Displacement based finite element formulations over polygons: a comparison between Laplace interpolants, strain smoothing and scaled boundary polygon formulation

Sundararajan Natarajan · Ean Tat Ooi · Irene Chiong · Chongmin Song

Received: date / Accepted: date

Abstract Three different displacement based finite element formulations over arbitrary polygons are studied in this paper. The formulations considered are: the conventional polygonal finite element method (FEM) with Laplace interpolants, the cell-based smoothed polygonal FEM with simple averaging technique and the scaled boundary polygon formulation. For the purpose of numerical integration, we employ the sub-triangulation for the polygonal FEM and classical Gaussian quadrature for the smoothed FEM and for the scaled boundary polygon formulation. The accuracy and the convergence properties of these formulations are studied with a few benchmark problems in the context of linear elasticity and the linear elastic fracture mechanics. The extension of scaled boundary polygon to higher order polygons is also discussed.

Keywords Polygonal finite elements · scaled boundary polygon formulation · strain projection · patch test · linear elastic fracture mechanics · generalized stress intensity factor.

1 Introduction

The finite element method (FEM) is a versatile technique for the numerical approximation of solutions of partial differential equations (PDEs). Traditional FEM simulations rely on strictly tetrahedral or hexahedral meshes in 3D (or triangular, quadrilateral meshes in 2D). In generating a FE mesh, a balance is required between the accuracy and the flexibility in the mesh generation. For example, triangulation of a domain is relatively easy when compared with quadrangulation, whilst the quadrilateral mesh is more accurate than the triangular mesh. It is relatively easy to construct interpolants over standard shapes, viz., triangles and tetraehdral. The use of standard shapes, viz., triangles
(or quadrilaterals) and tetrahedrals (or hexahedral) simplifies the approach, however, allowing only a few element shapes can be too restrictive, because

- it may require sophisticated meshing algorithm to generate high-quality meshes, esp. with quadrilaterals, for meshing complex geometries;
- it may require complex remeshing to capture topological changes, for instance due to discontinuous surface growth.

1.1 Background

With a goal to decrease the constraints imposed on finite element meshes different methods have been introduced. For instance, the meshfree methods [1], the partition of unity methods (PUMs) [2,3], the smoothed finite element methods (SFEM) [4,5] and the recent, isogeometric analysis [6]. However, the PUMs and the SFEM still require the domain to be discretized using a combination of triangles and quadrilaterals in 2D, unless, the boundary is defined implicitly. This causes additional difficulties in imposing boundary conditions and ensuring an accurate definition of the boundary of the domain [7,8]. The introduction of the isogeometric analysis has revolutionized the analysis procedure. It circumvents the need to discretize the domain with standard shapes and provides a natural link with the CAD models.

On another, related front, generalizations of FEM on arbitrary polygonal and polyhedral meshes have been the subject of increasing attention in the research community, both in computational physics [9,10,11,12,13,14] and in computer graphics [15,16].

**Polygonal FEM** In polygonal finite elements, the use of elements with more than four sides can provide flexibility, especially in meshing and accuracy [17]. Ghosh et al., [9] developed the Voronoi cell finite element model (VCFEM) to model the mechanical response of heterogeneous microstructures of composites and porous materials with heterogeneities. The VCFEM is based on the assumed stress hybrid formulation and was further developed by Tiwary et al., [18] to study the behaviour of microstructures with irregular geometries. Rashid and Gullet [19] proposed a variable element topology finite element method (VETFEM), in which the shape functions are constructed using a constrained minimization procedure. Sukumar [20] used Voronoi cells and natural neighbor interpolants to develop a finite difference method on unstructured grids. Biabanaki and Khoei [21] employed the polygonal FEM technique to model the large deformation response of interfaces. In their approach, the polygonal finite elements were generated from a non-conforming regular mesh. Wachspress interpolants were employed over the polygons. Kraus et al., [14] presented a polygonal FEM based on the constrained adaptive Delaunay tessellation. The polygonal elements can also be used as transition elements to simplify meshing or to describe the microstructure of polycrystalline alloys [11,12,13] in a rather straightforward manner. However, approximation functions on polygonal elements are usually non-polynomial, which introduces difficulties in numerical integration. Improving numerical integration over polytopes have gained increasing attention [17,22,23,24].

**Polygonal SFEM** In the stabilized conforming nodal integration [25], the strain is written as the divergence of a spatial average of the standard (compatible) strain field - i.e., the symmetric gradient of the displacement field. This concept was incorporated into the FEM by Liu [26] and extensively studied in [27,5,28]. Depending on the number and geometry of the subcells used, a spectrum of methods, each exhibiting a set of unique properties. Interested readers are referred to the literature [4] and references therein. Generalizations of the SFEM to arbitrarily shaped polygons were reported in [29,28].

**Scaled boundary polygon** Wolf and Song [30] introduced the scaled boundary finite element method (SBFEM) for elasto-statics and elasto-dynamic problems. The SBFEM reduces the governing PDE to a set of ordinary differential equations. Like the FEM, no fundamental solution is required and like the boundary element method, the spatial dimension is reduced by one, since only the boundary need to be discretized, resulting in a decrease in the total degrees of freedom. The SBFEM relies on defining a ‘scaling center’ from which the entire domain is visible. By exploiting the unique feature of the scaling center, the method allows the computation of stress intensity factor directly from their definitions. Natarajan and Song [31] combined the extended FEM and the SBFEM, thus, circumventing the need to know a priori the enrichment functions, required by the former. Recently, Ooi et al., [32] employed scaled
boundary formulation in polygonal elements to study crack propagation, but its performance and application in the context of linear elasticity was not studied.

1.2 Objective

The generalization of finite elements over polygonal and polyhedral elements is a subject of increasing attention in the research community. Apart from the aforementioned formulations, recent studies, among others include developing polygonal elements based on the virtual nodes [33] and the virtual element methods [34]. To the author’s knowledge, a comparison of different displacement based formulations over arbitrary polygons has not been reported yet in the literature. The main objective of the paper is to study the accuracy and the convergence properties of different displacement based finite element formulations, esp., the polygonal FEM, the polygonal SFEM and the scaled boundary polygon in the context of linear elasticity and linear elastic fracture mechanics. The shape functions used in these formulations are: Laplace interpolants for the polygonal FEM, simple averaging technique for the polygonal nSFEM and Gauss-Legendre shape functions for the scaled boundary polygon formulation. The scaled boundary polygons with higher order shape functions is also studied.

1.3 Outline

The paper is organized as follows. Section 2 presents the governing equations and weak form for 2D static elasticity problem. The different displacement based finite element formulations considered in this study are also discussed. In Section 3 with a few benchmark problems from linear elasticity, the accuracy and the convergence properties of various finite element techniques are studied. Later, the scaled boundary polygon formulation is applied to study problems in linear elastic fracture mechanics. The results from scaled boundary polygon formulation are compared with results available in the literature, for example with the extended FEM, followed by concluding remarks in the last section.

2 Overview of finite element techniques over polygons

2.1 Governing equations and weak form

For a 2D static elasticity problem defined in the domain $\Omega$ bounded by $\Gamma = \Gamma_u \cup \Gamma_t$, $\Gamma_u \cap \Gamma_t = \emptyset$, in the absence of body forces, the governing equation is given by:

$$\nabla_s \cdot \sigma = 0 \quad \text{in} \quad \Omega$$

(1)

with the following conditions prescribed on the boundary:

$$\begin{align*}
    u &= \bar{u} \quad \text{in} \quad \Gamma_u \\
    \sigma \cdot n &= \bar{t} \quad \text{on} \quad \Gamma_t
\end{align*}$$

(2)

where $\sigma$ is the stress tensor. The discrete equations for this problem are formulated from the Galerkin weak form:

$$\int_{\Omega} (\nabla_s u)^T D (\nabla_s \delta u) \, d\Omega - \int_{\Omega} (\delta u)^T b \, d\Omega - \int_{\Gamma_t} (\delta u)^T \bar{t} \, d\Gamma = 0$$

(3)

where $u$ and $\delta u$ are the trial and the test functions, respectively and $D$ is the material constitutive matrix. The FEM uses the following trial and test functions:

$$u^h(x) = \sum_{I=1}^{NP} N_I(x) d_I, \quad \delta u^h(x) = \sum_{I=1}^{NP} N_I(x) \delta d_I$$

(4)
where $NP$ is the total number of nodes in the mesh, $\mathbf{N}$ is the shape function matrix and $\mathbf{d}_I$ is the vector of degrees of freedom associated with node $I$. Upon substituting Equation (4) into Equation (3) and invoking the arbitrariness of $\delta \mathbf{u}$, we obtain the following discretized algebraic system of equations:

$$\mathbf{K} \mathbf{d} = \mathbf{f}$$

(5)

where

$$\mathbf{K} = \int_{\Omega^h} \mathbf{B}^T \mathbf{D} \mathbf{B} \, d\Omega$$

$$\mathbf{f} = \int_{\Omega^h} \mathbf{N}^T \mathbf{b} \, d\Omega + \int_{\Gamma^h} \mathbf{N}^T \mathbf{t} \, d\Gamma$$

(6)

where $\mathbf{K}$ is the stiffness matrix and $\Omega^h$ is the discretized domain formed by the union of elements $\Omega_e$. The stiffness matrix is computed over each element and assembled to the global matrix. The size of the stiffness matrix depends on the number of nodes in an element.

2.2 Generalization to arbitrary polygons

The growing interest in the generalization of FE over arbitrary meshes has opened up a new area of finite elements called ‘polygonal finite elements’. In polygonal finite elements, the number of sides of an element is not restricted to three or four as in the case of 2D.

**Polygonal mesh** The Voronoi tessellation is a fundamental geometrical construct to generate a polygonal mesh covering a given domain. Polygonal meshes can be generated from Voronoi diagrams. The Voronoi diagram is a subdivision of the domain into regions $V(p_I)$, such that any point in $V(p_I)$ is closer to node $p_I$ than to any other node. Figure (1) shows a Voronoi diagram of a point $P$. The first order Voronoi $V(N)$ is a subdivision of the Euclidean space $\mathbb{R}^2$ into convex regions, mathematically:

$$T_I = \{ \mathbf{x} \in \mathbb{R}^2 : d(\mathbf{x}, \mathbf{x}_I) < d(\mathbf{x}, \mathbf{x}_J) \forall J \neq I \}$$

(7)

where $d(\mathbf{x}_I, \mathbf{x}_J)$, the Euclidean matrix, is the distance between $\mathbf{x}_I$ and $\mathbf{x}_J$. The quality of the generated polygonal mesh depends on the randomness in the scattered points. Figure (2) shows a typical Voronoi tessellation of two sets of scattered data set. The quality of a polygonal mesh determines the accuracy of the solution [17]. To improve the
quality of the Voronoï tessellation, the generating point of each Voronoï cell can be used as its center of mass, leading to a special type of Voronoï diagram, called the centroidal Voronoï tessellation (CVT) \cite{35}. Sieger et al., \cite{36} presented an optimizing technique to improve the Voronoï diagrams for use in FE computations. A polygonal mesh can also be constructed from a triangular mesh by connecting the centroid of all the triangular elements circumventing a particular node \cite{32}. Figure (3) shows a sequence of steps required to generate a polygonal mesh from an underlying triangular mesh. This procedure is also adopted in the edge based smoothed FEM \cite{28}. In the following sub-sections we briefly discuss three different displacement based formulations.

2.3 Polygonal Finite Element Method (Polygonal FEM)

In polygonal finite elements, the domain is discretized with elements having arbitrary number of edges. The shape functions are constructed over each polygonal element. The construction of shape functions over arbitrary polygons can be broadly categorized as:

- Length and area measures (Wachspress interpolants) \cite{37,38}.
- Natural neighbour interpolants (Sibson, Laplace interpolant) \cite{39,40}.
- Harmonic coordinates \cite{16,41}.
- Maximum entropy approximants \cite{42}.
- Mean value coordinates \cite{15}.
For recent advances in polygonal elements, interested readers are referred to the literature [43] and the references therein. In this study, we consider only Laplace interpolants with isoparametric mapping. The Laplace interpolant is also called the natural neighbor interpolant [39,40,17]. It provides a natural weighting function for irregularly spaced nodes. For a point $P$ with $n$ natural neighbors, the Laplace shape functions for node $P_i$ can be written as [17]:

$$\phi_i(x) = \frac{\alpha_i(x)}{\sum_{j=1}^{n} \alpha_j(x)} , \quad \alpha_j(x) = \frac{s_j(x)}{h_j(x)} , \quad x \in \mathbb{R}^2$$

(8)

where $\alpha_i(x)$ is the Laplace weight function, $s_j(x)$ is the length of the Voronoi edge associated with $P$ and node $P_i$ and $h_j(x)$ is the Euclidean distance between $P$ and $P_i$ (see Figure 1). Listing 1 gives a MATLAB® function to compute the Laplace interpolants for a point $P$.  

Listing 1 Laplace Interpolant over convex polytope

```matlab
function [phi,dphi,dphiy]=shape(x,y,coord)
shapedenom=0; n=length(coord(:,1)); % number of sides of the polygon
phi = zeros(n,1); sumder = zeros(n,1);
for i=1:n %loop over number of vertices
    a1 = coord(i,1); a2 = coord(i,2);
    if(i==1)
        nodem = n; nodep = i+1;
    elseif(i==n)
        nodem = n-1; nodep = 1;
    else
        nodem = i-1; nodep = i+1;
    end
    %circumcenter of a triangle formed by points: (a1,b1,x) & (a2,b2,y)
    b1 = coord(nodem,1); b2 = coord(nodem,2);
    [v1,v2,v1x,v1y,v2x,v2y] = circum(b1,b2,a1,a2,x,y);
    %circumcenter of a triangle formed by points: (a1,c1,x) & (a2,c2,y)
    c1 = coord(nodep,1); c2 = coord(nodep,2);
    [s1,s2,s1x,s1y,s2x,s2y]=circum(a1,a2,c1,c2,x,y);
    SI = sqrt((v1-s1)^2+(v2-s2)^2); HI = sqrt((x-a1)^2+(y-a2)^2);
    SI_v1 = ((v1-s1)*(v1x-s1x) + (v2-s2)*(v2x - s2x))/SI;
    SI_v2 = ((v1-s1)*(v1y-s1y) + (v2-s2)*(v2y - s2y))/SI;
    HI_v1 = (x-a1)/HI; HI_v2 = (y-a2)/HI;
    shapenum = SI/HI;
    der(1) = (SI_v1 - shapenum*HI_v1)/HI;
    der(2) = (SI_v2 - shapenum*HI_v2)/HI;
    shapedenom = shapedenom + shapenum;
    sumder(1) = sumder(1) + der(1); sumder(2) = sumder(2) + der(2);
    shapefun(i) = shapenum;
    phix(i) = der(1); phiy(i) = der(2);
end
phi = shapefun./shapedenom;
dphix = (phix - phi.*sumder(1))/shapedenom;
dphiy = (phiy - phi.*sumder(2))/shapedenom;

function [v1,v2,v1x,v1y,v2x,v2y] = circum(a1,a2,b1,b2,x,y)
D = (a1-x)*(b2-y) - (b1-x)*(a2-y);
t1 = ((a1-x)*(a1x) + (a2-y)*(a2y))*b2-y) + (b1-x)*(a2-y))*(a1-x);
term1 = (1/2)*(t1-t2); v1 = term1/D;
t4 = ((b1-x)*(b1x) + (b2-y)*(b2y))*(a1-x);
t5 = ((a1-x)*(a1x) + (a2-y)*(a2y))*(b1-x);
term2 = (1/2)*(t4-t5); v2 = term2/D;
dx = a2-b2; dy = b1-a1;
term3 = (1/2)*((b1^2-a1^2)+(b2^2-a2^2));
```

1 When using the isoparametric formulation, the Wachspress interpolants and the Laplace interpolants are identical [17]. Hence, the interpretation of the results with the Laplace interpolants are applicable when Wachspress interpolants are used.

2 Original source code in FORTRAN is available from [44].
2.3.1 Numerical integration

The computation of the stiffness matrix and the force vectors involves evaluating the integrals given by Equation (6). The shape functions over arbitrary polygons are rational polynomials and hence the formulation of efficient integration rules poses a unique challenge. This has received much attention in the recent years [17,22,23,28] as evidenced by the growing literature. One potential solution is to sub-divide the physical element into triangles and then use the well-known quadrature on a triangle for numerical integration [17]. The purpose of subdivision is solely for the purpose of numerical integration and does not introduce additional unknowns. Although, straightforward, this process involves a two-level isoparametric mapping and relies on the positivity of the Jacobian matrix involved in the transformation. Lyness and Monegato [45] presented quadrature rules for regions with hexagonal symmetry. Natarajan et al., [22] proposed a numerical integration technique over arbitrary polygons based on complex mapping. This procedure eliminates the need for a two level isoparametric mapping while guaranteeing the positivity of the Jacobian, but is restricted to only 2D. Sommariva and Vianello [46] presented a Gauss-like cubature over arbitrary polygons. Mousavi et al., [23] presented a numerical algorithm based on group theory and optimization scheme to compute quadrature rules over arbitrary polygons.

Once the stiffness matrix and the force vector over each element is computed, then the conventional FE procedure is adopted for the assemblage process and the solution of the system of equations [17,22]. Listing 2 presents the MATLAB® function to compute the stiffness matrix. The shape functions are computed by Laplace interpolants using the routine given in Listing 1.

### Listing 2 Computation of the stiffness matrix using conventional Polygonal FEM

```matlab
function [kmat] = getKmatPolyFEM(element,node,matmtx,W,Q)
for iel = 1:size(element,1)
    econ = element{iel}; nn = length(econ); ndof=2;
    gindex = []; gindex=reshape([ndof*econ-1;ndof*econ],1,1);
    for igp = 1:size(W,1) % loop over integration points
        % shape function and derivatives in canonical domain
        [N,dphir,dphis] = shape(Q(igp,1),Q(igp,2),polycoord(length(econ)));
        JO = [dphir';dphis']*node(econ,:);
        dNdx = inv(JO)*[dphir';dphis'] ;
        % strain-displacement matrix
        B = zeros(3,ndof*nn); dNdx = dNdx' ;
        B(1,1:ndof*nn) = dNdx(:,1)' ; B(2,1:ndof*nn) = dNdx(:,2)' ;
        B(3,1:ndof*nn) = dNdx(:,2)' ; B(3,2:ndof*nn) = dNdx(:,1)' ;
        % compute the stiffness matrix
        kmat(gindex,gindex) = kmat(gindex,gindex) + B'*matmtx*B*W(igp)*det(JO) ;
    end % end loop over integration points
end % end loop over elements
```

2.4 Polygonal Smoothed Finite Element Method (Polygonal SFEM)

Similar to the Polygonal FEM, the Polygonal SFEM, also discretizes the domain into polygonal elements with arbitrary edges. However, the stiffness matrix given by Equation (6), is computed by using the strain projection procedure. In this approach, the strain field, $\boldsymbol{\varepsilon}_{ij}$, used to compute the stiffness matrix is computed from a weighted average of the
standard strain field $\varepsilon_{ij}^h$. At a point $x_C$ in an element $\Omega^h$, the smoothed strain field is given by:

$$
\bar{\varepsilon}_{ij}^h = \int_{\Omega^h} \varepsilon_{ij}^h(x) \Phi(x - x_C) \, dx
= \int_{\Omega^C} Bq \Phi(x - x_C) \, dx = \Bar{B}q
$$

where $q$ is a vector of degrees of freedom, $\Phi$ is a smoothing function is given by

$$
\Phi = \begin{cases}
\frac{1}{A_C} & x_C \in \Omega_C \\
0 & x_C \notin \Omega_C 
\end{cases}
$$

where $A_C$ is the area of the subcell. The smoothed strain-displacement matrix $\Bar{B}$ is given by:

$$
\Bar{B} = \frac{1}{A_C} \int_{\Omega^C} \begin{bmatrix}
\frac{\partial N_i}{\partial x} & 0 \\
0 & \frac{\partial N_i}{\partial y} \\
\frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x}
\end{bmatrix} \, d\Omega
$$

Using the divergence theorem, we obtain

$$
\Bar{B} = \frac{1}{A_C} \int_{\Gamma^C} \begin{bmatrix}
0 & 0 \\
0 & n_yN_i \\
n_xN_i & n_xN_i
\end{bmatrix} \, d\Gamma
$$

where $N_i$ is the shape function at node $I$. The smoothed stiffness matrix is given by [4]:

$$
\Bar{K} = \Bar{B}^T D \Bar{B} A_C
$$

This process of smoothing the gradient field is known as ‘cell-based smoothed finite element method’ (CSFEM) in the literature [26]. Until now, no assumption has been made on the shape of the element. The procedure outlined so far is general and is applicable to polygons of arbitrary shapes [29]. Other smoothing techniques [4] can also be extended to polygons, for example, Nguyen-Thoi et al., [28] applied the edge based smoothing technique over polygons for problems in solid mechanics. In this study we study the properties of the CSFEM over arbitrary polygons.

**Construction of Shape functions** Due to the process of strain smoothing, only the shape function is involved in the calculation of the field gradients and hence the stiffness matrix. Moreover, no analytical form of the shape functions or its derivatives is required. In this study, we use the simple averaging technique outlined in [29] to compute the shape function over arbitrary polygons. The construction of the shape functions is as follows: for a general polygonal element, a central point $O$ is located by (see Figure 4):

$$
(x_O, y_O) = \frac{1}{n} \sum_{I=1}^{n} (x_I, y_I)
$$

where $n$ is the number of nodes of the polygonal element. The shape function at point $O$ is given by $[\frac{1}{n}, \cdots, \frac{1}{n}]$ with size $1 \times n$. In this study, the shape function along each edge is linear and one Gauss point is sufficient to evaluate the integral in Equation (11). For the midpoint on the side of the element, its shape functions are evaluated averagely using two related nodes while for the interior point, its shape functions are evaluated using the point $O$ and the related node. Once the shape functions are evaluated, the stiffness matrix can be computed by following the procedure outlined in [26, 29].
Choice of number of subcells When the strain smoothing is applied to quadrilateral elements, each element can be subdivided into number of triangular or quadrilateral subcells. Liu et al., [26] showed that upper and lower bound solutions in SFEM can be obtained by varying the number of smoothing cells. In case of arbitrary polygons, Dai et al., [29] showed that for the stiffness matrix to be non-singular and stable, an \( n \)-sided polygon should be sub-divided into \( n \) triangular subcells. In this study, we adopt \( n \) triangular subcells for the construction of the stiffness matrix. A simple Matlab\textsuperscript{®} function to compute the smoothed stiffness matrix is given in Listing 3.

### Listing 3 Smoothed stiffness matrix

```matlab
function [kmatrix] = getKmatrixPolySFEM(element, node, matmtx, \( W \), Q)
  for iel = 1:numelem
    gecon = element(iel); % get current element connectivity
   coord = node(gecon,:); nn = length(gecon); ndof = 2;
    tri = delaunay(coord(:,1),coord(:,2)); % triangulate to get subcells
    N = [eye(nn); 1/nn*ones(1,nn)]; % shape function at nodes and at the centroid
    gindex = []; gindex = reshape([ndof*gecon-1; ndof*gecon],1,[]);
    for isc = 1:size(tri,1) % loop over subcells...
      B = zeros(nn,2);
      ncon = tri(isc,:); nodes = length(ncon);
      xl = x(ncon,1); yl = x(ncon,2);
      [geom,iner,cmo] = polygeom(xl,yl); % compute the area of the subcell...
      for iside = 1:size(tri,2) % loop over the sides of the subcell
        xy1 = [xl; xl(1)]; xy2 = [yl; yl(1)]; xys = [xy1 xy2];
        shape = [N(ncon,:); N(ncon(1,:))]; % shape function
        lside = norm(xys(iside+1,:)-xys(iside,:)); % side length
        nx = 1/2*(xy2(iside+1)-xy2(iside))/lside; % nx
        ny = -1/2*(xy1(iside+1)-xy1(iside))/(lside); % ny
        phi = mean(shape(iside,:),shape(iside+1,:)); % shape function
        % construct the B matrix...
        for inode = 1:nn
          B(inode,1) = B(inode,1) + phi(inode)*nx*lside/geom(1);
          B(inode,2) = B(inode,2) + phi(inode)*ny*lside/geom(1);
        end
      end
  end
end
```

Fig. 4 Construction of simple averaging shape functions for a polygonal element. \( O - 1 - 2 \) is subcell.
\[ \eta = 1 \]
\[ S(x_2, y_2), \eta = 1 \]
\[ (x_1, y_1), \eta = 1 \]
\[ \xi = 1 \]

**Radial Lines**

**Scaling Centre**

**Line element**

**Similar curve to \( S \)**

(a) Uncracked polygon

**Scaling Centre**

**Radial Lines**

**Similar curve to \( S \)**

(b) Cracked polygon

**Fig. 5** Polygon representation by the SBFEM.

\begin{align*}
\text{Bmat}(1: \text{ndof} \times \text{nn}, 1) &= \text{B}(; , 1); \quad \text{Bmat}(2: \text{ndof} \times \text{nn}, 2) = \text{B}(; , 1); \\
\text{Bmat}(1: \text{ndof} \times \text{nn}, 3) &= \text{B}(; , 2); \quad \text{Bmat}(2: \text{ndof} \times \text{nn}, 3) = \text{B}(; , 1); \\
% stiffness matrix
\text{kmat(\text{gindex}, \text{gindex})} &= \text{kmat(\text{gindex}, \text{gindex})} + \text{Bmat} \times \text{matmx} \times \text{Bmat'} \times \text{geom}(1); \\
\end{align*}

**2.5 Polygonal Scaled Boundary Finite Element Method (Polygonal SBFEM)**

Polygonal elements can be conveniently developed from the scaled boundary FEM (SBFEM). The SBFEM \cite{30} is a semi-analytical computational technique that reduces the governing partial differential equations to a set of ordinary differential equations. In the SBFEM, a local radial-circumferential-like coordinate system is introduced. Numerical solutions are sought around the circumferential direction using conventional FEM, whilst in the radial direction, the solution is defined by smooth analytical functions. A scaling centre \( O \) is selected at a point from which the whole boundary of the domain is visible. This ‘scaling requirement’ can always be satisfied by sub-structuring, i.e. dividing the structure into smaller subdomains.

Figure (5) shows a polygon modeled by the SBFEM. There is no restriction to the number of sides that a polygon can have. The radial coordinate \( \xi \) is defined from the scaling centre \( (\xi = 0) \) to the polygon boundary \( (\xi = 1) \). Each edge on the polygon boundary is discretised using 1D finite elements with local coordinate \( \eta \). The local coordinate is defined in the range \(-1 \leq \eta \leq 1\) and is similar to the standard finite elements.

**Remark 1** The geometry of the polygon has to only satisfy the aforementioned scaling requirement. This condition is automatically satisfied for all convex polygons and many concave polygons.

**Displacement approximation** The displacements of a point in a polygon is approximated by:

\[ u(\xi, \eta) = N(\eta)u(\xi) \quad (14) \]

where \( u(\xi) \) are radial displacement functions and \( N(\eta) \) is the shape function matrix in the circumferential direction. Along the circumferential direction, the standard 1D Gauss-Lobatto-Lagrange shape functions can be used.
Remark 2 The shape functions \( N(\eta) \) can be of any order and does not affect the way in which the stiffness matrix and load vectors are evaluated. Higher order polygons to be constructed conveniently using the SBFEM by simply increasing the order of the shape functions in \( N(\eta) \).

By following the procedure outlined in [30, 47], the following ODE is obtained:

\[
E_0 \xi^2 u(\xi, \xi) + (E_0 + E_1 - E_1^T) \xi u(\xi, \xi) - E_2 u(\xi) = 0
\]

(15)

where \( E_0, E_1 \) and \( E_2 \) are coefficient matrices given by:

\[
E_0 = \int_{\eta} B_1(\eta)^TDB_1(\eta) |J(\eta)| d\eta,
\]

(16)

\[
E_1 = \int_{\eta} B_2(\eta)^TDB_1(\eta) |J(\eta)| d\eta,
\]

\[
E_2 = \int_{\eta} B_2(\eta)^TDB_2(\eta) |J(\eta)| d\eta.
\]

They are evaluated element-by-element on the polygon boundary and assembled over a polygon. This process is similar to the standard FE procedure of assemblage. The boundary nodal forces are related to the displacement functions by:

\[
f = (E_0 \xi^2 u(\xi, \xi) + E_1^T u(\xi))|_{\xi=1}
\]

(17)

The stiffness matrix \( K \) in case of the SBFEM is given by:

\[
K = \Phi_q \Phi_u^{-1}
\]

(18)

where \( \Phi_u \) and \( \Phi_q \) represent the modal displacements and forces, respectively, computed from the eigenvalue decomposition of the following Hamiltonian matrix [30]:

\[
Z = \begin{bmatrix}
E_0^{-1}E_1^T & -E_0^{-1}
E_1 & E_0^{-1}E_2 & -E_0^{-1}E_1
\end{bmatrix}
\]

(19)

The corresponding eigenvalues are given by: \( \Lambda_n = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_n) \). For a 2 noded line element, the coefficient matrices given by Equation (16) can be written explicitly. The one dimensional shape functions up to order 3 is given in Listing 4. A simple Matlab\textsuperscript{®} function is given in Listing 5 to compute the coefficient matrices and the corresponding stiffness matrix.

**Listing 4** One dimensional shape functions

```matlab
function [shp] = shapeFunction1d(ennodes, x)
    % shp(1,j): shape function N_j(eta) of j-th node
    % shp(2,j): derivative of shape function N_j(eta),eta
    % ennodes: number of nodes of element. x: local coordinate eta
    shp = zeros(2,ennodes);
    switch ennodes
    case(2) % 2-node line element
        shp = [0.5d0*(1.d0-x) 0.5d0*(1.d0+x);-0.5d0 0.5d0];
    case(3) % 3-node line element
        shp = [0.5d0*(-1.d0+x)*x 1.d0-x*x 0.5d0*(1.d0+x)*x; -0.5d0+x -2*x 0.5d0+x];
    case(4) % Shape functions of 4-node line element
        shp(1,1) = -0.125d0*(-1.d0+x)*(-1.d0+5.d0*x*x); shp(1,2) = 0.625d0*(-1.d0+x)*x*(-1.d0+2.2360679774997897d0*x); shp(1,3) = -0.27950849718747371d0*(-1.d0+x)*x*(2.2360679774997897d0+5.d0*x); shp(1,4) = 0.125d0*( 1.d0+x)*(-1.d0+5.d0*x*x); shp(2,1) = 0.125d0*( 1.d0+(10.d0-15.d0*x)*x); shp(2,2) = 0.625d0*(-2.2360679774997897d0+(-2.d0+6.7082039324993691d0*x)*x); shp(2,3) = -0.27950849718747371d0*(-5.d0+(4.47213595499957933d0+15.d0*x)*x); shp(2,4) = 0.125d0*(-1.d0+(10.d0+15.d0*x)*x);
    end
```
Remark 3  When the SBFEM is applied to arbitrary polygons, the stiffness matrix is computed directly using Equation (18). An explicit form of the shape functions or its derivatives are not required for linear problems. For nonlinear problems, the shape functions can be derived by solving the the elasticity equation [48].

Listing 5  Scaled boundary polygon formulation: stiffness matrix

```matlab
function [kmat] = getKmatPolySBFEM(element, node, matmtx, W, Q)
for iel = 1:numelem
    % loop over the elements
    econ = element{iel}; nn = length(econ); ndof = 2;
    gindex = []; gindex = reshape([ndof*econ-1;ndof*econ],1,[]);
    EleUnknown = length(econ)*ndof; E0 = zeros(EleUnknown,EleUnknown);
    E1 = zeros(EleUnknown,EleUnknown); E2 = zeros(EleUnknown,EleUnknown);
    for isub = 1:size(subecon,1) % loop over the sides of the polygon
        lecon = subecon(isub,:); xynodes = xyc(lecon,:);
        lindex = reshape([ndof*lecon-1;ndof*lecon],1,[]);
        for igp = 1:size(W,1) % loop over the integration points...
            shp = shapeFunction1d(length(lecon), Q(igp,:));
            N = shp(1,:); dNdxi = shp(2,:); % shape functions and its derivatives
            Jac = [N' dNdxi']’*xynodes(:,1:2); % Jacobian
            % small b1 and b2 terms...
            xetaN = Jac(2,1)*N; yetaN = Jac(2,2)*N;
            xetadN = Jac(1,1)*dNdxi; yetadN = Jac(1,2)*dNdxi;
            b1eta = [yetaN; zeros(1,length(N)); -xetaN; zeros(1,length(N)); -xetaN; yetaN];
            b1eta = 1/det(Jac)*reshape(b1eta,3,[]);
            b2eta = [-yetadN; zeros(1,length(N)); xetadN; zeros(1,length(N)); xetadN; -yetadN];
            b2eta = 1/det(Jac)*reshape(b2eta,3,[]);
            % coefficient matrices.... lindex - local index
            E0(lindex,lindex)=E0(lindex,lindex)+b1eta'*matmtx*b1eta*det(Jac)*W(igp);
            E1(lindex,lindex)=E1(lindex,lindex)+b2eta'*matmtx*b1eta*det(Jac)*W(igp);
            E2(lindex,lindex)=E2(lindex,lindex)+b2eta'*matmtx*b2eta*det(Jac)*W(igp);
        end
    end
    % compute the stiffness matrix...
    nd = ndof*nn; id = 1:nd;
    m = E0\[E1 - eye(nd)]; Z = [m; E1*m(:,id)-E2 E1*m(:,nd+id)];
    [v, d] = eig(Z); lambda = diag(d); [~, idx] = sort(real(lambda),’ascend’);
    lambda = lambda(idx); v = v(:, idx(id));
    v(:,end-1:end) = 0; v(1:2:nd, end-1) = 1; v((1:2:nd)+1,end) = 1;
    % stiffness matrix...
    kmat(gindex,gindex) = kmat(gindex,gindex) + real(v(nd+id,:)/v(id,:));
end
```

3 Numerical Examples

In the first part of this section, the results from the three different finite element techniques discussed above are compared using three benchmark problems in the context of linear elasticity. The results are compared with the analytical solution where available. In the later part of the section, problems involving strong discontinuity (e.g. plate with a crack, two cracks emanating from a hole) are solved using the scaled boundary polygon formulation. The results from the scaled boundary polygon formulation are compared with the results from other techniques taken from literature [49, 50,51]. We employ the following convention while discussing the results:

- **Polygonal FEM** - conventional polygonal FEM with Laplace interpolants. For the purpose of numerical integration, sub-triangulation is employed and sixth order Dunavant quadrature rule [52] over each triangle is employed.
- **Polygonal nSFEM** - $n$-sided smoothed finite element method. A simple averaging technique is employed to compute the shape functions and the corresponding stiffness matrix. In this study, along each edge, the shape functions are assumed to be linear and hence, only one Gauss point is sufficient to integrate the terms in the stiffness matrix.
Polygonal SBFEM - scaled boundary polygon formulation is employed over each polygon. In this case, only the polygon boundary is discretized and 1D shape functions along the boundary are used to approximate the displacement field. Hence, no special numerical integration technique is required to compute the stiffness matrix.

The built-in Matlab\textsuperscript{®} function \texttt{voronoin} and Matlab\textsuperscript{®} functions in PolyTop \cite{53} for building the mesh-connectivity are used to create the polygonal meshes. For the purpose of error estimation and convergence studies, the relative error, $L^2$ and $H^1$ norms, are used. The displacement norm is given by:

$$\| u - u^h \|_{L^2(\Omega)} = \sqrt{ \int_{\Omega} [(u - u^h) \cdot (u - u^h)] \, d\Omega}$$ (20)

where $u^h$ is the numerical solution and $u$ is the analytical solution or a reference solution. The energy norm is given by:

$$\| u - u^h \|_{H^1(\Omega)} = \sqrt{ \int_{\Omega} [(\varepsilon - \varepsilon^h)^T D (\varepsilon - \varepsilon^h)] \, d\Omega}$$ (21)

3.1 Patch test

In this section, the equilibrium path test is performed to study the accuracy and the convergence properties of different finite element techniques. Consider a uniaxial stress $\sigma = 1$, under the condition of plate stress applied in the $y$–direction on the top edge of a unit square, whilst essential boundary conditions are applied at the bottom edge as shown in Figure (6). The exact displacement field is given by:

$$u(x,y) = \frac{v}{E} (1 - x)$$

$$v(x,y) = \frac{y}{E}$$ (22)
Figure (13) shows the relative error in displacement norm $L^2$ as a function of total number of degrees of freedom (dofs) for the different formulations. The same mesh is used when comparing the performances of the different polygonal elements. It is seen that the $L^2$-error is very small, $1e-8$ in case of polygonal FEM, but does not decrease down to machine precision. This can be attributed to the integration error as discussed in [24]. The order of accuracy is similar to that reported in [17] for polygonal FEM with Laplace interpolants. In case of the polygonal nSFEM, the method fails to pass the patch test. Again, this can be attributed to the numerical integration, in which the simple averaging technique introduces additional errors when computing the stiffness matrix. The scaled boundary polygon formulation satisfies the patch test down to machine precision. This can be attributed to the semi-analytical nature of the SBFEM technique.

Higher order interpolants can be conveniently formulated using the scaled boundary polygon formulation. This involves only increasing the order of the one-dimensional shape function $N(\eta)$ in Equation (14). The process is simpler compared with the other polygon formulations reported in the literature. For example, the approach developed by Sukumar [42] requires the solution of an optimization problem. Table 1 presents the relative error in the displacement norm $L^2$ for different orders of shape functions along each edge of the polygonal element. It is seen that the polygon SBFEM satisfies the linear patch test down to machine precision.

Table 1 Relative error in the displacement norm for the equilibrium patch test.

| Number of Polygons | Order of shape functions $p = 1$ | Order of shape functions $p = 2$ | Order of shape functions $p = 3$ |
|--------------------|----------------------------------|----------------------------------|----------------------------------|
| 1                  | $1.1955e-15$                    | $2.4484e-14$                    | $2.4442e-14$                    |
| 10                 | $1.1228e-14$                    | $7.1956e-14$                    | $3.4006e-14$                    |
| 20                 | $1.5151e-14$                    | $5.5886e-14$                    | $3.9309e-14$                    |
| 30                 | $1.7593e-14$                    | $4.7998e-14$                    | $3.2566e-14$                    |
3.2 Cantilever beam

A two-dimensional cantilever beam subjected to a parabolic shear load at the free end is examined as shown in Figure (8). The geometry is: length $L = 10$, height $D = 2$. The material properties are: Young’s modulus, $E = 3 \times 10^7$, Poisson’s ratio $\nu = 0.25$ and the parabolic shear force $P = 150$. The exact solution for displacements are given by:

$$u(x,y) = \frac{Py}{6EI} \left[ (9L-3x)x + (2+\nu) \left( y^2 - \frac{D^2}{4} \right) \right]$$

$$v(x,y) = -\frac{P}{6EI} \left[ 3\nu y^2 (L-x) + (4+5\nu) \frac{D^2x}{4} + (3L-x)x^2 \right]$$

where $I = D^3/12$ is the moment of inertia, $\overline{E} = E$, $\overline{\nu} = \nu$ and $\overline{E} = E/(1-\nu^2)$, $\overline{\nu} = \nu/(1-\nu)$ for plane stress and plane strain, respectively. Figure (9) shows a sample polygonal mesh used for this study. The numerical convergence of the relative error in the displacement norm and the relative error in the energy norm is shown in Figure (10). The results from different approaches are compared with the available analytical solution. The numerical integration error is present in both the Polygonal FEM and the Polygonal nSFEM, but is more pronounced in case of the Polygonal nSFEM. Both the Polygonal FEM and the Polygonal SBFEM yields optimal convergence in $L^2$ and $H^1$, whilst the polygonal nSFEM yields sub-optimal convergence rate. This can be attributed to the error in the numerical integration introduced by the simple averaging technique. It is seen that with mesh refinement, all the methods converge to the exact solution. An estimation of the convergence rate is also shown. From Figure (10), it can be observed that the scaled boundary polygon formulation yields more accurate results. It is also noted that in case of the polygonal SBFEM, no special numerical integration technique is required to compute the stiffness matrix and extension to higher order shape function is straightforward. Figure (10) also shows the error in the displacement norm when quadratic shape functions are used along each edge within the scaled boundary polygon formulation. It is seen that as the order of the shape functions is increased, the error decreases while the convergence rate increases.

3.3 Infinite plate with a circular hole

In this example, consider an infinite plate with a traction free hole under uniaxial tension ($\sigma = 1$) along $x-$axis Figure (18). The exact solution of the principal stresses in polar coordinates $(r, \theta)$ is given by:

$$\sigma_{11}(r, \theta) = 1 - \frac{r^2}{2\pi} \left( \frac{1}{3}(\cos 2\theta + \cos 4\theta) \right) + \frac{3r^4}{2\pi} \cos 4\theta$$

$$\sigma_{22}(r, \theta) = -\frac{r^2}{2\pi} \left( \frac{1}{3}(\cos 2\theta - \cos 4\theta) \right) - \frac{3r^4}{2\pi} \cos 4\theta$$

$$\sigma_{12}(r, \theta) = -\frac{r^2}{\pi} \left( \frac{1}{3} (\sin 2\theta + \sin 4\theta) \right) + \frac{3r^4}{2\pi} \sin 4\theta$$

(24)
where $a$ is the radius of the hole. Owing to symmetry, only one quarter of the plate is modeled. Figure (12) shows a typical polygonal mesh used for the study. The material properties are: Young’s modulus $E = 10^5$ and Poisson’s ratio $\nu = 0.3$. In this example, analytical tractions are applied on the boundary. The domain is discretized with polygonal elements and along each edge of the polygon, the shape function is linear. The convergence rate in terms of the displacement norm is shown in Figure (13). It can be seen that the Polygonal SBFEM yields more accurate results when compared with the Polygonal FEM and the Polygonal nSFEM. The convergence rate of the polygonal FEM and the polygonal SBFEM are similar. The convergence rate of the polygonal nSFEM is inferior compared with the polygonal FEM and the polygonal SBFEM. All the techniques converge to exact energy with mesh refinement. The relative error in terms of the displacement norm for the polygon SBFEM with quadratic shape functions is shown in Figure (13). It is observed that the error decreases as the order of the shape functions is increased.
Fig. 10 Bending of thick cantilever beam: Convergence results for (a) the relative error in the displacement norm ($L^2$) and (b) the relative error in the energy norm. The rate of convergence is also shown, where $m$ is the average slope. In case of the polygonal SBFEM, $p$ denotes the order of the shape functions along each edge of the polygon.
3.4 Application to linear elastic fracture mechanics

The polygonal finite element techniques discussed above can be applied to problems in linear elastic fracture mechanics. However, to accurately capture the singularity at the crack tip, a very fine mesh in combination with singular elements at the crack tip is usually required. This poses numerical difficulties when the crack evolves. Another possibility is to augment the existing finite element approximation basis with additional functions [49]. In the literature, the latter method is referred to as the ‘extended finite element method’ (XFEM). The flexibility provided by the XFEM, also has certain difficulties, such as numerical integration and blending of enriched and non-enriched elements [3]. In this section, we discuss the application of the polygonal SBFEM to problems with strong discontinuities and singularities. By exploiting the special characteristics of the scaling center [30], the stress intensity factors can be computed directly. When modeling a cracked/notched structure, the scaling centre is placed at the crack tip. The straight crack/notch edges are formed by scaling the nodes A and B on the boundary and are not discretized (see Figure (5)). In this study, a polygonal mesh that conforms to the crack surface is generated.

Calculation of the stress intensity factors. A unique feature of the scaled boundary polygon formulation is that stress singularities of any kind e.g. in cracks notches and material junctions, if present, are analytically represented in the radial displacement functions \( u(\xi) \). When a crack is modelled by a polygon element with the scaling centre chosen at its crack tip in Figure (5), some of the eigenvalues in \( \Lambda_n \) satisfy \(-1 < \lambda(\Lambda_n) < 0\). The stress field \( \sigma(\xi, \eta) \) can be expressed as [30]:

\[
\sigma(\xi, \eta) = \Psi_{\sigma}(\eta) \xi^{-\Lambda_n} \mathbf{c}
\]

where \( \Lambda_n = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_n) \) are the eigenvalues, \( \mathbf{c} = \Phi_u^{-1} \mathbf{u}_b \), \( \mathbf{u}_b \) is the displacement vector on the boundary \( \xi = 1 \), obtained by solving the algebraic system of equations. The stress mode \( \Psi_{\sigma}(\eta) \) is defined as

\[
\Psi_{\sigma}(\eta) = \mathbf{D} (-\mathbf{B}_1(\eta) \Phi_u \Lambda_n + \mathbf{B}_2(\eta) \Phi_u) \]

It can be discerned from Equation (25) that these eigenvalues lead to singular stresses at the crack tip. This enables the stress intensity factors (SIF) to be computed directly from their definitions. For a crack that is aligned with the Cartesian coordinate axes shown in Figure (14), the SIFs are defined as:

![Fig. 11 Infinite plate with a circular hole.](image-url)
Fig. 12 Plate with a circular hole: domain discretized with polygonal elements: (a) 100 elements; (b) 200 elements; (c) 400 elements and (d) 800 elements.

\[
\begin{bmatrix}
K_1 \\
K_{II}
\end{bmatrix}
= \lim_{r \to 0} \begin{bmatrix}
\sqrt{2\pi r} \sigma_{\gamma x} \big|_{\theta = 0} \\
\sqrt{2\pi r} \tau_{\gamma x} \big|_{\theta = 0}
\end{bmatrix}
\]  

Equation (27)

In the limit when \( r \to 0 \) or \( \xi \to 0 \), Equation (25) shows that the stresses vanish when the real parts of eigenvalues in \( \Lambda_n, \tilde{\lambda} (\Lambda_n) \leq -1 \). The diagonal submatrix \( \Lambda_n^{(s)} = -0.5I \) becomes dominant. For this diagonal submatrix, the singular stress field \( \sigma^{(s)}(\xi, \eta) \) can be expressed as:

\[
\sigma^{(s)}(\xi, \eta) = \Psi^{(s)}(\eta(\theta)) \xi^{-0.5} c^{(s)}
\]

where \( \Psi^{(s)}(\eta(\theta)) = \begin{bmatrix} \Psi_{\sigma_{xx}}(\eta(\theta)) & \Psi_{\sigma_{xy}}(\eta(\theta)) & \Psi_{\tau_{xy}}(\eta(\theta)) \end{bmatrix}^T \) and \( c^{(s)} \) are the stress modes and integration constants corresponding to the diagonal submatrix in \( \Lambda_n \) satisfying \(-1 < \tilde{\lambda} (\Lambda_n) < 0\). Substituting the stress components
in Equation (28) at angle $\theta = 0$ into Equation (27) and using the relation $\xi = \frac{r}{r_0}$ at $\theta = 0$, the SIF is

$$\begin{bmatrix} K_I \\ K_{II} \end{bmatrix} = \sqrt{\frac{2\pi \xi_0}{L_0}} \begin{bmatrix} \Psi_0^{(i)}(\eta(\theta = 0))c^{(i)} \\ \Psi_{\xi_0}(\eta(\theta = 0))c^{(i)} \end{bmatrix}$$

(29)

3.4.1 Plate with double edge crack in tension

The plate with double edge crack subjected to a uniform tension at both ends as shown in Figure (15) is considered. In the computations, the ratio of the crack length, $a$, to the width of the plate, $H$, is $a/H = 0.25$. The material properties
of the plate are: Young’s modulus, $E = 200$ GPa and Poisson’s ratio, $\nu = 0.3$. In this example, plane stress conditions are assumed. The empirical mode I SIF that is given by:

$$K_{I}^{\text{ref}} = C\sigma\sqrt{\pi a}$$

where $C$ is a correction factor. For $a/b > 0.4$, $b = H/2$, the correction factor is given by [54]:

$$C = 1.12 + 0.203\left(\frac{a}{b}\right) - 1.197\left(\frac{a}{b}\right)^2 + 1.930\left(\frac{a}{b}\right)^3$$

The above correction factor is for an infinite plate and with an accuracy of 2% [54]. For the chosen parameters, the reference normalized SIF is, $K_I/\sqrt{\pi a} = 1.1635$. The plate discretized using a polygon mesh of arbitrary number of sides. For the cracked polygon, each edge is discretized with twelve elements so that the angular variation of the SIF can be captured accurately. Table 2 presents the results from the Polygonal SBFEM with linear and quadratic shape functions along the each edge of the element. The results from the Polygonal SBFEM are compared with the XFEM results [49]. The number of nodes is maintained approximately the same as reported in [49]. It can be seen that the results from the Polygonal SBFEM are in good agreement with the results in the literature. It is noted that, no special functions are required in case of the polygonal SBFEM and the SIF can be computed directly as outlined above. In Table 3, the results from the Polygonal SBFEM are compared with the Body Force Method (BFM) [50] with $L/H = 3$. It is observed that the results from the Polygonal SBFEM are in very good agreement with the BFM and also with the empirical relation given in [54]. Also, the error decreases as the order of the shape functions is increased.

### 3.4.2 Inclined crack in plate

In the next example, a plate with an angled crack subjected to far field bi-axial stress field, $\sigma$ (see Figure 16) with $a/w = 0.1$, $\sigma_1 = 1$ and $\sigma_2 = 2$ is considered. In this example, the mode I and the mode II SIFs, $K_I$ and $K_{II}$, respectively,
Table 2 Normalized mode I SIF for double-edge cracked plate in tension for \( L/H = 2, \ a/H = 0.25 \).

| Ref. [49] | Polygonal SBFEM (\( p = 1 \)) | Polygonal SBFEM (\( p = 2 \)) |
|-----------|-----------------|-----------------|
| Number of nodes | \( K_I/\sqrt{\pi a} \) | Number of nodes | \( K_I/\sqrt{\pi a} \) | Number of nodes | \( K_I/\sqrt{\pi a} \) |
| Mesh 1     | -                | 116 1.1582      | 255 1.1702      |
| Mesh 2     | -                | 148 1.1599      | 381 1.1702      |
| Mesh 3     | 402 1.1622       | 400 1.1662      | 946 1.1703      |
| Mesh 4     | 1002 1.1625      | 1050 1.1679     | 2549 1.1703     |
| Mesh 5     | 2001 1.1631      | 2012 1.1679     | 4934 1.1703     |

Table 3 Normalized mode I SIF for double-edge cracked plate in tension for \( L/H = 3, \ a/H = 0.25 \).

| Number of Polygons | Polygonal SBFEM, \( K_I/\sqrt{\pi a} \) | Poly = 1 | Poly = 2 |
|--------------------|----------------------------------|---------|---------|
| 194                | 1.16531                          | 1.16925 |
| 637                | 1.16680                          | 1.16927 |
| 1327               | 1.16843                          | 1.16927 |
| Ref. [50]          | 1.16925                          | 1.16925 |

are obtained as a function of the crack angle \( \beta \). For the loads shown, the analytical SIF for an infinite plate are given by [55,49]:

\[
K_I = (\sigma_2 \sin^2 \beta + \sigma_1 \cos^2 \beta) \sqrt{\pi a} \\
K_{II} = (\sigma_2 - \sigma_1) \sin \beta \cos \beta \sqrt{\pi a}
\]  

(32)

Fig. 16 Plate with an oblique crack: geometry and boundary conditions.

The material properties of the plate are: Young’s modulus, \( E = 200 \) GPa and Poisson’s ratio, \( \nu = 0.3 \). In this example, the plate is discretized with polygon meshes containing approximately 300 polygons for all values of beta.
considered. For the element containing the crack tip, five elements (linear or quadratic) were used along each edge to capture the angular variation of the SIF accurately. The variation of the mode I and the mode II SIF with crack orientation $\beta$ are presented in Table 4 and Figure (17). The results from the XFEM [49] is also listed in the table, which is for the case of $a/w = 10$. The influence of the order of the shape functions and the plate width-to-the crack length ratio is studied using the polygonal SBFEM. It is observed the results compared very well with the reference solutions. Table 5 presents the influence of the $a/w$ on the numerical SIF in case of the Polygonal SBFEM with quadratic elements. It is seen that, as the plate width-to-the crack length ratio is increased, the numerical SIFs approach the analytical solutions (see Equation (32)) for the infinite plate.

Table 4 Mode I and Mode II SIF for a plate with an inclined crack.

| SIF   | Method                        | $\beta$  |         |         |
|-------|-------------------------------|----------|---------|---------|
|       |                               | 90°      | 45°     | 0°      |
| $K_I$ | Equation [52]                 | 2.5066   | 1.8800  | 1.2533  |
|       | Ref. [49]                     | 2.5171   | 1.8782  | 1.2585  |
|       | Polygonal SBFEM ($p = 1$)    | 2.5378   | 1.9052  | 1.2689  |
|       | Polygonal SBFEM ($p = 2$)    | 2.5415   | 1.9039  | 1.2708  |
|       | Polygonal SBFEM ($p = 3$)    | 2.5416   | 1.9039  | 1.2708  |
|       | Polygonal SBFEM ($p = 4$)    | 2.5416   | 1.9039  | 1.2708  |
| $K_{II}$ | Equation [52]              | 0.0000   | 0.6266  | 0.0000  |
|       | Ref. [49]                     | 0.0000   | 0.6260  | 0.0000  |
|       | Polygonal SBFEM ($p = 1$)    | 0.0000   | 0.6295  | 0.0000  |
|       | Polygonal SBFEM ($p = 2$)    | 0.0000   | 0.6323  | 0.0000  |
|       | Polygonal SBFEM ($p = 3$)    | 0.0000   | 0.6324  | 0.0000  |
|       | Polygonal SBFEM ($p = 4$)    | 0.0000   | 0.6324  | 0.0000  |

Table 5 Mode I and Mode II SIF for a plate with an inclined crack.

| SIF   | $\beta$ | Equation [52] | Polygonal SBFEM ($p = 2$) |
|-------|---------|---------------|---------------------------|
| $K_I$ | Infinite Plate | $a/w = 10$ | $a/w = 50$ |
|       | 90°     | 2.5066        | 2.5416                    | 2.5077        |
|       | 45°     | 1.8800        | 1.9039                    | 1.8810        |
|       | 0°      | 1.2533        | 1.2708                    | 1.2539        |
| $K_{II}$ | 90°     | 0.0000        | 0.0000                    | 0.0000        |
|       | 45°     | 0.6266        | 0.6324                    | 0.6265        |
|       | 0°      | 0.0000        | 0.0000                    | 0.0000        |

### 3.4.3 Two cracks emanating from a hole in a finite plate

As a last example, a plate with a centrally located hole of radius $r$ and two cracks emanating from a hole (see Figure (18)) is considered. The dimensions of the plate are: $H = 2W$ and the radius of the hole $r = 2W$. The normalized stress intensity factors are defined as: $F_I = K_I / \sigma \sqrt{\pi a}$ and $F_{II} = K_{II} / \sigma \sqrt{\pi a}$. The results from the polygonal SBFEM are compared with the results of Woo et al., [56] and Daux et al., [51]. In this example, 1D quadratic element is used along each edge of the polygonal element that contains the crack tip. Table 6 shows the computed SIFs $F_I$ and $F_{II}$ for several ratios of $a/W$ and for different angles $\theta$. It can be seen from Table 6 that the results from the polygonal
SBFEM are in very good agreement with the results in the literature. It is noted that, when compared to Daux et al., [51], no additional enrichment functions are required to resolve the singularity at the crack tip. Moreover, higher order elements can be employed for the element containing the crack tip. The scaled boundary polygon formulation when combined with the extended FEM, eliminates the need to compute the enrichment functions and improves the accuracy of the computed SIF. Interested readers are referred to [31] where the XFEM was combined with the SBFEM for quadrilateral elements.

4 Conclusions

In this paper, we compared three different displacement based finite element formulations over arbitrary polygons. The accuracy and the convergence properties of different approaches were studied with a few benchmark problems from linear elasticity. From the numerical examples, it is seen that the Polygonal SBFEM yields the most accurate results. Although, the Polygonal FEM yields optimal results, the numerical integration should be carried out accurately to improve the results, as discussed in [24]. The results from the Polygonal nSFEM are stiffer and yields sub-optimal convergence in both the displacement norm and the energy norm. It is noted that for this study, we had employed cell-based smoothed FEM, whilst other smoothing techniques are possible. The Polygonal SBFEM eliminates the need to compute the shape functions or to employ special numerical integration technique. Only the boundary of the polygon is discretized and the stiffness matrix is computed directly. Moreover, an extension to higher order shape functions along the boundary of the polygon is straightforward. When applied to linear elastic fracture mechanics, the scaled boundary polygon formulation yields accurate results. It is noted that the polygonal SBFEM does not require enrichment of the approximation basis and the stress intensity factors can be computed directly. This approach can readily be combined with the extended FEM. By combining the scaled boundary formulation with the XFEM, a priori knowledge of the asymptotic fields is not required.
Fig. 18 Two cracks emanating from a hole in a finite plate: (a) geometry and boundary conditions and (b) domain discretized with polygonal elements.

Table 6 Normalised stress intensity factors for two cracks emanating from a hole with various ratios \(a/W\) and for different angles \(\theta\).

| \(\theta\) | \(a/W\) | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
|-----------|--------|-----|-----|-----|-----|-----|-----|-----|
| \(F_I\)   | Ref. [51] | 1.090 | 1.214 | 1.283 | 1.394 | 1.577 | 1.902 | 2.639 |
|           | Ref. [55] | 1.078 | 1.216 | 1.286 | 1.396 | 1.574 | 1.892 | 2.498 |
| \(F_{II}\) | Ref. [51] | 0.730 | 0.872 | 0.948 | 1.038 | 1.155 | 1.315 | 1.543 |
|           | Ref. [55] | 0.730 | 0.872 | 0.948 | 1.038 | 1.155 | 1.315 | 1.543 |
| \(F_I\)   | Ref. [51] | 0.153 | 0.342 | 0.419 | 0.464 | 0.504 | 0.559 | 0.637 |
|           | Ref. [55] | 0.141 | 0.336 | 0.417 | 0.463 | 0.504 | 0.556 | 0.639 |
| \(F_{II}\) | Ref. [51] | 0.069 | 0.162 | 0.220 | 0.264 | 0.306 | 0.348 | 0.393 |
|           | Ref. [55] | 0.039 | 0.147 | 0.210 | 0.257 | 0.300 | 0.344 | 0.390 |
| \(F_{II}\) | Ref. [51] | 0.158 | 0.345 | 0.425 | 0.469 | 0.502 | 0.534 | 0.566 |
|           | Ref. [55] | 0.137 | 0.332 | 0.418 | 0.465 | 0.500 | 0.532 | 0.565 |
Acknowledgements

S Natarajan would like to acknowledge the financial support of the School of Civil and Environmental Engineering, The University of New South Wales for his research fellowship for the period Sep. 2012 onwards.

References

1. T. Fries, H. Matthies, Classification and overview of meshfree methods. Tech. Rep. D38106, Institute of Scientific Computing, Technical University, Braunschweig, Hans-Sommer-Strasse (2003)
2. C.A. Duarte, J.T. Oden, Numerical Methods for Partial Differential Equations 12, 673 (1996)
3. T.P. Fries, T. Belytschko, International Journal for Numerical Methods in Engineering 84(3), 253 (2010)
4. G. Liu, N.T. Trung, Smoothed Finite Element Methods (CRC Press, 2010)
5. S. Bordas, S. Natarajan, P. Kerfriden, C.E. Augarde, D.R. Mahapatra, T. Rabczuk, S.D. Pont, International Journal for Numerical Methods in Engineering 86, 637 (2011)
6. J.A. Cottrell, T.J. Hughes, Y. Bazilevs, Isogeometric analysis: Toward Integration of CAD and FEA (John Wiley & Sons, Ltd., 2009)
7. T. Belytschko, C. Parimi, N. Moës, S. Usui, N. Sukumar, International Journal for Numerical Methods in Engineering 56, 609 (2003)
8. N. Moummassi, S. Belouettar, E. Béchet, S. Bordas, D. Quoirin, M. Potier-Ferry, Computer Methods in Applied Mechanics and Engineering 200(5–8), 774 (2011)
9. S. Moorthy, S. Ghosh, Computer Methods in Applied Mechanics and Engineering 185, 37 (2000)
10. G. Dagauputa, Journal of Aerospace Engineering (ASCE) 16(1), 1 (2003)
11. K. Sze, N. Sheng, Finite Elements in Analysis and Design 42(2), 107 (2005)
12. P. Pavanikum, K. Jayabal, A. Arockiarajana, Interrated Ferroelectrics 120, 90 (2010)
13. K. Jayabal, A. Menzel, A. Arockiarajan, S. Srinivasan, Computational Mechanics 48, 421 (2011)
14. M. Kraus, A. Rajagopal, P. Steinmann, Computers & Structures 120, 33 (2013). DOI 10.1016/j.compstruc.2013.01.017
15. M.S. Floater, Computer Aided Geometric Design 20(1), 19 (2003)
16. J. Warren, S. Schaefer, A. Hirani, M. Desbrun, Advances in Computational Mechanics 27(3), 319 (2007)
17. N. Sukumar, A. Tabarraei, International Journal for Numerical Methods in Engineering 61, 2045 (2004)
18. A. Tiwary, C. Hu, S. Ghosh, Finite Elements in Analysis and Design 43, 504 (2007)
19. M. Rashid, P. Gullet, Computer Methods in Applied Mechanics and Engineering 190(11–12), 1509 (2000)
20. N. Sukumar, International Journal for Numerical Methods in Engineering 57, 1 (2003)
21. S. Biabanaki, A. Khoei, Computational Mechanics 50, 19 (2012). DOI 10.1007/s00466-011-0668-4
22. S. Natarajan, S. Bordas, D.R. Mahapatra, International Journal for Numerical Methods in Engineering 80, 103 (2009)
23. S. Mousavi, H. Xiao, N. Sukumar, International Journal for Numerical Methods in Engineering 82(1), 99 (2010)
24. C. Talischi, G.H. Paulino, (2013). DOI http://arxiv.org/pdf/1307.4423v1.pdf. In review
25. J. Chen, C. Wu, S. Yoon, Y. You, International Journal for Numerical Methods in Engineering 50, 435 (2001)
26. G. Liu, T. Nguyen, K. Dai, K. Lam, International Journal for Numerical Methods in Engineering 71(8), 902 (2007)
27. H. Nguyen-Xuan, S. Bordas, H. Nguyen-Dang, International Journal for Numerical Methods in Engineering 74, 175 (2008)
28. T. Nguyen-Thoi, G. Liu, H. Nguyen-Xuan, International Journal for Numerical Methods in Engineering 27, 1446 (2011)
29. K. Dai, G. Liu, T. Nguyen, Finite Elements in Analysis and Design 43, 847 (2007)
30. J. Wolf, C. Song, Computer Methods in Applied Mechanics and Engineering 190, 5551 (2001)
31. S. Natarajan, C. Song, International Journal for Numerical Methods in Engineering (2013). DOI 10.1002/nme.4557
32. E.T. Ooi, C. Song, F. Tin-Loi, Z. Yang, International Journal for Numerical Methods in Engineering 91, 319 (2012)
33. X. Yang, S. Wu, C. Zheng, J. Zhang, Appl. Math. Mech. 30(10), 1233 (2009)
34. L. da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. Marini, A. Russo, Mathematical Models and Methods in Applied Sciences 23, 199 (2013)
35. Q. Du, D. Wang, SIAM J Sci Comput 26(3), 737 (2005)
36. D. Sieger, P. Alliez, M. Botsch, in Proceedings of the 19th International Meshing Roundtable (2010), pp. 335–350
37. E. Wachspress, A rational basis for function approximation (Springer, New York, 1971)
38. M.S. Floater, A. Gillette, N. Sukumar, (2013). DOI http://arxiv.org/pdf/1306.4385v1.pdf. In review
39. R. Shibon, Mathematical Proceedings of the Cambridge Philosophical Society 87, 151 (1980)
40. N. Sukumar, B. Moran, T. Belytschko, International Journal for Numerical Methods in Engineering 43(5), 839 (1998)
41. J. Blachow, International Journal for Numerical Methods in Engineering (2013). DOI 10.1002/nme.4562
42. N. Sukumar, Computer Methods in Applied Mechanics and Engineering 263, 27 (2013)
43. N. Sukumar, E. Malsch, Archives of Computational Methods in Engineering 13(1), 129 (2006)
44. Polygonal finite elements: Laplace interpolants (FORTRAN implementation), http://dilbert.engr.ucdavis.edu/~suku/nem/nem_polygon.html
45. J. Lyness, G. Monegato, SIAM Journal on Numerical Analysis 14(2), 283 (1977)
46. A. Sommariva, M. Vianello, Journal of Computational and Applied Mathematics 231(2), 886 (2009). DOI 10.1016/j.cam.2009.05.014
47. A. Deeks, J. Wolf, Computational Mechanics 28, 489 (2002)
48. E. Ooi, C. Song, F. Tin-Loi, Computer Methods in Applied Mechanics and Engineering (2013). In review
49. A. Tabarraei, N. Sukumar, Computer Methods in Applied Mechanics and Engineering 197(5), 425 (2008)
50. A. Saimoto, F. Motomura, H. Nisitani, Key Engineering Materials 452–453, 405 (2010)
51. C. Daux, N. Moës, J. Dolbow, N. Sukumar, T. Belytschko, International Journal for Numerical Methods in Engineering 48, 1741 (2000)
52. D. Dunavant, International Journal for Numerical Methods in Engineering 21(6), 1129 (1985)
53. C. Talischi, G.H. Paulino, A. Pereira, I.F. Menezes, Struct. Multidiscip. Optim 45, 329 (2012)
54. H. Tada, P.C. Paris, G.R. Irwin, The stress analysis of cracks handbook (The American Society of Mechanics Engineers, New York, 2000)
55. M. Aliabadi, D. Rooke, D. Cartwright, International Journal of Fracture 34, 131 (1987)
56. C. Woo, Y. Wang, Y. Cheung, Engineering Fracture Mechanics 32, 279 (1989)