Implementation of arbitrary polyhedral elements for automatic dynamic analyses of three-dimensional structures

Lei Zhou¹,², Jianbo Li¹,²* & Gao Lin¹,²

The transition from the geometry to the mesh can be rather difficult, manual and time-consuming, especially for the large scale complex structures. The procedure of mesh generation needs massive human intervention making the automatic engineering analyses of structures from CAD geometry models hardly possible. This paper focuses on implementing a polyhedron element with arbitrary convex topology based on the Scaled Boundary Finite Element Method (SBFEM) in ABAQUS on the strength of the interface of UEL (the subroutine to define a general user-defined element) for automatic dynamic analyses of three-dimensional structures. This implementation empowers ABAQUS to analyze any model with arbitrary polyhedron elements. When the geometry of a structure is obtained from CAD, the dynamic analyses can be launched seamlessly and automatically. Cases of a cantilever subjected to a dynamic harmonic excitation with the traditional hexahedron element and this polyhedron element are compared to verify the accuracy of the UEL. Taking a practical example of the Soil-Structure Interaction analysis of a Nuclear Power Plant, the applicability and performance of this implementation are tested. The results of the two examples confirm that this polyhedron element based on SBFEM can be more accurate using much less degrees of freedom and its implementation in ABAQUS is robust and compatible.

As the development of the Computer Aided Design (CAD) and Computer Aided Engineering (CAE) tools in the industry, the design, safety assessment, health monitoring of some important modern infrastructures tend to need more and more refined and sophisticated geometry models and mesh models. Nevertheless, there is a gap between the CAD and the CAE holding a barrier between the geometry model for design and the mesh model for engineering analysis. Because except for some rarely used meshless methods, it is a difficult, manual and time-consuming process to generate a fine mesh from a complex geometry. During the manual mesh generation, the balance of accuracy and efficiency should be examined, especially for dynamic analysis in the time domain. A basic requirement is to assign heterogeneous distributed grid densities for the different location of the geometry model to balance the calculation accuracy and efficiency, which are case-sensitive, completely experimental and can be rather tedious. The automatic mesh generation technique with polyhedron elements can break this barrier due to the flexible topology that makes them highly adaptable to the complex shape and smooth transition of mesh density, which is the foundation of the automatic engineering analysis.

Since the arising of numerical methods for the Partial Differential Equation (PDE), the mesh generation has become a primary procedure for these methods that need spatial discretization like the Finite Element Method (FEM) and the Finite Difference Method (FDM). From a geometrical point of view, a mesh is the approximate shape representation of the domains or the media (or can be collectively referred to as configurations) of the original physical problem, which consists of many non-overlapping continuous small domains called elements. These elements have certain topology defined by geometric feature points such as vertices called nodes, where the degrees of freedom are defined on. Except that a numerical analysis cannot be carried on without a mesh, the quality, size and type of the mesh have a great influence on the reliability of the results as well.

For the last four decades, lots of mesh generation technologies have been developed from two-dimension (2D) domain to three-dimension (3D) domain on the strength of the computer graphics. During the 1950s, mesh generation techniques based on coordinate transformations firstly came up, such as the conformal mapping¹²

¹State Key Laboratory of Coastal and Offshore Engineering, Dalian University of Technology, 2 Linggong Road, Dalian 116024, China. ²Institute of Earthquake Engineering, Faculty of Infrastructure Engineering, Dalian University of Technology, 2 Linggong Road, Dalian 116024, China. *email: jianboli@dlut.edu.cn
Two kinds of interpolation schemes were developed. The first one was based on natural neighbor interpolations proposed by Sibson\textsuperscript{11,12}. Traversoni\textsuperscript{13} and Sambridge et al.\textsuperscript{14} introduced Sibson interpolations modified by Watson\textsuperscript{15} to the Galerkin-type natural neighbor, which was extended to continuum by Sukumar et al.\textsuperscript{16,17}. Belokov et al.\textsuperscript{16} derived the Laplacian and non-Sibson interpolants\textsuperscript{18,19} and the polygonal finite element interpolants\textsuperscript{20–22}. The other one was built on generalized barycentric coordinates, which were introduced by Wachspress\textsuperscript{22}. This kind of interpolants was further developed by Warren\textsuperscript{23,24}, Meyer et al.\textsuperscript{25}, Floater et al.\textsuperscript{26,27}, Lipman\textsuperscript{28}, and Warren et al.\textsuperscript{29}. However, the interpolation scheme generally needs considerable calculation to get adequate accuracy.

Another approach to constructing polyhedral elements is to utilize SBFEM. In mid-1990s, SBFEM also known as the consistent infinitesimal finite-element-cell method was brought up by Wolf and Song as an extension to FEM to simulate the wave propagation in the unbounded domain\textsuperscript{30–33}. As a similarity-based semi-analytical numerical approach for PDE, SBFEM has many advantages over the traditional FEM and Boundary Element Method (BEM) exhibiting high accuracy and flexibility\textsuperscript{34,35}. SBFEM just needs boundary discretization reducing the problem spatial dimension by one with no need for a fundamental solution and the solution along the radial direction is in a precise closed-form. For the unbounded media, SBFEM can perfectly satisfy the boundary condition of the radiation damping. The unit-impulse response matrix in the time domain and the dynamic-stiffness matrix in the frequency domain of the unbounded media can be directly obtained, which is equivalent to a time and space coupling artificial non-reflecting boundary\textsuperscript{36}. For the bounded media, the SBFEM is derived in a wedge media. The wedge media can be assembled into an arbitrary convex polyhedron (3D) or polygon (2D) super element revolving around the similarity center. If this assembly is not occlusive and the crack tip is positioned in the similarity center, the singularity of the field function can be directly described by the analytical solution along the radial crack surface. This superiority of mesh adaptability can be combined with the automatic mesh generation technique based on polyhedral elements.

In this paper, arbitrary polyhedron elements based on SBFEM for three dimensional continuum is developed and implemented in ABAQUS facilitating automatic dynamic analyses of structures. The governing equation of the bounded 3D elastic continuum in a polyhedral domain is derived from the SBFEM in “SBFEM governing equation for the elastic continuum” section. The FEM-coupled characteristic matrix of a polyhedral element can be obtained from the solution of the governing equation. The topology feature and corresponding data structure of a polyhedron is analyzed in “Topology and data structure of an arbitrary polyhedron element” section. The user element subroutine of UEL is developed associated with the Hilber-Hughes-Taylor (HHT) implicit time integration scheme in “Implementation in ABAQUS’ section. In “Numerical application” section, there are two examples for the UEL. A cantilever subjected to a harmonic excitation with the traditional hexahedron element and the SBFEM-based polyhedron element are compared to confirm the accuracy of the UEL. Then taking a
practical example of the soil-structure dynamic interaction analysis of a nuclear power plant, the applicability and performance of this implementation of polyhedron elements is tested.

SBFEM governing equation for the elastic continuum
SBFEM is a numerical method for solving PDE. The PDE is transformed into ordinary differential equations after specific spatial discretization, which is different from FEM, BEM, etc. SBFEM needs boundary discretization only decreasing the spatial dimension by 1 and remains analytical in the radial direction without the need of a foundation solution. The derivation of the SBFEM governing equation of elastic continuum is based on a wedge area defined as W-element. The governing equations of W-elements that shared the same scaling center can be assembled by degrees of freedom of adjacent W-elements (see Fig. 1) leading to a piece-wise boundary of the discretized domain defined as an S-element. The choice of the scaling center location can be pretty casual. The boundary of the discretized domain must be visible from the scaling center is the only geometry requirement that needs to be satisfied. The bounded domain can be divided into multiple S-elements to meet the requirement of the scaling center or to improve the poor geometry of W-elements (see Fig. 2). Apparently, the geometry of the S-element can be rather flexible for any convex polyhedron with only triangle or quadrilateral facets.

The detailed derivation of SBFEM governing equation is well documented in many literatures. For the sake of integrity, only key equations are given below. The equations in this section are given in linear algebraic notation for the convenience of reading and programming. The brace symbol of \{·\} denotes a matrix and the bracket symbol of \[·\] denotes a column vector.
Partial differential equations of dynamic elastic mechanics. According to the theory of continuum mechanics, three basic equations are derived as

\[ [A][[\sigma]] + \{f\} = \{0\}, \]
\[ \{\varepsilon\} = [L]\{u\}, \]
\[ \{\sigma\} = [D]\{\varepsilon\}, \]

where \([A]\) denotes the spatial partial derivative operator, \([\sigma]\) denotes the stress vector, \([\varepsilon]\) denotes the strain vector, \([f]\) denotes the body forces that include the inertial force for the dynamic analysis. The detailed definitions of them are

\[ [A] = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y} \end{bmatrix}, \]

\[ \{\sigma(x,y,z)\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix}, \]

\[ \{f(x,y,z)\} = \begin{Bmatrix} f_x(x,y,z) \\ f_y(x,y,z) \\ f_z(x,y,z) \end{Bmatrix} = \{p(x,y,z)\} - \rho\{\ddot{u}(x,y,z)\}. \]

The equilibrium equations (Eq. (1)), the geometric equations (Eq. (2)) and the constitutive equations (Eq. (3)) can be combined into one set of partial differential equations of 3D dynamic elastic mechanics with the displacement vector filed as the unknown variable as

\[ [L]^T[D][L]\{u\} + \{p\} - \rho\{\ddot{u}\} = \{0\}, \]

\[ [L] = [A]^T = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y} \end{bmatrix}, \]

where \([u(x,y,z)] = [u_x(x,y,z) \ u_y(x,y,z) \ u_z(x,y,z)]^T\) denotes the displacement vector filed, \([D]\) denotes the material property matrix, \([p]\) denotes the body forces except the d'Alembert inertial force, \(\rho\) denotes the density, \([L]\) denotes the spatial partial derivative operator.

The linear elastic material property matrix \([D]\) is given by

\[ [D] = \begin{bmatrix} \lambda + 2G & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2G & \lambda & 0 & 0 \\ 0 & 0 & \lambda + 2G & 0 & 0 \\ 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & G \end{bmatrix}, \]

\[ \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad G = \frac{E}{2(1+\nu)}, \]

where \(\lambda\) denotes the Lame parameter, \(E\) denotes the Young's modulus, \(G\) denotes the shear modulus, \(\nu\) denotes the Poisson's ratio.

Equation (7) represents a set of second order linear nonhomogeneous partial differential equations with constant coefficients in Cartesian coordinate, namely Navier equations. One way to solve the Navier equations analytically is to transform them into wave equations of potential functions by Helmholtz's theorem. The other way is to solve the Navier equations numerically by spatial and time discretization.

Two kinds of boundary conditions and one initial condition should be introduced to get the definite solution of the fundamental equation Eq. (7). One boundary condition is the geometry boundary condition on the surface \(S_u\) defined as

\[ \{u\} = \{\Pi\} \text{ on } S_u, \]
where \( \{ \mathbf{u} \} \) is the constant displacement on the surface of \( S_u \).

The other one is the surface traction boundary condition on the surface \( S_\sigma \) defined as

\[
\{ n \} \{ \sigma \} = \{ \mathbf{T} \} \quad \text{on} \quad S_\sigma,
\]

\[
\{ \mathbf{T} \}(x,y,z) = \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix},
\]

\[
[n(x,y,z)] = \begin{bmatrix} n_x & 0 & 0 & n_y & 0 & n_z \\ 0 & n_y & 0 & n_z & 0 \\ 0 & 0 & n_z & 0 & n_x \end{bmatrix},
\]

where \( \{ \mathbf{T} \} \) is the constant surface traction force per area on \( S_\sigma \), \( \{ n \} \) is the orientation cosine matrix, \( n_x(x,y,z) \), \( n_y(x,y,z) \), and \( n_z(x,y,z) \) are the orientation cosine of the outer normal vector of the surface \( S_\sigma \).

The initial condition can be defined as

\[
\{ \mathbf{u} \}(x,y,z) \text{ when } t = 0,
\]

\[
\{ \mathbf{\dot{u}} \}(x,y,z) \text{ when } t = 0,
\]

\[
\{ \mathbf{\ddot{u}} \}(x,y,z) \text{ when } t = 0.
\]

Coordinate mapping based on scaled boundary transformation. Taking one W-element as an example (see Fig. 3), the boundary is discretized with one plane four nodes linear finite element with circumferential axis of \( \eta \) and \( \zeta \) (see Fig. 4). Other 2D finite elements such as the quadratic element with eight nodes can be used for this boundary discretization as well \cite{51}.

The origin of the global Cartesian coordinate system of the nodes needs to be translated to the scaling center by

\[
\{ x_e \} = \{ \bar{x}_e \} - \bar{z}_0^0,
\]

\[
\{ y_e \} = \{ \bar{y}_e \} - \bar{y}_e^0,
\]

\[
\{ z_e \} = \{ \bar{z}_e \} - \bar{z}_0^0,
\]

where \( \{ \bar{x}_e \} \), \( \{ \bar{y}_e \} \), and \( \{ \bar{z}_e \} \) denote the global coordinates of the nodes, \( \bar{z}_0^0 \), \( \bar{y}_e^0 \), and \( \bar{z}_0^0 \) denote the coordinates of the scaling center in global Cartesian coordinate system, \( \{ x_e \} \), \( \{ y_e \} \), and \( \{ z_e \} \) denote the translated global coordinates of the nodes.

The mapping function between the global coordinates and the local coordinates of the points on the boundary is given by

Figure 3. Spatial discretization of the W-element.
where \( \{N(\eta, \zeta)\}^T = [N_1 \ N_2 \ N_3 \ N_4] \) denotes the shape function of the plane finite element, \( \{x_e\}, \{y_e\}, \) and \( \{z_e\} \) denote the global coordinates of the nodes that have been translated by Eq. (15), \( \hat{x}(\eta, \zeta), \hat{y}(\eta, \zeta), \) and \( \hat{z}(\eta, \zeta) \) denote the global coordinates of arbitrary point on the boundary determined by the local coordinates \( (\eta, \zeta) \).

The new local scale boundary coordinate is built for this W-element with the origin at the scaling center. Points on the plane finite element is scaled along the radical direction from the scaling center to the boundary, which is defined as the \( \xi \) axis of the local scale boundary coordinate. The mapping function between the global coordinates and the local coordinates of the points in the W-element is given by

\[
x(\xi, \eta, \zeta) = \xi \hat{x} = \xi \{N(\eta, \zeta)\}^T \{x_e\},
\]
\[
y(\xi, \eta, \zeta) = \xi \hat{y} = \xi \{N(\eta, \zeta)\}^T \{y_e\},
\]
\[
z(\xi, \eta, \zeta) = \xi \hat{z} = \xi \{N(\eta, \zeta)\}^T \{z_e\},
\]

(17)

where \( x(\xi, \eta, \zeta), y(\xi, \eta, \zeta), \) and \( z(\xi, \eta, \zeta) \) denote the local coordinates of the points in the W-element.

Equation (17) provides a mapping relation between the local and global coordinates. Based on the derivative method for compound function, the corresponding relationship of the derivative operators between two coordinates is given by

\[
\begin{bmatrix}
\frac{\partial}{\partial \xi} \\
\frac{\partial}{\partial \eta} \\
\frac{\partial}{\partial \zeta}
\end{bmatrix} = [J]^{-1} \begin{bmatrix}
\frac{\partial}{\partial x} \\
\frac{\partial}{\partial y} \\
\frac{\partial}{\partial z}
\end{bmatrix},
\]

(18)

where \([J]\) is the Jacobian matrix expressed as

\[
[J] = \begin{bmatrix}
\frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\
\frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\
\frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta}
\end{bmatrix}.
\]

(19)

The variable \( \xi \) can be isolated from Eq. (19) as

\[
[J(\xi, \eta, \zeta)] = \begin{bmatrix}
1 & 0 & 0 \\
0 & \xi & 0 \\
0 & 0 & \xi
\end{bmatrix} \begin{bmatrix}
x \\
y \\
z
\end{bmatrix} = \begin{bmatrix}
\hat{x} \\
\hat{y} \\
\hat{z}
\end{bmatrix},
\]

(20)

with the Jacobian matrix on the boundary \([\tilde{J}]\) defined as

\[
[\tilde{J}(\eta, \zeta)] = \begin{bmatrix}
\hat{\xi} \\
\hat{\eta} \\
\hat{\zeta}
\end{bmatrix} = \begin{bmatrix}
\hat{x}_\eta \\
\hat{y}_\eta \\
\hat{z}_\eta
\end{bmatrix},
\]

(21)

where \( (\cdot)_{,\eta} \) denotes differentiating with respect to \( \eta \), \( (\cdot)_{,\zeta} \) denotes differentiating with respect to \( \zeta \).

Based on the space analytical geometric theory, the geometry properties in the local coordinate system can be derived. The infinitesimal line vectors in the axis direction are given by

\[
\overrightarrow{d\xi} = d\xi \hat{x}, \quad \overrightarrow{d\eta} = \xi d\eta \hat{y}, \quad \overrightarrow{d\zeta} = \xi d\zeta \hat{z},
\]

(22)

where \( (\cdot) \) denotes a vector.

The infinitesimal volume is given by
\[ dV = \overrightarrow{d\xi} \cdot \overrightarrow{d\eta} \times \overrightarrow{d\zeta} = |\mathbf{J}| \, \overrightarrow{d\eta} \, d\zeta, \]  

where \(|\cdot|\) denotes the determinant of the matrix.

The normal vectors of the coordinate surfaces are given by

\[ \overrightarrow{g^2} = (\hat{\gamma}_n \hat{\gamma}_z - \hat{\gamma}_c \hat{\gamma}_n, \hat{\gamma}_n \hat{\chi}_z - \hat{\chi}_c \hat{\gamma}_n, \hat{\chi}_n \hat{\gamma}_z - \hat{\gamma}_c \hat{\chi}_n), \]
\[ \overrightarrow{g^3} = (\hat{\gamma}_c \hat{\chi}_n - \hat{\chi}_c \hat{\gamma}_n, \hat{\gamma}_c \hat{\chi}_z - \hat{\chi}_c \hat{\gamma}_z, \hat{\gamma}_c \hat{\gamma}_z - \hat{\chi}_c \hat{\gamma}_z), \]
\[ \overrightarrow{g^4} = (\hat{\gamma}_n \hat{\gamma}_z - \hat{\gamma}_c \hat{\gamma}_n, \hat{\gamma}_n \hat{\chi}_z - \hat{\chi}_c \hat{\gamma}_n, \hat{\chi}_n \hat{\gamma}_z - \hat{\gamma}_c \hat{\chi}_n), \]  

where \(\overrightarrow{g^2}, \overrightarrow{g^3}, \) and \(\overrightarrow{g^4}\) denote the normal vector of the coordinate surface of \(S^2, S^3\) and \(S^4\).

The length of the normal vectors \(g^2, g^3,\) and \(g^4\) and the unit normal vectors \(n^2, n^3,\) and \(n^4\) are defined as

\[ g^2 = \|\overrightarrow{g^2}\|, \quad g^3 = \|\overrightarrow{g^3}\|, \quad g^4 = \|\overrightarrow{g^4}\|, \]  

where \(|\cdot|\) denotes the length of the vector.

The infinitesimal area of the coordinate surfaces is given by

\[ ds^2 = \overrightarrow{d\eta} \times \overrightarrow{d\zeta} = g^2 d\eta d\zeta, \]
\[ ds^3 = \overrightarrow{d\zeta} \times \overrightarrow{d\xi} = g^3 d\zeta d\xi, \]
\[ ds^4 = \overrightarrow{d\xi} \times \overrightarrow{d\eta} = g^4 d\xi d\eta. \]

\([\mathbf{J}]^{-1}\) can be expressed as

\[ [\mathbf{J}]^{-1} = \frac{1}{|\mathbf{J}|} \begin{bmatrix} g^2 n^2_x & g^3 n^2_y & g^4 n^2_z \\ g^2 n^3_x & g^3 n^3_y & g^4 n^3_z \\ g^2 n^4_x & g^3 n^4_y & g^4 n^4_z \end{bmatrix}. \]

Substituting Eq. (28) into Eq. (18), the derivative operators in the local coordinates is given by

\[ \frac{\partial}{\partial \xi} = \frac{g^2}{|\mathbf{J}|} n^2_x \frac{\partial}{\partial \xi} + \frac{1}{g^2} \left( \frac{g^3}{|\mathbf{J}|} n^3_y \frac{\partial}{\partial \eta} + \frac{g^4}{|\mathbf{J}|} n^4_z \frac{\partial}{\partial \zeta} \right), \]
\[ \frac{\partial}{\partial \eta} = \frac{g^2}{|\mathbf{J}|} n^2_y \frac{\partial}{\partial \xi} + \frac{1}{g^2} \left( \frac{g^3}{|\mathbf{J}|} n^3_x \frac{\partial}{\partial \eta} + \frac{g^4}{|\mathbf{J}|} n^4_z \frac{\partial}{\partial \zeta} \right), \]
\[ \frac{\partial}{\partial \zeta} = \frac{g^2}{|\mathbf{J}|} n^2_z \frac{\partial}{\partial \xi} + \frac{1}{g^2} \left( \frac{g^3}{|\mathbf{J}|} n^3_x \frac{\partial}{\partial \eta} + \frac{g^4}{|\mathbf{J}|} n^4_y \frac{\partial}{\partial \zeta} \right). \]

Substituting Eq. (29) into Eq. (7), the spatial partial derivative operator \([L]\) in local coordinate system is given by

\[ [L] = [b^1] \frac{\partial}{\partial \xi} + \frac{1}{g^2} \left( [b^2] \frac{\partial}{\partial \eta} + [b^3] \frac{\partial}{\partial \zeta} \right), \]

where \([b^1], [b^2],\) and \([b^3]\) are defined as

\[ [b^1] = \begin{bmatrix} n^2_x & 0 & 0 \\ 0 & n^2_y & 0 \\ 0 & 0 & n^2_z \end{bmatrix}, \]
\[
\mathbf{b}^\eta = \frac{g^\eta}{[\mathbf{T}]} \begin{bmatrix}
n_n^\eta & 0 & 0 \\
0 & n_n^\eta & 0 \\
0 & 0 & n_n^\eta \\
n_n^\eta & n_n^\eta & n_n^\eta \\
0 & n_n^\eta & n_n^\eta \\
n_n^\eta & 0 & n_n^\eta
\end{bmatrix},
\]
\[
\mathbf{b}^\xi = \frac{g^\xi}{[\mathbf{T}]} \begin{bmatrix}
n_n^\xi & 0 & 0 \\
0 & n_n^\xi & 0 \\
0 & 0 & n_n^\xi \\
n_n^\xi & n_n^\xi & n_n^\xi \\
0 & n_n^\xi & n_n^\xi \\
n_n^\xi & 0 & n_n^\xi
\end{bmatrix},
\]
and they satisfy
\[
\left( \left| [\mathbf{I}] \left[ [\mathbf{b}^\xi]\right] \right| \right)_\xi + \left( \left| [\mathbf{I}] \left[ [\mathbf{b}^\eta]\right] \right| \right)_\eta = -2 \left| [\mathbf{I}] \left[ [\mathbf{b}]\right] \right|.
\]
Substituting Eq. (30) into Eqs. (2) and (3), the stress \{\sigma\} in local coordinate system is given by
\[
\{\sigma(\xi, \eta, \zeta)\} = \left[ D \right] \left( [\mathbf{b}^\eta] \left[ \mathbf{u}(\xi, \eta, \zeta) \right] \right)_\xi + \frac{1}{\xi} \left( [\mathbf{b}^\xi] \left[ \mathbf{u}(\xi, \eta, \zeta) \right] \right)_\eta + \left[ b^\zeta \right] \left[ \mathbf{u}(\xi, \eta, \zeta) \right]_\zeta.
\]
Considering the surface traction boundary condition in Eq. (10) to Eq. (12), the surface traction on the coordinate surface is defined as
\[
\{ \mathbf{t}^\xi \} = \left\{ t_x^\xi, t_y^\xi, t_z^\xi \right\}^T = \left[ n_x^\xi 0 0 n_x^\xi 0 n_x^\xi \\
0 n_y^\xi 0 n_y^\xi n_y^\xi 0 \\
0 0 n_z^\xi 0 n_z^\xi n_z^\xi \right] \begin{bmatrix}
\sigma_x \\
\sigma_y \\
\sigma_z \\
\tau_{xy} \\
\tau_{yz} \\
\tau_{xz}
\end{bmatrix},
\]
\[
\{ \mathbf{t}^\eta \} = \left\{ t_x^\eta, t_y^\eta, t_z^\eta \right\}^T = \left[ n_x^\eta 0 0 n_x^\eta 0 n_x^\eta \\
0 n_y^\eta 0 n_y^\eta n_y^\eta 0 \\
0 0 n_z^\eta 0 n_z^\eta n_z^\eta \right] \begin{bmatrix}
\sigma_x \\
\sigma_y \\
\sigma_z \\
\tau_{xy} \\
\tau_{yz} \\
\tau_{xz}
\end{bmatrix},
\]
\[
\{ \mathbf{t}^\zeta \} = \left\{ t_x^\zeta, t_y^\zeta, t_z^\zeta \right\}^T = \left[ n_x^\zeta 0 0 n_x^\zeta 0 n_x^\zeta \\
0 n_y^\zeta 0 n_y^\zeta n_y^\zeta 0 \\
0 0 n_z^\zeta 0 n_z^\zeta n_z^\zeta \right] \begin{bmatrix}
\sigma_x \\
\sigma_y \\
\sigma_z \\
\tau_{xy} \\
\tau_{yz} \\
\tau_{xz}
\end{bmatrix},
\]
where \{ \mathbf{t}^\xi \}, \{ \mathbf{t}^\eta \}, and \{ \mathbf{t}^\zeta \} denote the the surface traction on the coordinate surface of \xi, \eta, and \zeta respectively.

Compared with Eq. (31) to Eqs. (33), (36) to Eq. (38) can be reformulated as
\[
\{ \mathbf{t}^\xi \} = \frac{[\mathbf{J}]}{[\mathbf{g}^\xi]} [\mathbf{b}^\xi]^T \{ \sigma \},
\]
\[
\{ \mathbf{t}^\eta \} = \frac{[\mathbf{J}]}{[\mathbf{g}^\eta]} [\mathbf{b}^\eta]^T \{ \sigma \},
\]
\[
\{ \mathbf{t}^\zeta \} = \frac{[\mathbf{J}]}{[\mathbf{g}^\zeta]} [\mathbf{b}^\zeta]^T \{ \sigma \}.
\]

**Spatial discretization of the displacement field on the boundary.** Taking the same W-element in "Coordinate mapping based on scaled boundary transformation" section as example (see Fig. 5), the boundary is discretized with a finite element with circumferential axis of \eta and \zeta (see Fig. 4).
The displacement field on the boundary \( \{u(\eta, \zeta)\} \) can be expressed as
\[
\{u(\eta, \zeta)\} = \{N(\eta, \zeta)\} \{u^e\},
\]
where \( \{N(\eta, \zeta)\} = \left[ N_1(\eta, \zeta) I \ N_2(\eta, \zeta) I \ N_3(\eta, \zeta) I \ N_4(\eta, \zeta) I \right] \) denotes the shape function of the finite element, \( \{u^e\} \) denotes the displacement values of the nodes, \( [I] \) denotes a \( 3 \times 3 \) identity matrix.

The interpolation shown in Eq. (42) is applied to any coordinate surface of \( S^b \). The displacement field \( \{u(\xi, \eta, \zeta)\} \) in the W-element can be expressed as
\[
\{u(\xi, \eta, \zeta)\} = \{N(\eta, \zeta)\} \{u^e(\xi)\},
\]
where \( \{u^e(\xi)\} \) denotes the dislocation values along the rays starting from the scaling center to the nodes on the boundary.

Substituting Eq. (43) into Eq. (35), the discretization of the stress \( \{\sigma\} \) is given by
\[
\{\sigma\} = \{D\} \left( [B^1] \{u^e(\xi)\}_\eta + \frac{1}{\xi} [B^2] \{u^e(\xi)\}_\zeta \right),
\]
where \( [B^1(\eta, \zeta)] \) and \( [B^2(\eta, \zeta)] \) are defined as
\[
[B^1(\eta, \zeta)] = [b^1] [N],
\]
\[
[B^2(\eta, \zeta)] = [b^2] [N]_\eta + [b^3] [N]_\zeta.
\]

**Equivalent integral equations and Galerkin weighted residual method.** The SBFEM governing equation can be derived by the application of the weighted residual method or variation principle. In this section, the procedure of Galerkin weighted residual method is elaborated.

The equivalent integral form of Eq. (7) is given by
\[
\int_V [w_j]^T [b^1]^T \{\sigma\}_\eta dV + \int_V [w_j]^T \frac{1}{\xi} \left( [b^2]^T \{\sigma\}_\eta + [b^3]^T \{\sigma\}_\zeta \right) dV
\]
\[
- \int_V [w_j]^T \rho \{\ddot{u}\} dV + \int_V [w_j]^T \{p\} dV = \{0\},
\]
where \([w_j]^T\) denotes the arbitrary weight function defined in Eq. (48), \([\cdot]^T\) denotes the transpose of the matrix. \([w_j]^T\) is defined as...
Substituting Eq. (39) to Eq. (41) into Eq. (50) results in

\[
\int_{-1}^{1} \int_{-1}^{1} \left[ \int_{-1}^{1} [w_j]^T [b^1]^T [\sigma]_\xi \left[ \hat{f}(\eta, \zeta) \right] d\eta d\zeta \right] d\xi d\eta d\zeta
d\eta d\zeta d\xi
\]

\[
+ \int_{-1}^{1} \int_{-1}^{1} \left[ \int_{-1}^{1} [w_j]^T \frac{1}{2} \left( [b^2]^T [\sigma]_\eta + [b^3]^T [\sigma]_\zeta \right) \left[ \hat{f}(\eta, \zeta) \right] d\eta d\zeta \right] d\xi d\eta d\zeta d\xi
\]

\[
+ \int_{-1}^{1} \int_{-1}^{1} \left[ \int_{-1}^{1} [w_j]^T \{ p \} \left[ \hat{f}(\eta, \zeta) \right] d\eta d\zeta \right] d\xi d\eta d\zeta d\xi
\]

\[
- \int_{-1}^{1} \int_{-1}^{1} \left[ \int_{-1}^{1} [w_j]^T \rho(\hat{\nu}) \left[ \hat{f}(\eta, \zeta) \right] d\eta d\zeta \right] d\xi d\eta d\zeta d\xi = [0].
\]

Dealing the second term labeled II in Eq. (49) with Green's theorem results in

\[
II = \int_{-1}^{1} \int_{-1}^{1} \left[ \int_{-1}^{1} [w_j]^T \left( [b^2]^T [\sigma]_\eta + [b^3]^T [\sigma]_\zeta \right) \left[ \hat{f}(\eta, \zeta) \right] d\eta d\zeta \right] d\xi d\eta d\zeta d\xi
\]

\[
= \int_{-1}^{1} \int_{-1}^{1} \left( \int_{\partial S} [w_j]^T \left( \frac{1}{2} \{ p \} \right) \left[ \hat{f}(\eta, \zeta) \right] d\eta + \int_{\partial S} [w_j]^T \left( \frac{1}{2} \{ p \} \right) \left[ \hat{f}(\eta, \zeta) \right] d\eta \right) d\xi d\eta d\zeta d\xi
\]

where $\partial S$ denotes the boundary of surface $S$. Substituting Eq. (39) to Eq. (41) into Eq. (50) results in

\[
II = \int_{-1}^{1} \left( \int_{\partial S} \left( \frac{1}{2} [w_j]^T \left( \frac{1}{2} \{ p \} \right) \left[ \hat{f}(\eta, \zeta) \right] d\eta + \int_{\partial S} [w_j]^T \left( \frac{1}{2} \{ p \} \right) \left[ \hat{f}(\eta, \zeta) \right] d\eta \right) d\xi d\eta d\zeta d\xi
\]

where the surface traction boundary condition $\{ t^p \}$ and $\{ t^s \}$ is introduced naturally. Substituting Eq. (34) into Eq. (51) results in

\[
\begin{bmatrix}
[w_j(\xi, \eta, \zeta)] = w_j[I] = w_j \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}.
\]

where $j = 1, 2, \ldots, m$ denotes the number of weight functions and $m \rightarrow \infty$. Substituting Eq. (23) into Eq. (47) results in
where Eq. (43) as

Consider the Galerkin weighted residual method, the interpolation of

By applying the weighted residual method, the arbitrary weight function

Substituting Eq. (52) into Eq. (47), the equivalent integral weak form of Eq. (47) is given by

Substituting Eq. (52) into Eq. (47), the equivalent integral weak form of Eq. (47) is given by

Substituting Eq. (52) into Eq. (47), the equivalent integral weak form of Eq. (47) is given by

where the surface traction boundary condition \( \{ t^0 \} \) and \( \{ t^\xi \} \) is introduced naturally, \( \Gamma^s \) denotes the boudnary of surafce \( S^s \).

By applying the weighted residual method, the arbitrary weight function \( [w] \) in Eq. (53) is simplified to a particular weight function \([w]\), the approximate form of Eq. (53) is given by

where \([w]^T\) denotes the weight functions in the W-element, \(\{ \cdot \}^T\) denotes a row vector.

Considering the Galerkin weighted residual method, the interpolation of \([w]^T\) is the same as \( [u(\xi, \eta, \zeta)] \) in Eq. (43) as

\[
[w(\xi, \eta, \zeta)] = [N(\eta, \zeta)] \{ w^\circ(\xi) \},
\]
Substituting Eq. (43), Eqs. (29) and (55) into Eq. (54), the SBFEM governing equation in the time domain for 3D elastic continuum media is given as

\[ [E^0] \xi^2 \{ u \} = \left( 2[E^0] - [E^1] + [E^1]^T \xi \{ u \} + \left( [E^1] - [E^2] \right) \right) \{ u \} - [M^0] \xi^2 \{ \dot{u} \} \right] + \xi \{ T \} + \xi^2 \{ P \} = [0], \]  

(56)

where the constant coefficient matrix is given by

\[ [E^0] = \int_{-1}^{1} \int_{-1}^{1} \left[ B \right]^T [D] \left[ B \right] \{ \xi \} \, d\eta d\xi, \]

\[ [E^1] = \int_{-1}^{1} \int_{-1}^{1} \left[ B \right]^T [D] \left[ B \right] \{ \xi \} \, d\eta d\xi, \]

\[ [E^2] = \int_{-1}^{1} \int_{-1}^{1} \left[ B \right]^T [D] \left[ B \right] \{ \xi \} \, d\eta d\xi, \]

\[ [M^0] = \int_{-1}^{1} \int_{-1}^{1} \left[ N \right]^T [\eta] \{ \eta \} \, d\eta d\xi. \]

The governing equation of W-elements sharing the same scaling center can be assembled by the degree of freedom contributed by adjacent W-elements, which results in the governing equation of the S-element like Eq. (56) but with the assembled \{ u \}, \{ E^0 \}, \{ E^1 \}, \{ E^2 \}, and \{ M^0 \}.

**Solution of the SBFEM governing equation.** The SBFEM governing equation (Eq. (56)) is a system of second order Ordinary Differential Equations (ODEs) called the Euler-Cauchy equations. The solution of this system is given in a closed form. The first step is to introduce a dual variable to convert the system of second order ODEs to a system of one order with variables doubled in size. Based on the principle of virtual work, the internal node force \{ q(\xi) \} along the ray starting from the scaling center to the nodes on the boundary resulting from the surface traction \{ t^s \} on the coordinate surface of \( S^0 \) is given by

\[ [w(\xi)]^T \{ q(\xi) \} = \int_{S^0} \{ w(\xi, \eta, \zeta) \}^T \{ t^s (\xi, \eta, \zeta) \} \, dS^0, \]  

(58)

\[ \{ q(\xi) \} = [E^0] \xi^2 \{ u(\xi) \}_x + [E^1] \xi \{ u(\xi) \}. \]  

(59)

If the body forces vanish, Eq. (56) can be reformulated as

\[ \xi \{ X(\xi) \} = [Z] \{ X(\xi) \}. \]  

(60)

where

\[ \{ X(\xi) \} = \left\{ \begin{array}{c} \xi^{0.5} \{ u(\xi) \} \\ \xi^{-0.5} \{ q(\xi) \} \end{array} \right\}, \]

(61)

and

\[ [Z] = \left[ \begin{array}{cc} 0.5[I] & \left[ [E^0] \right]^{-1} [E^1]^T \\ [E^2] & \left[ [E^0] \right]^{-1} [E^1]^T [E^1] [E^0]^{-1} - 0.5[I] \end{array} \right]. \]

(62)

The eigenvalues and eigenvectors of the matrix \([Z]\) can be obtained by

\[ [Z] \{ \Phi \} = [\Phi] [\lambda], \]  

(63)

where \([\Phi] = \{ \phi_1 \} \{ \phi_2 \} \ldots \{ \phi_{2n} \} \) denotes a matrix consisted of the eigenvectors \{ \phi_1 \}, \{ \phi_2 \}, \ldots, \{ \phi_{2n} \}, \lambda_1, \lambda_2, \ldots, \lambda_{2n} denotes the eigenvalues, diag(\cdot) denotes a diagonal matrix. It is worth to notice that the eigenvalues in [\lambda] should be in descending order, so the eigenvectors in [\Phi]. The eigenvectors \{ \phi_1 \}, \{ \phi_2 \}, \ldots, \{ \phi_{2n} \} and the eigenvalues \lambda_1, \lambda_2, \ldots, \lambda_{2n} can be partitioned into two blocks as

\[ [\Phi] = \left[ \begin{array}{cc} [\Phi]^p \\ [\Phi]^c \end{array} \right], \]

(64)
\[
\begin{bmatrix}
\lambda
\end{bmatrix}
= \begin{bmatrix}
\lambda^u_m & 0^u_n \\
0^u_p & \lambda^u_v
\end{bmatrix},
\]
where the subscript \( u \) denotes the block related to \( \{u(\xi)\} \), the subscript \( q \) denotes the block related to \( \{q(\xi)\} \).

The solution of Eq. (60) considering boundary conditions is given by
\[
\{u(\xi)\} = \xi^{-0.5} [\Phi^u_b] [\Phi^u_b]^{-1} \{c\},
\]
\[
\{q(\xi)\} = \xi^{0.5} [\Phi^u_b] [\Phi^u_b]^{-1} \{c\},
\]
where \( \{c\} = [c_1 \ c_2 \ \cdots \ c_n]^T \) denotes the arbitrary integration constant, \( \xi^{i}\) denotes a diagonal matrix expressed as
\[
\xi^{i} = \begin{bmatrix}
\xi^{i_1} & 0 & \cdots & 0 \\
0 & \xi^{i_2} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & \cdots & \xi^{i_n}
\end{bmatrix},
\]

For the SBFEM polyhedron element, the coefficient matrix representing the stiffness and mass matrix of the S-element need to be derived. Substituting Eq. (67) into Eq. (66) to eliminate \( \{c\} \), the stiffness matrix \([K]\) is given by
\[
[K] = [\Phi^u_b] [\Phi^u_b]^{-1},
\]
where \([\cdot]^{-1}\) denotes the inverse of the matrix.

Based on the virtual work statement, the mass matrix \([M]\) of the S-element can be expressed as
\[
[M] = [\Phi^u_b]^{-T} [m] [\Phi^u_b]^{-1},
\]
where \([\cdot]^{-T}\) denotes the inverse of the transpose of the matrix, the value \(m_{ij}\) in the row \(i\) and column \(j\) of \([m]\) is defined as
\[
m_{ij} = m^0_{ij} = \frac{m^0_{ij}}{\lambda^p_i + \lambda^p_j + 2},
\]

where the \(m^0_{ij}\) is the value in the row \(i\) and column \(j\) of \([m^0]\) defined as
\[
m^0_{ij} = [m^0] = [\Phi^u_b]^{-T} [M^0] [\Phi^u_b]^{-1}.
\]

**Topology and data structure of an arbitrary polyhedron element.**
The polyhedron geometry is a 3D shape with straight edges, sharp corners and vertices, and any number greater than 2 of plane polygonal faces with any number greater than 2 of straight edges and vertices. The tetrahedron, pentahedron, hexahedron and heptahedron that widely used in FEM are just the special cases of the convex polyhedron. Vastly different from the FEM element geometry, the first step to construct a polyhedron element is to describe the uncertainty of the topology of varied kinds of polyhedron elements in a model with a uniform data structure that can be recorded in the intermediate file for compatible purpose.

**Essential information to describe the topology of an arbitrary polyhedron.** For example, Fig. 6 shows a polyhedron element that may be encountered in an octree mesh, which has nine facets, twenty edges and thirteen vertices. The coordinates of the vertices are the basic information as shown in Table 1. In addition, the connectivity of the vertices of each facet needs to be defined as shown in Table 2. These two sets of information are sufficient and necessary for the description of the topology of any polyhedron. It is worth to notice that the order of the vertices and facets of a polyhedron is insignificant and when there are multiple polyhedra, there is no need to record the correspondence between the local and global vertex ID like FEM.

**SBFEM discretization of the polyhedron.** The geometry of the S-element can be adapted for any convex polyhedron with only triangle or quadrilateral facets which is a special case of the polyhedron defined in “Essential information to describe the topology of an arbitrary polyhedron” section. Nevertheless, the polygonal facet of a polyhedron can always be converted into multiple triangle or quadrilateral facets by adding a vertex on this facet (see Fig. 6). After this treatment of the polyhedron facets, a point inside the polyhedron that can see all the points on the facets is chosen as the scaling center and the vertices are defined as nodes of this polyhedron domain. The scaling center and the nodes of each facet constitute a W-element. All these W-elements are assembled into an S-element representing this polyhedron domain. It is obvious that the number of W-elements is equal to the number of facets of the polyhedron and the number of the nodes of the S-element is equal to the number of vertices of the polyhedron. The data structure to describe any 3D SBFEM polyhedron element is summarized in Table 3.
Figure 6. Topology of a polyhedron element and the treatment of the polygon facet.

Table 1. Vertex information of the polyhedron element.

| Vertex ID | X coordinates | Y coordinates | Z coordinates |
|-----------|---------------|---------------|---------------|
| 1         | 0.0           | 0.0           | 0.0           |
| 2         | 0.0           | 10.0          | 0.0           |
| 3         | 10.0          | 10.0          | 0.0           |
| 4         | 10.0          | 5.0           | 0.0           |
| 5         | 10.0          | 0.0           | 0.0           |
| 6         | 10.0          | 10.0          | 5.0           |
| 7         | 10.0          | 5.0           | 5.0           |
| 8         | 10.0          | 0.0           | 5.0           |
| 9         | 0.0           | 0.0           | 10.0          |
| 10        | 0.0           | 10.0          | 10.0          |
| 11        | 10.0          | 10.0          | 10.0          |
| 12        | 10.0          | 5.0           | 0.0           |
| 13        | 10.0          | 0.0           | 10.0          |

Table 2. Facet information of the polyhedron.

| Facet ID | Vertex connectivity |
|----------|---------------------|
| 1        | 1, 2, 3, 4, 5       |
| 2        | 9, 13, 12, 11, 10   |
| 3        | 2, 10, 11, 6, 3     |
| 4        | 1, 9, 10, 2        |
| 5        | 1, 5, 8, 13, 9     |
| 6        | 7, 6, 11, 12       |
| 7        | 8, 7, 12, 13       |
| 8        | 4, 3, 6, 7         |
| 9        | 5, 4, 7, 8         |
Generation of the SBFEM polyhedron mesh. There are roughly three kinds of tools to obtain a polyhedron mesh: the self-developed program, the open-source or commercial mesh generator. Liu et al. brought up an automatic octree mesh generation algorithm from an STL geometry file in MATLAB for three-dimensional domain. Besides, at present OPENFOAM, FLUENT and STAR-CCM+ provide robust polyhedron mesh generators, the latter two of which are the most convenient. The general procedure to generate a polyhedron mesh from geometry with an out-of-box tool is illustrated in Fig. 7. Because the common FEM software for solid media does not support the polyhedron element, the preprocessor built in them cannot generate polyhedron meshes. Furthermore, the raw polyhedron mesh needs some extra processing steps such as classification, grouping, adding vertices before applied. The flow of geometry and mesh information between various software through files in compatible formats is inevitable. The Initial Graphics Exchange Specification (IGES) for the geometry information and Visualization Toolkit (VTK) for the mesh information are recommended. It should be noted that no matter which polyhedron mesh generator above is chosen or developed, it is required to store the polyhedron mesh in the intermediate file that will be imported into the UEL after the polyhedron mesh is obtained.

Implementation in ABAQUS

The SBFEM polyhedron element is introduced in ABAQUS/Standard as a user element. The user element is implemented on the element iteration level of the solver in the form of a FORTRAN subroutine named UEL. During each increment of the computation, ABAQUS will invoke the UEL when calculating the coefficient matrix of each user element on and on until the end of the analysis. Assembly of the global coefficient matrix, nonlinear iteration, result output, imposition of contact, constrain, boundary and load conditions are performed by ABAQUS automatically.

Basics of UEL in ABAQUS. UEL is essentially a function to compute specific variables with given variables. The subroutine name, declaration of the input and output variables are fixed (see “Appendix 1”). During the calculation of each element, the UEL will be called importing information of analysis type, time increment, element type, right-hand-side vectors, etc. to export the mass matrix, stiffness matrix, equivalent nodes force vector, solution-dependent state variables, etc.

There are some conventions of user subroutines in ABAQUS that should be emphasized. All user subroutines and the subroutines created by users must be stored in a single source code file or its compiled object file. By default, the user subroutine can call another subroutine that is built in FORTRAN compiler, created by users or provided by ABAQUS named utility routines. This restriction suggests that some complicated calculations in a user subroutine have to be implemented by source codes provided by users. And if there were multiple user elements, these user elements should be implemented by the same subroutine of UEL with conditional branch statements on element types. Although user subroutines cannot call each other, the data can be transferred through the user-defined COMMON block variables between them.

Associated INP and intermediate file. A convenient characteristic of ABAQUS is that the entire information of a FEM model is stored in an INP file, which is human-readable. This INP file is the only input required by the solver of ABAQUS. The user elements must be declared in the INP file before the calculation (see “Appendix 2”). The declaration of user elements provide the information of the element type, element set name, number of nodes, node connectivity, node degrees of freedom and element properties such as material parameters, etc. According to the format of the user element declaration in an INP file, the user elements need to be classified

| Element ID | $k$ |
|------------|-----|
| Coordinate of scaling center | $x_0, y_0, z_0$ |
| Number of facets | $m$ |
| Number of nodes | $n$ |

| Local node ID | Global node ID | Node coordinates |
|---------------|---------------|------------------|
| 1 | $N_1$ | $x_1, y_1, z_1$ |
| 2 | $N_2$ | $x_2, y_2, z_2$ |
| ... | ... | ... |
| $n$ | $N_n$ | $x_n, y_n, z_n$ |

| Facets | Number of nodes | Node connectivity$^a$ |
|--------|-----------------|----------------------|
| 1 | $J_1$ | $N_{11}, N_{12}, \ldots, N_{1J_1}$ |
| 2 | $J_2$ | $N_{11}, N_{12}, \ldots, N_{1J_2}$ |
| ... | ... | ... |
| $m$ | $J_m$ | $N_{m1}, N_{m2}, \ldots, N_{mJ_m}$ |

| Density | $\rho_k$ |
| Young’s modulus | $E_k$ |
| Poisson’s ratio | $v_k$ |

Table 3. Data structure of the SBFEM polyhedron element. $^a$The node sequence should be in a certain order pointing out of the region according to the right-hand rule.
into different types based on at least the number of nodes and the material properties and the node connectivity of polygonal facets of a polyhedron cannot be represented.

Due to the limitation of the INP file mentioned before, an intermediate file storing the topology information and material properties of arbitrary polyhedron elements is necessary. This file should be imported once the calculation is started and when iterating through the elements, the corresponding topology information of current user element should be introduced into UEL. With this intermediate file, any polyhedron mesh generator mentioned in "Generation of the SBFEM polyhedron mesh" section can be used and any mesh with arbitrary convex polyhedron shape can be calculated.

Algorithm of UEL for SBFEM polyhedron element. Although UEL has provided a specific interface, it is necessary to understand the calculation process of ABAQUS/Standard to get a clear picture of what information is passed in UEL and what information UEL should return under different circumstances. As shown in Fig. 8, calculation process of ABAQUS/Standard has a hierarchy of three levels. Level 1 denotes the calculation procedure of steps in an analysis. Level 2 denotes the calculation procedure of increments of a step. Level 3 denotes the actual calculation of the polyhedron element matrix and vector in an increment by calling UEL for each element (see the green block in Fig. 8). The utility routine of UEXTERNALDB plays an important role in Level 1 and 2 (see the red blocks in Fig. 8), providing adequate interventions into the different phases of the calculation process, which give us more freedom to realize more complex functions. To coincide with UEL, the intermediate file storing the topology and material information of user polyhedron elements is imported by subroutine UEXTERNALDB at the beginning of the analysis, which is then transferred into UEL by COMMON block variables when the calculation of the polyhedron element starts.

The behavior of UEL is controlled by the basic branching structure (see Fig. 9) judged by the variable of LFLAGS. Considering only the general static and direct-integration dynamic analysis, there are five different cases (see colored blocks in Fig. 9) requiring returned values respectively according to the Newton–Raphson Method (NRM) and HHT direct time integration [55].

The polyhedron element computation algorithm of UEL based on SBFEM is summarized in "Appendix 3". The assembly of the coefficient matrix of W-elements in STEP 2.3 and stiffness and mass matrix calculation that

Figure 7. General procedure of SBFEM polyhedron mesh generation by out-of-box tools.
include Eigen-decomposition are the major differences from FEM. Subroutines from LAPACK\textsuperscript{56} is appended to UEL in the code file to perform STEP 3. It is obvious that this UEL is compatible with the traditional FEM element topology built in ABAQUS, considering the tetrahedron, pentahedron and hexahedron as special cases of the polyhedron.

**Numerical application**

There are two examples for the test and application of the implementation of arbitrary polyhedral elements developed in "Implementation in ABAQUS" section. A cantilever subjected to a harmonic excitation with the traditional hexahedron element and the polyhedron element are compared to confirm the accuracy of the developed UEL. Then the polyhedron element based on SBFEM is applied in the soil-structure dynamic interaction analysis of a nuclear power plant, the applicability and performance of this implementation is tested.

**Cantilever subjected to a harmonic excitation.**  A 5 m long horizontal beam subjected to concentrated forces on the vertices at one end and fixed at the other end is considered (see Fig. 10). The homogenous elastic material properties of Young's modulus, density and Poisson's ratio are 20 GPa, 2000 kg/m$^3$, and 0.25 respectively. The harmonic concentrated forces in Y direction on the two vertices follow Eq. (73) defined as

$$F = A \sin \left( \frac{2\pi}{T} t \right).$$  

\text{(73)}
where $T$ denoting the period is assumed to be 0.05 s, $A$ denoting the amplitude is assumed to be 1000 N. $F$ denotes the concentrated force. The HHT direct time integration calculation is terminated at 0.25 s with a fixed time step of 0.001 s.

**Numerical model.** From the geometry, two sets of meshes with varied mesh density are generated. The first set of meshes is made of structured hexahedron elements, namely C3D8 element (3D linear solid element with 8 nodes) built and verified in ABAQUS/Standard and the other is made of unstructured polyhedron elements implemented by UEL based on SBFEM. The details of the meshes are summarized in Table 4, where label P1 to P4 denote the polyhedron meshes with different mesh density (see Fig. 11) that can only be calculated by UEL developed in “Implementation in ABAQUS” section and H1 to H7 denote the hexahedron meshes (see Fig. 12) that can be calculated by C3D8 (built-in element) or the UEL (labeled as H2_UEL and H6_UEL). Model analysis for the first 50 modes of vibration and implicit dynamic analysis are performed.

---

**Figure 9.** Basic branching structure of UEL for general static and direct-integration dynamic analysis.

**Figure 10.** Geometry, load and boundary conditions of the cantilever.
The results of model analysis and direct dynamic analysis of the cantilever have proven the accuracy of the UEL calculated by the C3D8 element and the user-defined polyhedron element based on SBFEM is less than 0.5%. The displacement responses in Y direction from the implicit dynamic analysis of model P1, P2, P3, and H6 are demonstrated in Fig. 17. Except the displacement values that are close to zero, the error between the model H2 and H6_UEL is equal to model H6 calculated by C3D8. In other words, polyhedron element based on SBFEM can get the same level of accuracy with less than 1% of the elements based on FEM, the most immediate advantage of which is the reduction in storage space for large-scale problems.

Table 4. Mesh information of the cantilever.

| Model name | Element number | Node number | Mesh type | Element type |
|------------|----------------|-------------|-----------|--------------|
| P1         | 2312           | 4663        | Polyhedron | UEL          |
| P2         | 4384           | 5977        | Polyhedron | UEL          |
| P3         | 17,880         | 28,103      | Polyhedron | UEL          |
| H1         | 5              | 24          | Hexahedron | C3D8         |
| H2         | 40             | 99          | Hexahedron | C3D8         |
| H3         | 117            | 224         | Hexahedron | C3D8         |
| H4         | 153            | 288         | Hexahedron | C3D8         |
| H5         | 625            | 936         | Hexahedron | C3D8         |
| H6         | 5000           | 6171        | Hexahedron | C3D8         |
| H7         | 625,000        | 652,851     | Hexahedron | C3D8         |
| H2_UEL     | 40             | 99          | Polyhedron | UEL          |
| H6_UEL     | 5000           | 6171        | Polyhedron | UEL          |

Figure 13 shows that the difference between the frequencies of models with varied mesh densities is no more than 9% for the first vibration mode and this difference is enlarged to less than 19% for the first 9 vibration modes, which tends to keep on increasing with the order. Even though when the mesh is coarse, the convergence speed is fast with the increasing of the element, the convergence speed is much slower near the convergence values leading to rather poor computation efficiency (see Fig. 14). For example, the highest frequency of model H6 and H7 are 1374.8 Hz and 1392 Hz, the computation times of which are 5 s and 1109 s. More than 220 times the computation time of model H6 brings less than 1.2% improvement of the results of model H7. This is a typical illustration of the need to pay great attention to the balance between mesh density and calculation efficiency. Since it is reasonable to assume that the results of model H7 are close enough to the convergence values, the true fundamental frequency of this cantilever is assumed to be 19.955 Hz.

To exclude the influence of the meshes, the same hexahedron meshes H2 and H6 are calculated by C3D8 and UEL respectively. Figure 15 shows the frequency comparison of the cantilever model H6_UEL calculated by UEL and H7 by C3D8. The consistency of the results of model H7 and H6_UEL confirms the accuracy of the UEL for the SBFEM-based polyhedron element with error less than 0.6% over the entire calculated frequency range. Figure 16 shows the frequency comparison of the cantilever mesh model H2 calculated by C3D8 and UEL. Compared to model H2 calculated by C3D8, model H2_UEL can get much better results that is almost equal to model H6 calculated by C3D8. In other words, polyhedron element based on SBFEM can get the same level of accuracy with less than 1% of the elements based on FEM, the most immediate advantage of which is the reduction in storage space for large-scale problems.

The displacement responses in Y direction from the implicit dynamic analysis of model P1, P2, P3, and H6 are demonstrated in Fig. 17. Except the displacement values that are close to zero, the error between the model calculated by the C3D8 element and the user-defined polyhedron element based on SBFEM is less than 0.5%. The results of model analysis and direct dynamic analysis of the cantilever have proven the accuracy of the UEL implementation of the SBFEM-based polyhedron element and the efficiency of the SBFEM for the solid medium.

Soil-structure dynamic interaction analysis of a nuclear power plant. To verify the performance and robustness of the implementation that has been primarily validated in “Cantilever subjected to a harmonic excitation” section, the application in the analysis of a practical engineering problem is necessary. A typical numerical elastic SSI analysis of a NPP with direct method in the time domain is taken as an example, which includes the simulation of the structure with traditional FEM elements, near field soil with the polyhedron elements, far field soil with an artificial non-reflecting boundary condition, external excitation input with site response analysis, and interaction between them with constrain or contact. For simplicity, the SSI analysis of a NPP is elastic. But it is enough to demonstrate the merit in automatic mesh generation and ability to be integrated seamlessly with other modules of the polyhedron element based on SBFEM and its implementation by UEL. The nuclear power plant is located on elastic rock soil with the Young’s modulus of 2.63E10 Pa, Poisson’s ratio of 0.25 and density of 1762.06 kg/m³. Two artificial seismic plane waves vibrating along the two horizontal directions with PGA of 0.9806 m/s² or 0.1 g (see Fig. 18) defined by the U.S. NRC RG 1.6057 response acceleration spectrum (see Fig. 19) are introduced into the SSI system from the bottom of the rock propagating upward along the vertical direction. This earthquake intensity just satisfies the minimum limit recommended by the...
IAEA guidelines for the SL-2 ground motion hazard level. The SSI numerical analysis model of NPP based on the direct method is illustrated in Fig. 20.

**Numerical model.** The SSI numerical analysis of NPP can be divided into three individual parts: the structure, the near-field and the far-field. The structure and the near-field are modeled by FEM. The far-field is modeled by the direct method. Benefiting from the FEM for unmatched mesh based on kinematic constraints, the structure and the near-field soil can be meshed separately.
Figure 12. Hexahedron mesh of the cantilever (taken model H6 as an example).

Figure 13. First 50 natural frequencies of the C3D8 meshes of the cantilever.

Figure 14. Fundamental frequencies of the C3D8 meshes of the cantilever.
Figure 15. Frequency comparison of the C3D8 and UEL for mesh model H6.

Figure 16. Frequency comparison of the C3D8 and UEL for mesh model H2.

Figure 17. Displacement response in Y direction of models calculated by C3D8 and UEL.
For simplicity, the structures on the nuclear island including the containment and the auxiliary buildings are meshed with shell elements S4R or S3 (see Fig. 21). The containment of this NPP is a double shell structure including an outer reinforced concrete vessel called shield building (SB) and an inner steel vessel. The foundation of the nuclear island is meshed with solid elements embedded in the near-field soil. The connection between the structure and the near-field soil is modeled by the tie constrain defined on the interface.

The geometry of the near-field soil can be very complex once geological features must be considered like cracks or faults, etc. Therefore, the solid element is the most reasonable choice to model the near-field soil. Besides, in order to ensure the accuracy, the direct method requires that the size of the near-field soil should be as large as possible and no less than twice of the size of the structure consuming a lot of computing resources. Tremendous efforts for the mesh generation are needed to balance the accuracy and efficiency of the numerical model. In this example, the near-field soil is meshed with octree solid elements automatically (see Fig. 22). Although the geometry of the near-field soil in the example is rather simple, the geological complexity of the soil should not cause any difficulties in the automatic mesh generation process.

The SSI effect is considered by applying an artificial non-reflecting boundary condition (NRBC) at the truncation of the near field soil. Specifically, a set of springs and dashpots tied to the ground along the three directions are added to the nodes on the interface between the near-field and far-filed soil representing the approximation of the viscoelastic NRBC. Unlike the substructure method, the corresponding input of exogenous excitations for a specific NRBC is needed. For viscoelastic boundary, the seismic load is imported by equivalent forces acting
on the nodes of the artificial boundary. The Rayleigh damping is applied to keep the damping ratio equal to 0.05 at the frequencies of 2.0 Hz and 15.0 Hz with the coefficient of $\alpha = 1.108797$ and $\beta = 0.000936$.

There are totally 260,499 elements, 167,221 nodes, and 549,177 degrees of freedom in this NPP model, the 13 section properties of which are summarized in Table 5. A transient dynamic analysis based on HHT implicit time integration is performed with a fixed time step of 0.01 s.

**Results and discussion.** To investigate the detailed acceleration and displacement responses of the NPP, five nodes are selected as monitoring points (see Fig. 23), the locations of which are listed in Table 6.

The time history of acceleration and displacement response of these nodes is compared. Figures 24 and 25 show the horizontal displacement and acceleration response of the nodes on the NPP foundation at the height of 33.576 m. It appears that the foundation is moving like a rigid plate without any rotation. The peak acceleration of the input movement at the foundation is about 0.09 g, which is a little less than the input movement at the substratum top (or the hard rock bottom). The site hardly changes the input movement due to the thin hard rock soil layer beneath the foundation indicating an insignificant SSI effect. The peak displacement of the nodes at the foundation is about 0.076 m, which can be recognized as a basic motion of the entire NPP. Figures 26 and 27 display the horizontal displacement and acceleration response of the nodes on the NPP foundation at different heights from the foundation to the top of the outer RC containment. It can be found in Fig. 26 that the peak displacement is 0.054 m (X direction) and 0.077 m (Y direction) at the height of 33.567 m, 0.058 m (X direction) and 0.081 m (Y direction) at the height of 69.671 m, 0.069 m (X direction) and 0.085 m (Y direction) at the height of 116.916 m, increasing along the vertical direction from the bottom to the top of the containment.

![Figure 19](image1.png)

**Figure 19.** Design response acceleration spectrum defined by RG 1.60.

![Figure 20](image2.png)

**Figure 20.** Numerical SSI model of NPP based on the direct method.
structure. In Fig. 27, the same trend that the peak accelerations increase from 0.85 m/s² (X direction) and 0.81 m/s² (Y direction) to 2.75 m/s² (X direction) and 3.43 m/s² (Y direction) to 5.47 m/s² (X direction) and 7.9 m/s² (Y direction) can be found. The geometry asymmetry is the main reason why there are differences in the response along the X and Y direction. The acceleration enlargement ratio from the foundation to the top is about 9.75, which is the result of the stiffness of the soil. Figure 28 illustrates the relative displacement response of the SB top to the SB foundation, eliminating the rigid transition movement due to the global site response. The peak values of the displacement in X direction and Y direction are 1.6 cm and 2.0 cm.

The floor response spectra of monitoring points at different heights with the damping ratio of 0.05 are listed in Fig. 29. The same trend of spectrum enlargement for the two horizontal directions can be observed. The peak spectrum values of the three nodes appear at the same frequency of 2.78 Hz, which is near the fundamental
Table 5. Section properties of the SSI model of the nuclear power plant.

| Section name | Thickness (m) | Density (kg/m³) | Young's modulus (Pa) | Poisson's ratio | Element type | Additional mass (kg/m²) | Location |
|--------------|--------------|-----------------|----------------------|----------------|--------------|------------------------|----------|
| S1           | 0.38         | 2300            | 3.35E10              | 0.2            | S4R/S3       | 1136.2                 | Structure|
| S2           | 0.912        | 2300            | 3.35E10              | 0.2            | S4R/S3       | 2728.88                | Structure|
| S3           | 0.6          | 2300            | 3.35E10              | 0.2            | S4R/S3       | 1794                   | Structure|
| S4           | 0.2286       | 2300            | 3.35E10              | 0.2            | S4R/S3       | 683.514                | Structure|
| S5           | 0.762        | 2300            | 3.35E10              | 0.2            | S4R/S3       | 2278.38                | Structure|
| S6           | 3.9768       | 2300            | 3.35E10              | 0.2            | S4R/S3       | 111,899.8              | Structure|
| S7           | 4.6048       | 2300            | 3.35E10              | 0.2            | S4R/S3       | 13,768.3               | Structure|
| S8           | 0.45         | 2300            | 3.35E10              | 0.2            | S4R/S3       | 1345.5                 | Structure|
| S9           | 0.048        | 7800            | 2.0E11               | 0.3            | S4R/S3       | 486.72                 | Structure|
| S10          | 0.042        | 7800            | 2.0E11               | 0.3            | S4R/S3       | 418.782                | Structure|
| S11          | -            | 2300            | 3.35E10              | 0.2            | C3D8R/C3D6   | -                      | Structure|
| S12          | -            | 1762.06         | 2.63E10              | 0.25           | UEL          | -                      | Soil     |

Figure 23. Locations of monitoring points of the nuclear power plant.
Table 6. Locations of monitoring points of the nuclear power plant.

| Node ID | Part  | X coordinate (m) | Y coordinate (m) | Z coordinate (m) |
|---------|-------|------------------|------------------|------------------|
| 18,624  | Structure | 115.5            | 131.355          | 33.576           |
| 9959    | Structure | 115.5            | 131.355          | 69.671           |
| 12,070  | Structure | 115.500          | 222.820          | 116.916          |
| 18,116  | Structure | 73.742           | 109.255          | 33.576           |
| 18,168  | Structure | 151.162          | 117.789          | 33.576           |

Figure 24. Horizontal displacement histories of monitoring points at the foundation. (a) X direction, (b) Y direction.

Figure 25. Horizontal acceleration histories of monitoring points at the foundation. (a) X direction, (b) Y direction.
frequency of the SSI system. In the frequency range between 0.25 and 1.0 Hz, the enlargement ratio along the height is less than 1.5. The enlargement ratio increases starting from 1.0 Hz and reaches its peak value of 13.33 at 2.78 Hz. This value decreases to about 9.3 at 100.0 Hz that is nearly equivalent to the acceleration enlargement ratio from the foundation to the top of the shield building.

The distribution of the peak acceleration (absolute values) can be seen from Fig. 30. The maximum PGA values arise at the top of the shield building. In general, the PGA values are enlarged along the vertical direction except regions near the foundation. The amplification ratio of the peak acceleration along three paths is detailed

![Graph](https://example.com/graph.png)  
**Figure 26.** Horizontal displacement histories of monitoring points at different heights (33.576 m, 69.671 m, 116.916 m). (a) X direction, (b) Y direction.

![Graph](https://example.com/graph.png)  
**Figure 27.** Horizontal acceleration histories of monitoring points at different heights (33.576 m, 69.671 m, 116.916 m). (a) X direction, (b) Y direction.
in Fig. 31. The nodes of the three paths that are parallel to Z axis are chosen as the monitoring points (see Fig. 23) and the height of the foundation bottom is set as zero. There is an upheaval of amplification ratio at the height of around 75 m where the water tank is located.

**Conclusion**

The results of two examples indicate that the implementation of SBFEM-based arbitrary polyhedron elements can be adapted to the automatic FEM analysis of static and dynamic elastic problems with polyhedron meshes generated by any tool. The first example conforms the accuracy and efficiency of the polyhedron element based on SBFEM and the associated UEL implementation in ABAQUS. Through the second example, it can be found that

---

**Figure 28.** Relative horizontal displacement histories of the points on the top of the shield building. (a) X direction, (b) Y direction.

---

**Figure 29.** Floor response spectrums of monitoring points at different heights (33.576 m, 69.671 m, 116.916 m).
after the geometry model of a structure is built, automatic dynamic analyses of complex or large structures can be achieved by the combination of polyhedron mesh generation technique and the implementation of arbitrary polyhedron elements in ABAQUS. On one hand, this implementation of SBFEM-based elements dispenses with the development of the pre/post-processor, matrix solver, time integration module, contact analysis module, nonlinear analysis module, etc. extending the usability of SBFEM in engineering analyses of bounded domain. On the other hand, this implementation also enhances the original function of ABAQUS to a great extent allowing engineering analyses with elements of almost any topology.

Figure 30. Distribution of the horizontal peak acceleration of the NPP structure. (a) X direction, (b) Y direction.

Figure 31. Amplification ratio of the peak acceleration along the Z direction.
Appendix 1: Interface of UEL and UEXTERNALDB

```fortran
SUBROUTINE UEL(RHS,AMATRX,SVARS,ENERGY,NDOFEL,NRHS,NSVARS,
                PROPS,NPROPS,COORDS,MCRD,NNODE,U,DU,V,A,JTYPE,TIME,DTIME,
                KSTEP,KINC,JELEM,PARAMS,NDLOAD,JDLTYP,ADLMAG,PREDEF,NPREDF,
                LFLAGS,MLVARX,DDLMAG,MDLOAD,PNEWDT,JPROPS,NJPROP,PERIOD)
  C
  INCLUDE 'ABA_PARAM.INC'
  C
  DIMENSION RHS(MLVARX,*),AMATRX(NDOFEL,NDOFEL),PROPS(*),
             SVARS(*),ENERGY(8),COORDS(MCRD,NNODE),U(NDOFEL),
             DU(MLVARX,*),V(NDOFEL),A(NDOFEL),TIME(2),PARAMS(*),
             JDLTYP(MDLOAD,*),ADLMAG(MDLOAD,*),DDLMAG(MDLOAD,*),
             PREDEF(2,NPREDF,NNODE),LFLAGS(*),JPROPS(*)
  C
  user coding to define RHS, AMATRX, SVARS, ENERGY, and PNEWDT
  RETURN
END

SUBROUTINE UEXTERNALDB(LOP,LRESTART,TIME,DTIME,KSTEP,KINC)
  C
  INCLUDE 'ABA_PARAM.INC'
  C
  DIMENSION TIME(2)
  C
  user coding to open files, close files, etc.
  RETURN
END
```

Appendix 2: INP format to define user elements

```plaintext
*USER ELEMENT, NODES=2, TYPE=U1, PROPERTIES=4, COORDINATES=3, I PROPERTIES=2, VARIABLES=12
  1, 2, 3
*ELEMENT, TYPE=U1, ELSET=UTRUS
  101, 101, 102
*UEL PROPERTY, ELSET=UTRUS
  1, 2, 0.002, 2.1E11, 0.3, 7200.
```

Appendix 3: Element computation algorithm of UEL

| Stiffness and mass matrix computation of the polyhedron element based on SBFEM |
| --- |
| **STEP 1:** Prepare S-element data, including shifted coordinates of nodes and scaling center (Eq. (15)), node connectivity of each W-element, etc. |
| **STEP 2:** For each W-element do: |
| **STEP 2.1:** Prepare W-element data and integration points data; |
| **STEP 2.2:** For each numerical integration point do: |
### Stiffness and mass matrix computation of the polyhedron element based on SBFEM

| Step | Description |
|------|-------------|
| 2.1  | Compute the shape functions and their derivatives. |
| 2.2  | Compute the determinant of the Jacobian matrix on the boundary by Eq. (21). |
| 2.3  | Compute $|b_1|^2, |b_2|^2, |b_3|^2$ by Eqs. (31)-(33). |
| 2.4  | Compute $[B]^T[B]$ by Eqs. (45), (46). |
| 2.5  | Compute material property matrix $[D]$ by Eq. (8). |
| 2.6  | Compute $[Z]^T$, $[Z]$, $[Z]^T$, and $[M]^T$ by Eq. (57). |
| 2.7  | Accumulate the $[Z]^T$, $[Z]$, $[Z]^T$, and $[M]^T$ values. |
| 3.0  | Assemble the $[Z]^T$, $[Z]$, $[Z]^T$, and $[M]^T$ values. |
| 4.0  | Compute the stiffness matrix $[K]$ by Eq. (69). |
| 5.0  | Compute the mass matrix $[M]$ by Eq. (70)-(72). |

Received: 18 October 2021; Accepted: 28 February 2022
Published online: 09 March 2022

### References
1. Moretti, G. Grid generation using classical techniques. In (ed., Langley, V. A.) (1980) [cited 2020 Aug 8]. Available from: [http://www.researchgate.net/publication/23912609_Grid_generation_using_classical_techniques](http://www.researchgate.net/publication/23912609_Grid_generation_using_classical_techniques).
2. Caughey, D. A. A systematic procedure for generating useful conformal mappings. *Int. J. Numer. Methods Eng.* 12(11), 1651–1657 (1978).
3. South, J. C. & Jameson, A. Relaxation solutions for inviscid axisymmetric transonic flow over blunt or pointed bodies. In AIAA Computational Fluid Dynamics Conference, *Palm Springs, California* (1973).
4. Baker, T. J. Mesh generation by a sequence of transformations. *Appl. Numer. Math.* 2(6), 515–528 (1986).
5. Sheehy, D. J., Armstrong, C. G. & Robinson, D. J. Computing the medial surface of a solid from a domain Delaunay triangulation. In *Proceedings of the Third ACM Symposium on Solid Modeling and Applications*. 201–212 (1995).
6. Sukumar, N. Automatic generation of hexahedral finite element meshes. *Comput. Aided Geom. Des.* 12(7), 693–707 (1995).
7. Zhu, J., Zienkiewicz, O., Hinton, E. & Wu, J. A new approach to the development of automatic quadrilateral mesh generation. *Int. J. Numer. Methods Eng.* 32(4), 849–866 (1991).
8. Blacker, T. D. & Stephenson, M. B. Paving: A new approach to automated quadrilateral mesh generation. *Int. J. Numer. Methods Eng.* 32(4), 811–847 (1991).
9. Dvorak, P. New element lifts time off CFD simulations. *Math. Des.* 7(5), 154–155 (2006).
10. Balafas, G. Polyhedral mesh generation for CFD-analysis of complex structures (Technische Universitat Munchen, 2014).
11. Sibson, R. A. A new vector identity for the Dirichlet tessellation. In *Mathematical Proceedings of the Cambridge Philosophical Society* 151–155 (Cambridge University Press, 1980).
12. Sibson, R. A. A brief description of natural neighbour interpolation. *Interpreting multivariate data*. 21–36 (1981).
13. Traversoni, L. Modified natural neighbor interpolant. In *Curves and Surfaces in Computer Vision and Graphics III* 196–203 (International Society for Optics and Photonics, 1992).
14. Sambridge, M., Braun, J. & McQueen, H. Geophysical parametrization and interpolation of irregular data using natural neighbours. *Geophys. J. Int.* 122(3), 837–857 (1995).
15. Watson, D. Compound signed decomposition, the core of natural neighbor interpolation in n-dimensional space. See [ftp://ftp.iagm.org/Waston/core.ps.gz](http://ftp.iagm.org/Waston/core.ps.gz) (2001).
16. Sukumar, N., Moran, B., Yu Semenov, A. & Belikov, V. Natural neighbour Galerkin methods. *Adv. Comput. Math.* 13(1), 129 (2006).
17. Sukumar, N. & Tabarraei, A. Conforming polygonal finite elements. *Int. J. Numer. Methods Eng.* 61(12), 2045–2066 (2004).
18. Sukumar, N. & Malsch, E. Recent advances in the construction of polygonal finite element interpolants. *Arch. Comput. Methods Eng.* 13(1), 129 (2006).
19. Fokker, M. & Floater, M. S. Mean value coordinates for arbitrary planar polygons. *ACM Transactions on Graphics (TOG)* 25(4), 1424–1441 (2006).
20. Floater, M. S. Mean value coordinates. *Comput. Aided Geom. Des.* 20(1), 19–27 (2003).
21. Lipman, Y., Levin, D. & Cohen-Or, D. Green coordinates. *ACM Trans. Graph.* (TOG) 27(3), 1–10 (2008).
22. Warren, J. Conformal coordinates for convex polytopes. *Adv. Comput. Math.* 6(1), 97–108 (1996).
23. Meyer, M., Barr, A., Lee, H. & Desbrun, M. Generalized barycentric coordinates on irregular polygons. *J. Graph. Tools* 7(1), 13–22 (2002).
24. Hornung, K. & Floater, M. S. Mean value coordinates for arbitrary planar polygons. *ACM Transactions on Graphics (TOG)* 23(4), 849–866 (2004).
25. Floater, M. S. Mean value coordinates. *Comput. Aided Geom. Des.* 20(1), 19–27 (2003).
26. Swee, L. & Desbrun, M. Interpolating the values of a function on an arbitrary set of points. *Comput. Math. Math. Phys.* 37(1), 9–15 (1997).
27. Sukumar, N. & Tabarraei, A. Conforming polygonal finite elements. *Int. J. Numer. Methods Eng.* 61(12), 2045–2066 (2004).
28. Sukumar, N. & Malsch, E. Recent advances in the construction of polygonal finite element interpolants. *Arch. Comput. Methods Eng.* 13(1), 129 (2006).
34. Li, M., Zhang, Y., Zhang, H. & Guan, H. Numerical stability and accuracy of the scaled boundary finite element method in engineering applications. ANZIAM J. 57(2), 114–137 (2015).
35. Wang, F., Lin, G., Zhou, Y. & Chen, D. Element-free Galerkin scaled boundary method based on moving Krige interpolation for steady heat conduction analysis. Eng. Anal. Bound. Elem. 1(106), 440–451 (2019).
36. Wolf, J. P. & Song, C. Finite-Element Modelling of Unbounded Media (Wiley, 1996).
37. Talebi, H., Saputra, A. & Song, C. Stress analysis of 3D complex geometries using the scaled boundary polyhedral finite elements. Comput. Mech. 58(4), 697–715 (2016).
38. Liu, Y., Saputra, A. A., Wang, J., Yin-Loi, F. & Song, C. Automatic polyhedral mesh generation and scaled boundary finite element analysis of STL models. Comput. Methods Appl. Mech. Eng. 313, 106–132 (2017).
39. Zhang, J. & Song, C. A polytree based coupling method for non-matching meshes in 3D. Comput. Methods Appl. Mech. Eng. 349, 743–773 (2019).
40. Ooi, E. T., Saputra, A., Natarajan, S., Ooi, E. H. & Song, C. A dual scaled boundary finite element formulation over arbitrary faceted star convex polyhedra. Comput Mech. 66(1), 27–47 (2020).
41. Zou, D., Chen, K., Kong, X. & Liu, J. An enhanced octree polyhedral scaled boundary finite element method and its applications in structure analysis. Eng. Anal. Bound. Elem. 84, 87–107 (2017).
42. Chen, K., Zou, D. & Kong, X. A nonlinear approach for the three-dimensional polyhedron scaled boundary finite element method and its verification using Koyna gravity dam. Soil Dyn. Earthq. Eng. 1(96), 1–12 (2017).
43. Huang, Y. J., Yang, Z. J., Liu, G. H. & Chen, X. W. An efficient FE-SBEF coupled method for mesoscale cohesive fracture modelling of concrete. Comput. Mech. 58(4), 635–655 (2016).
44. Zhang, Z., Dissanayake, D., Saputra, A., Wu, D. & Song, C. Three-dimensional damage analysis by the scaled boundary finite element method. Comput. Struct. 206, 1–17 (2018).
45. Zhang, Z., Liu, Y., Dissanayake, D. D., Saputra, A. A. & Song, C. Nonlocal damage modelling by the scaled boundary finite element method. Eng. Anal. Bound. Elem. 99, 29–45 (2019).
46. Zou, D., Sui, Y. & Chen, K. Plastic damage analysis of pile foundation of nuclear power plants under beyond-design basis earthquake excitation. Soil Dyn. Earthq. Eng. 136, 106179 (2020).
47. Yang, Z. J., Yao, F. & Huang, Y. J. Development of ABAQUS UEL/VUEL subroutines for scaled boundary finite element method for general static and dynamic stress analyses. Eng. Anal. Bound. Elem. 1(114), 58–73 (2020).
48. Wolf, J. P. & Song, C. The scaled boundary finite-element method—a primer: Derivations. Comput. Struct. 78(1–3), 191–210 (2000).
49. Song, C. & Wolf, J. P. The scaled boundary finite-element method—a primer: Solution procedures. Comput. Struct. 78(1–3), 211–225 (2000).
50. Deeks, A. J. & Wolf, J. P. A virtual work derivation of the scaled boundary finite-element method for elastostatics. Comput. Mech. 28(6), 489–504 (2002).
51. Song, C. & Wolf, J. P. Body loads in scaled boundary finite-element method. Comput. Methods Appl. Mech. Eng. 180(1–2), 117–135 (1999).
52. Baziyar, M. H. & Song, C. A continued-fraction-based high-order transmitting boundary for wave propagation in unbounded domains of arbitrary geometry. Int. J. Numer. Methods Eng. 74(2), 209–237 (2008).
53. Song, C. The Scaled Boundary Finite Element Method: Introduction to Theory and Implementation 504 (Wiley, 2018).
54. SIMULIA User Assistance 2018. (Dassault Systemes Simulia Corp., 2017).
55. Hilber, H. M., Hughes, T. J. & Taylor, R. L. Improved numerical dissipation for time integration algorithms in structural dynamics. Earthq. Eng. Struct. Dyn. 3(5), 283–292 (1977).
56. Anderson, E. et al. LAPACK Users’ Guide (SIAM, 1999).
57. U.S. Nuclear Regulatory Commission Regulatory Guide 1.60. Design Response Spectra for Seismic Design of Nuclear Power Plants (2014).
58. International Atomic Energy Agency (IAEA). SSG-9 Seismic Hazards in Site Evaluation for Nuclear Installations: Safety Guide (2010).
59. Liu, J., Du, Y., Du, X., Wang, Z. & Wu, J. 3D viscous-spring artificial boundary in time domain. Earthq. Eng. Eng. Vib. 5(1), 93–102 (2006).
60. Li, Y., Hu, Z., Li, Z., Lin, G., Zhou, L. Practical technique for analysing free-field response of horizontally layered viscoelastic half-space excited by vertically travelling seismic waves. Soil Dyn. Earthq. Eng. 151, 106968 (2021).

Acknowledgements
This research was supported by Grant 52178460 from the National Natural Science Foundation of China, Grant 51779222 from the National Natural Science Foundation of China, and DUT20TD209 from the Fundamental Research Funds for the Central Universities of China.

Author contributions
Lei Zhou debugged the program, analyzed the results, and wrote the manuscript. Jianbo Li developed the program. Gao Lin provided the theoretical guide and designed the examples. All authors reviewed the manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to J.L.
Reprints and permissions information is available at www.nature.com/reprints.
Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
