Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6628837, 22 pages
https://doi.org/10.1155/2021/6628837

Research Article

PSBFEM-Abaqus: Development of User Element Subroutine (UEL) for Polygonal Scaled Boundary Finite Element Method in Abaqus

Nan Ye,[1,2] Chao Su ,[1] and Yang Yang [3]

[1]*College of Water Conservancy and Hydropower Engineering, Hohai University, Nanjing 210098, China*
[2]*Water Conservancy Research Institute of Jilin Province, Changchun 130061, Jinlin, China*
[3]*Power China Kunming Engineering Corporation Limited, Kunming 650051, China*

Correspondence should be addressed to Yang Yang; yangyhhu@foxmail.com

Received 9 November 2020; Revised 10 July 2021; Accepted 24 August 2021; Published 16 September 2021

Academic Editor: Toqeer Mahmood

The polygonal scaled boundary finite element method (PSBFEM) is a novel method integrating the standard scaled boundary finite element method (SBFEM) and the polygonal mesh technique. This work discusses developing a PSBFEM framework within the commercial finite element software Abaqus. The PSBFEM is implemented by the User Element Subroutine (UEL) feature of the software. The details on the main procedures to interact with Abaqus, defining the UEL element, and solving the stiffness matrix by the eigenvalue decomposition are present. Moreover, we also develop the preprocessing module and the postprocessing module using the Python script to generate meshes automatically and visualize results. Several benchmark problems from two-dimensional linear elastostatics are solved to validate the proposed implementation. The results show that PSBFEM-UEL has significantly better than FEM convergence and accuracy rate with mesh refinement. The implementation of PSBFEM-UEL can conveniently use arbitrary polygon elements by the polygon/quadtree discretizations in the Abaqus. The developed UEL and the associated input files can be downloaded from https://github.com/hhupde/PSBFEM-Abaqus.

1. Introduction

The finite element method (FEM) is a reliable computational tool to solve partial differential equations (PDE) in science and engineering [1–4]. A domain of complex geometry is partitioned into a finite number of nonoverlapping subdomains of simplex shapes by introducing the concept of discretization [5]. Typically, the shape of the conventional two-dimensional finite element method is triangles or quadrilaterals. At present, the conventional FEM also faces several problems. For example, (a) the accuracy of the solution depends on the quality of the mesh and (b) requires sophisticated discretization techniques to generate high-quality meshes and to capture topological changes [6]. These elements used in conventional FEM must conform to the domain's boundary, which leads to difficulty in solving many complex problems.

To overcome the weakness above the conventional FEM, researchers proposed other alternatives methods, such as the meshfree method [7–9], the smoothed finite element method [6, 10], the Isogeometric Analysis (IGA) [11], Deep Neural Networks (DNNs) [12], and the polygonal finite element method [10, 13]. Polygonal element with more than four sides involves more nodes in their interpolation compared with a conventional FEM. Simultaneously, they generally exhibit superior solution accuracy [5]. Moreover, it is more flexible in the discretization of complex geometry. Therefore, these advantages have further motivated polygonal elements as an alternative to conventional FEM using triangles or quadrilaterals.

The scaled boundary finite element method (SBFEM) is an alternative method to construct polygonal elements. Song and Wolf developed the technique in the 1990s [14]. The scaled boundary finite element method is a semianalytical method that attempts to fuse the advantages and characteristics of FEM and the boundary element method (BEM) into one new approach. The SBFEM has been applied to many physical field problems, such as wave propagation [15, 16], heat conduction [17, 18], fracture [19, 20], acoustic [21], seepage [22, 23], elastoplastic [5], and fluid [24]. For these problems, the SBFEM presents more efficiency compared with the conventional FEM.

The polygonal scaled boundary finite element method (PSBFEM) is a novel method integrating the standard SBFEM and the polygonal mesh technique. This method is flexible in meshing complex geometries, and the use of polygons to discretize the computational domain naturally complements the SBFEM. Recently, an alternative mesh technique has been widely used in geometric discretization. In computational mechanics, the quadtree algorithm is usually used in large-scale simulations typical in the modeling of earthquake and ground motions [25], flood [26], and tsunami [27]. The quadtree algorithm is fast, efficient, and capable of achieving rapid and smooth transitions of element sizes between mesh refinement regions [28]. Mesh generation and adaptive refinement of quadtree meshes are straightforward [29]. Due to hanging nodes between two adjacent elements of different sizes, it is problematic that quadtree meshes are directly used to simulate within the finite element method's framework. However, the SBFEM only discretizes in the boundary of geometry. Hence, each cell in a quadtree mesh, regardless of hanging nodes, is treated as a generic polygon. This enables the structure of the quadtree to be exploited for efficient computations. The ability to assume any number of sides also enables the SBFEM to discretize curved boundaries better.

Although the PSBFEM has become quite mature, these studies only exist as a few stand-alone codes. The PSBFEM has not yet formed a part of commercial software at present. Hence, it is not easy to use the method in a specific community or laboratory. The commercial software Abaqus has powerful linear or nonlinear, static, or dynamic analysis capabilities. Also, Abaqus/Standard analysis provides a User Element Subroutine (UEL) to define an element with a very available option to interface with the code. Liang et al. [30] developed a UEL for dynamic analysis of saturated porous media based FEM. Kumbhar et al. [6] implemented the element based smoothed finite element method (CSFEM) by the UEL subroutines. Molnar et al. [31] implemented an implicit, staggered elastoplastic version of the phase-field approach Abaqus through the UEL. Therefore, the UEL can be used to extend the Abaqus to solve new problems conveniently.

Recently, Ya et al. [32] implemented an open-source polyhedral SBFEM element for three-dimensional and nonlinear problems in the commercial software Abaqus through the UEL. The code can use the polyhedral element to enhance the performances of ABAQUS for interfacial problems, and it significantly reduces the meshing burden encountered in the FEM. At the same time, it also implemented the technique of octree mesh generation, which is efficient and robust to mesh complex geometries and promise to integrate geometric models and numerical analysis in a fully automatic manner.

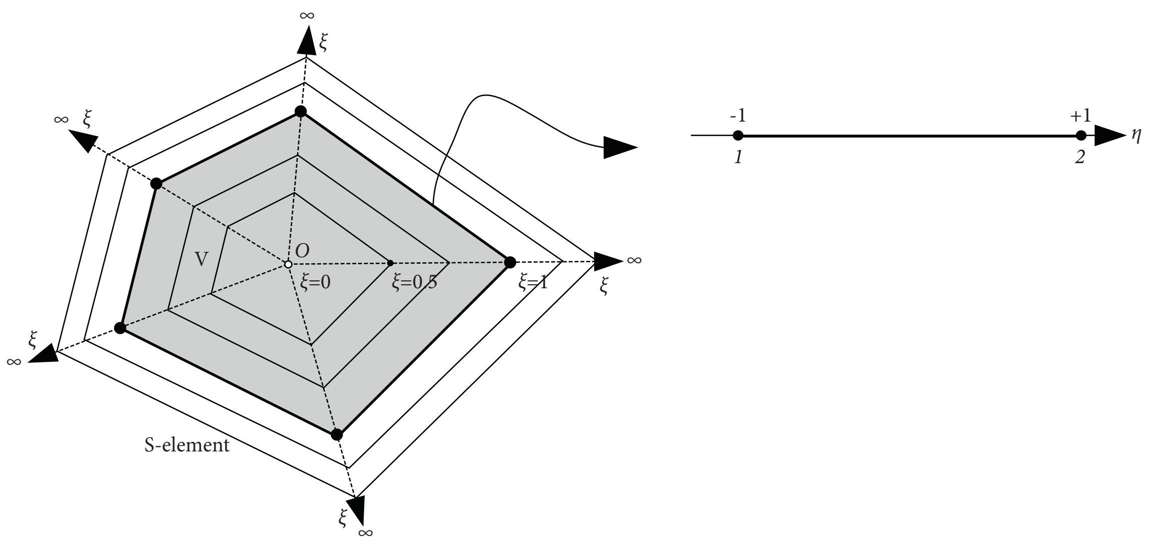
In engineering designs, to calculate models to be used as design tools, two-dimensional (2D) models are relatively easy to set up and have reasonably short computational times, which would allow sensitivity and optimization analyses [33]. There are no available subroutines of SBFEM to solve 2D problems in the commercial software Abaqus at present. However, the implementation of two-dimension can be easily derived from three-dimensional implementation. Hence, Ya et al. [32] provided a reference for developing the UEL to solve 2D problems using the polygonal scaled boundary finite element method with the polygon/quadtree meshes.

In addition, to solve the stiffness matrix, we need to perform the eigenvalue decomposition in the SBFEM. Ya et al. [32] used the Linear Algebra Package (LAPACK) [34] to perform the eigenvalue decomposition by adding the LAPACK source codes into UEL subroutines. Hence, LAPACK needs to be compiled every time. It will take extra time when calling UEL subroutines. Abaqus provides mathematical libraries (the Intel Math Kernel Library, MKL [35]) to perform the eigenvalue decomposition. The user can directly use MKL only by modifying the Abaqus environment file, which can avoid taking extra time when calling UEL subroutines.

In this paper, we implement the PSBFEM technique with Abaqus and provide the details. For simplicity, we consider a two-dimensional linear elastostatics problem using the PSBFEM technique. This work is organized as follows: the basic principles of the SBFEM are introduced in Section 2. Section 3 describes the implementation of PSBFEM by the Abaqus UEL subroutine. Moreover, we also develop the preprocessing and postprocessing module using the Python script. Section 4 presents a detailed convergence study by comparing the convergence and accuracy rates with the conventional FEM using several numerical benchmarks in two-dimensional linear elastostatics. Finally, the concluding remarks of this work are summarized in Section 5.

2. Theory

2.1. The SBFEM Equations in the 2D Linear Elastostatics. The fundamental difference between SBFEM and FEM is introducing a scaling center O [36]. As shown in Figure 1, a scaling center can be directly visible from any point on the whole boundary of the S-element. By a scaling center, we can transform the scaled boundary coordinates to polar coordinates. Hence, the S-element can be described by a radial coordinate ξ and a circumferential coordinate η.



▨ Bounded domains
□ Unbounded domains

Figure 1: Concept of scaled boundary finite element method.

In this section, we consider a two-dimensional isotropic linear elastostatics problem. The governing equation and the boundary conditions can be expressed as

$$\begin{aligned} \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \mathbf{b} &= 0, \quad \text{in } \Omega, \\ \mathbf{u} &= \overline{u}, \quad \text{on } \Gamma_u, \\ \boldsymbol{\sigma} \cdot \mathbf{n} &= \overline{t}, \quad \text{on } \Gamma_t, \end{aligned} \tag{1}$$

where $\boldsymbol{\nabla}$ is the differential operator, $\boldsymbol{\sigma}$ is the Cauchy stress tensor, $\mathbf{b}$ is the body force, Ω is the computational domain. Γ_u is the displacement boundary conditions, and Γ_t is the surface traction boundary conditions. The displacement $\overline{u}$ and the surface traction $\overline{t}$ are imposed on the domain boundaries Γ_u and Γ_t with the outward unit normal n.

As illustrated in Figure 2, the SBFEM presents a local coordinate system (ξ, η). The coordinates of a point (x, y) along the radial line and inside the domain can be expressed as follows [36]:

$$x = \xi [N(\eta)] \{x\}, \tag{2a}$$

$$y = \xi [N(\eta)] \{y\}, \tag{2b}$$

where ξ,η are the scaled boundary coordinates in two dimensions, ξ is a radial coordinate, and η is the circumferential coordinate.

The differential operator can be transformed from the Cartesian coordinate system x, y into the scaled boundary coordinates system ξ, η as follows:

$$\nabla = [b_1] \frac{\partial}{\partial \xi} + \frac{1}{\xi} [b_2] \frac{\partial}{\partial \eta}, \tag{3}$$

with

$$[b_1] = \frac{1}{|J_b|} \begin{bmatrix} y_{b,\eta} & 0 \\ 0 & -x_{b,\eta} \\ -x_{b,\eta} & y_{b,\eta} \end{bmatrix}, \tag{4a}$$

$$[b_2] = \frac{1}{|J_b|} \begin{bmatrix} -y_b & 0 \\ 0 & x_b \\ x_b & -y_b \end{bmatrix}, \tag{4b}$$

where the Jacobian matrix at the boundary $(\xi = 1)$ can be expressed as

$$[J_b] = \begin{bmatrix} x_b & y_b \\ x_{b,\eta} & y_{b,\eta} \end{bmatrix} = x_b y_{b,\eta} - y_b x_{b,\eta}. \tag{5}$$

A comma followed by a subscript is used to denote partial differentiation to the variable in the subscript. The displacement field $u(\xi, \eta)$ at any point in SBFEM coordinates is written as

$$\{u(\xi, \eta)\} = [N_u(\eta)] \{u(\xi)\}, \tag{6}$$

where $u(\xi)$ is radial displacement functions along a line connecting the scaling center O and a node at the boundary. $[N_u(\eta)]$ is the shape function matrix

$$[N_u(\eta)] = \begin{bmatrix} N_1(\eta) & 0 & N_2(\eta) & 0 & \dots & 0 & N_m(\eta) & 0 \\ 0 & N_1(\eta) & 0 & N_2(\eta) & & \dots & 0 & N_m(\eta) \end{bmatrix}. \tag{7}$$

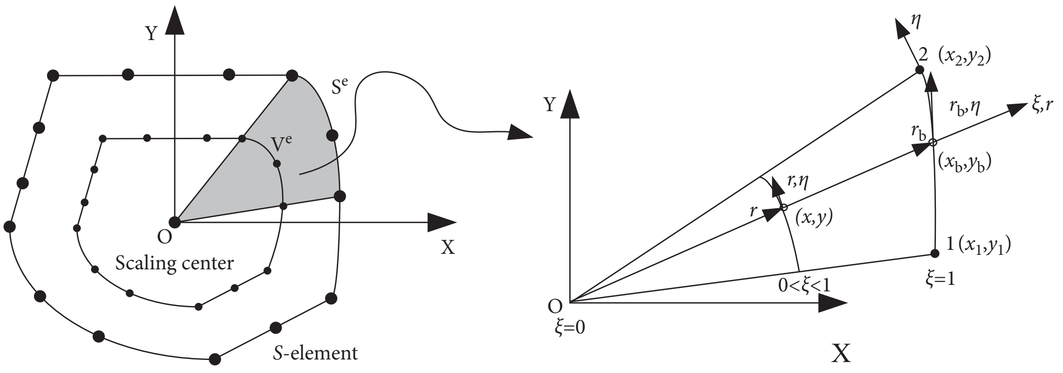


Figure 2: The SBFEM local coordinates.

The strain-displacement transition matrices $[B_1] = [B_1(\eta)]$ and $[B_2] = [B_2(\eta)]$ are introduced:

$$[B_1] = [b_1][N_u], \tag{8a}$$

$$[B_2] = [b_2][N_u]_{,\eta}. \tag{8b}$$

The strain field $\{\varepsilon\}$ is expressed in the scaled boundary coordinates as

$$\{\varepsilon\} = \left([B_1]\{u(\xi)\}_{,\xi} + \frac{1}{\xi}[B_2]\{u(\xi)\} \right). \tag{9}$$

The stress field can be express as

$$\{\sigma\} = [D]\left([B_1]\{u(\xi)\}_{,\xi} + \frac{1}{\xi}[B_2]\{u(\xi)\} \right), \tag{10}$$

where $[D]$ is an elasticity matrix and has the form

$$[D] = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}, \tag{11}$$

and for plane stress cases, E is Young's modulus, and ν is Poisson's ratio.

According to the virtual work principle, the radial displacements function $u(\xi)$ is the solution of the SBFEM equation in displacement [36]:

$$[E_0]\xi^2\{u(\xi)\}_{,\xi\xi} + \left([E_0] + [E_1]^T - [E_1]\right)\xi\{u(\xi)\}_{,\xi} - [E_2]\{u(\xi)\} + F(\xi) = 0, \tag{12}$$

where $\{u(\xi)\}_{,\xi\xi}$ is the second partial derivative with respect to the variable ξ, $\{u(\xi)\}_{,\xi}$ is the first partial derivative with respect to variable ξ, $[E_i]$, $i = 0, 1, 2$, are the coefficient matrices, and $F(\xi)$ is a load vector.

In the derivation, the governing equations of linear elasticity are weakened in the circumferential direction, while the strong form remains in the radial direction. The coefficient matrices $[E_i]$, $i = 0, 1, 2$, depends only on the geometry and material properties. These coefficient matrices can be given as follows:

$$[E_0] = \int_{-1}^{+1} [B_1]^T [D][B_1]|J_b|\,d\eta, \tag{13a}$$

$$[E_1] = \int_{-1}^{+1} [B_2]^T [D][B_1]|J_b|\,d\eta, \tag{13b}$$

$$[E_2] = \int_{-1}^{+1} [B_2]^T [D][B_2]|J_b|\,d\eta. \tag{13c}$$

When $F(\xi) = 0$, the solution of equation (12) is second-order ordinary differential equations that can be obtained by introducing the variable:

$$\{X(\xi)\} = \begin{Bmatrix} \{u(\xi)\} \\ \{q(\xi)\} \end{Bmatrix}, \tag{14}$$

where $q(\xi)$ is the internal force vector. Equation (12) can be transformed into a first-order ordinary differential equation with twice the number of an unknown as

$$\xi\{X(\xi)\}_{,\xi} - [Z_p]\{X(\xi)\} = 0, \tag{15}$$

where the coefficient matrix $[Z_p]$ is a Hamiltonian matrix. The solution for a bounded domain is obtained using the positive eigenvalues of $[Z_p]$. Hence, $[Z_p]$ can be expressed as

$$[Z_p] = \begin{bmatrix} -[E_0]^{-1}[E_1]^T & [E_0]^{-1} \\ [E_2] - [E_1][E_0]^{-1}[E_1]^T & [E_1][E_0]^{-1} \end{bmatrix}. \tag{16}$$

The solution of the equation (15) can be obtained by the mathematical theorem and eigenvalue decomposition technique. The eigenvalue decomposition of $[Z_p]$ is expressed as

$$[Z_p]\begin{bmatrix} [\Phi_u^{(n)}] & [\Phi_u^{(p)}] \\ [\Phi_q^{(n)}] & [\Phi_q^{(p)}] \end{bmatrix} = \begin{bmatrix} [\Phi_u^{(n)}] & [\Phi_u^{(p)}] \\ [\Phi_q^{(n)}] & [\Phi_q^{(p)}] \end{bmatrix}\begin{bmatrix} [\lambda^{(n)}] & 0 \\ 0 & [\lambda^{(p)}] \end{bmatrix}. \tag{17}$$

The real parts of eigenvalues $[\lambda^{(n)}]$ are negative, and that of $[\lambda^{(p)}]$ are positive. $[\Phi_u^{(n)}]$ and $[\Phi_u^{(p)}]$ are the transformation matrices corresponding to the modal displacements and forces, respectively. The general solution of equation (15) can be written as

$$X(\xi) = \begin{bmatrix} [\Phi_u^{(n)}] & [\Phi_u^{(p)}] \\ [\Phi_q^{(n)}] & [\Phi_q^{(p)}] \end{bmatrix} \begin{bmatrix} \xi^{-[\lambda^{(n)}]} & 0 \\ 0 & \xi^{-[\lambda^{(p)}]} \end{bmatrix} \begin{Bmatrix} \{c^{(n)}\} \\ \{c^{(p)}\} \end{Bmatrix}, \tag{18}$$

where $\{c^{(n)}\}$ and $\{c^{(p)}\}$ are the integration constants. For a super element with $0 \le \xi \le 1$, the solution of equation (15) is

$$\{u(\xi)\} = [\Phi_u^{(n)}]\xi^{-[\lambda^{(n)}]}\{c^{(n)}\}, \tag{19}$$

$$\{q(\xi)\} = [\Phi_q^{(n)}]\xi^{-[\lambda^{(n)}]}\{c^{(n)}\}, \tag{20}$$

where the integration constants $\{c^{(n)}\}$ can be extracted from the nodal displacements on the boundary $\{u_b\} = \{u\}(\xi = 1)$ as

$$\{c^{(n)}\} = [\Phi_u^{(n)}]^{-1}\{u_b\}. \tag{21}$$

Eliminating the integration constants $\{c^{(n)}\}$ at $\xi = 1$ is given

$$\{F\} = \{q(\xi = 1)\} = [\Phi_q^{(n)}][\Phi_u^{(n)}]^{-1}\{u_b\}. \tag{22}$$

Hence, the stiffness matrix of the S-element can be written as

$$[K] = [\Phi_q^{(n)}][\Phi_u^{(n)}]^{-1}. \tag{23}$$

The displacement field $u(\xi, \eta)$ inside subdomains by a line element on the S-element can be given

$$\{u(\xi, \eta)\} = [N_u(\eta)][\Phi_u^{(n)}]\xi^{-[\lambda^{(n)}]}\{c^{(n)}\}. \tag{24}$$

2.2. The Element of PSBFEM. The PSBFEM is inherently appropriate for modeling polygons and has other promising capabilities. Because the PSBFEM is discretized only in the boundary, and an element of PSBFEM can assume more complex shapes than a finite element method, the PSBFEM element is much more flexible than the FEM element. With such an important advantage, the PSBFEM is more applicable to complex geometries than other methods. Figure 3 shows the supported element type in the PSBFEM-UEL. In the PSBFEM-UEL, we can use the Abaqus element type, such as CPS3, CPE3, CPS4, and CPE4. Besides, the PSBFEM-UEL also provides the complex element type: polygonal elements and complex quadrilateral element (quadtree discretization). Hence, it is an effective tool to solve complex elements in numerical analysis.

2.3. Automatic Meshes Generation. This section uses a Python script to automatically generate PSBFEM elements by the Delaunay triangulation [37]. This algorithm is robust and efficient, and it has been integrated into Abaqus CAE. The algorithm can be used to triangulate any set of points on a two-dimensional plane. The triangulated mesh is then used to generate the polygon elements. The detail of the polygon generation algorithm mainly contains two steps. Firstly, we automatically generate triangular mesh by the Abaqus CAE. Secondly, considering each triangle interior node as the center of the polygon element, a polygonal element can be generated by connecting the centroids of all the triangular elements circumventing. More detail of this algorithm is presented in Figure 4. Finally, the PSBFEM-UEL can be directly used with a polygon mesh generator to analyze complex geometry problems.

The quadtree decomposition is a tree data structure in which each parent has precisely four children [38]. The quadtree meshes are fast, efficient, and capable of achieving rapid and smooth transitions of element sizes between mesh refinement regions. The quadtree mesh can provide vital support in the preprocessing for the adaptive analysis of the SBFEM. We developed a quadtree mesh automatic generation code by a Python script. A simple example of quadtree discretization with three levels is illustrated in Figure 5. The quadtree algorithm is described in more detail in [28, 38].

3. UEL Implementation of SBFEM in Abaqus

The Abaqus/Standard analysis provides a programming interface UEL to define the customized elements. In this section, we present the major implementation details of UEL for the PSBFEM in Abaqus. The Abaqus solver can be carried out with the UEL subroutine by the command:

$$\textbf{abaqus job} = \langle\textbf{input file name}\rangle\textbf{user} = \langle\textbf{UEL subroutine file}\rangle. \tag{25}$$

Figure 6 shows the framework sketch of implementing the PSBFEM within Abaqus. This system contains three parts: preprocessing module, PSBFEM-UEL module, and postprocessing module. A Python script is used in preprocessing to generate polygon SBFEM mesh and define the loading and boundary conditions. Finally, we can obtain an Abaqus input file by preprocessing. The Abaqus CAE does not support the visualization of the UEL element. Thus, a Python script is provided in the postprocessing to extract the VTU format results and visualize them in the software Paraview [39].

3.1. The Implementation of PSBFEM-UEL. The most critical work of UEL is to update the contribution of the element to the internal force vector RHS and the stiffness matrix AMATRX according to the information from ABAQUS/Standard analysis. In this paper, the UEL code for implementing the PSBFEM is written in FORTRAN77. Figure 7 shows an overall implementation of the UEL subroutine. Based on the input file's connectivity information, the UEL computes the scaling centers and transforms the global coordinate into the local coordinate. Equation (13) computes the coefficient matrices $[E_0]$, $[E_1]$, and $[E_2]$, which are used to construct the Hamilton matrix $[Z_p]$ by equation (16). The two eigenvector matrices ($[\Phi_q^{(n)}]$, $[\Phi_u^{(n)}]$) are constructed by

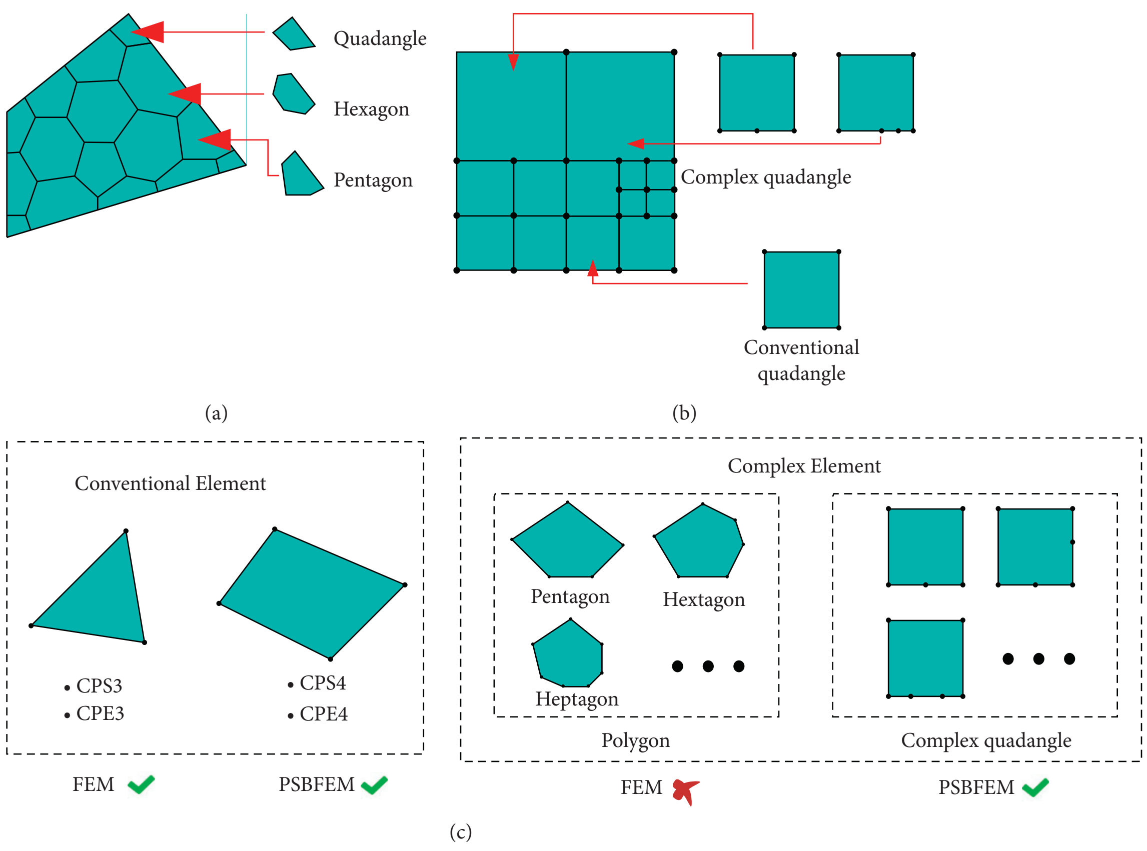


Figure 3: The supported element type in the PSBFEM-UEL; (a) a polygon discretization illustrating the mesh generation using PSBFEM-UEL element; (b) a quadtree discretization illustrating the mesh generation using PSBFEM-UEL element; (c) the PSBFEM-UEL element library.

eigenvalue decomposition. Finally, we can obtain the stiffness matrix $[K]$ of the PSBFEM element. Moreover, to avoid calculating the stiffness matrix $[K]$ at each incremental step and decrease the calculation cost, we only calculate the stiffness matrix $[K]$ at the first incremental step and store it in the state variable. The stiffness matrix $[K]$ is read directly at the next incremental step.

To solve the stiffness matrix, we need to employ the eigenvalue decomposition (see equation (17)). At present, many mathematical libraries to perform the eigenvalue decomposition exist. In this work, we use the Intel Math Kernel Library (MKL) [35] to decompose the eigenvalue. In the Abaqus/Standard analysis, we can directly use MKL by modifying the Abaqus environment file.

3.2. Defining the UEL Elements. Abaqus's input file usually contains a model information section (such as defining nodes, elements, the active degrees of freedom, and materials). At present, this information cannot be set in Abaqus CAE and must be defined through an input file. The Abaqus provides the keyword ∗USER ELEMENT to defined a new user element. The main contents of defining the user element are as follows:

(1) Assigning an element type key to a user-defined element and the number of nodes. The element type key must be of the form "Un" in Abaqus/Standard analysis, where 'n' is a positive integer that identifies the element type uniquely. In this implementation of PSBFEM, the integer is equal to the number of nodes of the element.

(2) Defining the element properties and assigning an Abaqus material to the user element.

(3) Defining the number of degrees of freedom per node and the active degrees of freedom at the nodes.

Listing 1: user element definition for the arbitrary polygons c.f. Figure 8.

(1) *USER ELEMENT, NODES = 3, TYPE = U3, PROPERTIES = 2, COORDINATES = 2

(2) 1, 2

(3) *ELEMENT, TYPE = U3, ELSET = E3

(4) 2,1,2,7

(5) 4,5,6,4

(6) *UEL PROPERTY, ELEST = E3

(7) 1000, 0.2

(8) *USER ELEMENT, NODES = 4, TYPE = U4, PROPERTIES = 2, COORDINATES = 2

(9) 1,2

(10) *ELEMENT, TYPE = U4, ELSET = E4

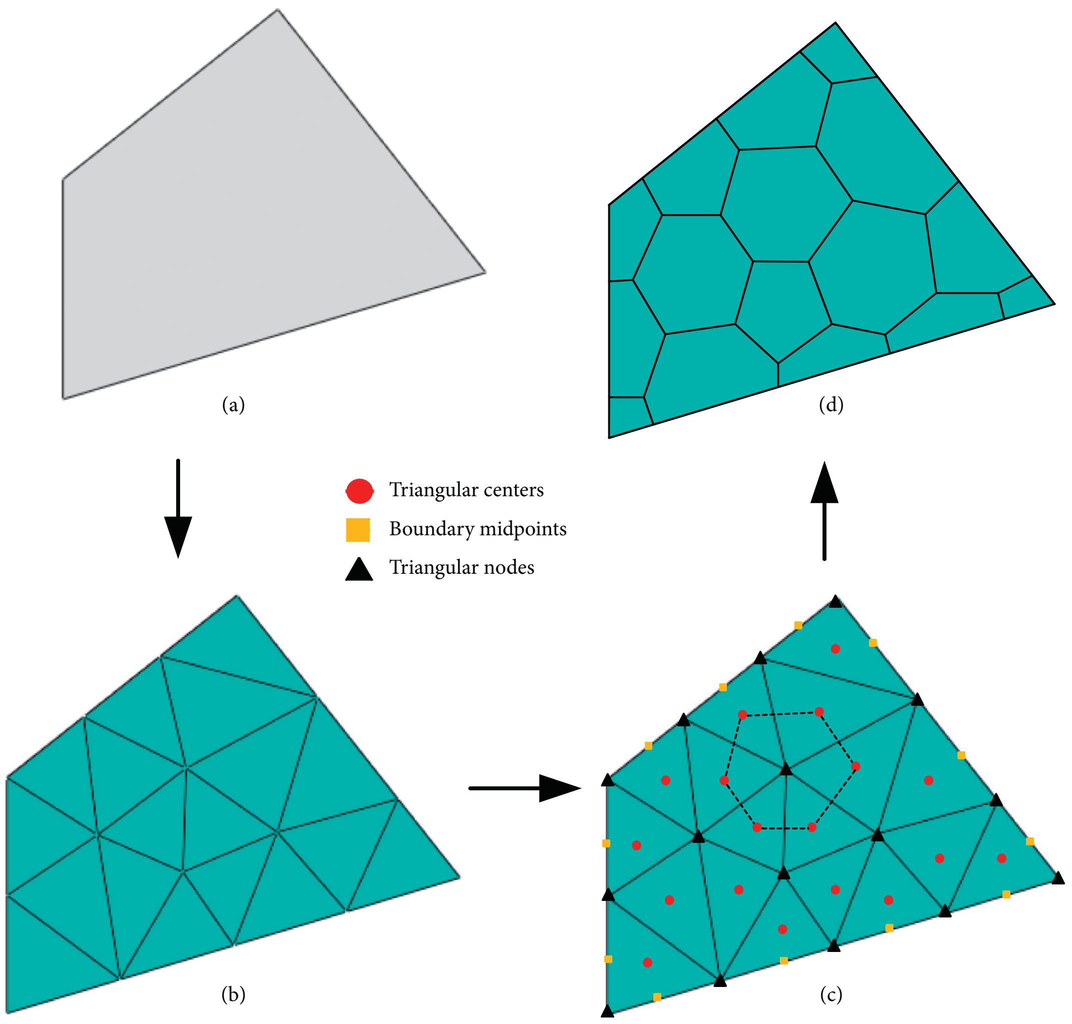


Figure 4: Generation of a polygon element from a triangular element: (a) arbitrary geometric model; (b) generating triangular meshes by Abaqus CAE; (c) construction of polygon meshes; and (d) final SBFEM polygon meshes.

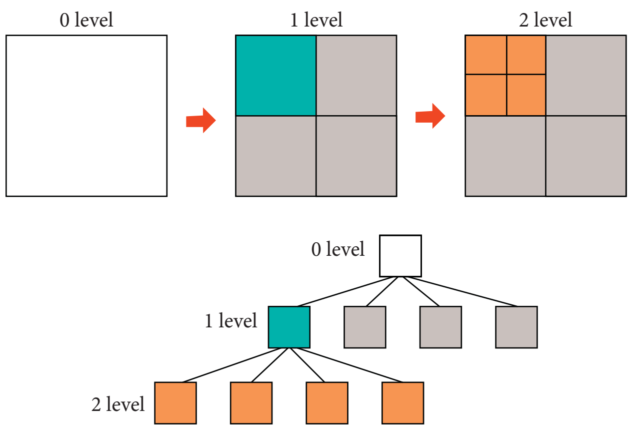


Figure 5: A quadtree discretization with three levels.

(11) 1,1,7,6,5

(12) *UEL PROPERTY, ELEST = E1

(13) 1000,0.2

(14) *USER ELEMENT, NODES = 5, TYPE = U5, PROPERTIES = 2, COORDINATES = 2

(15) 1,2

(16) *ELEMENT, TYPE = U5, ELSET = E5

(17) 3,2,3,4,6,7

(18) *UEL PROPERTY, ELEST = E1

(19) 1000,0.2

In the case of PSBFEM, we present a simple polygonal mesh of PSBFEM (see Figure 8). This mesh consists of three element types: triangular element (U3), quadrilateral element (U4), and Pentagon element (U5). In the input file (see listing 1), 1 ~ 19 is the line number; the actual input file does not contain the line number. Lines 1 ~ 7 are used to define two triangular elements (U3). Line 1 assigns the element type, the number of nodes, the number of element properties, and the number of freedom degrees per node. Line 2 sets the active degrees of freedom. Lines 3 ~ 5 define the element sets' E3'. Lines 6 ~ 7 set the element properties (Young's modulus and Poisson's ratio) of "E3". Similarly, other types of elements are defined using the same way.

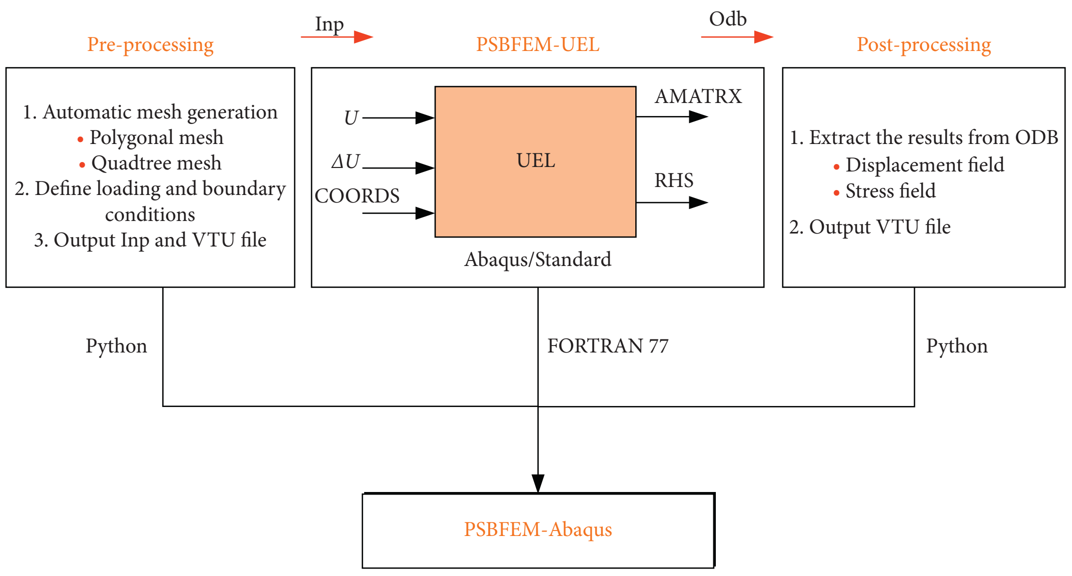


Figure 6: Framework sketch of implementing the PSBFEM within Abaqus.

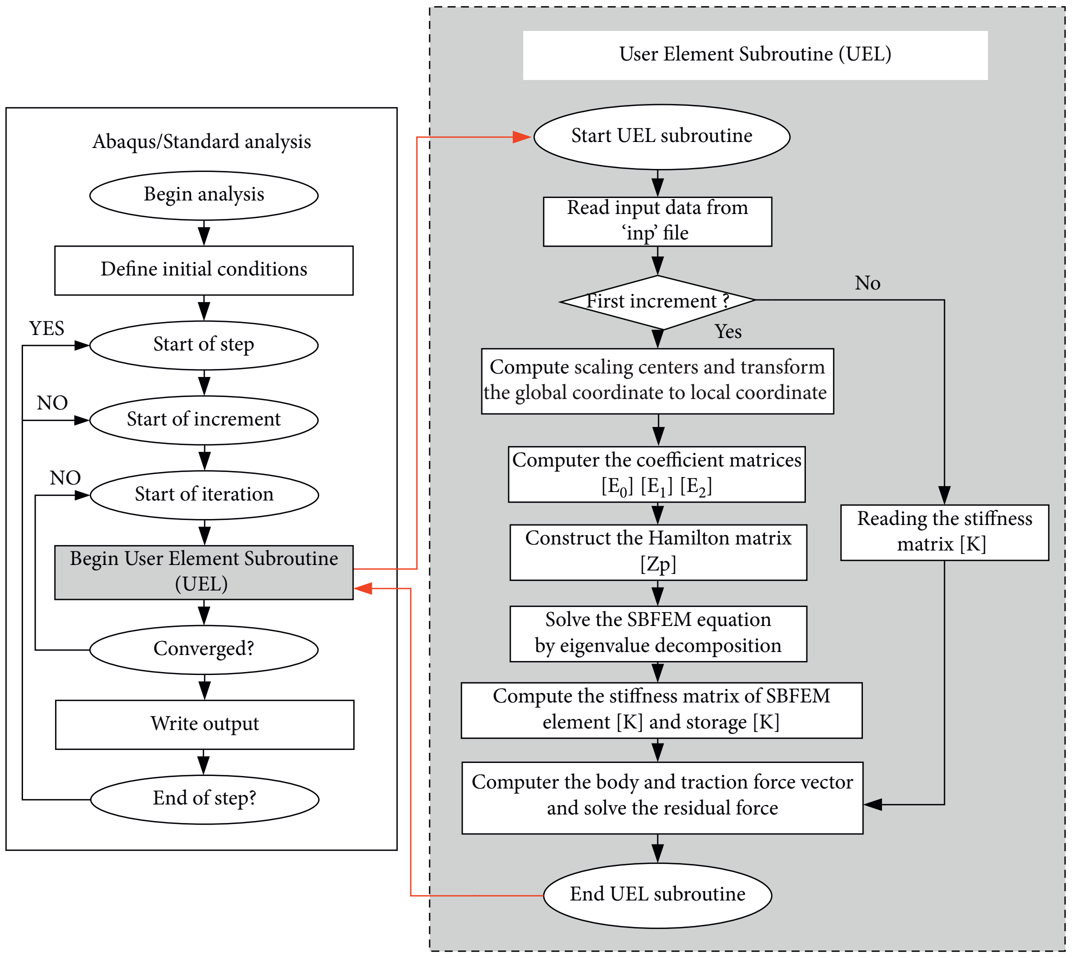

User Element Subroutine (UEL)
Abaqus/Standard analysis
Begin analysis
Define initial conditions
YES
Start of step
NO
Start of increment
NO
Start of iteration
Begin User Element Subroutine (UEL)
Converged?
Write output
End of step?
Start UEL subroutine
Read input data from 'inp' file
First increment ?
No
Yes
Compute scaling centers and transform the global coordinate to local coordinate
Computer the coefficient matrices [E0] [E1] [E2]
Construct the Hamilton matrix [Zp]
Solve the SBFEM equation by eigenvalue decomposition
Compute the stiffness matrix of SBFEM element [K] and storage [K]
Computer the body and traction force vector and solve the residual force
Reading the stiffness matrix [K]
End UEL subroutine

Figure 7: Framework sketch of implementing the UEL.

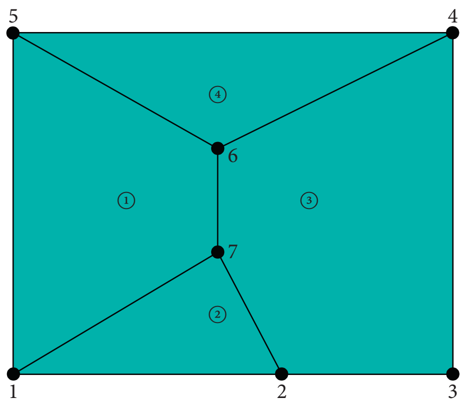


Figure 8: A simple example of the SBFEM polygonal mesh.

4. Numerical Examples

In this section, we validate the convergence and accuracy of the implementation by solving a few benchmark problems. Moreover, the results of the PSBFEM are compared with the FEM. The FEM analysis uses the commercial finite element software Abaqus. For validation, the relative error L^2 norms in the displacement are computed as follows:

$$e_u = \|\mathbf{u}-\mathbf{u}\|^h_{L^2(\Omega)} = \frac{\sqrt{\int_\Omega \left(\mathbf{u}-\mathbf{u}^h\right)^{\mathrm{T}}\left(\mathbf{u}-\mathbf{u}^h\right)\mathrm{d}\Omega}}{\sqrt{\int_\Omega \mathbf{u}^{\mathrm{T}}\mathbf{u}\mathrm{d}\Omega}}, \tag{26}$$

where $\mathbf{u^h}$ is the numerical solution and $\mathbf{u}$ is the analytical or reference solution.

The relative error in the energy as given by

$$e_{\text{enery}} = \|\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}\|^h_{L^2(\Omega)} = \frac{\sqrt{\int_\Omega \left(\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}^h\right)^{\mathrm{T}} D\left(\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}^h\right)\mathrm{d}\Omega}}{\sqrt{\int_\Omega \boldsymbol{\varepsilon}^{\mathrm{T}} D\boldsymbol{\varepsilon}\mathrm{d}\Omega}}, \tag{27}$$

where $\varepsilon^{\mathbf{h}}$ is the numerical solution and ε is the analytical or reference solution.

4.1. A Two-Dimensional Cantilever Beam. A two-dimensional cantilever beam of height $H = 1.0\,\mathrm{m}$ and length $L = 5.0\,\mathrm{m}$ subjected to a uniform loading $P = 10\,\mathrm{kPa}$ is considered, as shown in Figure 9(a). The material properties are given by Young's modulus $E = 2\,\mathrm{GPa}$ and Poisson's ratio $\nu = 0$. The right boundaries are constrained without displacement $(\Delta X = 0, \Delta Y = 0)$. The domain is discretized with the quadrangle and arbitrary polygonal elements. A convergence study is performed by mesh refinement. The meshes are refined successively following the sequence $n = 2, 4, 8, 16, 32$. The element size is chosen as $(h = H/n)$. A representative mesh is presented in Figures 9(b) and 9(c). In this work, the Abaqus uses the CPS4 element. The analytical solution of the stress fields can be expressed as [40–42]

$$\sigma_x = q\frac{y}{h}\left(4\frac{y^2}{h^2}-\frac{3}{5}-6\frac{x^2}{h^2}\right), \tag{28a}$$

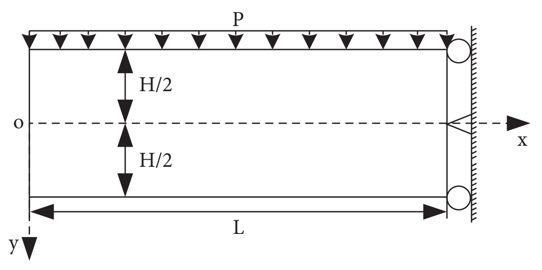
$$\sigma_y = -\frac{q}{2}\left(1-3\frac{y}{h}+4\frac{y^3}{h^3}\right), \tag{28b}$$

$$\tau_{xy} = -\frac{3q}{2}\frac{x}{h}\left(1-4\frac{y^2}{h^2}\right). \tag{28c}$$

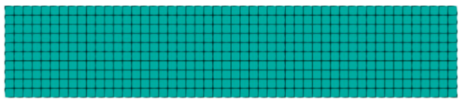
The results of the relative error in the vertical displacement u_y for the point O are given in Table 1. It can be observed that the errors decrease as the mesh refinement. The errors of PSBFEM are less than the FEM at the same element size. Moreover, the polygonal element shows higher accuracy than the quadrilateral element in the PSBFEM because the polygonal element has more nodes for the same element size. Figure 10 shows that the results of mesh size sensitivity for three element types. It is clear that these elements would obtain more accurate results when mesh sizeis less than 1/8 m. The convergence of the relative error in the displacement and the energy norm with mesh refinement are presented in Figures 11 and 12. It is observed that the PSBFEM converges to an exact solution with an optimal convergence rate.

Moreover, Figure 13 shows the contours of the vertical displacement u_y of Abaqus CPS4, PSBFEM-Quad, and PSBFEM polygon. It is clearly shown that the results are virtually the same for the FEM and PSBFEM. In addition, Figure 14 presents the computational cost comparison of the developed UEL and Abaqus standard elements. Due to the increment size setting affecting the solving time, we use the automatic incrementation type. The comparison is evaluated with an Intel Core i7-4710MQ CPU running at 2.50 GHz and 4.0 GB of RAM. The total CPU time is normalized. It is noted that computation costs for the PSBFEM-UEL and Abaqus standard elements are comparable and of the same trend. Moreover, the computational cost of PSBFEM-UEL is slightly higher than the Abaqus standard element. It is due to the semianalytical method of SBFEM being bound to generate relatively more computations in its concept than FEM [43].

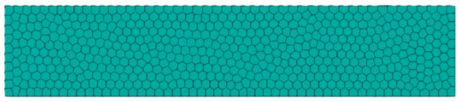
4.2. Infinite Plate with a Circular Hole. In this example, an infinite plate with a circular hole of radius a under remote uniaxial tension σ_x is considered as shown in Figure 15(a). Considering the symmetry of geometry, we only analyze a quarter of the infinite plate, as shown in Figure 15(b). The analytical solution of stress fields in the polar coordinates (r, θ) is expressed as [36]



(a)



(b)



(c)

Figure 9: A cantilever beam subjected to a uniform loading: (a) geometry and boundary conditions; (b) the quadrilateral mesh; (c) the polygonal mesh; the discretization of the domain by a Python script, and visualization mesh by the software Paraview.

Table 1: The relative error in the vertical displacement u_y of a cantilever beam subjected to a uniform loading with mesh refinement.

Element size (m)	FEM		PSBFEM-Quad		PSBFEM-Polygon	
	DOFs	e_u (%)	DOFs	e_u (%)	DOFs	e_u (%)
0.5	66	1.13×10^{-1}	66	8.75×10^{-2}	82	8.20×10^{-2}
0.25	210	3.11×10^{-2}	210	2.37×10^{-2}	322	1.86×10^{-2}
0.125	738	8.01×10^{-3}	738	6.08×10^{-3}	1282	3.30×10^{-3}
0.0625	2754	2.02×10^{-3}	2754	1.53×10^{-3}	5124	7.73×10^{-4}
0.03125	10626	5.07×10^{-4}	10626	3.83×10^{-4}	20480	1.96×10^{-4}

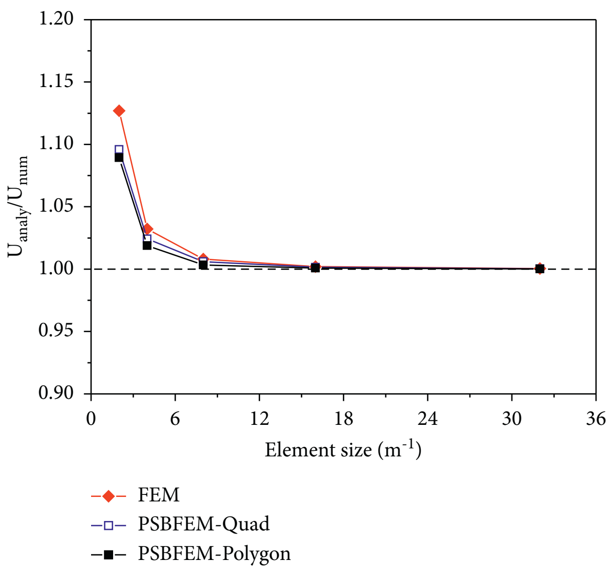


Figure 10: Sensitivity analysis of mesh sizes (U_{analy} denotes the analytical solution of displacement; U_{num} denotes the numerical solution of displacement).

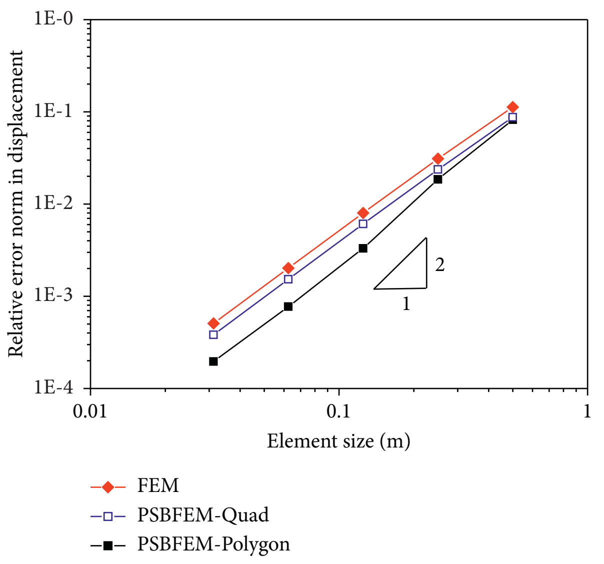


Figure 11: Convergence of the relative error in the displacements of a cantilever beam is subjected to uniform loading.

$$\sigma_x = \frac{p}{2}\left(2 - \frac{a^2}{r^2}\left(3\cos 2\theta + \left(2 - 3\frac{a^2}{r^2}\right)\cos 4\theta\right)\right), \quad (29a)$$

$$\sigma_y = -\frac{pa^2}{2r^2}\left(\cos 2\theta - \left(2 - 3\frac{a^2}{r^2}\right)\cos 4\theta\right), \quad (29b)$$

$$\tau_{xy} = -\frac{pa^2}{2r^2}\left(\sin 2\theta + \left(2 - 3\frac{a^2}{r^2}\right)\sin 4\theta\right), \quad (29c)$$

and the displacement fields can be expressed as

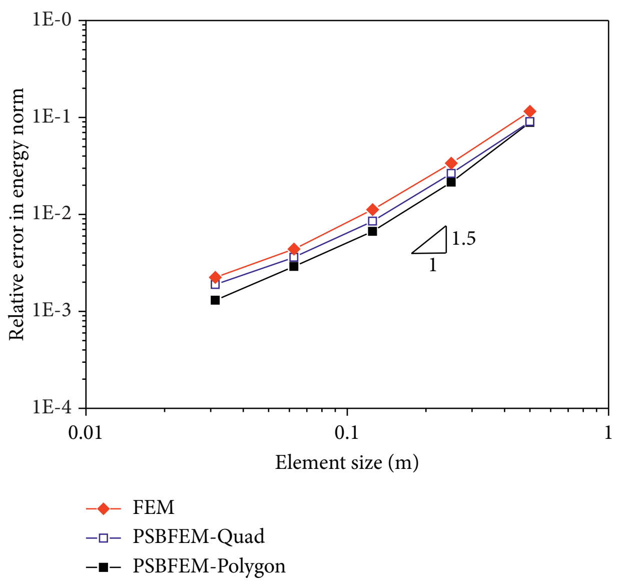
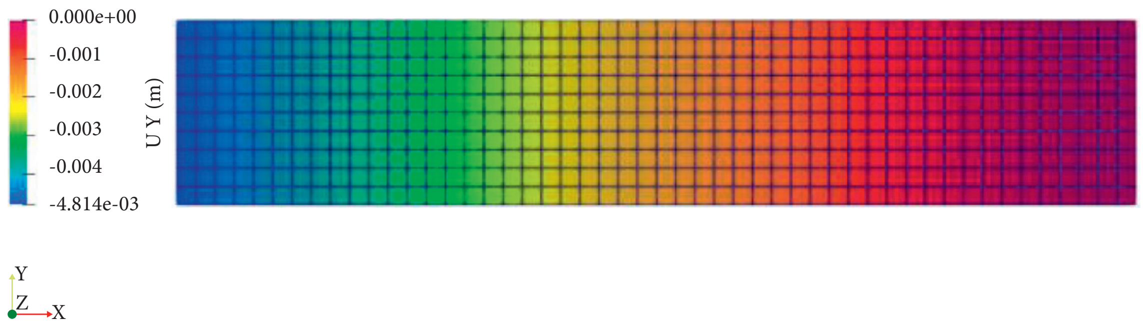
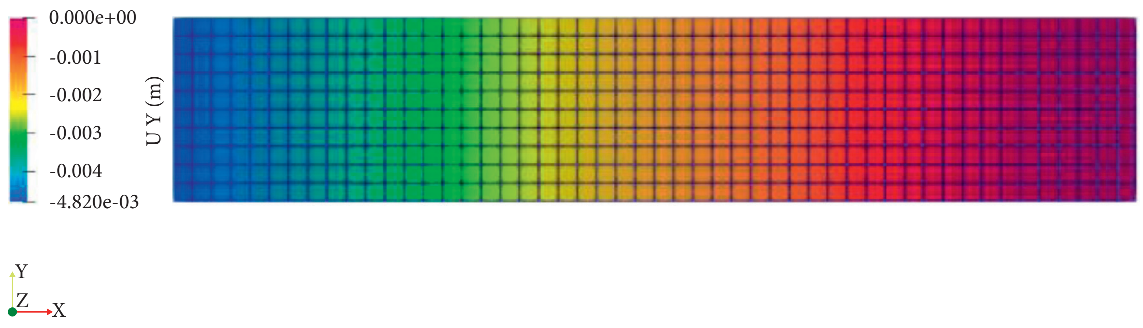


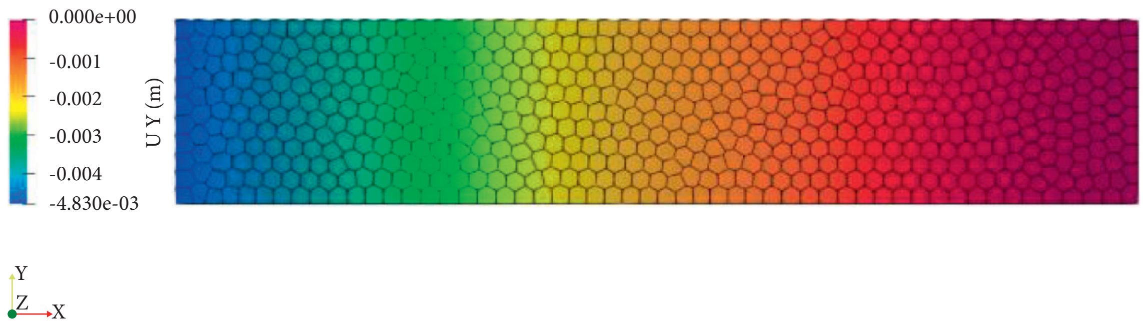
Figure 12: Convergence of relative error in the energy for a cantilever beam is subjected to uniform loading.



(a)



(b)



(c)

Figure 13: Contour plots of displacements u_y in a cantilever beam: (a) Abaqus CPS4; (b) PSBFEM Quad; (c) PSBFEM Polygon.

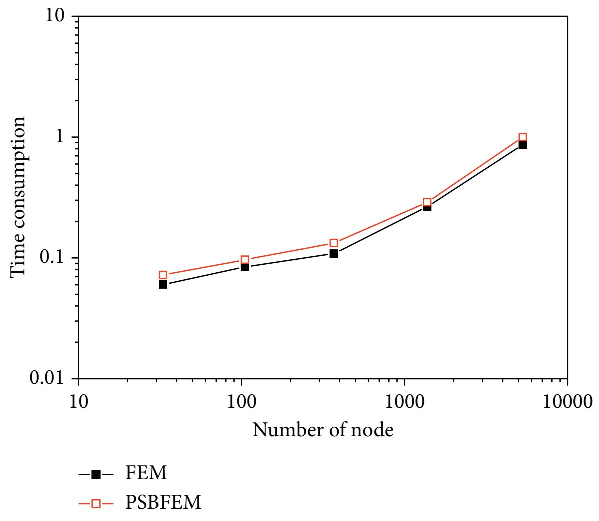


Figure 14: The time consumption comparison of PSBFEM-UEL between Abaqus standard elements.

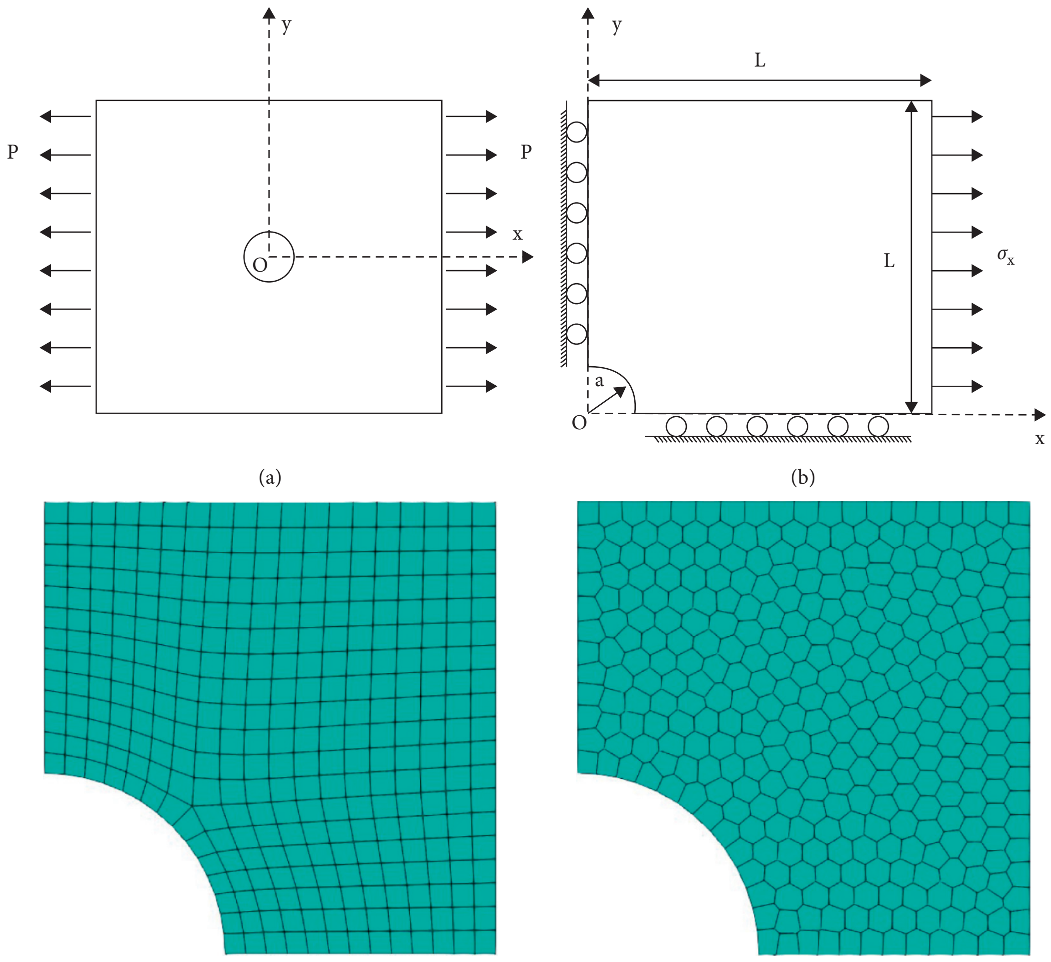


Figure 15: Infinite plate with a circular hole: geometry and boundary conditions: (a) a circular hole; (b) a quarter of the circular hole; (c) the quadrilateral mesh; and (d) the polygonal mesh.

$$u_x = \frac{Pa}{8G}\left(\frac{r}{a}(1+\kappa)\cos\theta + \frac{2a}{r}((1+\kappa)\cos\theta + \cos 3\theta) - \frac{2a^3}{r^3}\cos 3\theta\right), \tag{30a}$$

$$u_y = \frac{Pa}{8G}\left(\frac{r}{a}(\kappa-3)\sin\theta + \frac{2a}{r}((1-\kappa)\sin\theta + \sin 3\theta) - 2\frac{a^3}{r^3}\sin 3\theta\right). \tag{30b}$$

The shear modulus G and Kolosov's constant κ can be expressed as

$$G = \frac{2E}{2(1+\nu)}, \quad k = \frac{3-\nu}{1+\nu}, \tag{31}$$

where a is the radius of the hole. The circular hole radius is 0.4 m, the length of the quarter of the circle hole is 1.0 m, and remote tension $\sigma_x = 1$ kPa. The domain is discretized using quadrilateral and polygonal meshes, respectively. The problem is also modeled by the plane stress, and the material properties are given by Young's modulus $E = 1 \times 10^5$ Pa and Poisson's ratio $\nu = 0.25$. The vertical displacement of bottom boundaries ($\Delta Y = 0$) and the horizontal displacement of left boundaries ($\Delta X = 0$) are constrained without displacement.

The results of PSBFEM and FEM in terms of convergence behavior and accuracy are compared and demonstrated. Figures 16 and 17 show the convergence of the relative error norms decrease in displacement and the energy norm with increasing elements. It is observed that the PSBFEM element is significantly more accurate than the FEM element. Besides, PSBFEM Polygon has a more slightly fast convergence rate.

From the results, it is clear that all the methods asymptotically converge to the analytical solution with mesh refinement. It is noted that the PSBFEM requires fewer DOFs when compared to conventional FEM. Besides, Figure 18(a) provides that the contours of the vertical displacement u_y of Abaqus CPS4, PSBFEM-Quad, and PSBFEM polygon in an infinite plate with a circular hole. The results show a good agreement for the FEM and PSBFEM. Moreover, Figures 18(b) and 18(c) also show that the distribution of stress component σ_y and strain component $\varepsilon_y$obtained from the PSBFEM and the FEM by Abaqus. Similarly, good agreement between the three sets of results is observed.

4.3. Pressurized Thick Cylinder Modeled by a Quarter-Annulus Model.

4.3. Pressurized Thick Cylinder Modeled by a Quarter-Annulus Model. In this example, we consider a benchmark problem of a quarter-thick cylinder subjected to internal pressure at the inner circular edge. The analytical solution of stress fields in the polar coordinates is [44]

$$\sigma_{rr} = \frac{R_a^2 P}{R_a^2 - R_b^2}\left(1 - \frac{R_a^2}{r^2}\right), \tag{32a}$$

$$\sigma_{\theta\theta} = \frac{R_a^2 P}{R_a^2 - R_b^2}\left(1 + \frac{R_a^2}{r^2}\right), \tag{32b}$$

$$\sigma_{r\theta} = 0, \tag{32c}$$

and the displacement fields can be expressed as

$$u_{\text{rad}} = \frac{R_a^2 P r}{E\left(R_b^2 - R_a^2\right)}\left(1 - \nu + \left(\frac{R_b}{r}\right)^2 (1+\nu)\right), \tag{33a}$$

$$u_x = u_{\text{rad}}\cos\theta, \tag{33b}$$

$$u_y = u_{\text{rad}}\sin\theta. \tag{33c}$$

where R_a and R_b are the inner and the outer radius of the cylinder, respectively, with $R_a = 1$ m and $R_b = 2$ m. P is the pressure exerted along the inner circular edge, with $P = 1000$ Pa. The material properties are given by Young's modulus $E = 1 \times 10^5$ Pa and Poisson's ratio $\nu = 0.30$. The geometry and boundary conditions are presented in Figure 19(a). The domain is discretized with the quadrilateral and polygonal elements, as shown in Figures 19(b) and 19(c).

Figures 20 and 21 show the convergence of the relative error in both the displacement and the energy norm with mesh refinement. Results show that all the techniques asymptotically converge to analytical solutions with reducing element size, and the PSBFEM Polygon shows slightly accurate results. Moreover, a good agreement between PSFEM and FEM is observed in Figure 22.

4.4. Square Body with Multiple Holes. To highlight the flexibility of the PSBFEM element and quadtree algorithm in handling mesh, a unit square body ($L = 1.0$ m) with four circular holes is considered, as shown in Figure 23(a). The material properties are Young's modulus $E = 10$ GPa and Poisson's ratio $\nu = 0.25$. The bottom edge is constrained without displacement ($\Delta X = 0, \Delta Y = 0$), and a uniform tension $P = 1$ MPa is applied on the top edge of the square. The four circular holes locate ($X = 0.5$ m, $Y = 0.75$ m, $r = 0.05$ m), ($X = 0.25$ m, $Y = 0.5$ m, $r = 0.15$ m), ($X = 0.5$ m, $Y = 0.25$ m, $r = 0.05$ m), and ($X = 0.75$ m, $Y = 0.5$ m, $r = 0.15$ m), respectively, to form a hole cluster. For the error estimation, the relative error L^2 norms in the displacement are computed as equation (26). Moreover, a simple regression is investigated to compare the results of FEM and PSBFEM.

The quadtree mesh is generated by setting the same number of mesh seeds on every hole, as shown in Figure 23(c). It is clearly present that the mesh transition between the holes of different sizes is effectively handled. The square body with multiple holes is modeled with 1632 quadtree elements, and the total nodes are 2478.

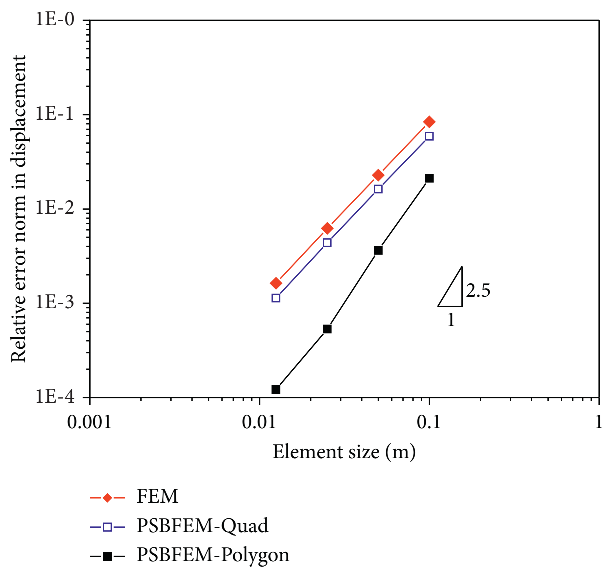


FIGURE 16: Convergence of the relative error in the displacement for an infinite plate with a circular hole.

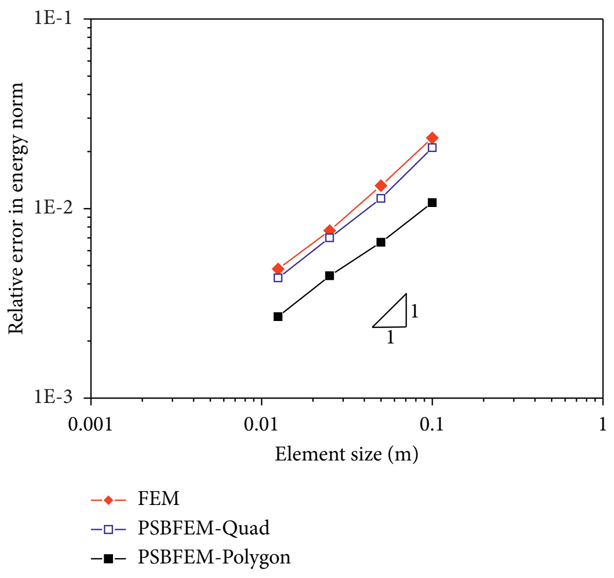


FIGURE 17: Convergence of the relative error in the energy for an infinite plate with a circular hole.

Moreover, this problem is also analyzed with a similar number of nodes (2532 nodes) using the Abaqus CPS4 element. Due to the symmetry of geometry, only the results on the right side of the top edge are provided for comparison. Results in Figure 24 show the comparison between quadtree PSBFEM and Abaqus CPS4 in the vertical displacement u_y. A simple regression on the Abaqus CPS4 and quadtree PSBFEM had presented $R^2 = 0.99$, and the error in the L^2 norm is 0.0097. The strain energy of FEM is 85.934 kJ, and the strain energy of PSBFEM is 86.5917 kJ. The relative error of strain energy is 0.76%. Hence, these results indicate PSBFEM accuracy and reliability for the quadtree mesh. Also, the contour plots of the vertical displacement u_y obtained from the PSBFEM analysis and the FEM analysis by Abaqus are shown in Figure 25. It is noted that the contour plots present a good agreement.

4.5. A Square Plate for the Complex Geometry. A square body with a rabbit shape cavity, as shown in Figure 26. The size of the square plate is $L = 4$ m. The material properties are Young's modulus $E = 1$ MPa and Poisson's ratio $v = 0.25$. The left edge is constrained without displacement

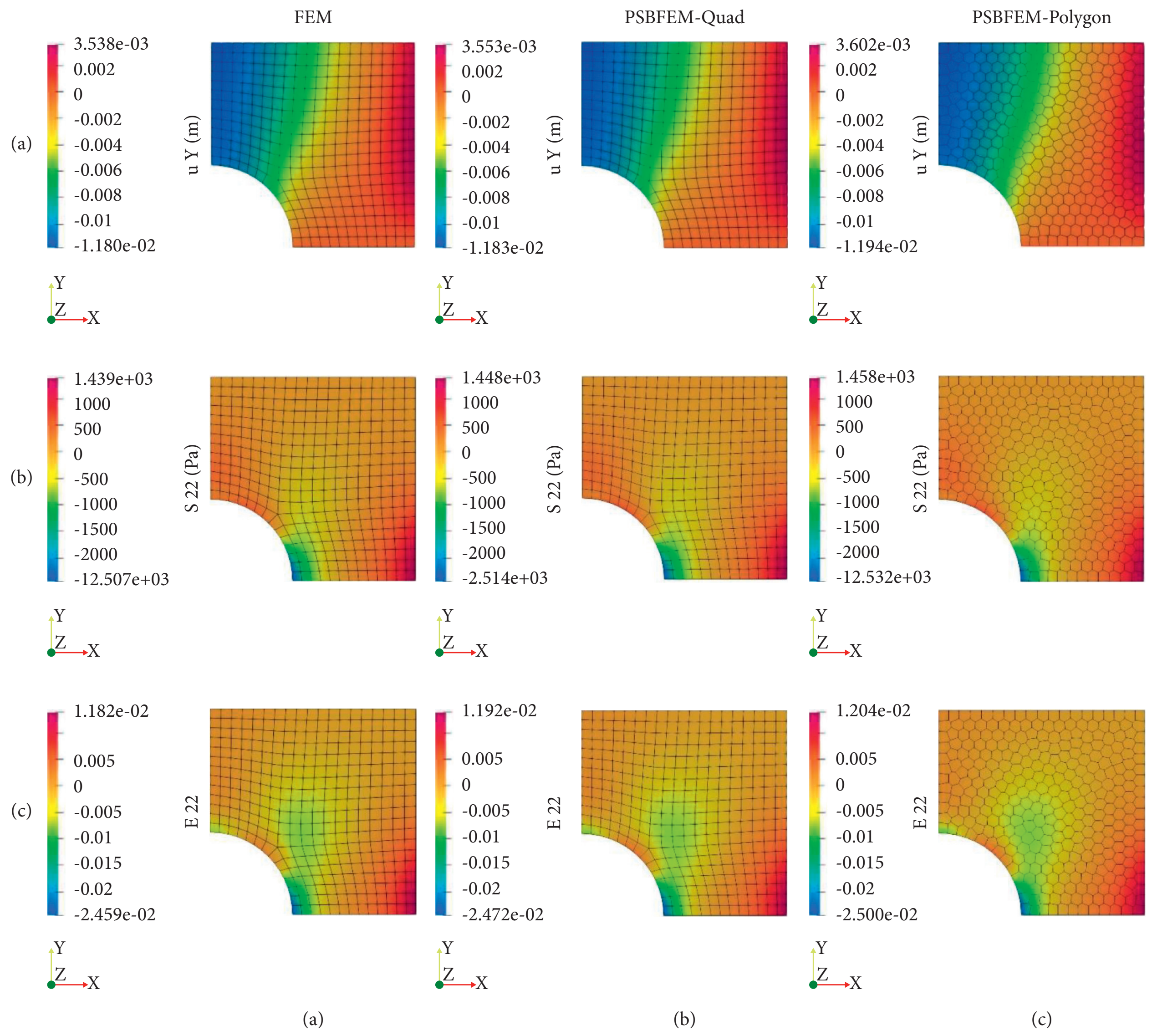


FIGURE 18: Contour plots of results in an infinite plate with a circular hole: (a) vertical displacement u_y; (b) stress component σ_y; and (c) strain component ε_y.

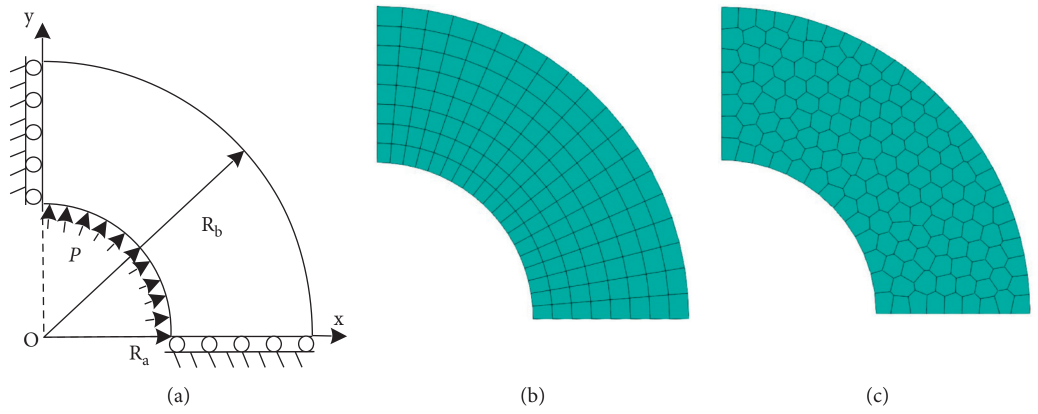


FIGURE 19: A quarter-thick cylinder subjected to internal pressure at the inner circular edge: (a) the geometry and boundary conditions for a quarter-annulus model; (b) the quadrilateral mesh; and (c) the polygonal mesh.

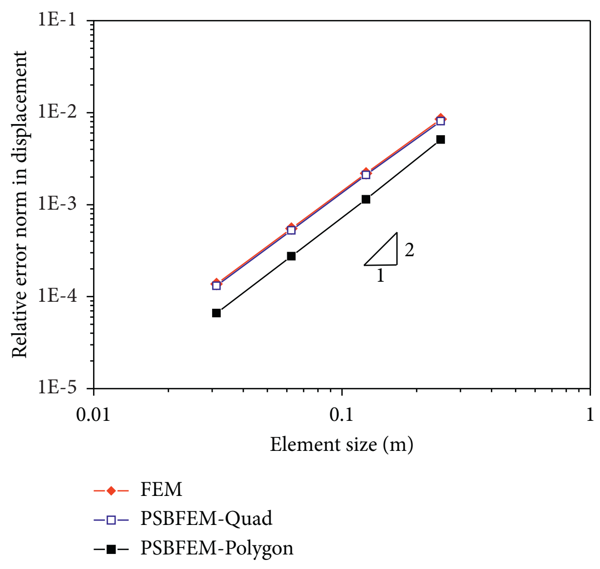


Figure 20: Convergence of the relative error in the displacement for a quarter-thick cylinder.

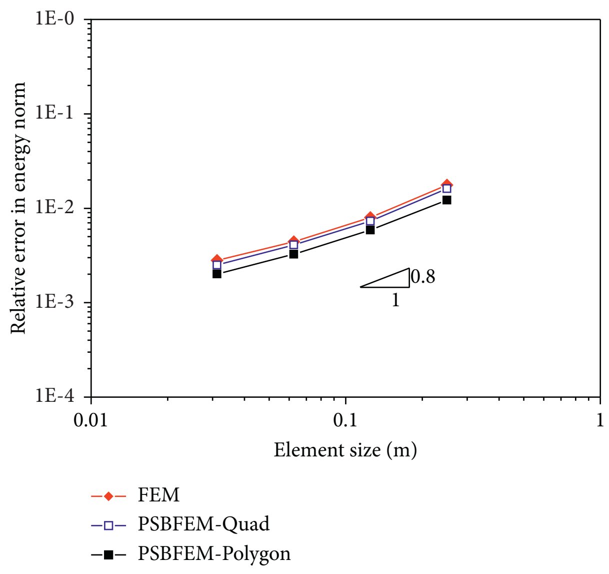
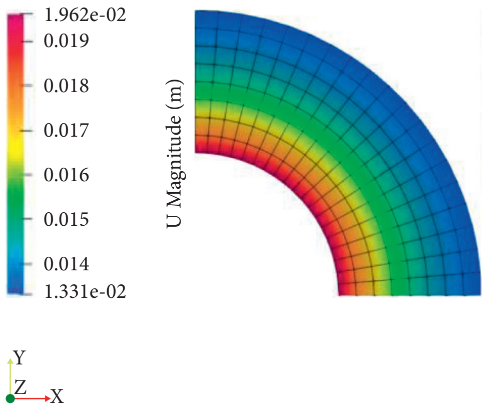


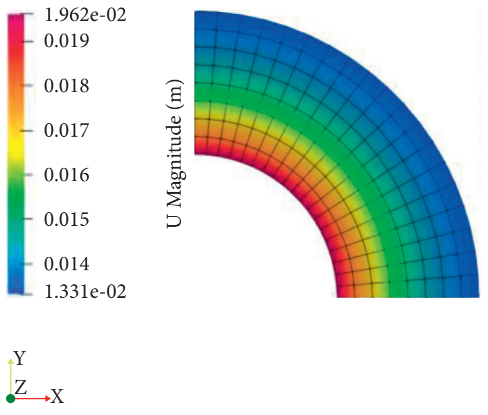
Figure 21: Convergence of the relative error in the energy for a quarter-thick cylinder.

$(\Delta X = 0, \Delta Y = 0)$, and the right edge is applied a horizontal displacement $U = 0.1$ m, as shown in Figure 26. We chose four nodes, A, B, C, D, to compare results, as shown in Figure 26. The square body with a rabbit shape cavity is modeled with 3147 quadtree elements. Moreover, this problem is also analyzed with a similar number of elements using the Abaqus CPS4 element, and the total elements are 3358, as shown in Figure 27.

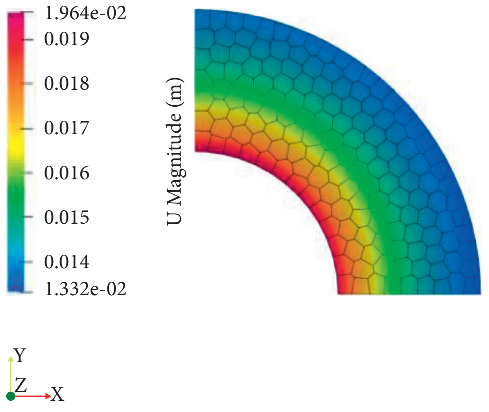
Table 2 shows the relative error in the horizontal displacement for the PSBFEM and FEM. The relative errors are less than 0.5%. Moreover, the strain energy of FEM is 4.304 kJ, and the strain energy of PSBFEM is 4.299 kJ. The relative error of strain energy is 0.12%. Figure 28 presents that the contour plots present a good agreement for FEM and PSBFEM. Therefore, the quadtree mesh of PSBFEM has sufficient accuracy and reliability.



(a)

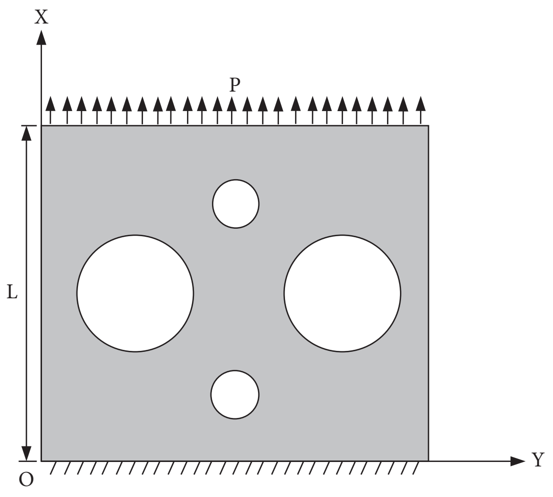


(b)

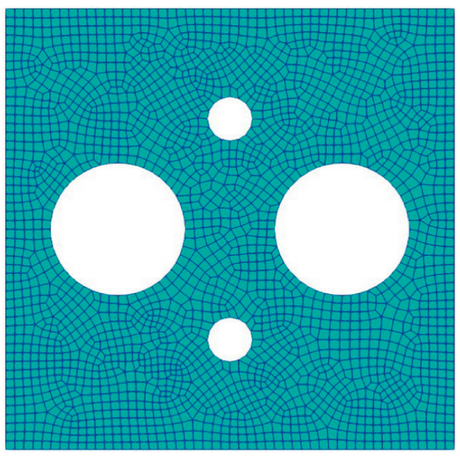


(c)

Figure 22: Contour plots of resultant displacements in a quarter-thick cylinder: (a) Abaqus CPS4; (b) PSBFEM Quad; and (c) PSBFEM Polygon.



(a)



(b)

Figure 23: Continued.

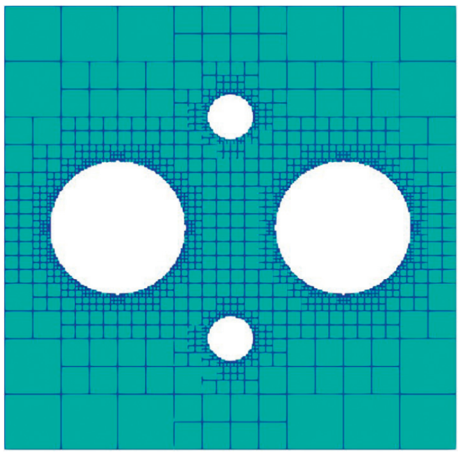

(c)

Figure 23: Schematic diagram of the square body with multiple holes under uniaxial tension: (a) the geometry and boundary conditions; (b) Abaqus mesh; and (c) Quadtree mesh.

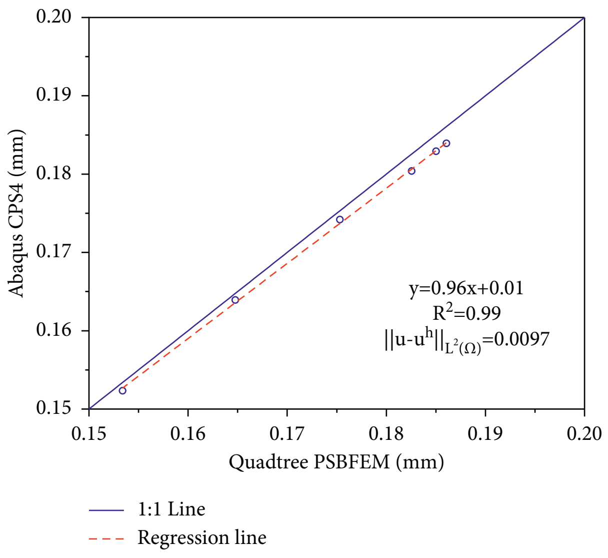
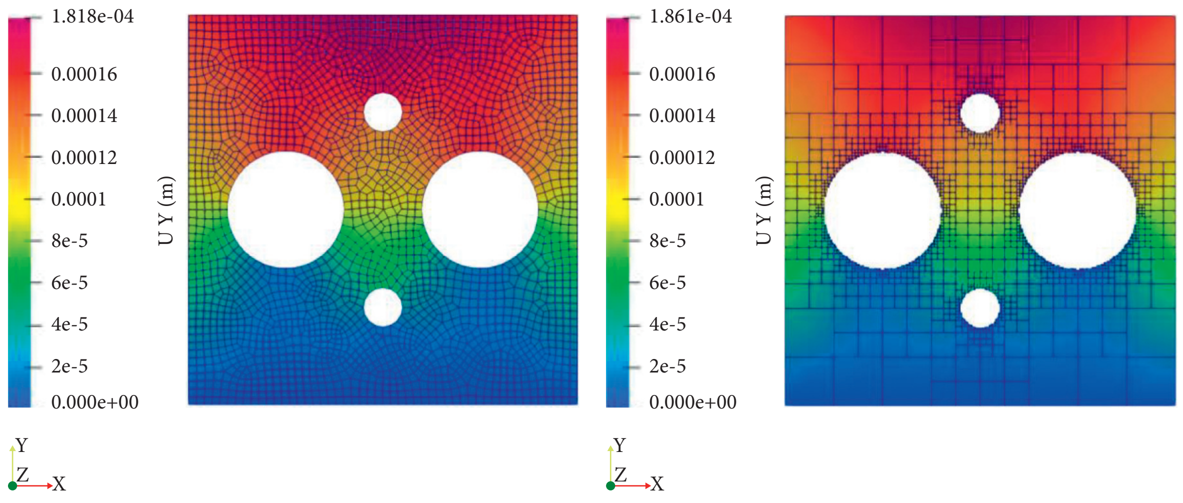


Figure 24: Comparison between quadtree PSBFEM and Abaqus CPS4 in the vertical displacement u_y on the top edge.



(a) (b)

Figure 25: Contour plots of displacements u_y in a square body with multiple holes: (a) Abaqus CPS4 and (b) Quadtree PSBFEM.

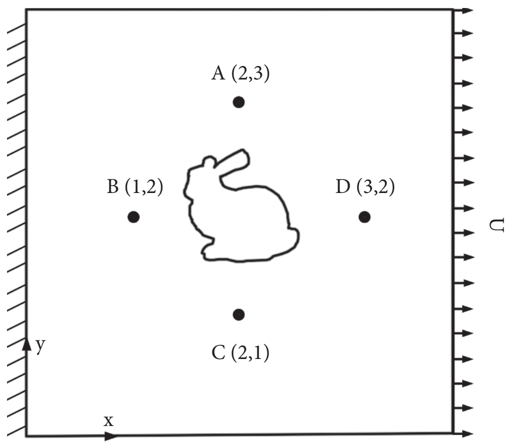


Figure 26: The geometry and boundary conditions of the square body with a rabbit shape cavity.

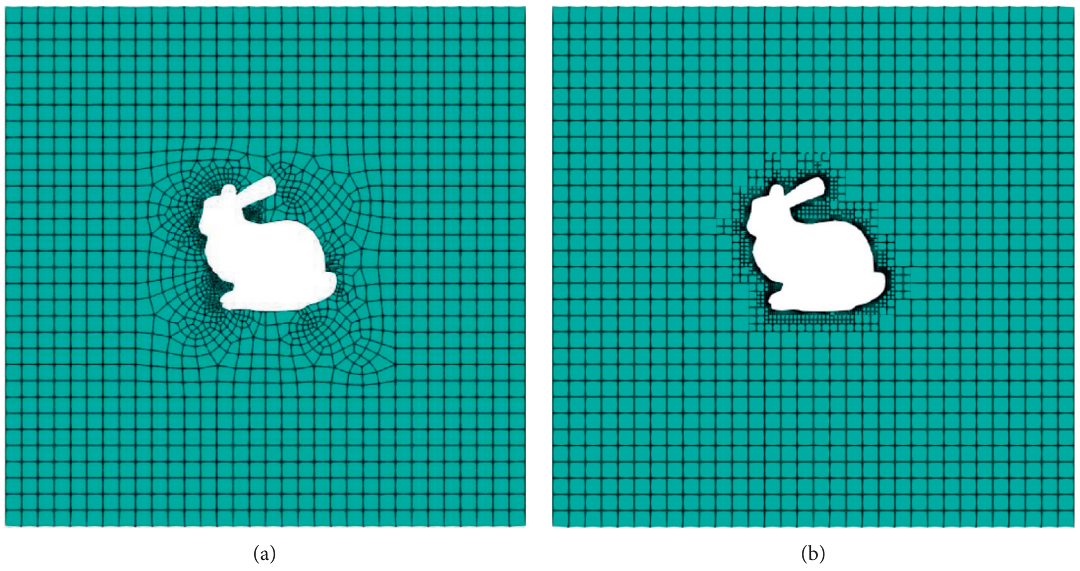


Figure 27: The meshes of a square plate with a rabbit shape cavity: (a) Abaqus mesh and (b) Quadtree mesh.

Table 2: The relative error for the PSBFEM and FEM in the horizontal displacement.

Point	Horizontal displacement u_x		
	FEM (m)	PSBFEM (m)	e_u (%)
A	0.05042	0.05032	0.20
B	0.01270	0.01264	0.47
C	0.04959	0.04956	0.06
D	0.08600	0.08616	0.19

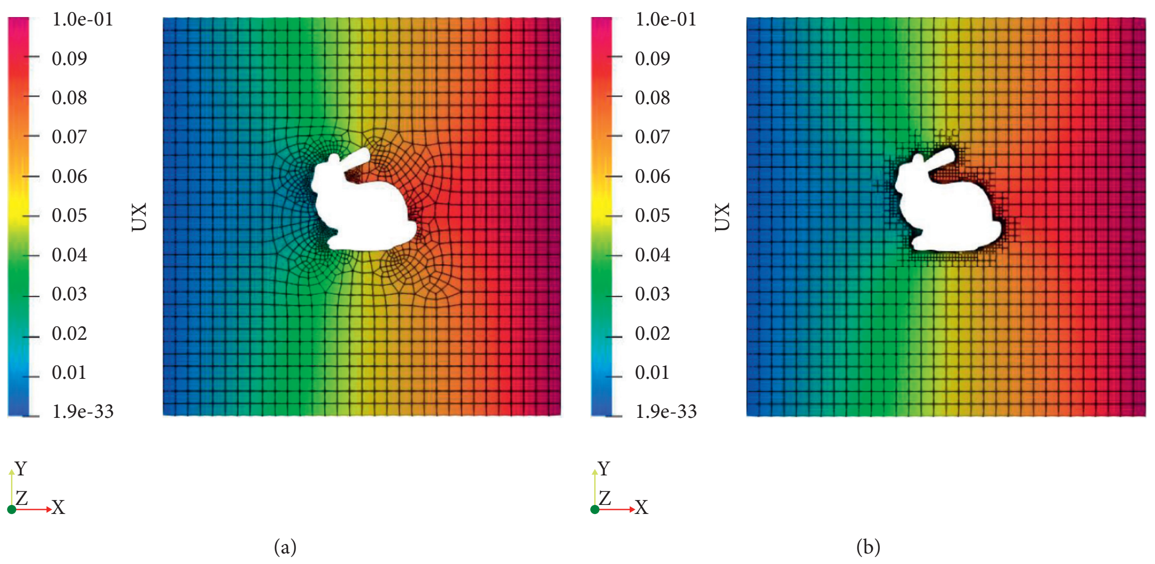


Figure 28: Contour plots of horizontal displacements in a square plate with a rabbit shape cavity: (a) Abaqus CPS4 and (b) Quadtree PSBFEM.

5. Conclusions

This paper implements the PSBFEM of two-dimensional linear elastostatic problems within the Abaqus/Standard analysis by the UEL subroutine. This work mainly focuses on the main procedures to interact with Abaqus, defining the UEL element in the input file, and solving the stiffness matrix by the eigenvalue decomposition in the UEL implementation procedure. Also, we discuss the automatic mesh generation of polygon/quadtree and the visualization of results by the Paraview.

The implementation of PSBFEM is validated against the FEM by solving a few benchmark problems. The results demonstrate that PSBFEM-UEL has a significantly better than FEM convergence rate. Moreover, the polygon mesh has a higher accuracy rate than the quadrangle mesh in the PSBFEM-UEL. Notably, the implementation of PSBFEM can conveniently use arbitrary polygon elements by the polygon/quadtree discretizations in the commercial finite element software Abaqus. In the future, the scope of this approach developed here can be extended to higher-order elements. The source code of the implementation can be downloaded from https://github.com/hhupde/PSBFEM-Abaqus with input files of numerical examples presented in this work.

Abbreviation

FEM:	The finite element method
PDE:	Partial differential equations
IGA:	Isogeometric analysis
DNNs:	Deep neural networks
SBFEM:	The scaled boundary finite element method
PSBFEM:	The polygonal scaled boundary finite element method
UEL:	User element subroutine
MKL:	Intel math kernel library
CPS3:	3-node linear plane stress elements
CPE3:	3-node linear plane strain elements
CPS4:	4-node bilinear plane stress elements
CPE4:	4-node bilinear plane strain elements
RAM:	Random-access memory
CPU:	Central processing unit.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (Grant no. 51579089).

References

[1] C. Xu, Z. Huo, and G. Giunta, “Numerical simulation of gravity anomaly based on the unstructured element grid and finite element method,” *Mathematical Problems in Engineering*, vol. 2020, Article ID 3604084, 9 pages, 2020.

[2] Q. Li, J. Xing, R. Tang, and Y. Zhang, “Finite-element method for calculating the sound field in a tank with impedance boundaries,” *Mathematical Problems in Engineering*, vol. 2020, Article ID 6794760, 8 pages, 2020.

[3] M. Iqbal, K. Masood, A. Aljuhni, and A. Ahmad, “Generalized finite element method with time-independent enrichment functions for 3D transient heat diffusion problems,” *International Journal of Heat and Mass Transfer*, vol. 149, Article ID 118969, 2020.

[4] M. Iqbal, K. Alam, A. Ahmad, S. Maqsood, H. Ullah, and B. Ullah, “An enriched finite element method for efficient solutions of transient heat diffusion problems with

multiple heat sources," *Engineering with Computers*, pp. 1–17, 2021.

[5] E. T. Ooi, C. Song, and F. Tin-Loi, "A scaled boundary polygon formulation for elasto-plastic analyses," *Computer Methods in Applied Mechanics and Engineering*, vol. 268, pp. 905–937, 2014.

[6] P. Y. Kumbhar, A. Francis, N. Swaminathan, R. K. Annabattula, and S. Natarajan, "Development of user element routine (UEL) for cell-based smoothed finite element method (CSFEM) in Abaqus," *International Journal of Computational Methods*, vol. 17, no. 2, Article ID 1850128, 2018.

[7] N. T. Nguyen, T. Q. Bui, and T. T. Truong, "Transient dynamic fracture analysis by an extended meshfree method with different crack-tip enrichments," *Meccanica Journal of the Italian Association of Theoretical and Applied Mechanics*, vol. 52, pp. 2363–2390, 2017.

[8] X. You, W. Li, and Y. Chai, "A truly meshfree method for solving acoustic problems using local weak form and radial basis functions," *Applied Mathematics and Computation*, vol. 365, Article ID 124694, 2020.

[9] Y. Chai, C. Cheng, W. Li, and Y. Huang, "A hybrid Finite element-Meshfree method based on partition of unity for transient wave propagation problems in homogeneous and inhomogeneous media," *Applied Mathematical Modelling*, vol. 85, pp. 192–209, 2020.

[10] A. Francis, A. Ortiz-Bernardin, S. P. A. Bordas, and S. Natarajan, "Linear smoothed polygonal and polyhedral finite elements," *International Journal for Numerical Methods in Engineering*, vol. 109, pp. 1–28, 2017.

[11] V. P. Nguyen, C. Anitescu, S. P. A. Bordas, and T. Rabczuk, "Isogeometric analysis: an overview and computer implementation aspects," *Mathematics and Computers in Simulation*, vol. 117, pp. 89–116, 2015.

[12] E. Samaniego, C. Anitescu, S. Goswami et al., "An energy approach to the solution of partial differential equations in computational mechanics via machine learning: concepts, implementation and applications," *Computer Methods in Applied Mechanics and Engineering*, vol. 362, Article ID 112790, 2020.

[13] T. Vu-Huu, C. Le-Thanh, H. Nguyen-Xuan, and M. Abdel-Wahab, "Stabilization for equal-order polygonal finite element in incompressible fluid flow computation," *Computers, Materials and Continua*, vol. 62, no. 3, pp. 1109–1123, 2020.

[14] C. Song and J. P. Wolf, "The scaled boundary finite-element method-alias consistent infinitesimal finite-element cell method-for elastodynamics," *Computer Methods in Applied Mechanics and Engineering*, vol. 147, no. 3-4, pp. 329–355, 1997.

[15] H. Gravenkamp, A. A. Saputra, C. Song, and C. Birk, "Efficient wave propagation simulation on quadtree meshes using SBFEM with reduced modal basis," *International Journal for Numerical Methods in Engineering*, vol. 110, no. 12, pp. 1119–1141, 2017.

[16] H. Gravenkamp, S. Natarajan, and W. Dornisch, "On the use of NURBS-based discretizations in the scaled boundary finite element method for wave propagation problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 315, pp. 867–880, 2017.

[17] L. Fengzhi and R. Penghao, "A novel solution for heat conduction problems by extending scaled boundary finite element method," *International Journal of Heat and Mass Transfer*, vol. 95, pp. 678–688, 2016.

[18] Y. He, X. Dong, and H. Yang, "A new adaptive algorithm for phase change heat transfer problems based on quadtree SBFEM and smoothed effective heat capacity method," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 75, no. 2, pp. 111–126, 2019.

[19] A. W. Egger, E. N. Chatzi, and S. P. Triantafyllou, "An enhanced scaled boundary finite element method for linear elastic fracture," *Archive of Applied Mechanics*, vol. 87, no. 10, pp. 1667–1706, 2017.

[20] A. Egger, U. Pillai, K. Agathos et al., "Discrete and phase field methods for linear elastic fracture mechanics: a comparative study and state-of-the-art review," *Applied Sciences*, vol. 9, no. 12, Article ID 2436, 2019.

[21] L. Liu, J. Zhang, C. Song, C. Birk, A. A. Saputra, and W. Gao, "Automatic three-dimensional acoustic-structure interaction analysis using the scaled boundary finite element method," *Journal of Computational Physics*, vol. 395, pp. 432–460, 2019.

[22] J. Liu, J. Li, P. Li, G. Lin, T. Xu, and L. Chen, "New application of the isogeometric boundary representations methodology with SBFEM to seepage problems in complex domains," *Computers and Fluids*, vol. 174, pp. 241–255, 2018.

[23] A. Johari and A. Heydari, "Reliability analysis of seepage using an applicable procedure based on stochastic scaled boundary finite element method," *Engineering Analysis with Boundary Elements*, vol. 94, pp. 44–59, 2018.

[24] W. Wang, Y. Peng, Z. Wei, Z. Guo, and Y. Jiang, "High performance analysis of liquid sloshing in horizontal circular tanks with internal body by using IGA-SBFEM," *Engineering Analysis with Boundary Elements*, vol. 101, pp. 1–16, 2019.

[25] J. Bielak, O. Ghattas, and E. J. Kim, "Parallel octree-based finite element method for large-scale earthquake ground motion simulation," *Computer Modeling in Engineering*, vol. 10, pp. 117–128, 2005.

[26] Q. Liang, G. Du, J. W. Hall, and A. G. Borthwick, "Flood inundation modeling with an adaptive quadtree grid shallow water equation solver," *Journal of Hydraulic Engineering*, vol. 134, pp. 1603–1610, 2015.

[27] S. Popinet, "Quadtree-adaptive tsunami modelling," *Ocean Dynamics*, vol. 61, no. 9, pp. 1261–1285, 2011.

[28] E. T. Ooi, H. Man, S. Natarajan, and C. Song, "Adaptation of quadtree meshes in the scaled boundary finite element method for crack propagation modelling," *Engineering Fracture Mechanics*, vol. 144, pp. 101–117, 2015.

[29] S. H. Huo, Y. S. Li, S. Y. Duan, X. Han, and G. R. Liu, "Novel quadtree algorithm for adaptive analysis based on cell-based smoothed finite element method," *Engineering Analysis with Boundary Elements*, vol. 106, pp. 541–554, 2019.

[30] J. Liang and J. Liang, "A user-defined element for dynamic analysis of saturated porous media in ABAQUS," *Computers and Geotechnics*, vol. 126, Article ID 103693, 2020.

[31] G. Molnár, A. Gravouil, R. Seghir, and J. Réthoré, "An open-source Abaqus implementation of the phase-field method to study the effect of plasticity on the instantaneous fracture toughness in dynamic crack propagation," *Computer Methods in Applied Mechanics and Engineering*, vol. 365, Article ID 113004, 2020.

[32] S. Ya, S. Eisenträger, C. Song, and J. Li, "An open-source ABAQUS implementation of the scaled boundary finite element method to study interfacial problems using polyhedral meshes," *Computer Methods in Applied Mechanics and Engineering*, vol. 381, Article ID 113766, 2021.

[33] K. M. Norén-Cosgriff, T. I. Bjørnarå, B. M. Dahl, and A. M. Kaynia, "Advantages and limitation of using 2-D FE modelling for assessment of effect of mitigation measures for

railway vibrations," *Applied Acoustics*, vol. 155, pp. 463–476, 2019.

[34] E. Anderson, Z. Bai, C. Bischof et al., *LAPACK Users' Guide*, SIAM, Philadelphia, PA, USA, 1999.

[35] E. Wang, Q. Zhang, B. Shen et al., "Intel Math Kernel Library," in *High-Performance Computing on the Intel®Xeon Phi™*, pp. 167–188, Springer, New York, NY, USA, 2014.

[36] C. Song, *The Scaled Boundary Finite Element Method: Introduction to Theory and Implementation*, John Wiley and Sons, Hoboken, NJ, USA, 2018.

[37] E. T. Ooi, C. Song, F. Tin-Loi, and Z. Yang, "Polygon scaled boundary finite elements for crack propagation modelling," *International Journal for Numerical Methods in Engineering*, vol. 91, no. 3, pp. 319–342, 2012.

[38] C. Jansari, K. Kannan, R. K. Annabattula, and S. Natarajan, "Others Adaptive phase field method for quasi-static brittle fracture using a recovery based error indicator and quadtree decomposition," *Engineering Fracture Mechanics*, vol. 220, Article ID 106599, 2019.

[39] U. Ayachit, *The Paraview Guide: A Parallel Visualization Application*, Kitware, Inc., New York, NY, USA, 2015.

[40] X. A. Liu, *DDA Based Complete and High Order Polynomial Displacement Approximation Method in Elastic Mechanics and its Cases Verification*, Changjiang River Scientific Research Institute, Hubei, China, 2013.

[41] J. R. Barber, *Elasticity*, Springer, New York, NY, USA, 2010.

[42] X. Q. Wang, Y. S. Liu, and Y. J. Feng, "Analytical solution of the elastic mechanics for cantilever beam under the even loads," *Journal of Gansu Sciences*, vol. 25, pp. 80–83, 2013.

[43] B. Yu, P. Hu, A. A. Saputra, and Y. Gu, "The scaled boundary finite element method based on the hybrid quadtree mesh for solving transient heat conduction problems," *Applied Mathematical Modelling*, vol. 89, pp. 541–571, 2021.

[44] S. Timoshenko and J. N. Goodier, *Theory of Elasticity*, pp. 279–291, New York McGraw—Hil1, New York, NY, USA, 1970.