

Integration of singular enrichment functions in the generalized/extended finite element method for three-dimensional problems

Kyoungsoo Park, Jeronimo P. Pereira, C. Armando Duarte^{*,†} and Glaucio H. Paulino

Newmark Laboratory, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 North Mathews Ave., Urbana, IL 61801, U.S.A.

SUMMARY

A mapping method is developed to integrate weak singularities, which result from enrichment functions in the generalized/extended finite element method. The integration scheme is applicable to 2D and 3D problems including arbitrarily shaped triangles and tetrahedra. Implementation of the proposed scheme in existing codes is straightforward. Numerical examples for 2D and 3D problems demonstrate the accuracy and convergence properties of the technique. Copyright © 2008 John Wiley & Sons, Ltd.

Received 12 August 2008; Revised 30 October 2008; Accepted 17 November 2008

KEY WORDS: GFEM/X-FEM; BEM; weakly singular integration; mapping method; branch functions; generalized FEM; extended FEM; boundary element method; fracture; singular enrichment functions

1. INTRODUCTION

Physical phenomena can be simulated by several numerical methods, such as finite element methods (FEMs) and boundary element methods (BEMs). These numerical methods are developed on the basis of governing differential equations. For instance, weak formulations in the FEM are obtained from governing differential equations, and are approximated by using shape functions. In the standard FEM, shape functions are usually based on polynomials. FEM shape functions can be

*Correspondence to: C. Armando Duarte, Newmark Laboratory, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 North Mathews Ave., Urbana, IL 61801, U.S.A.

†E-mail: caduarte@uiuc.edu

Contract/grant sponsor: Department of Civil and Environmental Engineering (CEE)

Contract/grant sponsor: Computational Science and Engineering (CSE) program at the University of Illinois at Urbana-Champaign (UIUC)

Contract/grant sponsor: National Science Foundation; contract/grant number: DMS-0611094

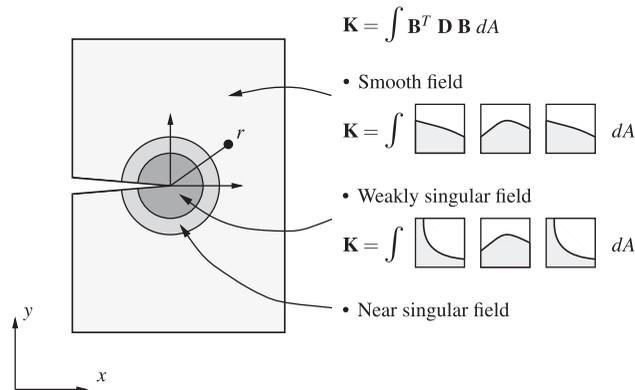


Figure 1. Example of singular behavior in linear elastic fracture mechanics problems. The notations are the following: \mathbf{K} denotes the element stiffness matrix; \mathbf{B} is the strain–displacement matrix; \mathbf{D} indicates the constitutive matrix; A is the element area.

generalized/extended by introducing custom-built enrichment functions using the partition of unity methodology [1–3]. Accordingly, the resulting method is named either generalized FEM (GFEM) [4, 5] or extended FEM (X-FEM) [6–8] in the technical literature. In the BEM, boundary integral equations are formulated using singular Green’s functions to solve the physical problems of interest [9–11].

In the standard FEM, numerical integrations are relatively easy compared with the BEM, because the integrands usually consist of smooth functions such as polynomials while the BEM in general involves singular Kernels. The standard Gaussian integration, therefore, provides accurate results for given polynomial orders.

When non-polynomial enrichment functions, such as Heaviside functions, singular functions, or trigonometric functions, are used in the GFEM/X-FEM, numerical integration should be performed with care. For example, in linear elastic fracture problems, polynomial functions can be used for the smooth field, while singular functions can be introduced for the weakly singular field to capture the crack tip behavior, as described in Figure 1. On the basis of the Westergaard stress functions, branch functions are utilized as enrichment functions [12, 13]. Owing to the branch functions, the strain–displacement matrix (\mathbf{B}) contains the $1/\sqrt{r}$ singularity, and therefore, some entries in the integrand of the element stiffness matrix (\mathbf{K}) have the $1/r$ singularity. To obtain accurate numerical integration of the element stiffness matrix, many integration points are necessary when standard quadrature rules for polynomials are employed, which lead to the increase of computational cost [14–16]. Recently, Bechet *et al.* [14] utilized the polar transformation technique to integrate singular functions for 2D problems. Laborde *et al.* [15] also developed a new integration rule, called the ‘almost polar integration’, for 2D problems. Xiao and Karihaloo [17] utilized the adaptive integration algorithm (DECUHR [18]) with higher-order Gauss-Legendre quadrature for 2D problems. They stated that ‘an adaptive control of the integration error using DECUHR is very time-consuming’ (p. 1381) and that ‘a very high-order quadrature is required for all elements enriched with crack tip branch functions’ (p. 1404). If the enrichment functions satisfy the equilibrium equations, the domain integral used to compute \mathbf{K} can be transformed into equivalent

boundary integrals. Ventura *et al.* [19] have shown that this form is computationally more efficient than the domain form when the enrichment functions are weakly singular or discontinuous.

In the BEM, numerical integrations are an important issue because of inherent singularities. The singularities in the integrand make computational simulations much less accurate if one uses standard quadrature rules. Whenever elements contain a source (e.g. loading) point, the integrand becomes singular for $r \rightarrow 0$, where r is the distance from a source point. When the integral of a singular function exists, and is continuous at the singularity, we name the integrand as weakly singular [9, 20, 21]. For example, at $x=0$, the $\ln|x|$ function is singular, but the integrand $\int \ln|x| dx = x \ln|x| - x + c$, where c is a constant, is continuous, as verified by applying the rule of L'Hospital: $\lim_{x \rightarrow 0} x \ln|x| = \lim_{x \rightarrow 0} \ln|x|/(1/x) = \lim_{x \rightarrow 0} (1/x)/(-1/x^2) = 0$. Weakly singular techniques for 2D problems are used to integrate the $\ln r$ singularity while weakly singular techniques for 3D problems are used to integrate the $1/r$ singularity. A review of general singular integration techniques can be found in References [9, 11]. Moreover, the so-called singular and hypersingular boundary integrals need to be evaluated for the symmetric Galerkin boundary element method (SGBEM) [11, 22], and the interested reader is referred elsewhere for further details [23, 24]. In order to reduce errors from numerical integration of singularities, several singular integration techniques have been developed in conjunction with the BEM [9, 11, 25]. For the weak singularity in 3D BEM, transformation techniques have been widely utilized so that the Jacobian cancels the singularity. The polar transformation can be utilized to integrate the singularity exactly. However, it results in curved integration intervals, which leads to an additional transformation to the reference square domain. This can result in higher computation (especially for nonlinear problems), and accumulated round off error [26]. The Lachat-Watson transformation was developed on the basis of special solutions of boundary integral equations [27]. Nagarajan and Mukherjee [26] developed a mapping method to evaluate integrals of $O(1/r)$, over two-dimensional triangular (curved or straight) domains. These singular integral techniques are developed only for a 2D domain because the integration of 3D domains is not necessary for the BEM [11]. In other words, the BEM reduces the dimension of the problem by one—thus 3D problems are treated by means of 2D integrations, while 2D problems are treated by means of 1D integrations.

Integration of singular functions in 3D has been an open issue in the GFEM/X-FEM literature. In recent papers published in 2005 in this journal, Laborde *et al.* [15] indicated that extension of their work to 'tridimensional problems is an open question, since the singularities are difficult to express' (p. 380). Similarly, Bechet *et al.* [14] remarked that they 'do not have singular integration scheme yet available for arbitrary 3D meshes' (p. 1053). The present paper addresses such issue by means of a tailored mapping method to handle 3D singular integration, which is applicable to arbitrarily shaped (i.e. straight edge or curved edge) tetrahedra elements. The present mapping method is also used to integrate 2D domains. Our GFEM/X-FEM mapping method is an extension of the BEM work by Nagarajan and Mukherjee [26]. The proposed mapping method leads to convergence of strain energy and stress intensity factors for fracture problems, while requiring less integration points (to obtain accurate results) than standard quadrature rules. For comparison purposes, convergence analysis for fracture problems are performed using both the proposed mapping method and standard quadratures (e.g. Keast [28]). For the same level of accuracy, the proposed mapping method requires considerably less number of integration points than standard quadrature rules. Moreover, its implementation in existing codes is straightforward.

The present paper is organized as follows. First, convergence of different types of branch functions used in the GFEM/X-FEM is investigated. The mapping method for both 2D and 3D singular integrations is presented in Section 3. Next, the performance of the mapping method is compared with that of standard quadrature rules through theoretical and engineering examples. Finally, Section 5 concludes the paper. Two appendices supplement the text. Appendix A provides a MATLAB code to generate integration points and weights for singular integration in 2D, which can be readily extended to 3D. Appendix B provides quadrature rules for both 2D and 3D.

2. GFEM/X-FEM FOR FRACTURE MECHANICS

In computational simulations of linear elastic fracture problems, the mesh quality is an important issue because of the singularity at a crack tip. When we utilize the standard FEM, not only the crack geometry should correspond to the finite element mesh, but the mesh should also be refined around a crack tip in order to obtain accurate results. In addition to these techniques, the GFEM/X-FEM provides extra flexibility in generating finite element meshes and improving the solution quality. These flexibilities and capabilities are achieved through the concept of a partition of unity in conjunction with local enrichment functions. Details of the GFEM/X-FEM for this class of problems are provided elsewhere [4, 6–8, 29, 30]. In this section, we focus only on the enrichment functions and their singular behavior.

2.1. Enrichment functions

In the standard FEM, a crack is described by double nodes along crack surfaces. Thus, finite element meshes should correspond to crack surfaces. In the GFEM/X-FEM, a crack can be represented by a discontinuous function, and therefore, finite element meshes do not need to coincide with the geometry of crack surfaces. A node whose support (or cloud) has a crack is enriched by a discontinuous (e.g. Heaviside) function. The enrichment cases of the Heaviside function with respect to the crack geometry are shown in Figure 2.

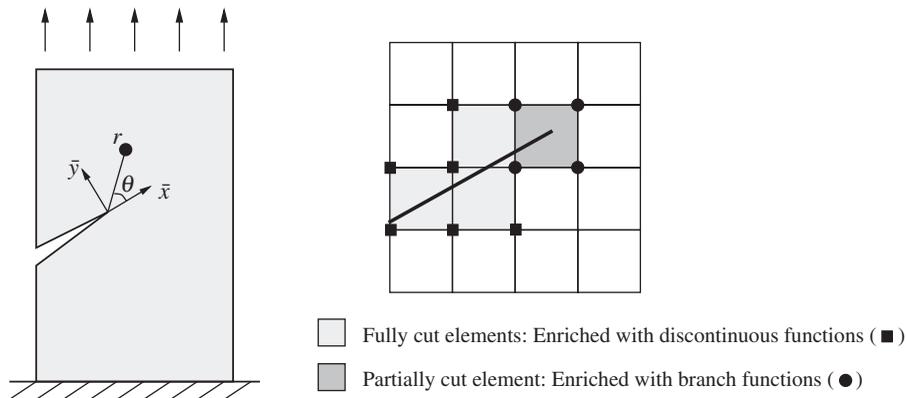


Figure 2. Enrichment cases with respect to the crack geometry.

However, the Heaviside function is unable to represent a crack that partially cuts an element (e.g. dark gray element in Figure 2). In this case, one can utilize branch functions to enrich a node whose support (or cloud) has a crack tip. Since branch functions are based on the Westergaard stress functions, the branch functions are able to represent singular stress behavior, and to provide accurate computational results. Oden and Duarte [12] introduced branch functions (these functions are hereafter denoted **OD**),

$$\begin{aligned} \mathbf{L}_{\text{tip}-\bar{x}} &= \left\{ \sqrt{r} \left[\left(\kappa - \frac{1}{2} \right) \cos \frac{\theta}{2} - \frac{1}{2} \cos \frac{3\theta}{2} \right], \sqrt{r} \left[\left(\kappa + \frac{3}{2} \right) \sin \frac{\theta}{2} + \frac{1}{2} \sin \frac{3\theta}{2} \right] \right\} \\ \mathbf{L}_{\text{tip}-\bar{y}} &= \left\{ \sqrt{r} \left[\left(\kappa + \frac{1}{2} \right) \sin \frac{\theta}{2} - \frac{1}{2} \sin \frac{3\theta}{2} \right], \sqrt{r} \left[\left(\kappa - \frac{3}{2} \right) \cos \frac{\theta}{2} + \frac{1}{2} \cos \frac{3\theta}{2} \right] \right\} \end{aligned} \quad (1)$$

where r and θ are polar coordinates at the crack tip (see Figure 2), κ is a material constant ($3-4\nu$), and ν is the Poisson ratio. The $\mathbf{L}_{\text{tip}-\bar{x}}$, $\mathbf{L}_{\text{tip}-\bar{y}}$ branch functions are used as enrichments around a crack tip along the local \bar{x} and \bar{y} direction, respectively. Enrichments for 3D problems are available in References [4, 30, 31]. Belytschko and Black [13] utilized branch functions (these functions are hereafter denoted **BB**),

$$\mathbf{L}_{\text{tip}} = \left\{ \sqrt{r} \sin \frac{\theta}{2}, \sqrt{r} \cos \frac{\theta}{2}, \sqrt{r} \sin \frac{\theta}{2} \sin \theta, \sqrt{r} \cos \frac{\theta}{2} \sin \theta \right\} \quad (2)$$

which are used as enrichments around a crack tip for both the \bar{x} and \bar{y} directions. The same branch functions are utilized for 3D fracture problems [8]. In the case of 2D problems, the **OD** branch functions introduce four additional local enrichment functions per node, while the **BB** branch functions generate eight additional local enrichment functions per node.

2.2. Comparison between **OD** and **BB** branch functions

The **OD** and **BB** branch functions lead to singular entries in the integrand of the element stiffness matrix, and thus are difficult to integrate with standard quadrature rules. This section compares the performance of these two family of functions, while Section 3 presents a mapping method for the numerical integration of these functions for 2D and 3D problems. In order to compare the **OD** and **BB** branch functions, a single edge notched tension test is investigated, as shown in Figure 3. The elastic modulus (E) is 1 and the Poisson ratio (ν) is 0.3. Uniform traction ($T=1$) is applied under the assumption of unity thickness and the plane strain condition, as illustrated in Figure 3. For this boundary value problem, the reference strain energy (U_{ref}) is taken as 9.198545583 [32].

Three different cases are investigated with respect to mesh discretization. First, the finite element mesh grid consists of an even number of elements along the horizontal and vertical directions such as 4×8 , 8×16 , 16×32 and 32×64 (Case I). In this case, the crack matches the finite element meshes. Next, the finite element mesh grid consists of an even number of elements along the horizontal direction and an odd number of elements along the vertical direction such as 4×9 , 8×17 , 16×33 and 32×65 (Case II). In this case, the crack cuts through elements while the crack tip is located at an element boundary (or edge). Finally, the mesh grid consists of an odd number of elements along the horizontal and vertical directions such as 5×9 , 9×17 , 17×33 and 33×65 (Case III). In this case, both the crack surface and the crack tip are located within elements. Figure 4 illustrates examples of finite element meshes for each case, and local enrichment types. Solid rectangles illustrate the local enrichment of the Heaviside function, and solid circles describe

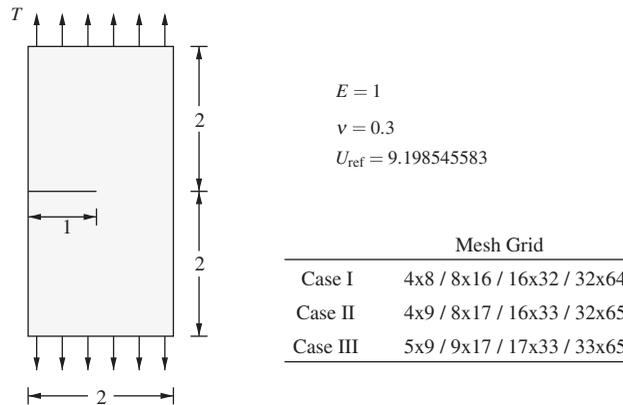


Figure 3. Geometry of problem and mesh grids for each case.

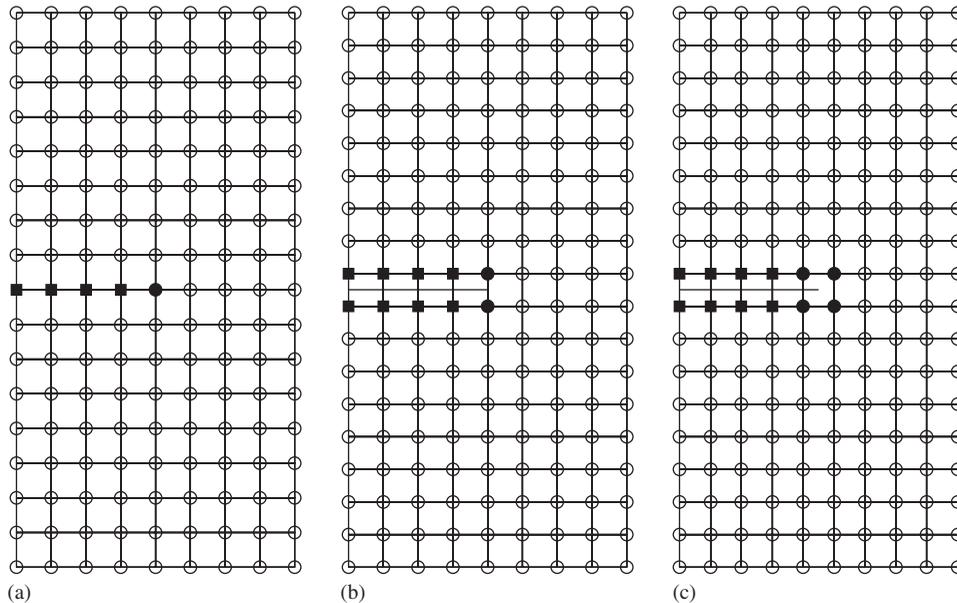


Figure 4. Finite element meshes for each case and local enrichment function types: (a) case I with the mesh grid of 8×16 ; (b) case II with the mesh grid of 8×17 ; and (c) case III with the mesh grid of 9×17 .

the local enrichment of the branch functions. With respect to the local enrichment types, one can introduce different numerical integration schemes during computational implementation, as discussed in the following section. The numerical integration of the stiffness matrix of elements enriched with singular functions is done using the technique described in Section 2.3 and a large number of integration points. Thus, the results presented below are not affected by integration errors.

Figure 5 compares the relative errors in the energy norm with respect to the branch functions such as the OD branch functions and the BB branch functions. For all the cases, the BB branch functions provide slightly lower errors in the energy norm than the OD branch functions. This is because the BB branch functions provide 8 degrees of freedom (DOFs) per node while the OD branch functions provide 4 DOFs per node in two-dimensional problems. In other words, the BB branch functions span a slightly larger solution space. The h -version provides convergence rate close to the theoretical values (0.25), while the p -version results in higher convergence rates (1.1 ~ 1.8) either for the OD or for the BB branch functions. The domain of influence, union of the support (or cloud) enriched with branch functions, progressively increases for each case (case I, case II, case II), and so does the convergence rates for the p -version. This fact is also investigated by Bechet *et al.* [14] and Laborde *et al.* [15].

Additionally, the total number of evaluation is estimated by multiplying the number of integration points by the number of \mathbf{K} entries. For instance, when we use polynomial order $p=1$ and the OD branch functions, we have 6 DOFs per node around the crack tip. Then, the number of \mathbf{K} entries is 576. If one uses 100 integration points, the total number of evaluations is 57 600. Similarly, when we use the BB branch function, we have 10 DOFs per node around the crack tip, and therefore, the number of \mathbf{K} entries is 1600. The total number of evaluations becomes 160 000 with the 100 integration points. One observation we can make is that the choice between the OD branch functions and the BB branch functions does not affect much the solution time since the size of the system of equations is about the same. However, it may affect more significantly the CPU time to integrate the stiffness matrix. Especially in 3D, the BB branch functions lead to 6 more DOFs per node than the OD branch functions. The difference in the number of evaluations required to integrate \mathbf{K} may be significant. The elements with singular enrichment functions may require many integration points and thus, the number of DOFs in these elements has a significant impact on the CPU time required to compute their stiffness matrix.

Based on the above investigation, hereafter the OD enrichment functions are employed. The general conclusions presented in this paper, however, hold for both the OD and BB enrichment functions.

2.3. Implementation of numerical integration

An element with a crack is subdivided into several triangular elements in order to avoid discontinuity in the integration domain [29, 33, 34]. When a crack fully cuts an element, we have two sub-domains, i.e. upper and lower part of an element. For each sub-domain, one obtains the centroid, and then creates triangles by connecting the centroid to the vertices of a sub-domain. The number of triangles is the same as the number of vertices of a sub-domain. Next, when a crack partially cuts an element, a crack tip is virtually extended to an edge so that one is able to perform the same procedure as the previous case, as shown in Figure 6. Three-dimensional implementation follows similar ideas, as described by Pereira *et al.* [35], in which integration sub-element edges fit the actual crack front. Alternatively, quadrature rules for triangular and tetrahedral elements are available when the integrand consists of a quadratic function times a Heaviside function [36].

Numerical integration schemes of subdivided triangular elements are described in Figure 6. When elements are enriched with polynomials or discontinuous functions (e.g. white sub-triangles), the standard quadrature rule is utilized with respect to given polynomial orders. When branch functions are enriched and sub-triangles do not have singularities (e.g. light gray triangles), the standard quadrature rule is also used. Sub-triangles around the crack tip (e.g. 4 dark gray elements) possess

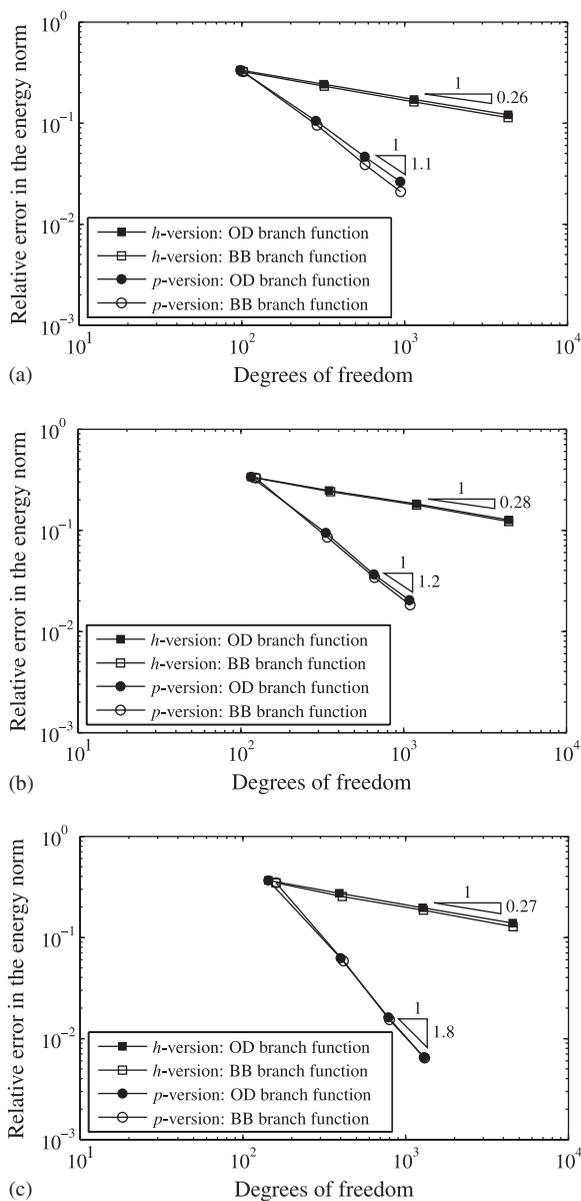


Figure 5. Comparison between the OD branch functions and the BB branch functions: (a) case I; (b) case II; and (c) case III.

singularities, and thus a mapping method is utilized so that one reduces the error from numerical integrations. The detailed computational implementation of the mapping method is discussed in the following section.

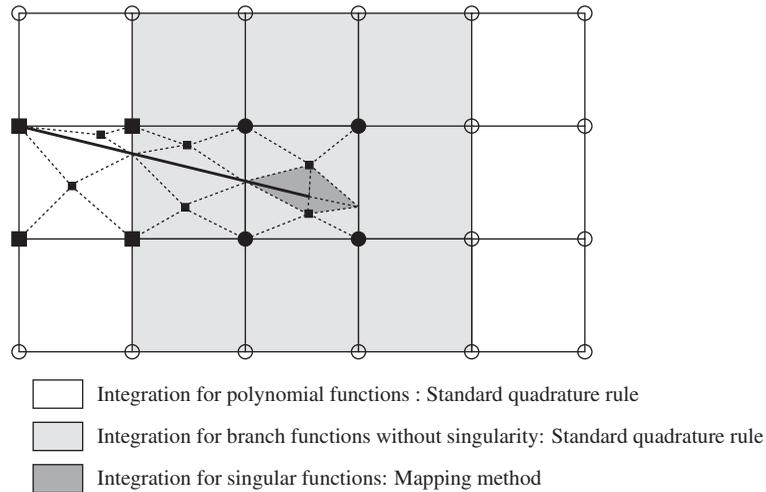


Figure 6. Numerical integration schemes with respect to enrichment functions.

3. MAPPING METHOD

In this section, a mapping method for singular integration in 2D and 3D domains is presented. For a 3D mapping method, two types of singularities are addressed in a tetrahedral element: (1) elements with an edge on the singularity and (2) elements with a node on the singularity.

3.1. 2D Mapping

For two-dimensional integrals with $1/r$ singularity, a mapping method is introduced by Nagarajan and Mukherjee [26]. The mapping method eliminates the singularity, and therefore, one can perform accurate numerical integration with the standard Gaussian integration points. The mapping is given as

$$T_M : (\bar{\rho}, \bar{\theta}) \mapsto (\bar{r}, \bar{s})$$

where

$$\bar{r} = \bar{\rho} \cos^2 \bar{\theta}, \bar{s} = \bar{\rho} \sin^2 \bar{\theta} \tag{3}$$

The inverse mapping (T_M^{-1}) transforms a right triangle (Figure 7 (b)) into a rectangle (Figure 7 (c)). Because the Jacobian of the transformation (J_M) is $\bar{\rho} \sin(2\bar{\theta})$, the $1/r$ singularity is eliminated in the integrand.

The procedure of the mapping method is described as follows. First, a triangular element, which has singularity at a vertex (e.g. dark gray shading in Figure 6), is mapped from the Cartesian coordinate (x, y) system (Figure 7 (a)) to the area coordinate (\bar{r}, \bar{s}) system (Figure 7 (b)). When we perform this transformation, the node, which corresponds to the crack tip, should be assigned as the origin in the area coordinate system. Next, the mapping transforms the area coordinate system into the rectangular coordinate $(\bar{\rho}, \bar{\theta})$ system.

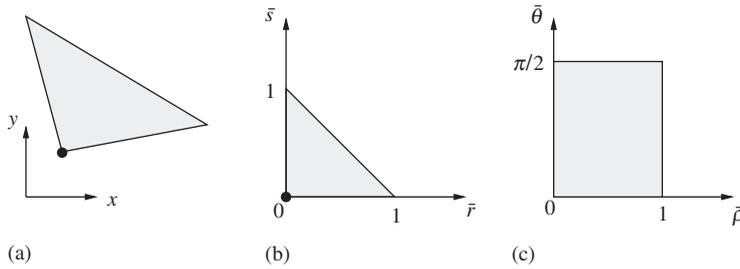


Figure 7. (a) Triangle with a node on the singularity in the Cartesian (physical) coordinate (x, y) system; (b) triangle in the area coordinate (\bar{r}, \bar{s}) system; and (c) transformed rectangle in the mapped coordinate $(\bar{\rho}, \bar{\theta})$ system.

3.1.1. *Computational implementation.* The numerical integration of the mapping method is implemented within the same framework of the standard numerical integration. For the standard integration, subdivided triangular elements (e.g. white and light gray shading in Figure 6) are integrated in the area coordinate (\bar{r}, \bar{s}) system,

$$I = \int_A f(x, y) dA = \int_A \bar{f}(\bar{r}, \bar{s}) J_t d\bar{r} d\bar{s} \approx \sum_i^{n_t} \bar{f}(\bar{r}_i, \bar{s}_i) J_t W_i \tag{4}$$

where n_t is the number of integration points for a triangle, and J_t is the Jacobian, which relates between the physical coordinates (x, y) system and the master (area) coordinates (\bar{r}, \bar{s}) system. The standard triangular integration points (\bar{r}_i, \bar{s}_i) and the weights (W_i) are listed in the reference by Dunavant [37].

The mapping method requires the transformation (T_M^{-1}) from the triangular domain to the rectangular domain. Then, subdivided triangular elements are integrated in the mapped coordinates system,

$$\begin{aligned} I &= \int_A \bar{f}(\bar{r}, \bar{s}) J_t d\bar{r} d\bar{s} = \int_0^1 \int_0^{\pi/2} \hat{f}(\bar{\rho}, \bar{\theta}) J_t \bar{\rho} \sin 2\bar{\theta} d\bar{\rho} d\bar{\theta} \\ &= \int_{-1}^1 \int_{-1}^1 \tilde{f}(\zeta, \eta) J_t \tilde{J}_M(\zeta, \eta) J_q d\zeta d\eta \approx \sum_i^{n_{q1}} \sum_j^{n_{q1}} \tilde{f}(\zeta_i, \eta_j) J_t \tilde{J}_M(\zeta_i, \eta_j) J_q W_i W_j \end{aligned} \tag{5}$$

where n_{q1} is the number of quadrature points for 1D Gaussian quadrature. While $\tilde{J}_M(\zeta, \eta)$ is equal to the transformation of $J_M = \bar{\rho} \sin(2\bar{\theta})$ to the integration coordinate system (ζ, η) , J_q is the Jacobian that relates between the mapped coordinates $(\bar{\rho}, \bar{\theta})$ system and the master coordinates (ζ, η) system for a rectangle. Note that the Jacobian (J_q) is a constant ($\pi/8$).

For the computational implementation, the above expression can be written as

$$\sum_i^{n_{q1}} \sum_j^{n_{q1}} \tilde{f}(\zeta_i, \eta_j) J_t \tilde{J}_M(\zeta_i, \eta_j) J_q W_i W_j = \sum_i^{n_q} \bar{f}(\bar{r}_i, \bar{s}_i) J_t \tilde{W}_i \tag{6}$$

where n_q is the number of quadrature points for the mapping method. The integration points (\bar{r}_i, \bar{s}_i) , which satisfy $\tilde{f}(\zeta_i, \eta_j) = \bar{f}(\bar{r}_i, \bar{s}_i)$, are obtained by transforming the integration points

(ξ_i, η_j) in the master coordinates system. Moreover, for the given integration points (\bar{r}_i, \bar{s}_i) , the quantity $(\tilde{J}_M(\xi_i, \eta_j) J_q W_i W_j)$ is considered as the transformed weights (\tilde{W}_i) . A pseudo code to generate the transformed integration points and weights is provided in Appendix A, and examples of integration points and weights are listed in Table BI.

In summary, the transformation procedures are eliminated in the computational implementation of the mapping method, and therefore, one can utilize an existing code by simply modifying the integration rule for elements which have a singularity. In other words, the mapping method (or transformation) generates the special integration rules (i.e. integration points and weights), which are able to efficiently handle the singularities.

3.2. 3D mapping

This section presents extensions of the mapping method to three-dimensional cases. For 3D elements, the singularity occurs either along an edge (Figure 8(a)) or at a node (Figure 9(a)). Mappings for these two cases are presented below.

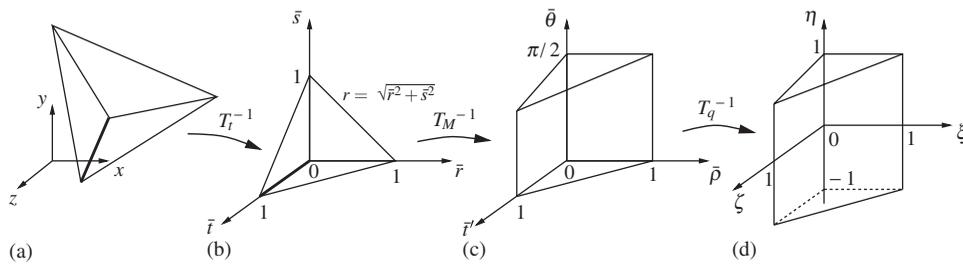


Figure 8. (a) Tetrahedron (Ω_{tet}) with an edge on the singularity in the Cartesian (physical) coordinate (x, y, z) system; (b) tetrahedron ($\hat{\Omega}_{tet}$) in the volume coordinate $(\bar{r}, \bar{s}, \bar{t})$ system; (c) transformed prism (Ω_{prism}) in the mapped coordinate $(\bar{\rho}, \bar{\theta}, \bar{t})$ system; and (d) master prism ($\hat{\Omega}_{prism}$) in the master coordinate (ξ, η, ζ) system.

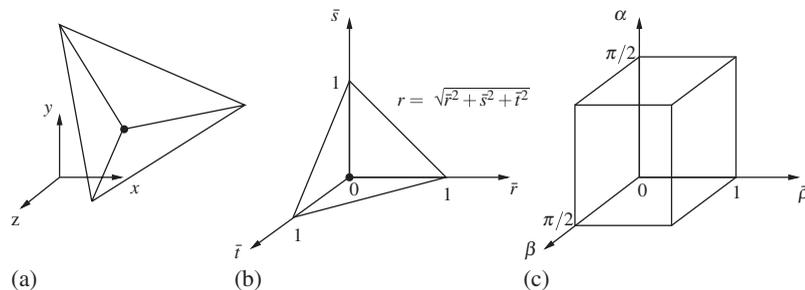


Figure 9. (a) Tetrahedron with a node on the singularity in the Cartesian (physical) coordinate (x, y, z) system; (b) tetrahedron in the volume coordinate $(\bar{r}, \bar{s}, \bar{t})$ system; and (c) transformed hexahedron in the mapped coordinate $(\bar{\rho}, \alpha, \beta)$ system.

3.2.1. *Elements with edge on singularity.* Let

$$T_t: (\bar{r}, \bar{s}, \bar{t}) \mapsto (x, y, z) \tag{7}$$

denote the transformation from master (volume) coordinates of a tetrahedral element to global (physical) coordinates. The mapping T_t is implemented using standard tetrahedral shape functions. The Jacobian of this mapping, J_t , is, in the case of elements with straight edges, constant over the element.

We assume, without loss of generality, that the element connectivities are such that the element edge on the singularity is mapped by T_t^{-1} to the edge along the \bar{t} axis, as illustrated in Figure 8(b). Thus, this case is similar to the two-dimensional one because the singularity is only influenced by \bar{r} and \bar{s} . Therefore, one utilizes the following mapping:

$$T_M: (\bar{\rho}, \bar{\theta}, \bar{t}') \mapsto (\bar{r}, \bar{s}, \bar{t})$$

where

$$\bar{r} = \bar{\rho} \cos^2 \bar{\theta}, \quad \bar{s} = \bar{\rho} \sin^2 \bar{\theta}, \quad \bar{t} = \bar{t}' \tag{8}$$

The inverse mapping, T_M^{-1} , leads to the integration domain shown in Figure 8(c). The (\bar{r}, \bar{s}) plane is transformed to the $(\bar{\rho}, \bar{\theta})$ plane. With the increase of \bar{t}' from 0 to 1, the integration domain of the $(\bar{\rho}, \bar{\theta})$ plane decreases from $(1 \times \pi/2)$ to $(0 \times \pi/2)$. Within this integration domain, the singularity is eliminated because the Jacobian of T_M is equal to $\bar{\rho} \sin 2\bar{\theta}$.

An additional mapping is used to transform the prism shown in Figure 8(c) to the master prism shown in Figure 8(d),

$$T_q^{-1}: (\bar{\rho}, \bar{\theta}, \bar{t}') \mapsto (\xi, \eta, \zeta) \tag{9}$$

Standard quadrature rules for a prism can be defined using the product of an 1D Gauss quadrature and a rule for triangular elements. In our computations, we use Dunavant rules [37]. The Jacobian of transformation T_q is constant with $J_q = \pi/4$.

3.2.2. *Elements with a single node on singularity.* For the case shown in Figure 9(a), the following mapping can be used:

$$\bar{r} = \bar{\rho} \cos^2 \alpha, \quad \bar{s} = \bar{\rho} \cos^2 \beta, \quad \bar{t} = \bar{\rho} \cos^2 \gamma \tag{10}$$

On a plane, which is normal to a vector $\{1, 1, 1\}$, the proposed mapping satisfies the following condition:

$$\cos^2 \alpha + \cos^2 \beta + \cos^2 \gamma = 1 \tag{11}$$

The substitution of the above expression into $\bar{t} = \bar{\rho} \cos^2 \gamma$ leads to $\bar{t} = \bar{\rho}(1 - \cos^2 \alpha - \cos^2 \beta)$. Then, the Jacobian is given as

$$\begin{vmatrix} -\bar{\rho} \sin(2\alpha) & 0 & \cos^2 \alpha \\ 0 & -\bar{\rho} \sin(2\beta) & \cos^2 \beta \\ \bar{\rho} \sin(2\alpha) & \bar{\rho} \sin(2\beta) & 1 - \cos^2 \alpha - \cos^2 \beta \end{vmatrix} = \bar{\rho}^2 \sin(2\alpha) \sin(2\beta) \tag{12}$$

and the transformed domain is described in Figure 9(c). Because the Jacobian has a $\bar{\rho}^2$ term, the singularity is eliminated within the integration domain.

3.2.3. Elements with a single node on singularity: Alternative approach. In the approach presented in the previous section, the quadrature is performed in a hexahedral element. Thus, for the same quadrature order, the number of quadrature points used in that approach is much larger than the one from Section 3.2.1. For a rule able to integrate an integrand of polynomial degree $p=4$, for example, the quadrature rule for a hexahedral element requires 27 points while the corresponding rule for a prism has only 18 points. Numerical experiments performed in Section 4.3 show that integration errors introduced by elements with an edge on a crack front is more pronounced than for elements with a single node on a crack front.

Based on the above, we propose to use the integration rule presented in Section 3.2.1 for elements with *either* an edge or a single node on the crack front. Let, in the later case, \mathbf{x}_α denote the node on the crack front. Let e_{close} denote the element edge *with node* \mathbf{x}_α that makes the smallest angle with the singularity line (crack front). We assume, without loss of generality, that the element connectivities are such that element edge e_{close} is mapped by T_t^{-1} to the edge along the \bar{t} axis, as illustrated in Figure 8(b). The rule for elements with an edge on the crack front is then used, without modifications, for this element. The performance of this approach is investigated in Section 4.3.

3.3. Computational implementation

The computation implementation of quadrature rules for tetrahedral elements can be done using the same ideas described in Section 3.1.1. The case of an element with an edge on a crack front is considered in this section.

Let $f(x, y, z)$ be a function we want to integrate over the tetrahedral Ω_{tet} . We assume that this function has a line singularity and that the tetrahedral element has an edge along this line. The integral of $f(x, y, z)$ can be computed as follows using the quadrature rule presented in Section 3.2.1

$$\begin{aligned}
 I &= \int_{\Omega_{\text{tet}}} f(x, y, z) \, d\mathbf{x} = \int_{\hat{\Omega}_{\text{tet}}} \bar{f}(\bar{r}, \bar{s}, \bar{t}) J_t \, d\bar{r} \, d\bar{s} \, d\bar{t} \\
 &= \int_{\Omega_{\text{prism}}} \check{f}(\bar{\rho}, \bar{\theta}, \bar{t}') J_t J_M(\bar{\rho}, \bar{\theta}) \, d\bar{\rho} \, d\bar{\theta} \, d\bar{t}' \\
 &= \int_{\hat{\Omega}_{\text{prism}}} \tilde{f}(\zeta, \eta, \zeta) J_t \tilde{J}_M(\zeta, \eta, \zeta) J_q \, d\zeta \, d\eta \, d\zeta \\
 &\approx \sum_{i=1}^{n_q} \tilde{f}(\zeta_i, \eta_i, \zeta_i) \tilde{J}_M(\zeta_i, \eta_i, \zeta_i) J_q J_t W_i \\
 &= \sum_{i=1}^{n_q} \bar{f}(\bar{r}_i, \bar{s}_i, \bar{t}_i) J_t \tilde{W}_i
 \end{aligned} \tag{13}$$

where Ω_{tet} , $\hat{\Omega}_{\text{tet}}$, Ω_{prism} and $\hat{\Omega}_{\text{prism}}$ are illustrated in Figures 8(a)–(d), respectively, J_t , J_M and J_q are the Jacobian of the transformations T_t , T_M and T_q defined in Section 3.2.1, respectively, and

$$\begin{aligned}
 \bar{f}(\bar{r}, \bar{s}, \bar{t}) &= f \circ T_t(\bar{r}, \bar{s}, \bar{t}) \\
 \check{f}(\bar{\rho}, \bar{\theta}, \bar{t}') &= \bar{f} \circ T_M(\bar{\rho}, \bar{\theta}, \bar{t}')
 \end{aligned} \tag{14}$$

$$\begin{aligned} \tilde{f}(\xi, \eta, \zeta) &= \check{f} \circ T_q(\xi, \eta, \zeta) \\ \tilde{J}_M(\xi, \eta, \zeta) &= J_M \circ T_q(\xi, \eta, \zeta) \\ \tilde{W}_i &= \tilde{J}_M(\xi_i, \eta_i, \zeta_i) J_q W_i \end{aligned}$$

where ‘o’ denotes the composition of two functions.

The quadrature points, (ξ_i, η_i, ζ_i) , and weights, $W_i, i = 1, \dots, n_q$, are defined in the master prism element $\hat{\Omega}_{\text{prism}}$. These points are mapped to quadrature points $(\bar{r}_i, \bar{s}_i, \bar{t}_i)$ defined in $\hat{\Omega}_{\text{tet}}$ using

$$(\bar{r}_i, \bar{s}_i, \bar{t}_i) = T_M \circ T_q(\xi_i, \eta_i, \zeta_i), \quad i = 1, \dots, n_q \tag{15}$$

These points, together with quadrature weights $\tilde{W}_i, i = 1, \dots, n_q$, can be used in the master element $\hat{\Omega}_{\text{tet}}$ to integrate the singular function $\tilde{f}(\bar{r}, \bar{s}, \bar{t})$. Thus, as pointed out in Section 3.1.1, the numerical integration of \tilde{f} can be accurately performed by simply replacing the quadrature rule used in an existing FEM code by the rule defined above.

Table BII lists quadrature rules for $n_q = 1, 6, 8, 18, 21, 48$. These rules are based on prism rules that can integrate polynomials of degree $p = 1, 2, 3, 4, 5, 6$, respectively. The weights listed there, \hat{W}_i , were scaled by the volume of the master tetrahedron, $\hat{\Omega}_{\text{tet}}$, i.e.

$$\hat{W}_i = \tilde{W}_i / (\frac{1}{6}) \tag{16}$$

They should be scaled back before their use.

The sum $\sum_i^{n_q} \hat{W}_i$ is also listed in Table BII. We can observe that the weights do *not* add to 1.0, as one would expect. The quadrature rule in the η -direction of the master prism $\hat{\Omega}_{\text{prism}}$ must integrate the trigonometric function $\sin 2\theta$ from the Jacobian J_M . Thus, a high enough rule in the η -direction should be used, regardless of the integrand function \tilde{f} . We can observe that the 18 point rule is able to integrate a constant function with an accuracy of 0.069445%. The deviation of $\sum_i^{n_q} \hat{W}_i$ from 1.0 is exactly the same in Tables BI and BII since the Jacobian J_M is the same in both cases.

In the case of fracture mechanics problems, the singular functions to be integrated involve trigonometric functions, as discussed in Section 2.1. Thus, a moderate number of points must be used anyway. The same is the case when high-order GFEM shape functions, like those introduced in references [35, 38] are used. The integration of J_M does not introduce any overhead in the computations.

Rules with $n_q = 6$ and $n_q = 8$ use the same 1D Gauss rule in the η -direction. Thus, they provide the same accuracy for the integration of the Jacobian J_M . The same holds for other rules like those with $n_q = 18$ and $n_q = 21$.

It is desirable that the quadrature rules exactly integrate at least a constant function. This can be achieved by scaling the weights such that they add to 1.0. Define

$$\tilde{\tilde{W}}_i = \hat{W}_i / \left(\sum_j^{n_q} \hat{W}_j \right), \quad i = 1, \dots, n_q \tag{17}$$

The effect of the scaling is discussed in Section 4.1, and we use these scaled weights in all computations presented in Section 4.3.

4. EXAMPLES

Three classes of examples are presented in this section:

- Theoretical assessment.
- Integration of branch functions.
- Effect of integration errors on quantities of interest.
 - Mode I fracture analysis on a coarse mesh.
 - Mode I fracture analysis of an edge-cracked bar using locally refined meshes.
 - Mixed-mode fracture analysis.

The aim of the theoretical assessment is to demonstrate the characteristics of the mapping method for integration of scalar functions over a triangular area. The integration of branch functions assesses the convergence of their element stiffness matrix entries for fracture problems. Finally, the effect of integration error on quantities of interest addresses practical fracture problems on coarse and fine meshes, including both Mode I and mixed-mode problems.

4.1. Theoretical assessment

In order to demonstrate the efficiency of the singular mapping method, convergence of the following singular integration is investigated:

$$I = \int_A f(x, y) dA \quad (18)$$

where A is the area of a triangle. A triangle has vertices at $(0,0)$, $(1,0)$ and $(0,1)$ in x - y physical coordinates. Within the triangular domain (shown in Figure 7(a)), a scalar function ($f(x, y)$) such as 1 , $1/r$ and $1/\sqrt{r}$ (where $r = \sqrt{x^2 + y^2}$) is integrated by using either the standard triangular

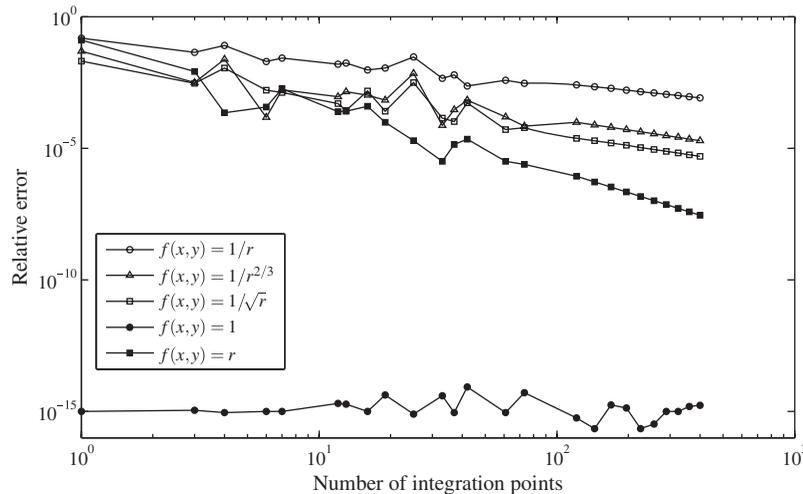


Figure 10. Convergence of numerical integration for the standard triangular integration.

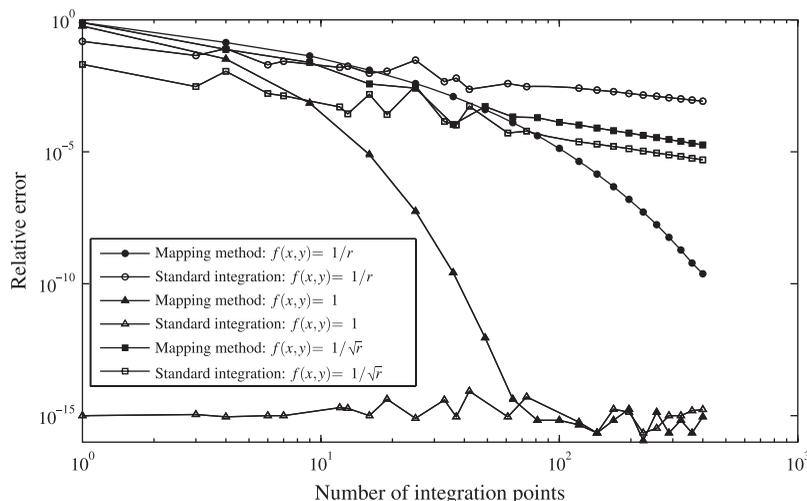


Figure 11. Comparison between the mapping method and the standard integration.

integration [37] or the singular mapping method. Figure 10 demonstrates convergence of a standard triangular integration with respect to the number of integration points. It is clear that when the integrand has higher singularity, convergence of the relative error is slower. The singular mapping method is compared with the standard integration, shown in Figure 11. Both the standard triangular integration and the mapping method result in almost the same convergence rate (or slope) for the $1/\sqrt{r}$ singularity, while the mapping method leads to the higher convergence rate for the $1/r$ singularity. The standard integration provides exact integration with any number of integration points for a constant function. However, the mapping method requires several integration points to obtain accurate results. This is because the summation of the weights of the mapping method is not exactly equal to one (although it converges to one), as shown in Table BI.

In order to obtain the integration rule that provides exact integration of a constant function, the weights are scaled so that they add to 1.0, as discussed previously. Figure 12 illustrates the convergence of a constant function ($f(x, y) = 1$) and a singular function ($f(x, y) = 1/r$) with respect to different integration schemes: the mapping method with the scaled weights, the mapping method without scaling and the standard triangular integration. The mapping method with scaled weights results in the exact integration of a constant function, as we expected. Moreover, the mapping method with scaled weights provides almost the same convergence for the $1/r$ singularity as the one without the scaled weights.

4.2. Integration of branch functions

We investigate convergence of the element stiffness matrix entries, which is associated with the integration of the branch functions. In this investigation, the OD branch functions are used for the evaluation of the element stiffness matrix. Figure 13 illustrates the convergence of the Frobenius-norm of the element stiffness matrix associated with a singular sub-triangle (i.e. gray sub-triangle in the figure). It is clear that the mapping method converges exponentially while standard quadrature converges algebraically.

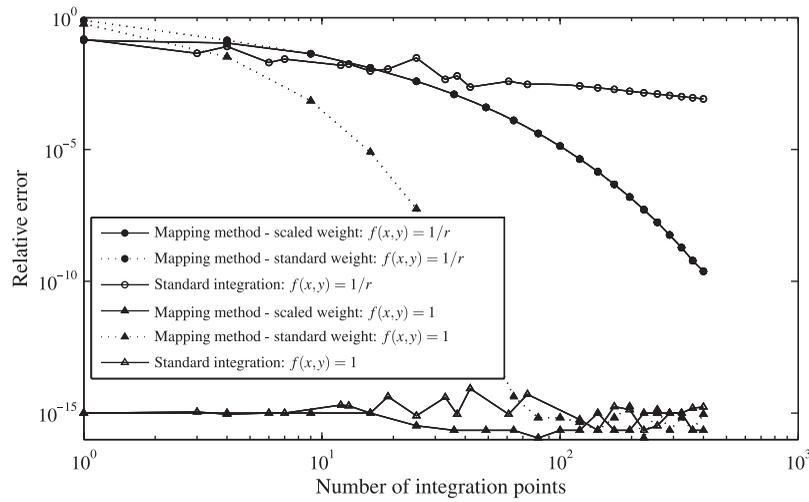


Figure 12. The effect of the scaled weight in the mapping method.

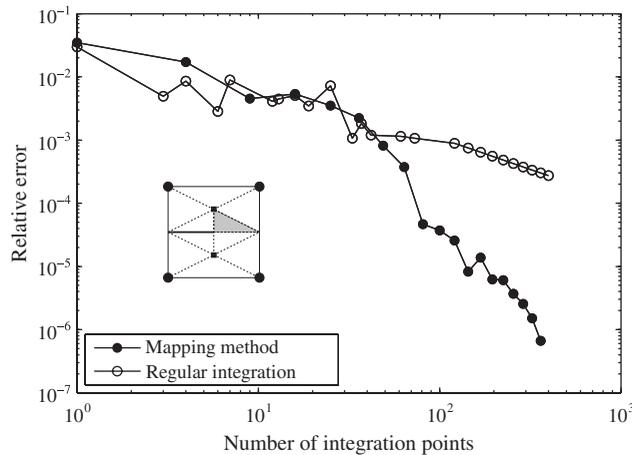


Figure 13. Convergence of the Frobenius-norm of the element stiffness matrix associated with a singular sub-triangle (shaded triangle).

In summary, Figures 10–13 also show that the mapping method provides a more monotonic convergence and that it is as accurate as the standard quadrature for low-accuracy integration and much more accurate in the high-accuracy range. Given that the computational cost of both quadratures is exactly the same (for the same number of points) and the implementation as suggested in the paper is as straightforward as the standard quadrature, we have a good case in favor of the mapping method.

4.3. 3D analysis and effect of integration errors on quantities of interest

In this section, we investigate the performance of the proposed 3D quadrature rules. Our focus is on the influence of integration errors introduced by elements enriched with the OD branch functions on quantities of interest for engineering analysis. We compare the strain energy and stress intensity factors computed with different integration strategies. The main goal is to select the most efficient quadrature rule for tetrahedral elements based on the target error level of quantities of interest. The analysis developed in this section is presented in three examples. In the first one, we analyze the worst-case scenario from the numerical integration point of view, i.e. very large elements enriched with singular functions. In this case, the 3D model is subjected to appropriate boundary conditions such that the exact values for strain energy and stress intensity factors are known. This approach resembles the so-called boundary layer models in fracture mechanics [39–41]. In the second example, we perform the integration error analysis for typical meshes used in engineering analysis, i.e. elements with singular enrichments are small. The reference solution used in this case is fully 3D. The third example presents a mixed-mode fracture analysis. Again, the 3D model is subjected to appropriate boundary conditions to obtain exact values for strain energy and stress intensity factors. In this case, typical tetrahedral finite element meshes are also used in the discretization.

4.3.1. Mode I fracture—benchmark for the mapping method. In this example, we solve an edge-cracked problem in which the Neumann boundary conditions are the tractions computed from the first term of the Mode I Westergaard solution. Using these boundary conditions and setting the Poisson ratio to zero, one can ensure that the exact solution in the domain of analysis is given by the first term of the asymptotic expansion. Thus, the strain energy as well as the stress intensity factor for this problem can be analytically defined. Figure 14 illustrates the domain of analysis and the boundary conditions applied. The main objective of the analysis presented below is to investigate the performance of the three-dimensional version of the proposed integration scheme. In particular, the effectiveness of the numerical quadrature for edge singularity is presented in Section 3.2.1.

GFEM shape functions of degree $p=4$ are used at all nodes, except at those on the crack front. Crack front nodes have GFEM shape functions of degree $p=3$ as well as the 3D counterpart of the OD singular functions (1). Further details on these enrichments can be found in [4, 30, 31, 35].

The integration rule applied to an element is selected according to its position with respect to the crack front. In the mesh shown in Figure 14, there are six elements with one edge along the crack front (elements with edge singularity), 12 elements with one node on the crack front (elements with node singularity) and six elements with no nodes on the crack front. The later set is integrated using Keast integration rule [28] of order 6 and 24 points. This rule integrates exactly the stiffness matrix of these elements. It corresponds to rule $K.3$ defined in Table I.

In the first part of the analysis presented here, we concentrate on the numerical integration of elements with edge singularity. We compute the normalized strain energy and normalized stress intensity factors using the GFEM discretization described above and three numerical integration strategies: Strategies 1, 6 and 7 described in Table II. Each strategy corresponds to a sequence of pairs of integration rules of increasing accuracy. The baseline is Strategy 1 that uses rules $K.2, \dots, K.10$ for elements with vertex or edge singularity (cf. Table II). Strategies 6 and 7 use rules $M.1, \dots, M.14$ and $K.2, \dots, K.10$, respectively, for elements with edge singularity and rule $K.10$ for elements with vertex singularity. This ensures that the integration errors introduced by

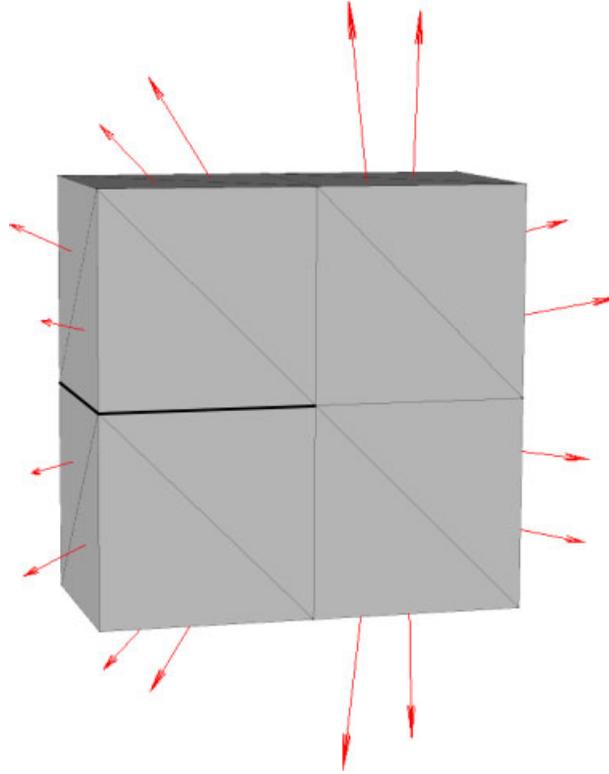


Figure 14. Edge-cracked panel with Mode *I* boundary conditions from asymptotic expansion.

these elements have no influence in the analysis. Rule *K.3* is used for elements without singular enrichment functions. Table I provides details on all rules used on this and subsequent sections. Rules of lower order than those listed in the table are not considered since they would not be able to integrate the polynomial shape functions used in the 3D GFEM discretizations considered in this paper.

Figures 15(a) and (b) show the variation of the normalized strain energy (U/U_{exact}) and normalized stress intensity factors ($K_I/K_{I\text{exact}}$) in semi-log scale, respectively, for Strategies 1, 6 and 7. The horizontal axes show the sum of the number of quadrature points used by *all* 24 elements of the GFEM discretization. Figures 16(a) and (b) show a zoom in of these plots.

From Figures 15(a) and 16(a), we observe that Strategy 6 quickly converges to the strain energy value corresponding to the highest rule used in this strategy (rule *M.14*). In contrast, the convergence of Strategy 7 to the value corresponding to rule *K.10* is quite slow. Another observation is that the strain energy values computed with Keast rules are larger than the exact one while those computed with the mapping method are, in general, smaller than the exact one. This indicates that the error in the case of the mapping method is mostly due to discretization errors while in the case of Keast rules it is also strongly affected by integration errors. Comparing Strategies 1 and 6, we can observe that integration errors introduced by elements with vertex singularity are not dominant since corresponding points on these curves show approximately the

Table I. Integration rules for tetrahedral elements. Rules *K.1–K.5* are Keast rules [28]. Rules *K.6–K.10* are based on the tensor product of 1D Gaussian rules.

Rule	Keast and tensor product rules		Rule	Mapping method of Section 3.2.1	
	Num. points	p_{Keast}		Num. points	p_{map}
<i>K.1</i>	11	4	<i>M.1</i>	18	4
<i>K.2</i>	15	5	<i>M.2</i>	21	5
<i>K.3</i>	24	6	<i>M.3</i>	48	6
<i>K.4</i>	31	7	<i>M.4</i>	52	7
<i>K.5</i>	45	8	<i>M.5</i>	80	8
<i>K.6</i>	125	9	<i>M.6</i>	95	9
<i>K.7</i>	216	11	<i>M.7</i>	150	10
<i>K.8</i>	343	13	<i>M.8</i>	198	11
<i>K.9</i>	512	15	<i>M.9</i>	231	12
<i>K.10</i>	729	17	<i>M.10</i>	259	13
			<i>M.11</i>	336	14
			<i>M.12</i>	488	15
			<i>M.13</i>	549	17
			<i>M.14</i>	730	19
			<i>M.15</i>	1331	21

p_{Keast} = polynomial order integrated by Keast or tensor product rule.

p_{map} = polynomial order integrated by prism quadrature used to generate the integration points used in the mapping method.

Table II. Integration strategies used on elements with singular shape functions.

Strategy	Integration rule	
	Elements with edge on front	Elements with vertex on front
1	<i>K.2, ..., K.10</i>	<i>K.2, ..., K.10</i>
2	<i>M.1, ..., M.10</i>	<i>K.1, ..., K.10</i>
3	<i>M.1, ..., M.8</i>	<i>K.3, ..., K.10</i>
4	<i>M.1, ..., M.7</i>	<i>K.4, ..., K.10</i>
5	<i>M.1, ..., M.14</i>	<i>M.1, ..., M.14</i>
6	<i>M.1, ..., M.14</i>	<i>K.10</i>
7	<i>K.2, ..., K.10</i>	<i>K.10</i>

Each strategy corresponds to a sequence of pairs of integration rules of increasing accuracy. They combine Keast (or tensor product) rules and the mapping method applied to elements with edge or node on the crack front. For instance, Strategy 1 applies the Keast rule for all elements, Strategy 2 applies the mapping method for elements with edge singularity and the Keast rule for elements with nodal singularity, Strategy 5 applies the mapping method for elements with edge singularity and elements with nodal singularity, and so forth.

same error level. Similar observations can be made for the results of the Mode I stress intensity factor (K_I) shown in Figures 15(b) and 16(b). The main difference in behavior is that, in the case of stress intensity factor, the integration errors seem to be under control in all strategies. In the case of strain energy, however, only Strategy 6 was able to control the effect of integration errors. This indicates that stress intensity factors are apparently less sensitive to integration errors in the computation of the stiffness matrix than the strain energy.

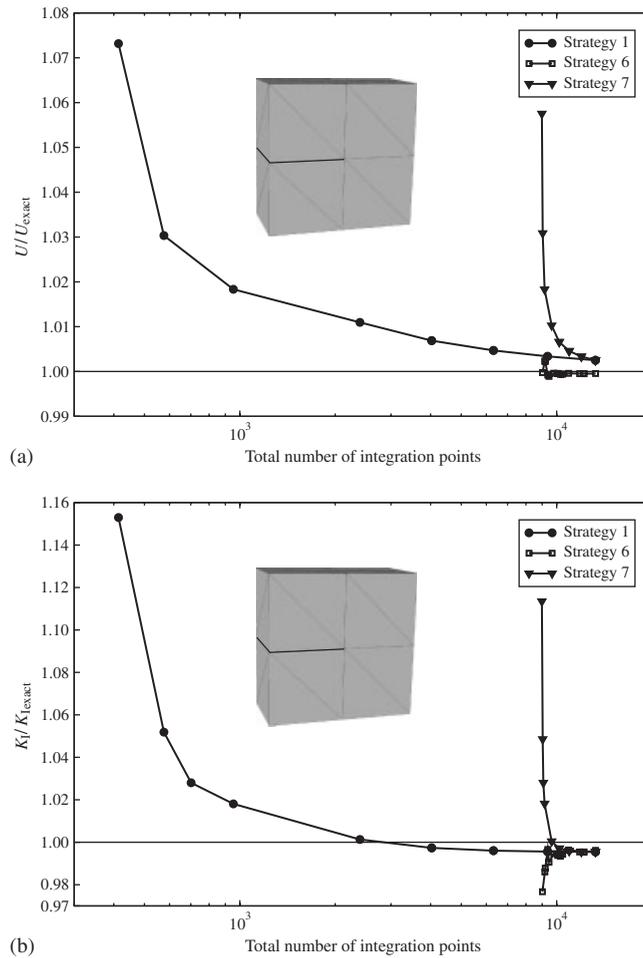


Figure 15. Normalized energy and stress intensity factor (SIF) for Strategies 1, 6 and 7. The horizontal axes show the sum of the number of quadrature points used by all 24 elements of the GFEM discretization: (a) normalized energy. Semi-log scale and (b) normalized SIF. Semi-log scale.

The results above show that the mapping method applied to elements with edge on the crack front can integrate singular functions using far fewer points than Keast integration rules. This is confirmed by the numerical experiments presented in Sections 4.3.2 and 4.3.3.

Strategies 6 and 7 cannot be used in practical computations since they use a very large number of integration points on elements with vertex singularities. Below, and in Sections 4.3.2 and 4.3.3, we focus on the selection of the best scheme to integrate these elements and on the overall reduction of the number of integration points for a given target error level. Strategy 1 is again used as a reference. We consider Strategies 2–5 described in Table II. The only difference among these strategies is the integration rule used for elements with vertex singularity. All of them use the mapping method for elements with edge singularity.

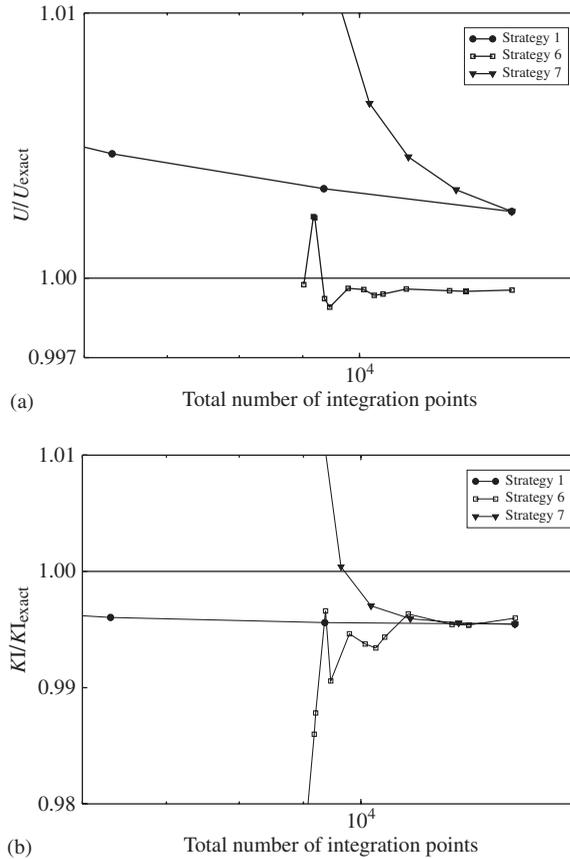


Figure 16. Zoom in of plots of Figures 15(a) and (b): (a) normalized energy. Semi-log scale and (b) normalized SIF. Semi-log scale.

Figures 17(a) and (b) show the variation of the normalized strain energy and normalized stress intensity factors in semi-log scale, respectively, for Strategies 1–5. In Figure 17(a), one can observe that for Strategies 2–5, after a large enough number of integration points the error is controlled only by the discretization error. This behavior is similar to Strategy 6 (cf. Figure 15(a)). Before this point, the error in strain energy is controlled by integration errors on elements enriched with Westergaard functions. Strategies 2 and 5 converge quicker than Strategies 3 and 4 to the strain energy value corresponding to the highest rule used. Similar observations can be made for the results shown in Figure 17(b). Strategy 5 is particularly attractive because it exhibits a more monotonic behavior than the other strategies. In this strategy, we apply the edge singular rule for elements with either edge or vertex singularity, as described in Sections 3.2.1 and 3.2.3. For those elements with nodal singularity, the integration rule is set to the edge that is connected to the singular vertex and has smallest angle with the crack front. This approach has proved to be efficient even in elements where the smallest angle is around 45°.

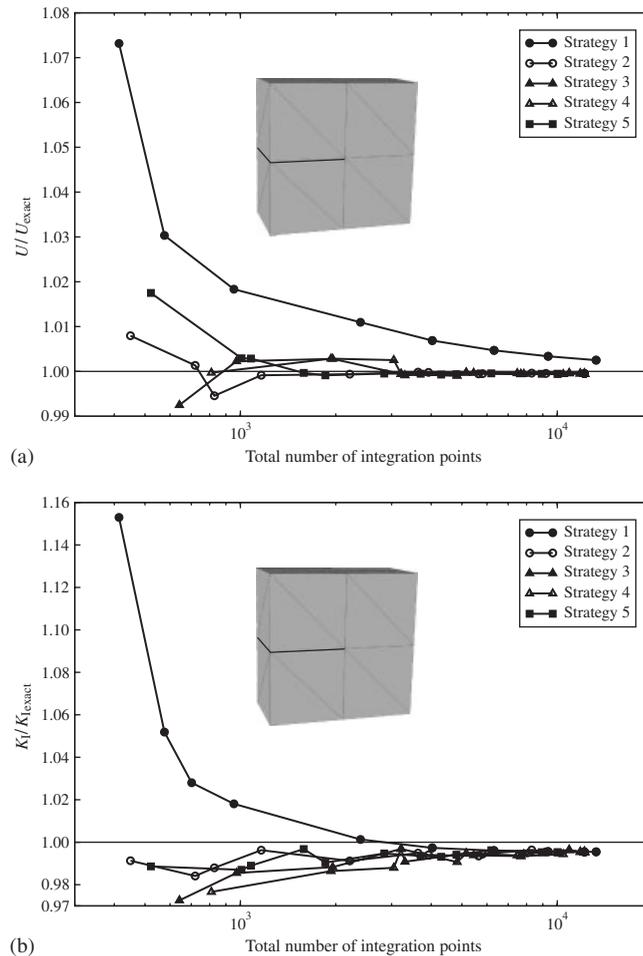


Figure 17. Normalized strain energy and stress intensity factor (SIF) for Strategies 1, 2, 3, 4 and 5 with respect to the sum of the total number of integration points used by all 24 elements: (a) normalized energy. Semi-log scale and (b) normalized SIF. Semi-log scale.

Based on the results presented above, only Strategies 1, 2 and 5 are considered in the following examples. There, we use meshes with element sizes typically used in fracture mechanics problems and select the quadrature rule based on the target error level of quantities of interest.

4.3.2. *Edge-cracked bar—selection of integration order.* In this section, an edge-cracked bar problem is considered. The model consists of a rectangular bar subjected to a uniform tensile traction applied at both ends with a through-the-thickness edge crack as illustrated in Figure 18. Nodal displacement boundary conditions are applied only to prevent rigid body motion. The geometric parameters are set as $h/t=0.875$, $a/t=0.5$, $w/t=1.5$. The Poisson ratio and Young’s modulus are set as $\nu=\frac{1}{3}$ and $E=1.0$, respectively.

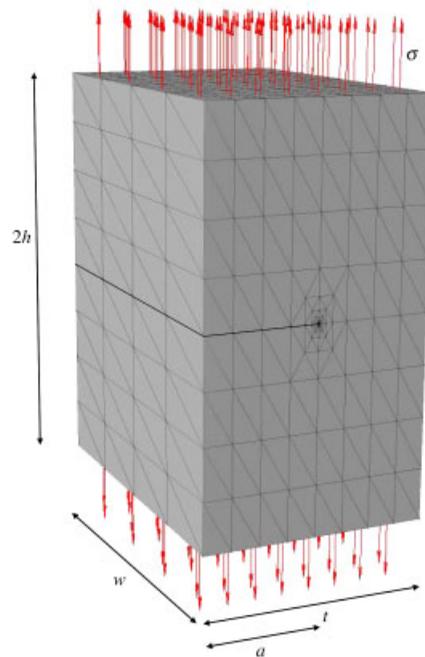


Figure 18. Edge-cracked bar: domain dimensions, boundary conditions and mesh with localized refinement.

This problem was also solved in References [42] and [43] using the standard FEM with quarter-point elements along the crack front and boundary element techniques, respectively. In this section, we focus our analysis on the convergence of the strain energy with respect to the number integration points and on the effect of integration errors on p -extensions performed on meshes with localized refinement along the crack front. Integration Strategies 1, 2 and 5 (cf. Section 4.3.1) are used. The analysis is aimed at selecting integration rule pairs from these strategies. As demonstrated below, this selection depends on the target error level of the quantities of interest and the level of refinement used near the crack front.

A closed-form solution is not available for this problem. However, one can use *a posteriori* error estimates [44] to compute a reference value for the strain energy. The computed reference value for the strain energy is, for $a=0.5$,

$$U_{\text{ref}} = 1.73475171991776 \times 10^{-5}$$

The discretizations used in the present analysis are based on the p -version of the GFEM presented in [4], i.e. hierarchical polynomial enrichments. A mesh with double nodes is used to represent the discontinuity. Crack front nodes are also enriched with the 3D counterpart of the OD branch functions (1) as described in [4, 30, 35].

Two meshes, denoted hereafter Mesh 1 and Mesh 2, are used in the simulations. In both cases, the meshes are locally refined along the crack front as shown in Figure 18 for the case of Mesh 1. The ratio of element size to crack size, (L_e/a) , at the crack front of Mesh 1 is in the range $3.9 \times 10^{-3} \leq L_e/a \leq 1.4 \times 10^{-2}$. This discretization leads to 2048 elements with an edge on the

Table III. Details on meshes for edge-cracked bar.

Mesh	1	2
$\min(L_e/a)$	3.9×10^{-3}	1.6×10^{-2}
$\max(L_e/a)$	1.4×10^{-2}	5.7×10^{-2}
# Elem. w/edge sing.	2048	512
# Elem. w/vertex sing.	6144	1024

(L_e/a) denotes the ratio of element size to crack size at the crack front.

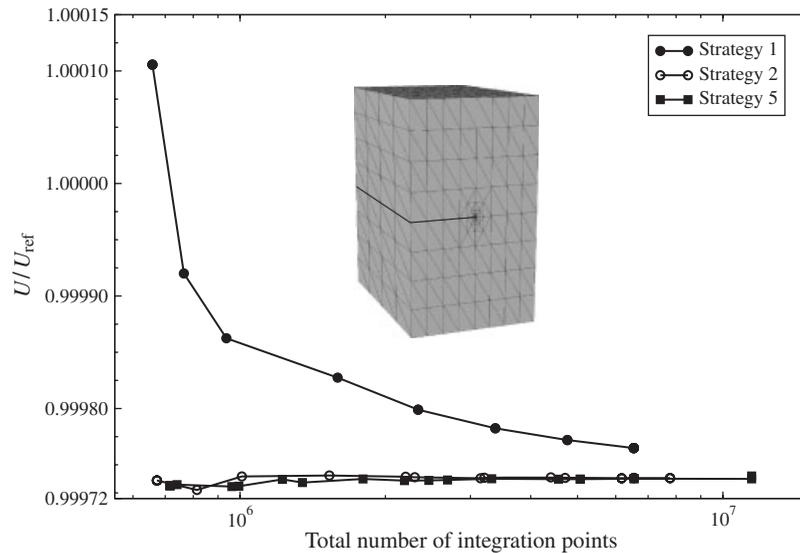


Figure 19. Convergence of strain energy with respect to the total number of integration points for elements enriched with the OD branch functions, i.e. elements with either edge or node on crack front. Semi-log scale.

crack front and 6144 elements with a node on the crack front. Details on Meshes 1 and 2 are provided in Table III.

The first goal of this analysis is to select integration rules for elements with edge or vertex singularities such that the effect of integration errors on the computed strain energy is much smaller than the discretization error. This analysis is performed by applying integration Strategies 1, 2 and 5 to Mesh 1 enriched with $p=4$ polynomials and the OD branch functions. We analyze the convergence of the strain energy with respect to the total number of integration points used in *all* elements that have a node enriched OD branch functions. Figure 19 presents the results. One can observe that, for a sufficiently large number of integration points and Strategy 2 or 5, the error in strain energy is controlled only by the discretization error. The difference between the results in Strategy 1 and the plateau provided by Strategies 2 and 5 can be regarded as due to integration errors of the singular shape functions which, in this case, is small due to the level of refinement applied to the crack front.

Table IV. Integration rule pairs used on elements with edge or vertex singularity.

Rule pair ID	Integration rule and number of points	
	Elements with edge on front	Elements with vertex on front
1.5	<i>K.5</i> (45 points)	<i>K.5</i> (45 points)
2.3	<i>M.3</i> (48 points)	<i>K.3</i> (24 points)
5.1	<i>M.1</i> (18 points)	<i>M.1</i> (18 points)

These pairs are members of integration strategies 1, 2 or 5. Each strategy corresponds to a sequence of integration rule pairs as discussed in Section 4.3.1. The naming convention used to identify a pair is based on the sequence/strategy number to which it belongs and the index of the rules it uses.

Table V. Total number of integration points used in elements with singular enrichments.

Rule pair ID	Mesh 1	Mesh 2
1.5	368 640	69 120
2.3	245 760	49 152
5.1	147 456	27 648

Based on the above results, we select one integration rule pair from each strategy for further investigation. These pairs are identified hereafter as 1.5, 2.3 and 5.1. Details are provided in Table IV. Table V lists, for Meshes 1 and 2 and integration pairs 1.5, 2.3 and 5.1, the total number of integration points used in elements with singular enrichments. We can observe that integration rule pair 5.1 leads to a considerably smaller number of integration points than other pairs.

The second goal of this analysis is to investigate the effect of integration errors on the computed strain energy when integration rule pairs 1.5, 2.3 and 5.1 are used. The p -version of the GFEM and Meshes 1 and 2 are used. In both cases, the polynomial order of the GFEM functions are in the range $1 \leq p \leq 4$. The OD enrichment functions are used at nodes on the crack front as before. Figure 20 shows the results of p -extensions on Mesh 1. Each curve in the figure corresponds to a *single* integration rule pair. Thus, in contrast with previous convergence plots, the integration rule is fixed for each curve and the discretization is changed by increasing the p order of the shape functions. The strain energy and energy norm are plotted with respect to the number of DOFs of Mesh 1. One can observe that integration rule pairs 2.3 and 5.1 deliver exponential convergence rates and virtually the same error values for strain energy as well as energy norm. Rule pair 1.5 leads to a slight deviation from curves for pairs 2.3 and 5.1 when the polynomial order of the approximation is greater than 3 and shows a faster convergence rate. This small deviation is due to integration errors. The difference in the relative error of strain energy for pairs 1.5 and 5.1, when $p=4$, is around 0.01%. In the case of the relative error in energy norm, the difference is about 0.5%. These results show that pair 5.1 (based on the mapping method) is very efficient since it can achieve the same error level as pair 1.5 (based on Keast rules) but using far less integration points (cf. Table V).

In the case of integration rule pair 1.5, the integration over elements with singular enrichments is performed using Keast rules only. As shown in Figure 19, the integration errors when Keast rules are used may cancel out the discretization error. This behavior is also manifested in the results shown in Figure 20. The integration errors are improving the convergence rate of the curve for

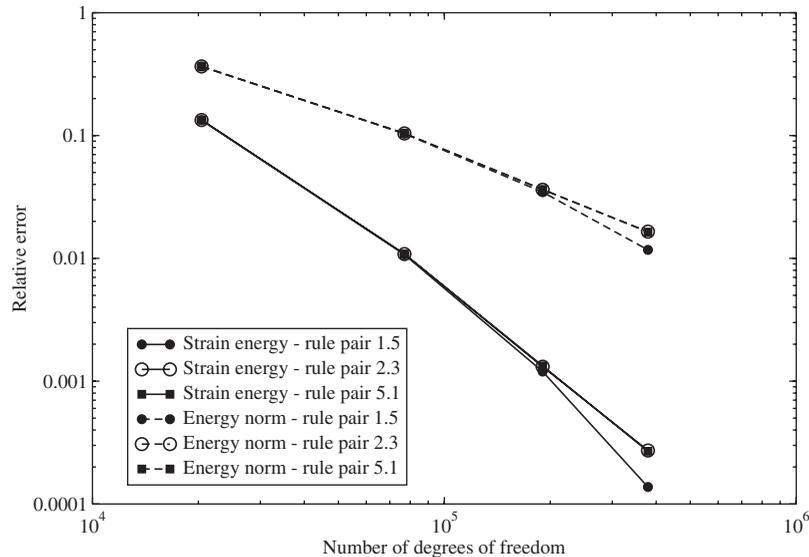


Figure 20. Convergence analysis of strain energy and energy norm with respect to the number of degrees of freedom for Mesh 1. Log–log scale. The p -version of the GFEM is used.

pair 1.5. This, however, is *not always* the case. Integration errors *can also deteriorate convergence rates*. To illustrate this point, we repeat the p -convergence analysis done above using Mesh 2, which is coarser than Mesh 1. Details on this mesh are listed in Table III.

Figure 21 presents the p -convergence analysis using Mesh 2. One can observe that the convergence rate when pair 1.5 is used deteriorates due to integration errors. In this case, integration errors when using integration rule pair 1.5 is greater than 0.1% in strain energy. In contrast with the results for Mesh 1, the integration errors on Mesh 2 lead to an overestimation of the strain energy when $p \geq 3$. The volume of the elements enriched with singular functions is larger in Mesh 2 than in Mesh 1 and this introduces a larger error in the overall solution when pair 1.5 is used. Therefore, when using pair 1.5 with a coarse mesh, one should increase the number of integration points for elements with singular enrichments according to the target error level that is required for the analysis in order to make the integration errors negligible.

Figure 21 also presents reference curves for strain energy and energy norm. These reference curves are computed with integration rule $M.15$ which uses 1331 integration points per element enriched with singular functions. We can observe that pairs 2.3 and 5.1 show virtually the same behavior as the reference curve. By using integration rules $M.1$ with just 18 integration points, it is possible to achieve the same level of accuracy as that of using a rule with 1331 integration points. Rule $M.1$ presents the best compromise between accuracy and computational cost among the integration strategies analyzed in this section. Of course, these conclusions may only be valid for the level of accuracy considered here (of the order of 0.01% or larger in energy).

4.3.3. Mixed-mode analysis—effect of integration errors on SIFs. This example considers the same model analyzed in Section 4.3.1 but subjected to Neumann boundary conditions from the first term of Mode I and Mode II expansion of the elasticity solution under the condition of the same

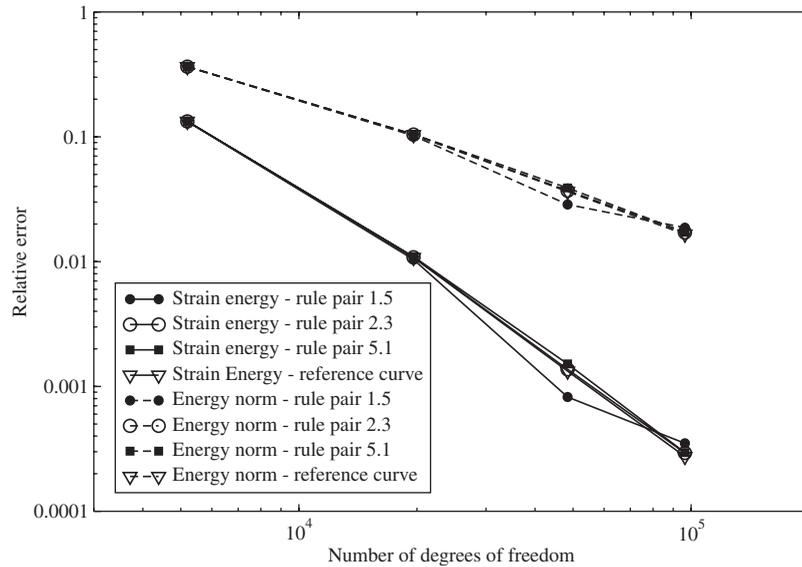


Figure 21. Convergence analysis of strain energy and energy norm with respect to the number of degrees of freedom for Mesh 2. Log–log scale.

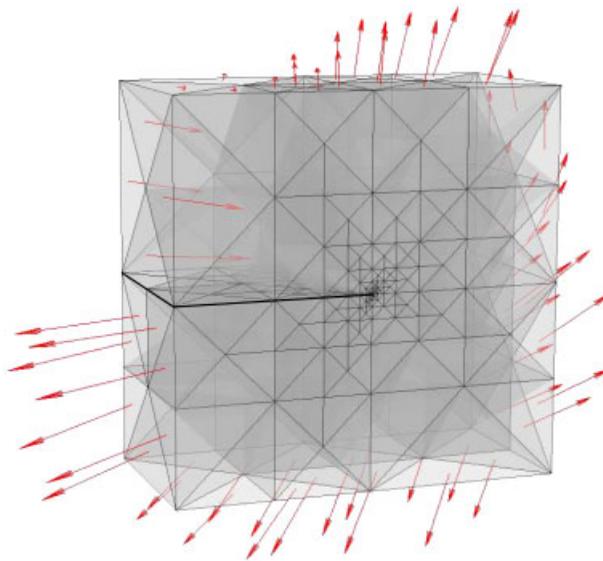


Figure 22. Edge-cracked panel with Mode I and Mode II boundary conditions from asymptotic expansion.

Mode I and Mode II stress intensity factors (i.e. $K_I = K_{II}$). Like in the case solved in Section 4.3.1, the strain energy as well as the stress intensity factors for this problem can be analytically defined. Figure 22 illustrates the mesh used in the analysis and the boundary conditions applied.

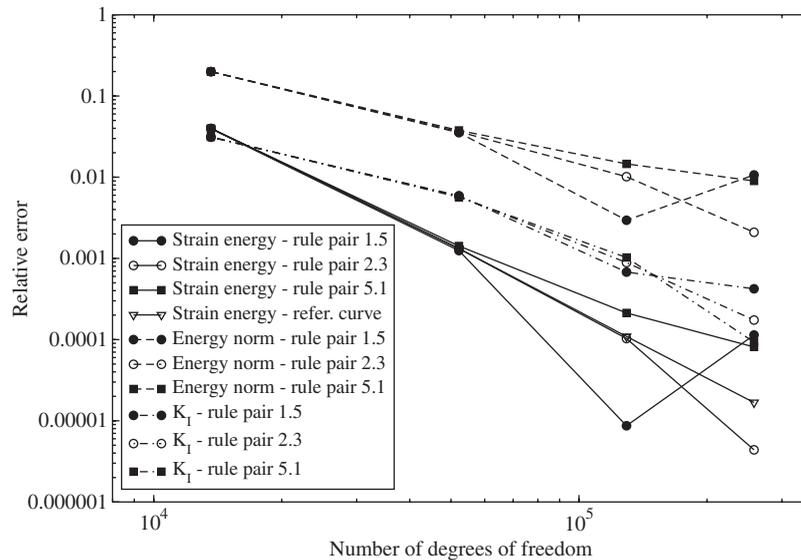


Figure 23. Convergence analysis of relative error in strain energy, energy norm and Mode I stress intensity factor with respect to the number of degrees of freedom. Log–log scale.

The main objective of this example is to investigate the performance of the integration rule pairs 1.5, 2.3 and 5.1 (cf. Table IV) in a mixed-mode fracture problem. We perform p -extensions on a tetrahedral mesh locally refined around the crack front as illustrated in Figure 22. The polynomial order of shape functions ranges from $p=1$ to 4. Like in previous sections, crack front nodes are also enriched with the 3D counterpart of the OD branch functions (1). The average ratio of element size to characteristic crack length is $L_e/a=1.7 \times 10^{-2}$. This ratio is representative of GFEM meshes used in Reference [35] to solve fracture mechanics problems. This discretization leads to 896 elements with edge singularity and 2180 elements with nodal singularity. The total number of integration points for the elements with singular enrichment is 138420, 95328 and 55368 for integration rule pairs 1.5, 2.3 and 5.1, respectively.

Figures 23 and 24 present the p -convergence analysis for the relative error in strain energy and energy norm compared with the relative error for Mode I and Mode II stress intensity factors, respectively. Both Figures also present a reference curve for the relative error in the strain energy. The reference curve is again obtained by using integration rule $M.15$ with 1331 integration points per element with singular enrichment. One can observe that, for the discretizations used in this example, the integration errors start to affect the convergence of the solution when the strain energy error is smaller than 0.1% in the case of pair 1.5 and about 0.01% in the case of pairs 2.3 and 5.1. Strategies 1.5 is highly affected by integration errors at strain energy error levels below 0.01%. The effect of integration errors is more pronounced in the case of pair 1.5 because no singular integration scheme is applied in this case. Integration pair 2.3 uses the mapping method for elements with an edge on the crack front and, therefore, leads to smaller integration errors than integration pair 1.5. Integration errors improve the convergence rate for rule pair 2.3 while it significantly deteriorates the convergence for rule pair 1.5. The difference in the relative error of strain energy between the reference curve and that of integration pair 5.1 is around 0.01% for

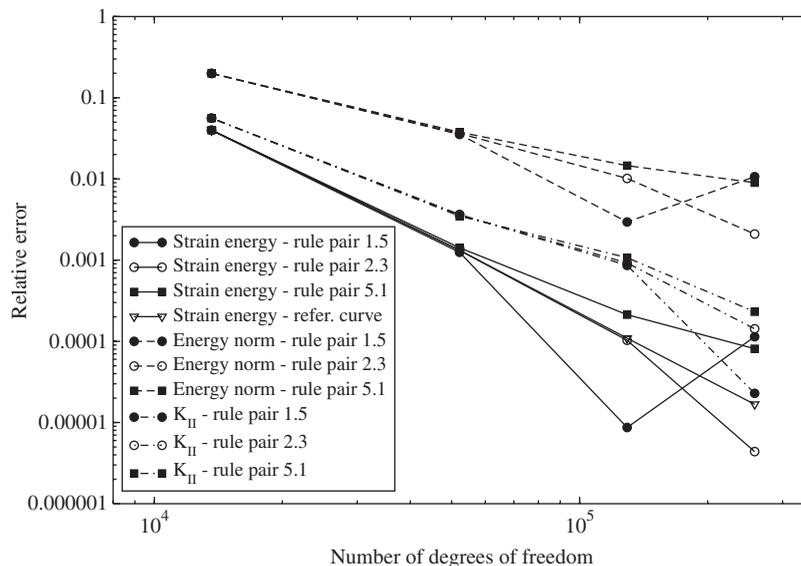


Figure 24. Convergence analysis of relative error in strain energy, energy norm and Mode II stress intensity factor with respect to the number of degrees of freedom. Log-log scale.

$p=3$ and 0.006% for $p=4$. These results show the efficiency of the proposed mapping method since it can achieve acceptable error levels by using significantly fewer integration points than the other rules considered here.

Mode I and II stress intensity factors show good agreement with analytical values. We can observe that both quantities exhibit super-convergence, i.e. their convergence rates are comparable with the convergence rates for strain energy [44–47]. Furthermore, the error levels for SIFs are always within the relative error in energy norm and the relative error in strain energy. We can also observe that the stress-intensity factors are less affected by integration errors. Integration errors start to affect the convergence of the solution when the SIF error is smaller than 0.1%, which happens only for $p=4$. This results suggest that, if the target error level of the analysis is less than 0.1% in SIF, one should use more integration points for the element with singular enrichments or a more refined mesh. In other words, the results indicate that the accuracy of the quadrature used in the GFEM/XFEM must be consistent with the target error level of quantities of interest. This is in contrast with, e.g. the h -version of the FEM where a single quadrature rule can provide exact integration and thus can be used for any target error level. Our results are, nevertheless, consistent with the numerical integration for meshless methods as demonstrated numerically and theoretically by Babuška *et al.* in [48].

5. CONCLUSIONS

A mapping method is developed to integrate weakly singular functions in the GFEM/X-FEM. The method is applicable for both 2D and 3D problems including arbitrarily shaped triangles and tetrahedra. For 2D problems, singularity occurs at a node, whereas for 3D problems, singularity

occurs either at a node or along an edge. The mapping method is able to handle all above cases. Moreover, one can easily implement the method in existing codes by replacing a standard integration rule with a quadrature rule generated by the mapping method for elements that have a singularity. Therefore, the method does not require extra computational cost.

The performance of the mapping method is assessed by three classes of examples: theoretical assessment, integration of branch functions, and 3D fracture problems. In the theoretical assessment, convergence of numerical integrations of scalar functions is investigated through the comparison of the mapping method with standard integration rules. The weights of a quadrature rule generated by the mapping method are scaled so that the weights add to 1.0, which leads to the exact integration of a constant function. The integration of branch functions demonstrates that the mapping method provides exponential convergence while the standard integration leads to algebraic convergence. In 3D fracture problems, this study investigates the effect of integration errors on quantities of interest such as the strain energy, the energy norm of the solution, and the Mode I and Mode II stress intensity factors. Several numerical integration strategies are presented, which are the combinations of Keast rules and the mapping method. The computational results illustrate that the mapping method reduces integration error significantly using far fewer integration points compared with standard integration rules (e.g. Keast rules). The accurate numerical integration (e.g. the mapping method) leads to a monotonic convergence in the case of p -extensions while integration errors can result in loss of convergence.

While in this paper we focus on the development and application of the mapping method to the GFEM/X-FEM method, the proposed quadrature rules are not limited to this case. The so-called enriched FEM proposed in the 1970s [49, 50] and further developed in, e.g. [51] also use singular functions from the asymptotic expansion of the elasticity solution in the neighborhood of cracks. Therefore, it also can benefit from the quadrature rules proposed here.

APPENDIX A

The following MATLAB code (2D_Singular_IntegrationPointsWeights) generates the integration points and weights for the integration of a weakly singular function in a master triangle—see Section 3.1.

```
function [r, s, wgt, n] = 2D_Singular_IntegrationPointsWeights (nIntPoints1D)
%-----
% K. Park, J.P. Pereira, C.A. Duarte, G.H. Paulino, 2008, Integration of
% singular enrichment functions in the generalized/extended finite element
% method for three-dimensional problems, International Journal for Numerical
% Methods in Engineering, X(X), XXX-XXX.
%-----
% Input parameters: nIntPoints1D
%   nIntPoints1D: The number of standard integration points in 1D
% Output parameters: r, s, wgt, n
%   r, s: Coordinates of the singular integration points in 2D
%   wgt : Weights of each singular integration point
%   n   : The total number of the singular integration points
% External functions
%   StandardGaussRuleloc: Provide the standard Gauss points
%   StandardGaussRulewgt: Provide weights of the standard Gauss points
```

```

% Variables
%   J      : Jacobian associated with rho-theta coord and ksi-eta coord
%   Area   : Area of a triangle
%   N      : Linear shape functions
%   ksi, eta : Standard integration points
%   iwgt, jwgt : Weights of the standard integration points
%   rho, theta : Integration points in rho-theta coord
%   Coord_rho_theta: Integration domain of rho-theta coord
%-----
n = 0;
J = pi/8;
Area = 0.5;
Coord_rho_theta = [0 , 0; 1, 0; 1, pi/2; 0, pi/2];
for ipt = 1 : nIntPoints1D
for jpt = 1 : nIntPoints1D
    % Obtain standard integration points and weights in the ksi-eta coord
    % with respect to the number of integration points in 1D (nIntPoints1D)
    ksi = StandardGaussRuleloc(nIntPoints1D, ipt);
    eta = StandardGaussRuleloc(nIntPoints1D, jpt);
    iwgt = StandardGaussRulewgt(nIntPoints1D, ipt);
    jwgt = StandardGaussRulewgt(nIntPoints1D, jpt);

    % Define linear shape functions
    N(1) = 0.25 * (1. - ksi)*(1. - eta);
    N(2) = 0.25 * (1. + ksi)*(1. - eta);
    N(3) = 0.25 * (1. + ksi)*(1. + eta);
    N(4) = 0.25 * (1. - ksi)*(1. + eta);

    % Integration points in the rho-theta coordinates
    rho = N*Coord_rho_theta(:,1);
    theta = N*Coord_rho_theta(:,2);

    % Integration points and weight in the r-s coordinates
    n = n + 1;
    r(n) = rho*cos(theta)*cos(theta);
    s(n) = rho*sin(theta)*sin(theta);
    wgt(n) = sin(2*theta)*rho*J*iwgt*jwgt/Area;
end
end

```

APPENDIX B: QUADRATURE RULES FOR TRIANGULAR AND TETRAHEDRAL ELEMENTS

Quadrature rules for triangular elements with singularity at a node are listed in Table BI, and quadrature rules for tetrahedral elements with singularity along an edge are listed in Table BII. The coordinates are given in the master area (\bar{r}, \bar{s}) and volume $(\bar{r}, \bar{s}, \bar{t})$ coordinate system, illustrated in Figures 7(b) and 8(b), respectively. For triangular elements, the singularity occurs at a node that corresponds to an origin in the area coordinate system. For tetrahedral elements, it is assumed that the line singularity is along the coordinate axis \bar{t} as illustrated in Figure 8(b). Higher-order rules can be obtained by contacting the authors.

Table BI. Transformed integration points and weights.

n_q	\bar{r}_i	\bar{s}_i	\bar{W}_i
1	0.25	0.25	1.5707963267949
4	0.08375842651774	$\sum \bar{W}_i$	1.5707963267949
	0.70491670807708	0.70491670807708	0.38168319089731
	0.02244300274473	0.08375842651774	0.38168319089731
	0.18888186266046	0.18888186266046	0.10227170276547
		$\sum \bar{W}_i$	0.10227170276547
9	0.25	0.25	0.96790978732557
	0.01550703374423	0.48449296625577	0.3102807559101
	0.48449296625577	0.01550703374423	0.06723616800137
	0.44364916731037	0.44364916731037	0.06723616800137
	0.02751873043233	0.85977960418841	0.34413949747987
	0.85977960418841	0.02751873043233	0.07457317486736
	0.05635083268963	0.05635083268963	0.07457317486736
	0.00349533705613	0.10920632832312	0.04371144740776
	0.10920632832312	0.00349533705613	0.00947203513434
	$\sum \bar{W}_i$	0.00947203513434	
16	0.16447738503377	0.50551313675866	1.0006944568039
	0.50551313675866	0.16447738503377	0.19263169249859
	0.00793785538829	0.66205266640414	0.19263169249859
	0.66205266640414	0.00793785538829	0.02583212102757
	0.08101472221835	0.24899475598922	0.02583212102757
	0.24899475598922	0.08101472221835	0.09488236364543
	0.00390985757197	0.3260996206356	0.09488236364543
	0.3260996206356	0.00390985757197	0.01272382892596
	0.22844713750833	0.70212101828869	0.01272382892596
	0.70212101828869	0.22844713750833	0.14271218426271
	0.01102510439984	0.91954305139719	0.14271218426271
	0.91954305139719	0.01102510439984	0.01913786027712
	0.01704496974379	0.05238687445918	0.01913786027712
	0.05238687445918	0.01704496974379	0.01064808642104
	0.00082260856042	0.06860923564255	0.01064808642104
	0.06860923564256	0.00082260856042	0.00142792005600
	$\sum \bar{W}_i$	0.00142792005600	
		$\sum \bar{W}_i$	0.99999211422886

Table BII. Transformed integration points and weights for tetrahedral with edge singularity.

n_q	\bar{r}_i	\bar{s}_i	\bar{t}_i	\hat{W}_i
1	0.1666666666666666	0.1666666666666666	0.3333333333333332	1.57079632679489
6			$\sum \hat{W}_i$	1.57079632679489
	0.14896642845625587	0.01770023821041065	0.66666666666666696	0.08065914894379719
	0.01770023821041067	0.14896642845625585	0.66666666666666696	0.08065914894379723
	0.59586571382502429	0.07080095284164270	0.16666666666666651	0.32263659577518921
	0.07080095284164279	0.59586571382502417	0.16666666666666651	0.32263659577518943
	0.14896642845625587	0.01770023821041065	0.16666666666666651	0.08065914894379719
0.01770023821041067	0.14896642845625585	0.16666666666666651	0.08065914894379723	
		$\sum \hat{W}_i$	$\sum \hat{W}_i$	0.967909787325567

Table BII. *Continued.*

8	0.29793285691251170	0.03540047642082130	0.3333333333333298	-0.27222462768531585
	0.03540047642082134	0.29793285691251164	0.3333333333333298	-0.27222462768531596
	0.17875971414750718	0.02124028585249280	0.5999999999999997	0.15123590426961994
	0.02124028585249282	0.17875971414750715	0.5999999999999997	0.15123590426962002
	0.53627914244252161	0.06372085755747840	0.1999999999999998	0.45370771280885979
	0.06372085755747848	0.53627914244252150	0.1999999999999998	0.45370771280886007
	0.17875971414750718	0.02124028585249280	0.1999999999999998	0.15123590426961994
	0.02124028585249282	0.17875971414750715	0.1999999999999998	0.15123590426962002
		$\sum \hat{W}_i$	0.967909787325567	
18	0.43211781432231888	0.01383067659364610	0.10810301816806999	0.04521044714518826
	0.22297424545798252	0.22297424545798247	0.10810301816806999	0.20863669260505907
	0.01383067659364609	0.43211781432231893	0.10810301816806999	0.04521044714518824
	0.10475030386689888	0.00335271430117109	0.44594849091596500	0.01095952983064100
	0.05405150908403500	0.05405150908403499	0.44594849091596500	0.05057592217631448
	0.00335271430117109	0.10475030386689891	0.44594849091596500	0.01095952983064099
	0.43211781432231888	0.01383067659364610	0.44594849091596500	0.04521044714518826
	0.22297424545798252	0.22297424545798247	0.44594849091596500	0.20863669260505907
	0.01383067659364609	0.43211781432231893	0.44594849091596500	0.04521044714518824
	0.08873606264364056	0.00284015086612994	0.81684757298045895	0.00456974040704402
	0.04578810675488526	0.04578810675488525	0.81684757298045895	0.02108838962657404
	0.00284015086612995	0.08873606264364056	0.81684757298045895	0.00456974040704403
	0.79151380722425546	0.02533376575620341	0.09157621350977052	0.04076147306796418
	0.40842378649022953	0.40842378649022942	0.09157621350977052	0.18810561415815038
	0.02533376575620345	0.79151380722425546	0.09157621350977052	0.04076147306796421
	0.08873606264364056	0.00284015086612994	0.09157621350977052	0.00456974040704402
	0.04578810675488526	0.04578810675488525	0.09157621350977052	0.02108838962657404
	0.00284015086612995	0.08873606264364056	0.09157621350977052	0.00456974040704403
		$\sum \hat{W}_i$	1.000694456803870	
21	0.32299531083717852	0.01033802249615442	0.3333333333333298	0.03403831005069069
	0.16666666666666651	0.16666666666666646	0.3333333333333298	0.15707963267948948
	0.01033802249615444	0.32299531083717852	0.3333333333333298	0.03403831005069071
	0.45556104639979350	0.01458101770532148	0.05971587178977000	0.02824910162929163
	0.23507103205255752	0.23507103205255744	0.05971587178977000	0.13036365497718521
	0.01458101770532146	0.45556104639979355	0.05971587178977000	0.02824910162929161
	0.05786383971194965	0.00185203207782034	0.47014206410511499	0.00358810636159924
	0.02985793589488500	0.02985793589488499	0.47014206410511499	0.01655835523137294
	0.00185203207782034	0.05786383971194965	0.47014206410511499	0.00358810636159924
	0.45556104639979350	0.01458101770532148	0.47014206410511499	0.02824910162929163
	0.23507103205255752	0.23507103205255744	0.47014206410511499	0.13036365497718521
	0.01458101770532146	0.45556104639979355	0.47014206410511499	0.02824910162929161
	0.09814520074965608	0.00314130657380041	0.79742698535308698	0.00578920842139905
	0.05064325366172826	0.05064325366172824	0.79742698535308698	0.02671597770230424
	0.00314130657380041	0.09814520074965610	0.79742698535308698	0.00578920842139905
	0.77269553101222443	0.02473145434086247	0.10128650732345650	0.04557834148939840
	0.39871349267654354	0.39871349267654343	0.10128650732345650	0.21033444752788957
	0.02473145434086246	0.77269553101222465	0.10128650732345650	0.04557834148939839
0.09814520074965608	0.00314130657380041	0.10128650732345650	0.00578920842139905	
0.05064325366172826	0.05064325366172824	0.10128650732345650	0.02671597770230424	
0.00314130657380041	0.09814520074965610	0.10128650732345650	0.00578920842139905	
		$\sum \hat{W}_i$	1.000694456803870	

Table BII. Continued.

n_q	\bar{r}_i	\bar{s}_i	\bar{t}_i	\hat{W}_i
48	0.24633326736932920	0.00295347780158129	0.50142650965817903	0.00516368078880538
	0.18808881678888075	0.06119792838202974	0.50142650965817903	0.03850588067500381
	0.06119792838202972	0.18808881678888075	0.50142650965817903	0.03850588067500381
	0.00295347780158129	0.24633326736932920	0.50142650965817903	0.00516368078880538
	0.49548575230108621	0.00594075735709286	0.24928674517091048	0.01038645850642608
	0.37833025917011581	0.12309625048806321	0.24928674517091048	0.07745245073076008
	0.12309625048806319	0.37833025917011581	0.24928674517091048	0.07745245073076008
	0.00594075735709287	0.49548575230108621	0.24928674517091048	0.01038645850642609
	0.24633326736932920	0.00295347780158129	0.24928674517091048	0.00516368078880538
	0.18808881678888075	0.06119792838202974	0.24928674517091048	0.03850588067500381
	0.06119792838202972	0.18808881678888075	0.24928674517091048	0.03850588067500381
	0.00295347780158129	0.24633326736932920	0.24928674517091048	0.00516368078880538
	0.06234155395686129	0.00074746053464071	0.87382197101699588	0.00056894408398100
	0.04760115937952347	0.01548785511197853	0.87382197101699610	0.00424265052479945
	0.01548785511197853	0.04760115937952347	0.87382197101699610	0.00424265052479945
	0.00074746053464071	0.06234155395686128	0.87382197101699588	0.00056894408398100
	0.86346917912602194	0.01035279189097402	0.06308901449150200	0.00788022835464816
	0.65930557398883038	0.21451639702816566	0.06308901449150200	0.05876334055615853
	0.21451639702816560	0.65930557398883038	0.06308901449150200	0.05876334055615853
	0.01035279189097407	0.86346917912602172	0.06308901449150200	0.00788022835464818
	0.06234155395686129	0.00074746053464071	0.06308901449150200	0.00056894408398100
	0.04760115937952347	0.01548785511197853	0.06308901449150200	0.00424265052479945
	0.01548785511197853	0.04760115937952347	0.06308901449150200	0.00424265052479945
	0.00074746053464071	0.06234155395686128	0.06308901449150200	0.00056894408398100
	0.30667548427742402	0.00367696675636000	0.05314504984481699	0.00456059725198525
	0.23416337383863913	0.07618907719514486	0.05314504984481700	0.03400865018077964
	0.07618907719514483	0.23416337383863913	0.05314504984481700	0.03400865018077964
	0.00367696675636001	0.30667548427742402	0.05314504984481699	0.00456059725198525
	0.05251540254899713	0.00062964729581986	0.63650249912139900	0.00078096102502669
	0.04009835956839377	0.01304669027642322	0.63650249912139900	0.00582367370707741
	0.01304669027642321	0.04009835956839377	0.63650249912139900	0.00582367370707741
	0.00062964729581986	0.05251540254899713	0.63650249912139900	0.00078096102502669
	0.62896140021332347	0.00754109890807558	0.31035245103378400	0.00935333856299657
	0.48024615934084435	0.15625633978055461	0.31035245103378400	0.06974841268276395
	0.15625633978055458	0.48024615934084435	0.31035245103378400	0.06974841268276395
	0.00754109890807559	0.62896140021332347	0.31035245103378400	0.00935333856299657
	0.62896140021332347	0.00754109890807558	0.05314504984481699	0.00935333856299657
	0.48024615934084435	0.15625633978055461	0.05314504984481700	0.06974841268276395
	0.15625633978055458	0.48024615934084435	0.05314504984481700	0.06974841268276395
	0.00754109890807559	0.62896140021332347	0.05314504984481699	0.00935333856299657
	0.30667548427742402	0.00367696675636000	0.63650249912139900	0.00456059725198525
	0.23416337383863913	0.07618907719514486	0.63650249912139900	0.03400865018077964
	0.07618907719514483	0.23416337383863913	0.63650249912139900	0.03400865018077964
	0.00367696675636001	0.30667548427742402	0.63650249912139900	0.00456059725198525
	0.05251540254899713	0.00062964729581986	0.31035245103378400	0.00078096102502669
	0.04009835956839377	0.01304669027642322	0.31035245103378400	0.00582367370707741
	0.01304669027642321	0.04009835956839377	0.31035245103378400	0.00582367370707741
	0.00062964729581986	0.05251540254899713	0.31035245103378400	0.00078096102502669
			$\sum \hat{W}_i$	0.999992114228862

The weights in the table were divided by the area (or volume) of the master triangular (or tetrahedral) element used in our computations. They should be properly scaled back prior to use. In our computations, we also scale the weights such that, for each quadrature rule, they add to 1.0 (cf. Section 3.3).

ACKNOWLEDGEMENTS

This paper resulted from the BEM class taught by Prof. Paulino [25] during the Fall/2004, and the Advanced FEM/GFEM class taught by Prof. Duarte [16] during the Fall/2007. We gratefully acknowledge the support from both the Department of Civil and Environmental Engineering (CEE) and Computational Science and Engineering (CSE) program at the University of Illinois at Urbana-Champaign (UIUC). C. A. Duarte also acknowledges the partial support of this work by the National Science Foundation under grant DMS-0611094.

REFERENCES

1. Babuska I, Melenk JM. The partition of unity method. *International Journal for Numerical Methods in Engineering* 1997; **40**(4):727–758.
2. Duarte CAM, Oden JT. An *hp* adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:237–262.
3. Oden JT, Duarte CA, Zienkiewicz OC. A new cloud-based *hp* finite element method. *Computer Methods in Applied Mechanics and Engineering* 1998; **153**:117–126.
4. Duarte CA, Babuška I, Oden JT. Generalized finite element methods for three dimensional structural mechanics problems. *Computers and Structures* 2000; **77**:215–232.
5. Strouboulis T, Copps K, Babuska I. Generalized finite element method: an example of its implementation and illustration of its performance. *International Journal for Numerical Methods in Engineering* 2000; **47**(8): 1401–1417.
6. Daux C, Moes N, Dolbow J, Sukumar N, Belytschko T. Arbitrary branched and intersecting cracks with the extended finite element method. *International Journal for Numerical Methods in Engineering* 2000; **48**(12): 1741–1760.
7. Sukumar N, Moes N, Moran B, Belytschko T. Extended finite element method for three-dimensional crack modelling. *International Journal for Numerical Methods in Engineering* 2000; **48**(11):1549–1570.
8. Moës N, Gravouil A, Belytschko T. Non-planar 3D crack growth by the extended finite element and level sets—part I: mechanical model. *International Journal for Numerical Methods in Engineering* 2002; **53**(11):2549–2568.
9. Gaul L, Kogl M, Wagner M. *Boundary Element Methods for Engineers and Scientists*. Springer: New York, 2003.
10. Sutradhar A, Paulino GH. A simple boundary element method for problems of potential in non-homogeneous media. *International Journal for Numerical Methods in Engineering* 2004; **60**(13):2203–2230.
11. Sutradhar A, Paulino GH, Gray LJ. *Symmetric Galerkin Boundary Element Method*. Springer: New York, 2008.
12. Oden JT, Duarte CA. Chapter: clouds, cracks and FEM's. In *Recent Developments in Computational and Applied Mechanics*, Reddy BD (ed.). International Center for Numerical Methods in Engineering, CIMNE: Barcelona, Spain, 1997; 302–321.
13. Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering* 1999; **45**:601–620.
14. Bechet E, Minnebo H, Moes N, Burgardt B. Improved implementation and robustness study of the X-FEM for stress analysis around cracks. *International Journal for Numerical Methods in Engineering* 2005; **64**(8): 1033–1056.
15. Laborde P, Pommier J, Renard Y, Salaun M. High-order extended finite element method for cracked domains. *International Journal for Numerical Methods in Engineering* 2005; **64**(3):354–381.
16. Duarte CA. *Lecture Notes: CEE 598 Advanced Finite Element Methods*, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Fall 2007.

17. Xiao QZ, Karihaloo BL. Improving the accuracy of XFEM crack tip fields using higher order quadrature and statically admissible stress recovery. *International Journal for Numerical Methods in Engineering* 2006; **66**(9):1378–1410.
18. Espelid TO, Genz A. The DECUHR: an algorithm for automatic integration of singular functions over a hyperrectangular region. *Numerical Algorithms* 1995; **8**(2–4):201–220.
19. Ventura G, Gracie R, Belytschko T. Fast integration and weight function blending in the extended finite element method. *International Journal for Numerical Methods in Engineering* 2009; **77**:1–29. DOI: 10.1002/nme.2387.
20. Casale MS, Sankar P. Misconceptions in the interpretation of singular integrals in mathematical physics. *International Journal for Numerical Methods in Engineering* 1993; **36**(7):1246–1247.
21. Huang Q, Cruse TA. Some notes on singular integral techniques in boundary element analysis. *International Journal for Numerical Methods in Engineering* 1993; **36**(15):2643–2659.
22. Bonnet M, Maier G, Polizzotto C. Symmetric Galerkin boundary element methods. *Applied Mechanics Reviews* 1998; **51**(11):669–704.
23. Sutradhar A, Paulino GH. Symmetric Galerkin boundary element computation of T-stress and stress intensity factors for mixed-mode cracks by the interaction integral method. *Engineering Analysis with Boundary Elements* 2004; **28**(11):1335–1350.
24. Sutradhar A, Paulino GH, Gray LJ. On hypersingular surface integrals in the symmetric Galerkin boundary element method: application to heat conduction in exponentially graded materials. *International Journal for Numerical Methods in Engineering* 2005; **62**(1):122–157.
25. Paulino GH. *Lecture Notes: CEE 598 Boundary Element Methods*. Department of Civil & Environmental Engineering, University of Illinois at Urbana-Champaign, Fall 2004.
26. Nagarajan A, Mukherjee S. A mapping method for numerical evaluation of two-dimensional integrals with $1/r$ singularity. *Computational Mechanics* 1993; **12**:19–26.
27. Lachat JC, Watson JO. Effective numerical treatment of boundary integral equations: a formulation for three-dimensional elastostatics. *International Journal for Numerical Methods in Engineering* 1976; **10**(5):991–1005.
28. Keast P. Moderate-degree tetrahedral quadrature formulas. *Computer Methods in Applied Mechanics and Engineering* 1986; **55**:339–348.
29. Dolbow J, Moes N, Belytschko T. Discontinuous enrichment in finite elements with a partition of unity method. *Finite Elements in Analysis and Design* 2000; **36**:235–260.
30. Duarte CA, Hamzeh ON, Liszka TJ, Tworzydło WW. A generalized finite element method for the simulation of three-dimensional dynamic crack propagation. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**(15–17):2227–2262. Available from: [http://dx.doi.org/10.1016/S0045-7825\(00\)00233-4](http://dx.doi.org/10.1016/S0045-7825(00)00233-4).
31. Pereira JP, Duarte CA, Jiao X, Guoy D. Generalized finite element method enrichment functions for curved singularities in 3D fracture mechanics problems. *Computational Mechanics*, accepted for publication. DOI: 10.1007/s00466-008-0356-1.
32. Duarte CA. The hp cloud method. *Ph.D. Thesis*, University of Texas at Austin, Austin, TX, U.S.A., 1996.
33. Moes N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering* 1999; **46**:131–150.
34. Simone A. Partition of unity-based discontinuous elements for interface phenomena: computational issues. *Communications in Numerical Methods in Engineering* 2004; **20**:465–478.
35. Pereira JP, Duarte CA, Guoy D, Jiao X. Hp-Generalized FEM and crack surface representation for non-planar 3-D cracks. *International Journal for Numerical Methods in Engineering* 2008. Available from: <http://dx.doi.org/10.1002/nme.2419>. DOI: 10.1002/nme.2419.
36. Holdych DJ, Noble DR, Secor RB. Quadrature rules for triangular and tetrahedral elements with generalized functions. *International Journal for Numerical Methods in Engineering* 2008; **73**(9):1310–1327.
37. Dunavant DA. High degree efficient symmetrical gaussian quadrature rules for the triangle. *International Journal for Numerical Methods in Engineering* 1985; **21**(6):1129–1148.
38. Duarte CA, Reno LG, Simone A. A high-order generalized FEM for through-the-thickness branched cracks. *International Journal for Numerical Methods in Engineering* 2007; **72**(3):325–351. Available from: <http://dx.doi.org/10.1002/nme.2012>.
39. Anderson TL. *Fracture Mechanics: Fundamentals and Applications*. CRC Press: Boca Raton, 1995.
40. Paulino GH, Kim J-H. A new approach to compute T-stress in functionally graded materials by means of the interaction integral method. *Engineering Fracture Mechanics* 2004; **71**(13–14):1907–1950.
41. Shim D-J, Paulino GH, Dodds Jr RH. A boundary layer framework considering material gradation effects. *Engineering Fracture Mechanics* 2006; **73**(5):593–615.

42. Raju JC, Newman Jr IS. Three dimensional finite-element analysis of finite-thickness fracture specimens. *Report TN D-8414*, NASA—Langley Research Center, Hampton, VA, May 1977; 1–40.
43. Li S, Mear ME, Xiao L. Symmetric weak-form integral equation method for three-dimensional fracture analysis. *Computer Methods in Applied Mechanics and Engineering* 1998; **151**:435–459.
44. Szabo B, Babuška I. *Finite Element Analysis*. Wiley: New York, 1991.
45. Szabo BA, Babuška I. Computation of the amplitude of stress singular terms for cracks and reentrant corners. In *Fracture Mechanics: Nineteenth Symposium, ASTM STP 969*, Cruse TA (ed.). Southwest Research Institute: San Antonio, TX, 1988; 101–124.
46. Pereira JP, Duarte CA. Extraction of stress intensity factors from generalized finite element solutions. *Engineering Analysis with Boundary Elements* 2005; **29**:397–413.
47. Pereira JP, Duarte CA. The contour integral method for loaded cracks. *Communications in Numerical Methods in Engineering* 2006; **22**(5):421–432. Available from: <http://dx.doi.org/10.1002/cnm.824>.
48. Babuška I, Banerjee U, Osborn JE, Li Q. Quadrature for meshless methods. *International Journal for Numerical Methods in Engineering* 2008; **76**:1434–1470.
49. Benzeley SE. Representation of singularities with isoparametric finite elements. *International Journal for Numerical Methods in Engineering* 1974; **8**:537–545.
50. Gifford LN, Hilton PD. Stress intensity factors by enriched finite elements. *Engineering Fracture Mechanics* 1978; **10**(3):485–496.
51. Ayhan AO, Nied HF. Stress intensity factors for three-dimensional surface cracks using enriched finite elements. *International Journal for Numerical Methods in Engineering* 2002; **54**:899–921.