lie on Γ

In this paper, we only employ first/second-order SSA to compute C_G . Hence, for lower order correction factors, the error in the approximation cannot be eliminated when approximating higher order moments. Thus, it is important to choose Ω_0 to closely approximate Ω so as to minimize this error.

Table 1: SSA of Moments

k	$D_{S_i}^k(\Gamma_0, V_N)$
1	$\int_{\Gamma_0} b_i V_N d\Gamma_0$
2	$\int_{\Gamma_0} [\nabla b_i]^T \mathbf{N} V_N^2 d\Gamma_0$

Table 2: TSA of Moments (for bivariate/trivariate polynomial basis)

	$D_T^1(\mathbf{X}_c)$	$f_1(\epsilon)$	$D_T^2(\mathbf{X}_c)$	$f_2(\epsilon)$
2D	$-X_c^p Y_c^q$	$\pi\epsilon^2$	$\frac{-1}{2\pi} \left[\binom{p}{2} X_c^{p-2} Y_c^q + \binom{q}{2} X_c^p Y_c^{q-2} \right]$	$\frac{1}{2}\pi^2\epsilon^4$
3D	$-X_c^p Y_c^q Z_c^r$	$\frac{4}{3}\pi\epsilon^3$	$\frac{-1}{4\pi} \left[\binom{p}{2} X_c^{p-2} Y_c^q Z_c^r + \binom{q}{2} X_c^p Y_c^{q-2} Z_c^r + \binom{r}{2} X_c^p Y_c^q Z_c^{r-2} \right]$	$\frac{8}{9}\pi^2\epsilon^5$

4.2. TSA of Moments

In general, the n_t^{th} -order topological correction factor (using TSA) for the i^{th} moment (M_i) over a ball of radius ϵ with center at \mathbf{X}_c can be written as [47, 48]

$$\begin{aligned} C_i^T &= f_1(\epsilon) D_{T_i}^1(\mathbf{X}_c) + f_2(\epsilon) D_{T_i}^2(\mathbf{X}_c) \\ &+ f_3(\epsilon) D_{T_i}^3(\mathbf{X}_c) + \dots f_{n_t}(\epsilon) D_{T_i}^{n_t}(\mathbf{X}_c) + \mathcal{R}(f_{n_t}(\epsilon)) \end{aligned} \quad (30)$$

where f_1, f_2, \dots, f_{n_t} are positive functions that decreases monotonically as $\epsilon \rightarrow 0^+$, $D_{T_i}^k$ is the k^{th} -order topological derivative of the i^{th} moment and $\mathcal{R}(f_{n_t}(\epsilon))$ is the remainder for this approximation. In general, the functions f_j should satisfy the following conditions [47]

$$\begin{aligned} \lim_{\epsilon \rightarrow 0} f_j(\epsilon) &\rightarrow 0 \quad \forall j \in 1, 2, \dots, n_t \\ \lim_{\epsilon \rightarrow 0} \frac{f_j(\epsilon)}{f_{j-1}(\epsilon)} &\rightarrow 0 \quad \forall j \in 2, 3, \dots, n_t \\ \lim_{\epsilon \rightarrow 0} \frac{\mathcal{R}(f_j(\epsilon))}{f_j(\epsilon)} &\rightarrow 0 \quad \forall j \in 1, 2, \dots, n_t \end{aligned} \quad (31)$$

The first ($D_{T_i}^1$) and second order ($D_{T_i}^2$) topological derivatives for moments (corresponding to bivariate/trivariate polynomials) along with the functions f_1 and f_2 are listed in Table. 2.

4.3. Finite sized circular/spherical holes

In 2D, we have the following second-order topological correction for a circular hole of radius ϵ at (X_c, Y_c)

with bivariate polynomial basis functions ($X^p Y^q$)

$$\begin{aligned} C_i^T &\approx -\pi\epsilon^2 [X_c^p Y_c^q] - \frac{1}{4}\pi\epsilon^4 \left[\binom{p}{2} X_c^{p-2} Y_c^q \right. \\ &\quad \left. + \binom{q}{2} X_c^p Y_c^{q-2} \right] \\ &= -\pi\epsilon^2 [X_c^p Y_c^q] \\ &\quad + \frac{\epsilon^2}{2} \left[\frac{\binom{p}{2} X_c^{p-2} Y_c^q + \binom{q}{2} X_c^p Y_c^{q-2}}{2} \right] \end{aligned} \quad (32)$$

Likewise, in 3D, we have the following second-order topological correction for a spherical hole of radius ϵ at (X_c, Y_c, Z_c) with trivariate polynomial basis functions ($X^p Y^q Z^r$)

$$\begin{aligned} C_i^T &\approx -\frac{4}{3}\pi\epsilon^3 [-X_c^p Y_c^q Z_c^r] - \frac{2}{9}\pi\epsilon^5 \left[\binom{p}{2} X_c^{p-2} Y_c^q Z_c^r \right. \\ &\quad \left. + \binom{q}{2} X_c^p Y_c^{q-2} Z_c^r + \binom{r}{2} X_c^p Y_c^q Z_c^{r-2} \right] \\ &= -\frac{4}{3}\pi\epsilon^3 \left([X_c^p Y_c^q Z_c^r] + \frac{\epsilon^2}{2} \right. \\ &\quad \left. \left[\frac{\binom{p}{2} X_c^{p-2} Y_c^q Z_c^r + \binom{q}{2} X_c^p Y_c^{q-2} Z_c^r + \binom{r}{2} X_c^p Y_c^q Z_c^{r-2}}{3} \right] \right) \end{aligned} \quad (33)$$

For details of first and second order TSA of moments in 2D/3D, we refer the reader to Appendix B. Now, from Eq.(32) and Eq.(33), one can deduce the follow-

ing more general form for the second-order correction factor

$$C_i^T \approx -\mu(B_\epsilon) \left[b_i(\mathbf{X}) \Big|_{\mathbf{x}=\mathbf{x}_c} + \frac{\epsilon^2}{2} \frac{\nabla^2(b_i(\mathbf{X}))}{2d} \Big|_{\mathbf{x}=\mathbf{x}_c} \right] \quad (34)$$

where μ and \mathbf{X}_c are respectively the measure and centroid of the ball $B_\epsilon \subset \mathbb{R}^d$ of radius ϵ . Here, $d = 2$ or 3 is the dimension of the domain Ω . It is important to note that this general form (Eq. 34) for topological corrections is only valid for bivariate/trivariate polynomial basis functions.

4.4. Finite sized arbitrary features

For a finite sized small negative/positive feature ω_l we define \mathbf{X}_c^l as the centroid of the feature ω_l and S as the sign of the feature

$$S(\omega_l) = \begin{cases} +1 & \text{if } \omega_l \text{ is a positive feature,} \\ -1 & \text{if } \omega_l \text{ is a negative feature.} \end{cases}$$

A feature ω_l can be classified positive or negative using morphological operations as discussed in the beginning of section 3. Now, we can define an equivalent topological correction factor for arbitrary small negative/positive feature as

$$C_i^T(\omega_l) = C_i^{T_l} \approx S(\omega_l) \mu(\omega_l) \left[b_i(\mathbf{X}) \Big|_{\mathbf{x}=\mathbf{x}_c^l} + \frac{\epsilon_l^2}{2} \frac{\nabla^2(b_i(\mathbf{X}))}{2d} \Big|_{\mathbf{x}=\mathbf{x}_c^l} \right] \quad (35)$$

For most primitive shapes ω_l , μ can be computed using closed form expressions (for e.g. $\mu = \frac{1}{3}\pi r_c^2 h_c$ for a cylinder of radius r_c and height h_c). For more general small features, μ can be estimated accurately by applying divergence theorem (over the original domain ω_l) or via first-order SSA (over an approximate polygonal/polyhedral domain ω_l^0) as a boundary integral. For a collection of small features $\{\omega_l\}_{l=1}^{n_l}$, one can simply use Eq.(35) and sum the result to obtain the total topological correction C^T as

$$C_i^T = \sum_{l=1}^{n_l} C_i^{T_l} = \sum_{l=1}^{n_l} S(\omega_l) \mu(\omega_l) \left[b_i(\mathbf{X}) \Big|_{\mathbf{x}=\mathbf{x}_c^l} + \frac{\epsilon_l^2}{2} \frac{\nabla^2(b_i(\mathbf{X}))}{2d} \Big|_{\mathbf{x}=\mathbf{x}_c^l} \right] \quad (36)$$

Note that the above approach is perfectly valid as we are dealing with topological sensitivity of integrals (moments), unlike topological sensitivity in boundary

value problems, principle of superposition is perfectly valid in our case. Thus, topological corrections can be computed independently for each of the features and can be added together as in Eq.(36).

Usually, the user specifies a relative (small) feature size δ and hence the approximations given in Eq.(35) yields best results when the feature size $f_s(\omega_l) \leq \frac{\delta}{2}$ (as otherwise the feature is not small as per our definition in section 3). The value of δ is usually problem dependent and can be estimated easily. For instance, in octree/quadtrees based integration, one can define δ as a fraction of the smallest leaf cell size h i.e. $\delta = \alpha_f h$ where $0 < \alpha_f < 1$. There are many ways to define the size $f_s(\omega_l)$ of an arbitrary feature ω_l . A standard way is to use local feature size based on medial axis transform as in [49, 50]. However, defining the feature size this way could lead to poor TSA correction estimates for long thin features.

An even better way is to conceptually replace the small feature with a ball B_{ϵ_l} centered at the centroid of the feature (\mathbf{X}_c^l) having the same measure as the small feature (as in [12]) i.e. $\mu(B_{\epsilon_l}) = \mu(\omega_l)$. The radius ϵ_l could be estimated as $\sqrt{\frac{\mu(\omega_l)}{\pi}}$ in 2D and $\sqrt[3]{\frac{3}{4\pi}\mu(\omega_l)}$ in 3D. This is consistent with the standard definition of TSA which considers the sensitivity of the quantity of interest w.r.t to the introduction of an infinitesimal “ball” in the domain. It is also important to note that TSA correction factors can be best applied for finite sized small features that are homeomorphic to a sphere (3D) / disc (2D). For features that are not homeomorphic to sphere (such as the torus), in principle, we could still apply TSA by breaking the small features into pieces that are homeomorphic to a ball. In this paper, we will always assume that the small features are homeomorphic to a sphere in 3D and disc in 2D. In order to compute TSA based correction factors for thin features, it is best to break them into small pieces whose size $\epsilon_l < \frac{\delta}{2}$. Alternatively, for thin features it could be more efficient to employ first/second-order SSA to compute C^T .

5. Algorithmic Details

The procedure follows essentially the same steps for 2D and 3D domains, and produces a set of integration nodes and weights that can be used to integrate any sufficiently smooth function over domain Ω . Conceptually, the procedure consists of the following three subtasks that we describe below in detail.

Initialization. involves setting the data structures required for solving the system of moment equations (Eq.(14)). Note that the dimensions of the matrix $[\mathbf{A}^\dagger]$ depend on the choice of basis functions and on the number of integration nodes, both of which depend on the required order of integration. Specifically, initialization consists of the following steps:

1. Choose a piecewise linear domain Ω_0 (polygon in 2D or polyhedron in 3D) that is a reasonable approximation of the original domain Ω with or without the small features. There are many ways to accomplish this as we have already explained in [1, 2, 3]. In this paper, we construct Ω_0 using quadtree/octree decomposition of Ω using marching squares/cubes algorithm.
2. Determine the order of integration (o) based on the function to be integrated. Choose a set of basis functions $\{b_i\}_{i=1}^m$ of order up to o . A simple choice is the trivariate polynomials $x^p y^r z^s$ (3D) or bivariate polynomials $x^p y^r$ (2D).
3. For the chosen order of integration (o), determine the number and location of quadrature nodes. The minimum number of quadrature points required is dictated by the solvability of moment fitting equations. By having as many equations (basis functions) as the number of unknowns (weights), we make matrix $[\mathbf{A}]$ invertible and thus eliminate the need for least squares solution. The minimum number of quadrature points required is equal to the number of basis functions (m) chosen in the previous step. However, in general, choosing more quadrature points (n) than this minimum number would result in a rectangular system that when solved in a least square sense results in $n-m$ quadrature points with zero weights. Moment fitting equations also offer a lot of flexibility in positioning the quadrature points. There are a number of ways to choose the quadrature points. One way is to use the tensor-product rule ensuring all points are inside the domain as explained in [51] (for e.g., see Fig. 9). Alternatively, quadrature points could be generated by randomly sampling points inside the domain or over the boundary as was demonstrated by us in our earlier work [2, 3]. In addition, it is best to ensure that all the quadrature nodes lie within the approximate polygon/polyhedron (Ω_0) as choosing points outside Ω_0 will lead to dealing with the discontinuity of an otherwise continuous integrand (as in characteristic func-

tion approach [52, 53, 54]) and thereby deteriorating SAQ's accuracy.

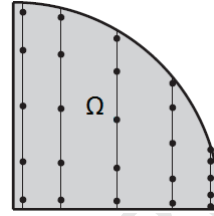


Figure 9: A 2D non-convex domain with numerous holes

Compute Moments The left hand side of Eq.(5) is a vector of m moments, one moment for each basis function. The SAQ evaluation procedure follows the development in section 3 and section 4, namely:

4. Determine the quadrature nodes and weights for each edge (2D) or face (3D) of the simplified domain Ω_0 in order to evaluate the moment approximations in Eq.(24). The correction factors in Eq.(24) can be obtained by using appropriate sensitivities from Table 1 and Table 2. It is important to note that these correction factors are usually a boundary integral of the form given by Eq.(16).

SSA Correction factor. The quadrature nodes and weights required to evaluate the SSA correction factors are not easy to determine accurately as SSA correction terms usually produces non-polynomial integrands (due to the presence of design velocity). However, one can arrive at a good estimate by looking at the curve/surface that the approximate edge/face approximates. If the edge/face approximates a surface of degree $2k - 1$ in the original boundary Γ , choosing the integration order to be $(o + k)$ (for example, $k = 2$ for conic sections and quadric surfaces) gives reasonably accurate results for SSA correction factors. The only requirement here is that k should be known *a priori* (or at least can be estimated *a posteriori*). A better method is to use well-known integrand adaptivity techniques for lines/triangles such as the one in [55, 56] (or using MATLAB's [57] *integral* and *quad2d* functions) although it is computationally a little expensive.

TSA Correction factor. For TSA correction factors, the sensitivities are relatively easy to

compute as it only involves the computation of the measure/size of the feature and sampling of the basis function (and/or its derivatives) at discrete points (see subsection 4.2). For simple features, such as a circle (or sphere) and ellipse (or ellipsoid), there are closed form expressions available for measure computations. For more general features, we use divergence theorem or first-order SSA to compute the measure ($\mu(\omega_l)$) approximately as a boundary integral. Thus, in this more general case, we choose an integration order of $k + 1$ to compute the boundary integrals arising in TSA corrections. Appropriate sign for the TSA correction factor can be determined by knowing if the feature is positive or negative. This can be easily accomplished in general by using the morphological opening and closing operations as discussed in the beginning of section 3. As already discussed in subsection 4.2, the size ϵ_l can be estimated as $\sqrt{\frac{\mu(\omega_l)}{\pi}}$ in 2D and $\sqrt[3]{\frac{3}{4\pi}\mu(\omega_l)}$ in 3D. If $\epsilon_l > \frac{\delta}{2}$, then it is best to break the small features into more manageable pieces for TSA computations. Alternatively, for such features, C^T could be computed using SSA without breaking it into smaller pieces.

5. Using the basis functions selected in step 1, evaluate the approximation to $\{\mathbf{M}\}$ over the original domain Ω from Eq.(24), by summing contribution over individual edges (2D) or faces (3D) of the approximate domain Ω_0 (using the quadrature nodes and weights determined in step 4).

Solve for weights and evaluate

6. Compute $[\mathbf{A}^\dagger]$ for the preselected set of basis functions (step 1) and quadrature nodes (step 2) using say QR factorization [44].
7. Using the $\{\mathbf{M}\}$ from Step 5 and $[\mathbf{A}^\dagger]$ from step 6, compute the approximate weights for the arbitrary domain Ω from $\{\mathbf{w}\} = [\mathbf{A}^\dagger]\{\mathbf{M}\}$ (Eq.(14)).
8. Finally, evaluate the approximation to the required integral by using these weights in Eq.(3) i.e. $\int_{\Omega} f(\mathbf{x})d\Omega \approx \sum_{k=1}^n w_k f(\mathbf{x}_k)$.

It should be clear that the same algorithm may be used in most situations, with steps 4 and 5 depending strongly on type, dimension, and representations of the geometric domain Ω and its approximation Ω_0 . Note that these two steps also dominate the computational complexity of the proposed approach.

6. Experimental Validation

The SAQ algorithm was implemented in MATLAB 7.10 (on a Intel Core i7 CPU with 2.8 GHz speed and 8 GB memory), which was also used for generating the reference solution via symbolic integration. In our earlier work [1, 2, 3], we have compared the efficacy of a particular class of SAQ corresponding to Fig. 7 (i.e. when the simplified domain (Ω) is homeomorphic to the original domain (Ω)) with several other standard integration techniques such as the

- Cartesian product or tensor-product rule scaled to fit inside the geometry of the cell as recommended in [42];
- Geometrically Adaptive integration method described in [51];
- quadrature rule obtained by solving the linear moment fitting equations (in least square sense) over a polygonized boundary (without correction) i.e. shape correction factors are obtained by neglecting the second term in Eq.(23).
- Characteristic function approach as described in [52, 53, 54]

for a variety of 2D/3D domains. We observed in [1, 2] that SAQ is computationally superior to all the above methods and often requires one or two quadtree/octree subdivisions to achieve a desired level of accuracy. We have also developed *a priori* and *a posteriori* estimates for this class of SAQ in [3]. Hence, in this paper, we will only study SAQ in its most general form when the simplified domain (Ω_0) is not homeomorphic to the original domain (Ω) (Fig. 6). Specifically, we will study the effect of applying SAQ to the leaf cells of quadtree/octree based integration of bivariate/trivariate polynomials over 2D/3D domains in the presence of numerous small circular/spherical holes.

6.1. Example 1 - Non-Convex domain with small holes

In this example, we integrate a cubic function $g = 10 + 0.1x + 0.4y - x^2 + 5xy + 2y^2 + 9x^3 - 10x^2y + 10xy^2 - 10y^3$ over the domain shown in Fig. 10 with 105 small holes of radius 0.025 ($\ll h = 0.25$). We fix the quadtree depth to one and integrate all interior cells using the usual box quadrature. For the leaf cells we generate quadrature points randomly such that they

lie completely inside the original (Ω) and simplified polygonal domain (Ω_0). We use first-order SSA for geometric corrections and first-order TSA/first-order SSA/divergence theorem for topological corrections [i.e. Eq.(20) - Eq.(22)]. Fig. 11 compares the integration of g over the non-convex domain using various correction factors by varying the number of edges of the approximating polygon (Ω_0) for the small features. From the plot it is clear that employing first-order TSA or SSA is superior to simply using divergence theorem (or zeroth order SSA) for topological corrections. In fact, first-order TSA only requires sampling of the basis function and area/volume computations which can be computed with very little effort for most simple shapes and hence is an attractive choice for C_i^T computations for small features. The relative error involved in ignoring all the small features is shown in Fig. 11 for the sake of comparison and is found to be as high as 16.201%.

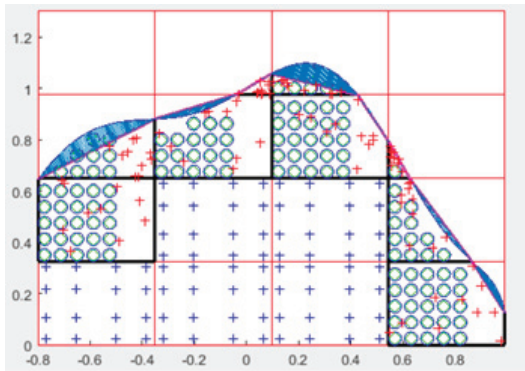


Figure 10: A 2D non-convex domain with numerous holes

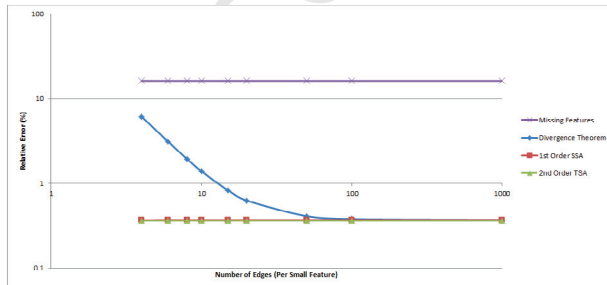


Figure 11: Non-Convex domain - Rel. Error Vs Number of Edges

6.2. Example 2 - Microstructures

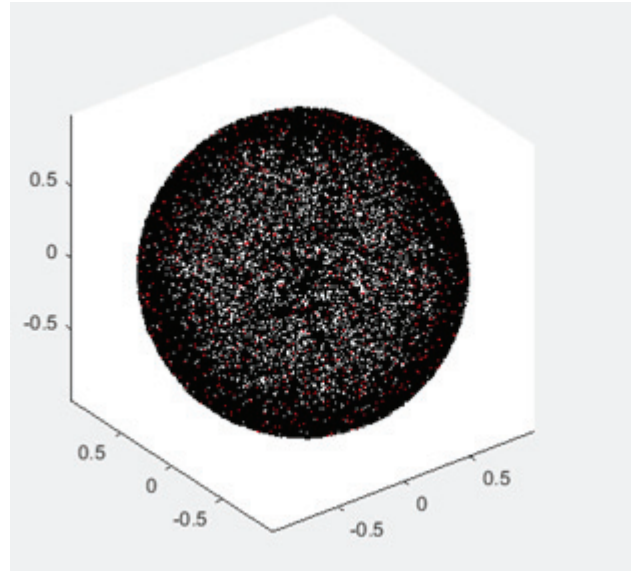


Figure 12: Unit sphere with 114,000 small spherical voids

In this example, we will consider the integral of a cubic function $h = -x^3 - y^3 - z^3 + 5x^2 + 6y^2 + 7z^2 + 8xy - 10xyz$ over a microstructure like domain using octree based integration. Specifically, we will assume the integration domain to be a unit sphere with several thousand small spherical voids as shown in Fig. 12. A typical leaf cell of the octree decomposition of this domain is shown in Fig. 13. As shown in the plot, the voids are generated randomly (both in size and position) over every leaf cell such that they don't overlap. Likewise, the quadrature points are generated randomly such that they lie within the domain. We use SAQ with first-order SSA based correction factor to account for the geometry of the unit sphere and first/second-order TSA to account for the spherical voids. The relative error in computing the integral using first-order TSA based correction, second-order TSA based correction, and ignoring the small features (for an octree of depth of one) are plotted in Fig. 14. It is clear from the plot that the first/second-order TSA predicts the integral accurately up to an error of 0.005 %. As expected, missing the features results in a relative error ranging from 0.013% to 0.264%.

Additionally, we also study the convergence of SAQ w.r.t. octree refinement by fixing the number of holes to be 114,000 and varying the depth of integration tree. From Fig. 15 we observe that SAQ converges

monotonically w.r.t. octree refinement. This example illustrates one of the important applications of SAQ that is to efficiently account for very small holes which is otherwise very difficult to account for in an octree based integration setting.

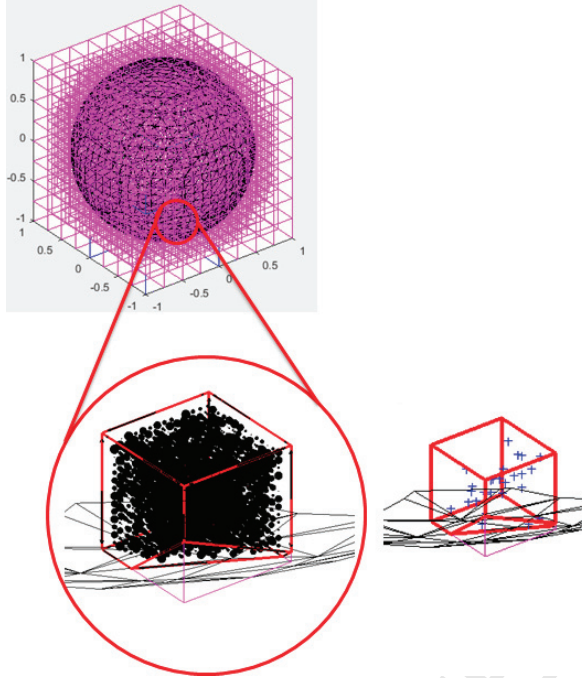


Figure 13: A typical leaf cell of the unit sphere contains several hundred voids (black dots denote holes); '+' marks denote the quadrature points in the cell (holes not shown for clarity)

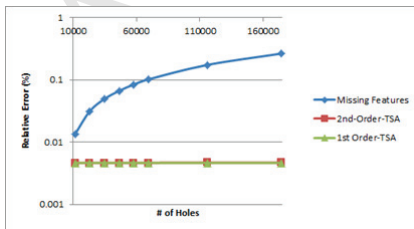


Figure 14: Error plot for various hole distribution

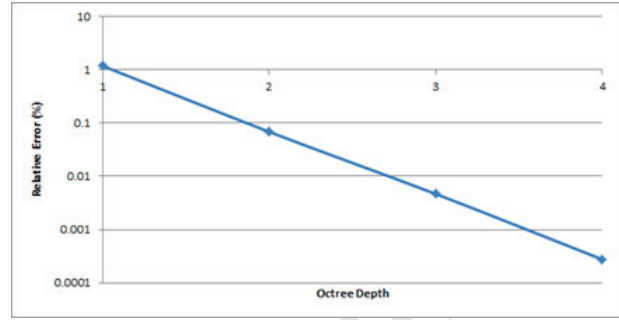


Figure 15: Convergence w.r.t. octree refinement

7. Conclusions and Open issues

We proposed a new integration technique called *Shape Aware Quadratures* (SAQ) using moment fitting equations [14, 15, 16], divergence theorem [22], and sensitivity analysis [8, 35, 12, 13] to efficiently integrate arbitrary integrable functions over arbitrary 2D/3D domains (Ω) even in the presence of small features. A simplified domain (Ω_0) was used to approximate the moments in moment fitting equations. Appropriate geometric and topological correction factors were used to correct for the shape deviation of Ω_0 from Ω . Shape correction factors were derived by the application of first/second-order Shape Sensitivity Analysis (SSA) [8] and Topological Sensitivity Analysis (TSA) [35] of the moment integrals. The use of shape correction factors in the moment fitting equations enabled the resulting quadrature rule “shape aware”.

We also note here that the method was formulated irrespectively of how the original domain Ω is represented, and it can be used with or without an integration mesh. SAQ offers a lot of flexibility in the choice of quadrature points and in the choice of basis functions as its formulation is based on the moment fitting equations. In fact, our experiments indicate that we can indeed generate quadrature points randomly either in the domain or on boundary as long as the moment matrix $[A]$ is invertible. The weights automatically adjusts itself depending on the position of quadrature points. A wide variety of basis functions can be chosen in forming the moment fitting equations depending on the type of integrand that arises in any given application.

The use of random quadrature points that lie both within the original (Ω) and approximate domain (Ω_0)

is found to give good results. However, it is important to more rigorously establish the effect of position of quadrature points on the moment matrix. Especially, it is important to study the effect of position of quadrature points on the conditioning and invertability of the moment matrix. Employing approximate Fekete points [27] as quadrature points could help eliminate this problem completely.

One of the main applications of SAQ is in immersed boundary methods such as the Finite Cell Method (FCM) as was demonstrated by us in our earlier work [2]. In such methods, Legendre, B-spline, and other types of polynomial basis functions are employed to study various field problems. Hence, it is more natural/useful to consider Legendre/Chebyshev/B-spline polynomial as basis function in constructing the moment fitting equations (for arbitrary 2D/3D domains) so as to improve the accuracy and relevance of SAQ to meshfree analysis.

The effect of other types of shape correction factors such as feature sensitivity [12] and modification sensitivity [13] and their comparison to TSA/SSA based correction factors requires further study. Also, application of SAQ to evolving level sets and non-linear problems is a straightforward and useful extension.

Acknowledgments

This research was supported by the National Science Foundation grants CMMI-1344205 and CMMI-1361862 and National Institute of Standards and Technology grants 70NANB14H248 and 70NANB14H232. The responsibility for errors and omissions lies solely with the authors.

References

- [1] V. Thiagarajan, V. Shapiro, Adaptively weighted numerical integration over arbitrary domains, *Computers and Mathematics with Applications* 67 (2014) 1682–1702.
- [2] V. Thiagarajan, V. Shapiro, Adaptively weighted numerical integration in the Finite Cell Method, *Computer Methods in Applied Mechanics and Engineering* 311 (2016) 250–279.
- [3] V. Thiagarajan, V. Shapiro, Error Analysis of Adaptively Weighted Numerical Integration, Tech. rep., Spatial Automation Laboratory (2017).
- [4] SOLIDWORKS, <http://www.solidworks.com/>, [Online; accessed 2016-06-07].
- [5] Schillinger, D and Ruess, M, 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 (2014) 391–455.
- [6] W. Gander, W. Gautschi, Adaptive quadrature—revisited, *BIT Numerical Mathematics* 40 (1) (2000) 84–101.
- [7] A. G. Requicha, Representations for rigid solids: Theory, methods, and systems, *ACM Comput. Surv.* 12 (4) (1980) 437–464.
- [8] K. Choi, N. Kim, *Structural Sensitivity Analysis and Optimization I: Linear Systems*, New York: Springer, 2005.
- [9] K. Choi, N. Kim, *Structural Sensitivity Analysis and Optimization II: Non-Linear Systems*, New York: Springer, 2005.
- [10] J. Sokolowski, A. Zochowski, On Topological Derivative in Shape Optimization, *SIAM Journal on Control and Optimization* 37 (1999) 1251–1272.
- [11] A. Novotny, R. A. Feijóo, E. Taroco, C. Padra, Topological-Shape Sensitivity Analysis, *Computer Methods in Applied Mechanics and Engineering* 192 (2003) 803–829.
- [12] S. Gopalakrishnan, K. Suresh, Feature sensitivity : A generalization of topological sensitivity, *Finite Elements in Analysis and Design* 44 (2008) 696–704.
- [13] L. Ming, G. Shuming, M. Ralph, Estimating effects of removing negative features on engineering analysis, *Computer-Aided Design* 43 (2011) 1402–1412.
- [14] J. Lyness, D. Jespersen, Moderate degree symmetric quadrature rules for the triangle, *IMA Journal of Applied Mathematics* 15 (1975) 19–32.

- [15] S. Wandzura, H. Xiao, Symmetric quadrature rules on a triangle, *Computers and Mathematics with Applications* 45 (2003) 1829–1840.
- [16] Z. Xiao, H. and Gimbutas, A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions, *Computers and Mathematics with Applications* 59 (2010) 663–676.
- [17] A. Sommariva, M. Vianello, Gauss–Green cubature and moment computation over arbitrary geometries, *Journal of Computational and Applied Mathematics* 231 (2009) 886–896.
- [18] Y. Lee, A. Requicha, Algorithms for computing the volume and other integral properties of solids. I. known methods and open issues, *Communications of the ACM* 25 (1982) 635–641.
- [19] F. Bernardini, Integration of polynomials over n-dimensional polyhedra, *Computer-Aided Design* 23 (1991) 51–58.
- [20] C. Cattani, A. Paoluzzi, Boundary integration over linear polyhedra, *Computer-Aided Design* 22 (1990) 130–135.
- [21] B. Mirtich, Fast and accurate computation of polyhedral mass properties, *Journal of Graphics Tools* 1 (1996) 31–50.
- [22] G. Dasgupta, Integration within polygonal finite elements, *J. Aerosp. Eng.* 16 (2003) 9–18.
- [23] A. Abedian, J. Parvizian, A. Düster, H. Khademzadeh, E. Rank, Performance of different integration schemes in facing discontinuities in the Finite Cell Method, *International Journal of Computational Methods* 10 (2013) 1–24.
- [24] R. Cools, P. Rabinowitz, Monomial cubature rules since stroud : a compilation, *J. Comput. Appl. Math.* 48 (1993) 309–326.
- [25] R. Cools, An encyclopaedia of cubature formulas, *Journal of Complexity* 19 (2003) 445–453.
- [26] A. Stroud, *Approximate Calculation of Multiple Integrals*, Prentice–Hall, Englewood Cliffs, NJ, 1971.
- [27] A. Sommariva, M. Vianello, Computing approximate Fekete points by QR factorizations of vandermonde matrices, *Computers and Mathematics with Applications* 57 (2009) 1324–1336.
- [28] S. Mousavi, N. Sukumar, Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons, *Computational Mechanics* 47 (2011) 535–554.
- [29] Mousavi S.E., Xiao H. and Sukumar N., Generalized Gaussian quadrature rules on arbitrary polygons, *International Journal for Numerical Methods in Engineering* 82 (2010) 99–113.
- [30] S. Mousavi, N. Sukumar, Generalized Gaussian quadrature rules for discontinuities and crack singularities in the Extended Finite Element Method, *Computer Methods in Applied Mechanics and Engineering* 199 (2010) 3237–3249.
- [31] J. Lasserre, Integration on a convex polytope, *Proceedings of the American Mathematical Society* 126 (1998) 2433–2441.
- [32] J. Lasserre, Integration and homogeneous functions, *Proceedings of the American Mathematical Society* 127 (1999) 813–818.
- [33] L. Ming, G. Shuming, Estimating defeaturing-induced engineering analysis errors for arbitrary 3d features, *Computer-Aided Design* 43 (2011) 1587–1597.
- [34] R. Feijoo, A. Novotny, E. Taroco, C. Padra, The topological derivative for poisson’s problem, *Mathematical Models and Methods in Applied Sciences* 13 (2003) 1925–1844.
- [35] R. Feijóo, A. A. Novotny, E. Taroco, C. Padra, The Topological Derivative For The Poisson’S Problem, *Math. Models Methods Appl. Sci.* 13 (2003) 1825–1844.
- [36] A. Novotny, R. A. Feijóo, E. Taroco, C. Padra, Topological Sensitivity Analysis for Three-Dimensional Linear Elasticity Problems, in: 6th World Congress on Structural and Multidisciplinary Optimization, 2005.
- [37] A. Novotny, R. A. Feijóo, E. Taroco, C. Padra, Topological-Shape Sensitivity Method : Theory and Applications, *Solid Mechanics and its Applications* 137 (2006) 469–478.
- [38] I. Turevsky, S. Gopalakrishnan, K. Suresh, Generalization of Topological Sensitivity and its Application to Defeaturing, in: *ASME-IDETC*, Las Vegas, 2007.

- [39] I. Turevsky, S. Gopalakrishnan, K. Suresh, De-featuring: A posteriori error analysis via feature sensitivity, *International Journal for Numerical Methods in Engineering* 76 (2008) 1379–1401.
- [40] L. Ming, G. Shuming, Z. Kai, A goal oriented error estimator for the analysis of simplified designs, *Computer Methods in Applied Mechanics and Engineering* 255 (2013) 89–103.
- [41] P. Soille, *Morphological Image Analysis: Principles and Applications*, 2nd Edition, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.
- [42] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992.
- [43] G. Strang, The Fundamental Theorem of Linear Algebra, *The American Mathematical Monthly* 100 (1993) 848–855.
- [44] L. Trefethen, D. Bau, *Numerical Linear Algebra*, SIAM, 1997.
- [45] J. Laguardia, E. Cueto, M. Doblaré, A natural neighbour Galerkin method with quadtree structure, *International Journal for Numerical methods in Engineering* 63 (2005) 789–812.
- [46] O. Klaas, M. Shephard, Automatic generation of octree-based three dimensional discretizations for partition of unity methods, *Computational Mechanics* 25 (2000) 296–304.
- [47] J. R. de Faria, A. Novotny, R. Feijóo, E. Taroco, C. Padra, Second order topological sensitivity analysis, *International Journal of Solids and Structures* 44 (14–15) (2007) 4958 – 4977.
- [48] M. Bonnet, Higher-order topological sensitivity for 2-D potential problems. Application to fast identification of inclusions, *International Journal of Solids and Structures* 46 (2009) 2275–2292.
- [49] N. Amenta, M. Bern, Surface reconstruction by Voronoi filtering, in: *Fourteenth annual symposium on Computational geometry*, 1998, pp. 39–48.
- [50] J. Ruppert, A delaunay refinement algorithm for quality 2-dimensional mesh generation, *Journal of Algorithms* 18 (3) (1995) 548 – 585.
- [51] B. Luft, V. Shapiro, I. Tsukanov, Geometrically adaptive numerical integration, in: *2008 ACM symposium on Solid and physical modeling*, NY, 2008, pp. 147–157.
- [52] R. Mittal, G. Iaccarino, Immersed boundary methods, *Annual Review of Fluid Mechanics* 37 (2005) 239–261.
- [53] P. J., D. A., R. E., Finite Cell Method : h- and p-extension for embedded domain methods in solid mechanics, *Computational Mechanics* 41 (2007) 121–133.
- [54] Y. Abdelaziz, A. Hamouine, A survey of the extended finite element, *Computers & Structures* 86 (2008) 1141–1151.
- [55] P. Gonnet, A Review of Error Estimation in Adaptive Quadrature, *ACM Computing Surveys* 44.
- [56] R. Cariño, I. Robinson, E. D. Doncker, Adaptive cubature over a collection of triangles using the d-transformation, *Journal of Computational and Applied Mathematics* 50 (1) (1994) 171 – 183.
- [57] MATLAB, version 7.10.0 (R2010a), The MathWorks Inc., Natick, Massachusetts, 2010.

Appendices

Appendix A. SSA of Moments

First-Order SSA of Moments. Recall that $M_i = \int_{\Omega} b_i(\mathbf{x}) d\Omega$. It is required to compute the shape sensitivity of the moments M_i i.e. $\left. \frac{dM_i}{dt} \right|_{t=0}$. In order to evaluate this, it is first required to transform the moment integral over the reference domain Ω_0 . This can be done by making use of the Jacobian matrix of the mapping $\mathbf{T}(\mathbf{X}, t)$. It can be easily established that $d\Omega = |J| d\Omega_0$ [8], where $|J|$ is the determinant of the Jacobian matrix. Thus, we have the following

$$\left. \frac{dM_i}{dt} \right|_{t=0} = \left. \frac{d}{dt} \left(\int_{\Omega_0} b_i(\mathbf{x}) |J| d\Omega_0 \right) \right|_{t=0} \quad (\text{A.1})$$

now, taking the derivative inside the integral and then applying the product and chain rule of differentiation

we get

$$\begin{aligned} \left. \frac{d}{dt} \left(\int_{\Omega_0} b_i(\mathbf{x}) |J| d\Omega_0 \right) \right|_{t=0} &= \int_{\Omega_0} \left. \frac{d}{dt} \left(b_i(\mathbf{x}) |J| \right) \right|_{t=0} d\Omega_0 \\ &= \int_{\Omega_0} \left(\nabla b_i(\mathbf{x}) \cdot \hat{\mathbf{V}}(\mathbf{x}) |J| + b_i(\mathbf{x}) \frac{d|J|}{dt} \right) d\Omega_0 \Big|_{t=0} \end{aligned} \quad (\text{A.2})$$

where $\hat{\mathbf{V}}(\mathbf{x}) = \frac{d\mathbf{x}}{dt}$ is the design velocity vector at any point $\mathbf{x} \in \Omega$. Now, using the fact that $\frac{d|J|}{dt} = |J| \nabla \cdot \hat{\mathbf{V}}(\mathbf{x})$ [8] in Eq.(A.2), we get

$$\left. \frac{d}{dt} \left(\int_{\Omega_0} b_i(\mathbf{x}) |J| d\Omega_0 \right) \right|_{t=0} = \int_{\Omega_0} \nabla \cdot (b_i(\mathbf{x}) \hat{\mathbf{V}}(\mathbf{x})) |J| d\Omega_0 \Big|_{t=0}$$

evaluating the RHS of the above at $t = 0$, we obtain the desired sensitivity as

$$\left. \frac{dM_i}{dt} \right|_{t=0} = \int_{\Omega_0} \nabla \cdot (b_i(\mathbf{X}) \hat{\mathbf{V}}(\mathbf{X})) d\Omega_0 \quad (\text{A.3})$$

where $\mathbf{X} \in \Omega_0$.

Applying the divergence theorem we obtain the following boundary form

$$\left. \frac{dM_i}{dt} \right|_{t=0} = \int_{\Gamma_0} b_i(\mathbf{X}) V_N(\mathbf{X}) d\Gamma_0 \quad (\text{A.4})$$

where $V_N = \hat{\mathbf{V}}(\mathbf{X}) \cdot \mathbf{N}(\mathbf{X})$ is the normal component of the design velocity at $\mathbf{X} \in \Gamma_0$.

Second-order SSA of Moments. Now, It is required to compute the second-order shape sensitivity of the moments M_i i.e. $\left. \frac{d^2 M_i}{dt^2} \right|_{t=0}$. This can be done by taking the first derivative of Eq.(A.4) as

$$\frac{d^2 M_i}{dt^2} = \frac{d}{dt} \int_{\Omega_0} \left(\nabla \cdot (b_i(\mathbf{x}) \hat{\mathbf{V}}(\mathbf{x})) |J| d\Omega_0 \right) \quad (\text{A.5})$$

taking the derivative inside the integral and then applying the product and chain rule of differentiation

we get:

$$\begin{aligned} \int_{\Omega_0} \frac{d}{dt} \left(\nabla \cdot (b_i(\mathbf{x}) \hat{\mathbf{V}}(\mathbf{x})) |J| \right) d\Omega_0 &= \int_{\Omega_0} \frac{d}{dt} \left(\nabla \cdot (b_i(\mathbf{x}) \hat{\mathbf{V}}(\mathbf{x})) \right) |J| \\ &\quad + \nabla \cdot (b_i(\mathbf{x}) \hat{\mathbf{V}}(\mathbf{x})) \frac{d|J|}{dt} d\Omega_0 \end{aligned} \quad (\text{A.6})$$

However,

$$\begin{aligned} \frac{d}{dt} \left(\nabla \cdot (b_i(\mathbf{x}) \hat{\mathbf{V}}(\mathbf{x})) \right) &= \frac{\partial}{\partial t} (\nabla \cdot (b_i(\mathbf{x}) \hat{\mathbf{V}}(\mathbf{x}))) \\ &\quad + \nabla \cdot (\nabla \cdot (b_i(\mathbf{x}) \hat{\mathbf{V}}(\mathbf{x}))) \cdot \hat{\mathbf{V}}(\mathbf{x}) \\ &= \nabla \cdot (b_i(\mathbf{x}) \frac{\partial}{\partial t} \hat{\mathbf{V}}(\mathbf{x})) \\ &\quad + \nabla \cdot (\nabla \cdot (b_i \hat{\mathbf{V}}(\mathbf{x}))) \cdot \hat{\mathbf{V}}(\mathbf{x}) \end{aligned} \quad (\text{A.7})$$

Since we assume a linear mapping, we have the following relationship for the total time derivative of the design velocity:

$$\frac{d}{dt} \hat{\mathbf{V}}(\mathbf{x}) = \frac{\partial}{\partial t} \hat{\mathbf{V}}(\mathbf{x}) + \nabla \hat{\mathbf{V}}(\mathbf{x}) \cdot \hat{\mathbf{V}}(\mathbf{x}) = 0 \quad (\text{A.8})$$

Thus, $\frac{\partial}{\partial t} \hat{\mathbf{V}}(\mathbf{x}) = -\nabla \hat{\mathbf{V}}(\mathbf{x}) \cdot \hat{\mathbf{V}}(\mathbf{x})$. Using this and $\frac{d|J|}{dt} = |J| \nabla \cdot \hat{\mathbf{V}}(\mathbf{x})$ [8] in Eq.(A.6) and subsequently substituting the result in Eq.(A.5), the second derivative reduces to:

$$\begin{aligned} \frac{d^2 M_i}{dt^2} &= \int_{\Omega_0} \left(-\nabla \cdot \left[b_i(\mathbf{x}) \nabla \hat{\mathbf{V}}(\mathbf{x}) \cdot \hat{\mathbf{V}}(\mathbf{x}) \right] \right. \\ &\quad + \nabla \cdot (\nabla \cdot (b_i(\mathbf{x}) \hat{\mathbf{V}}(\mathbf{x}))) \cdot \hat{\mathbf{V}}(\mathbf{x}) \\ &\quad + \nabla \cdot (b_i(\mathbf{x}) \hat{\mathbf{V}}(\mathbf{x})) \nabla \cdot (\hat{\mathbf{V}}(\mathbf{x})) \Big) |J| d\Omega_0 \\ &= \int_{\Omega_0} \left(\nabla \cdot \left[\nabla \cdot (b_i(\mathbf{x}) \hat{\mathbf{V}}(\mathbf{x})) \hat{\mathbf{V}}(\mathbf{x}) \right] \right. \\ &\quad \left. - \nabla \cdot \left[b_i(\mathbf{x}) \nabla \hat{\mathbf{V}}(\mathbf{x}) \cdot \hat{\mathbf{V}}(\mathbf{x}) \right] \right) |J| d\Omega_0 \end{aligned} \quad (\text{A.9})$$

evaluating the above at $t = 0$, we obtain the second-order shape sensitivity as:

$$\begin{aligned} \left. \frac{d^2 M_i}{dt^2} \right|_{t=0} &= \int_{\Omega_0} \left(\nabla \cdot \left[\nabla \cdot (b_i(\mathbf{X}) \hat{\mathbf{V}}(\mathbf{X})) \hat{\mathbf{V}}(\mathbf{X}) \right] \right. \\ &\quad \left. - \nabla \cdot \left[b_i(\mathbf{X}) \nabla \hat{\mathbf{V}}(\mathbf{X}) \cdot \hat{\mathbf{V}}(\mathbf{X}) \right] d\Omega_0 \right) \end{aligned} \quad (\text{A.10})$$

where $\mathbf{X} \in \Omega_0$.

Applying the divergence theorem we get the boundary form of the second order shape sensitivity as:

$$\begin{aligned} \left. \frac{d^2 M_i}{dt^2} \right|_{t=0} &= \int_{\Gamma_0} \left[\nabla \cdot [b_i(\mathbf{X}) \hat{\mathbf{V}}(\mathbf{X})] V_N(\mathbf{X}) \right. \\ &\quad \left. - b_i(\mathbf{X}) \left(\nabla \hat{\mathbf{V}}(\mathbf{X}) \cdot \hat{\mathbf{V}}(\mathbf{X}) \right) \cdot \mathbf{N}(\mathbf{X}) \right] d\Gamma_0 \\ &= \int_{\Gamma_0} \left[\nabla [b_i(\mathbf{X})] \hat{\mathbf{V}}(\mathbf{X}) \right. \\ &\quad + b_i(\mathbf{X}) \nabla \cdot (\hat{\mathbf{V}}(\mathbf{X})) \\ &\quad \left. - b_i(\mathbf{X}) \left(\mathbf{N}^T \nabla \hat{\mathbf{V}} \mathbf{N} \right) \right] V_N(\mathbf{X}) d\Gamma_0 \\ &= \int_{\Gamma_0} \left[\nabla [b_i(\mathbf{X})] \cdot \hat{\mathbf{V}}(\mathbf{X}) \right. \\ &\quad + b_i(\mathbf{X}) \left(\nabla \cdot (\hat{\mathbf{V}}(\mathbf{X})) \right. \\ &\quad \left. \left. - \mathbf{N}^T \nabla \hat{\mathbf{V}} \mathbf{N} \right) \right] V_N(\mathbf{X}) d\Gamma_0 \end{aligned} \quad (\text{A.11})$$

It can be easily proved [8] that $[\nabla \cdot (\mathbf{V}) - \mathbf{N}^T \nabla \mathbf{V} \mathbf{N}] = V_N(\mathbf{X}) \kappa(\mathbf{X})$, where κ is the curvature of Γ_0 in 2D and twice the mean curvature in 3D.

Thus,

$$\left. \frac{d^2 M_i}{dt^2} \right|_{t=0} = \int_{\Gamma_0} [\nabla b_i(\mathbf{X})]^T \mathbf{N} + b_i(\mathbf{X}) \kappa(\mathbf{X}) V_N^2(\mathbf{X}) d\Gamma_0 \quad (\text{A.12})$$

If we assume that the vertices/edges of the approximate polygon/polyhedron Ω_0 all lie on Γ as shown in Fig. 16, then $V_N = 0$ at the vertices/edges of Ω_0 . Also, $\kappa = 0$ in the interior of the edges/faces of the Γ_0 . Hence, we can easily prove that $\int_{\Gamma_0} \kappa(\mathbf{X}) V_N^2(\mathbf{X}) d\Gamma_0 = 0$ as the integrand $[\kappa(\mathbf{X}) V_N^2(\mathbf{X})]$ is identically zero. Hence, the above reduces to the following:

$$\left. \frac{dM_i^2}{dt^2} \right|_{t=0} = \int_{\Gamma_0} [\nabla b_i(\mathbf{X})]^T \mathbf{N} V_N^2(\mathbf{X}) d\Gamma_0 \quad (\text{A.13})$$

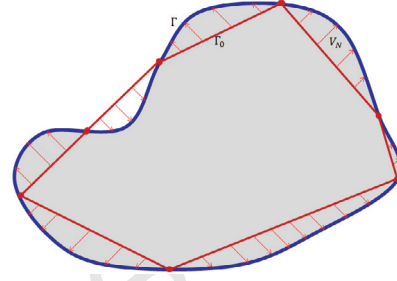


Figure 16: Vertices (black dots) of Γ_0 lying on Γ makes $V_N = 0$ at the vertices

Appendix B. TSA of Moments

First-order TSA of Moments. First-order topological sensitivity of moments at a point \mathbf{X}_0 can be established using the Topological-Shape Sensitivity definition as defined by Novotny et al. [11] as

$$D_{T_i}^1(\mathbf{X}_0) = \lim_{\epsilon \rightarrow 0} \frac{1}{f_1'(\epsilon) |V_N|} \left. \frac{dM_i}{dt} \right|_{t=0} \quad (\text{B.1})$$

From Eq.(A.4) we have already established the first-order SSA as

$$\left. \frac{dM_i}{dt} \right|_{t=0} = \int_{\Gamma_0} b_i(\mathbf{X}) V_N(\mathbf{X}) d\Gamma_0 \quad (\text{B.2})$$

where Γ_0 is now the circular boundary B_ϵ as shown in Fig. 17. Thus, we have

$$\left. \frac{dM_i}{dt} \right|_{t=0} = \int_{\partial B_\epsilon} b_i(\mathbf{X}) V_N(\mathbf{X}) dB_\epsilon \quad (\text{B.3})$$

Using Eq.(B.3) in Eq.(B.1), we get

$$D_T^1(\mathbf{X}_0) = \lim_{\epsilon \rightarrow 0} \frac{1}{f_1'(\epsilon) |V_N|} \int_{\partial B_\epsilon} b_i(\mathbf{X}) V_N(\mathbf{X}) dB_\epsilon \quad (\text{B.4})$$

Since V_N is a constant over the boundary of the ball

(∂B_ϵ) we have

$$\begin{aligned} D_T^1(\mathbf{X}_0) &= \lim_{\epsilon \rightarrow 0} \frac{V_N}{f_1'(\epsilon)|V_N|} \int_{\partial B_\epsilon} b_i(\mathbf{X}) dB_\epsilon \\ &= \lim_{\epsilon \rightarrow 0} \frac{\text{sign}(V_N)}{f_1'(\epsilon)} \int_{\partial B_\epsilon} b_i(\mathbf{X}) dB_\epsilon \\ &= \lim_{\epsilon \rightarrow 0} \frac{-1}{f_1'(\epsilon)} \int_{\partial B_\epsilon} b_i(\mathbf{X}) dB_\epsilon \quad (\text{B.5}) \end{aligned}$$

For 2D problems, $f_1(\epsilon) = \pi\epsilon^2$ and hence $f_1'(\epsilon) = 2\pi\epsilon$. If we assume bivariate basis functions of the form $b_i = X^p Y^q$ and then write out the above integral in polar coordinates we get

$$\begin{aligned} D_T^1(\mathbf{X}_0) &= \lim_{\epsilon \rightarrow 0} \frac{-1}{2\pi\epsilon} \int_0^{2\pi} (X_0 + \epsilon \cos(\theta))^p \\ &\quad (Y_0 + \epsilon \sin(\theta))^q \epsilon d\theta \\ &= -X_0^p Y_0^q \quad (\text{B.6}) \end{aligned}$$

where we have used the binomial expansion and applied the limit. Thus, in 2D the first-order TSA is given by

$$D_T^1(\mathbf{X}_0) = -X_0^p Y_0^q \quad \forall \mathbf{X}_0 \in \Omega \subset \mathbb{R}^2 \text{ with } f_1(\epsilon) = \pi\epsilon^2 \quad (\text{B.7})$$

Likewise in 3D, for trivariate polynomials of the form $b_i = X^p Y^q Z^r$ and $f_1(\epsilon) = \frac{4}{3}\pi\epsilon^3$, one can easily establish the first-order topological derivative to be

$$D_T^1(\mathbf{X}_0) = -X_0^p Y_0^q Z_0^r \quad \forall \mathbf{X}_0 \in \Omega \subset \mathbb{R}^3 \text{ with } f_1(\epsilon) = \frac{4}{3}\pi\epsilon^3 \quad (\text{B.8})$$

by using the fact that $X = X_0 + \epsilon \cos(\phi) \sin(\theta)$, $Y = Y_0 + \epsilon \sin(\phi) \sin(\theta)$ and $Z = Z_0 + \epsilon \cos(\theta)$ where $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi]$.

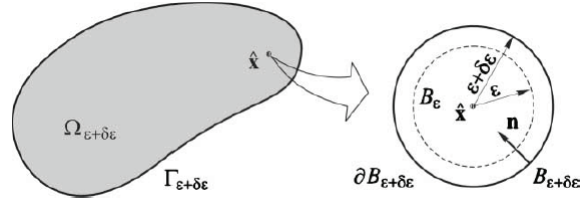


Figure 17: Reference and deformed domains for Topological-Shape Sensitivity Analysis [11]

Second-order TSA of Moments. The second-order TSA of moments can be defined using the Topological-Shape sensitivity method of [47] as

$$\begin{aligned} D_T^2(\mathbf{X}_0) &= \lim_{\epsilon \rightarrow 0} \frac{1}{f_2'(\epsilon)} \left[\frac{1}{|V_N|} \frac{dM_i}{dt} \right]_{t=0} \\ &\quad - f_1'(\epsilon) D_T^1(\mathbf{X}_0) \\ &= \lim_{\epsilon \rightarrow 0} \frac{1}{f_2'(\epsilon)} \left[\frac{1}{|V_N|} \int_{\partial B_\epsilon} b_i(\mathbf{X}) V_N(\mathbf{X}) dB_\epsilon \right. \\ &\quad \left. - f_1'(\epsilon) D_T^1(\mathbf{X}_0) \right] \\ &= \lim_{\epsilon \rightarrow 0} \frac{1}{f_2'(\epsilon)} \left[\text{sign}(V_N) \int_{\partial B_\epsilon} b_i(\mathbf{X}) dB_\epsilon \right. \\ &\quad \left. - f_1'(\epsilon) D_T^1(\mathbf{X}_0) \right] \\ &= \lim_{\epsilon \rightarrow 0} \frac{1}{f_2'(\epsilon)} \left[- \int_{\partial B_\epsilon} b_i(\mathbf{X}) dB_\epsilon \right. \\ &\quad \left. - f_1'(\epsilon) D_T^1(\mathbf{X}_0) \right] \quad (\text{B.9}) \end{aligned}$$

For 2D domains, if we choose the bivariate polynomials $b_i = X^p Y^q$ then $f_1'(\epsilon) D_T^1(\mathbf{X}_0) = -2\pi\epsilon X_0^p Y_0^q$ and $f_2(\epsilon) = \frac{1}{2}\pi^2\epsilon^4$ (this is chosen such that $f_1(\epsilon) \rightarrow 0$, $f_2(\epsilon) \rightarrow 0$ and $\frac{f_2(\epsilon)}{f_1(\epsilon)} \rightarrow 0$ as $\epsilon \rightarrow 0$). Therefore, Eq.(B.9) becomes

$$\begin{aligned}
 D_T^2(\mathbf{X}_0) &= \lim_{\epsilon \rightarrow 0} \frac{1}{2\pi^2\epsilon^3} \left[\right. \\
 &\quad - \int_0^{2\pi} (X_0 + \epsilon \cos(\theta))^p (Y_0 \\
 &\quad + \epsilon \sin(\theta))^q \epsilon d\theta + 2\pi\epsilon X_0^p Y_0^q \left. \right] \\
 &= \lim_{\epsilon \rightarrow 0} \frac{1}{2\pi^2\epsilon^2} \left[\right. \\
 &\quad - \int_0^{2\pi} (X_0 + \epsilon \cos(\theta))^p (Y_0 + \epsilon \sin(\theta))^q d\theta \\
 &\quad + 2\pi X_0^p Y_0^q \left. \right]
 \end{aligned}$$

Applying binomial theorem to the above and simplifying we get the second-order TSA in 2D as

$$D_T^2(\mathbf{X}_0) = \frac{-1}{2\pi} \left[\binom{p}{2} Y_0^q X_0^{p-2} + \binom{q}{2} X_0^p Y_0^{q-2} \right] \quad (\text{B.10})$$

The above equation is valid for all $\mathbf{X}_0 \in \Omega \subset \mathbb{R}^2$ with $f_2(\epsilon) = \frac{1}{2}\pi^2\epsilon^4$. For 3D domains if we choose the trivariate polynomials $b_i = X^p Y^q Z^r$ then $f_1'(\epsilon) D_T^1(\mathbf{X}_0) = -4\pi\epsilon^2 X_0^p Y_0^q Z^r$ and $f_2(\epsilon) = \frac{1}{2\epsilon} [\frac{4}{3}\pi\epsilon^3]^2$. Then, using Eq.(B.9) and applying binomial expansion and taking the limit we get the second-order TSA in 3D to be

$$\begin{aligned}
 D_T^2(\mathbf{X}_0) &= -\frac{1}{4\pi} \left[\binom{p}{2} X_0^{p-2} Y_0^q Z_0^r \right. \\
 &\quad + \binom{q}{2} X_0^p Y_0^{q-2} Z_0^r \\
 &\quad + \left. \binom{r}{2} X_0^p Y_0^q Z_0^{r-2} \right] \quad (\text{B.11})
 \end{aligned}$$

The above equation is valid for all $\mathbf{X}_0 \in \Omega \subset \mathbb{R}^3$ with $f_2(\epsilon) = \frac{8}{9}\pi^2\epsilon^5$