ShapeGen3D description

Adnan Shahriar
The University of Texas at San Antonio, San Antonio, TX 78256
adnan.shahriar@utsa.edu

ShapeGen3D organized into four segments, as presented in Fig. 5, to enable users to navigate, formulate and verify the accuracy of the formulation. Segment 1 is the input for the software that lets users choose the element family of interest and the desired interpolation order. Segment 2 includes the execution command to run the software and obtain the element formulation. Segment 3 allows users to store the element properties. Segment 4 represents the results and post-processing blocks that will enable users to observe and manipulate plots that can be used to verify the accuracy of the generated formulation visually. Detailed descriptions for each of these segments are provided below.

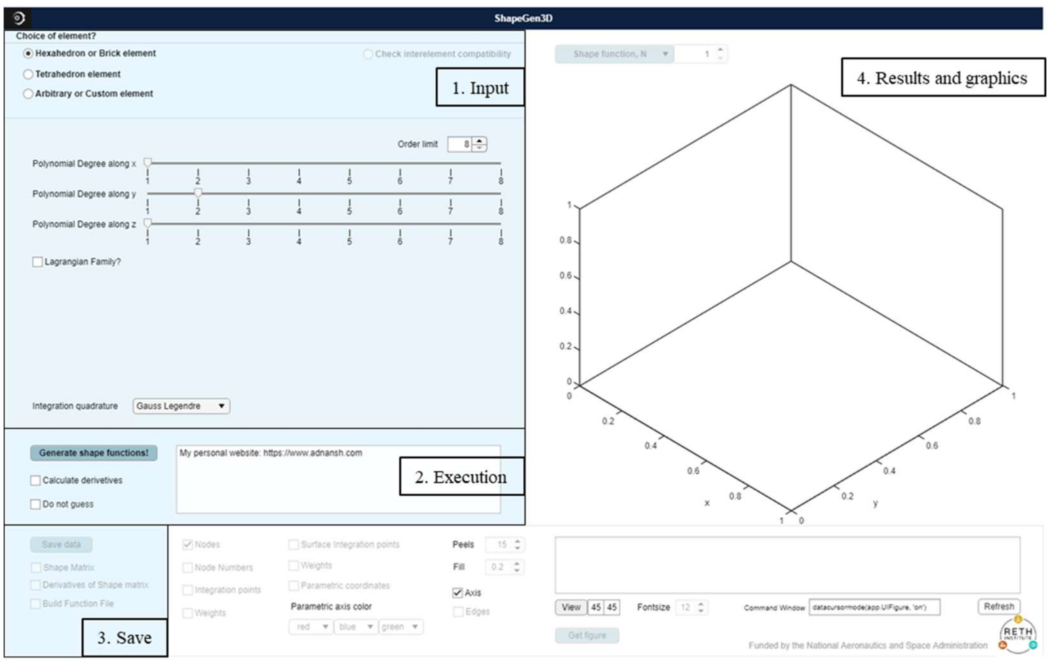

Figure 1. Software overview

A short description and a screenshot of the sub-options in Segment 1 are presented in Table 1 and Fig. 2, respectively. The element order and integration quadrature type must be defined for developing hexahedron and tetrahedron elements. On the other hand, in the case of arbitrary or custom elements, the nodal distribution must be inputted by users, and the software will determine the integration quadrature.

Table 1. Segment 1 sub-options and description

Seg 1: Input options	Sub-options	Description
Hexahedron or Brick element	Polynomial degree along x	m_x of Equation 6
	Polynomial degree along y	m_y of Equation 6
	Polynomial degree along z	m_z of Equation 6

	Lagrangian Family	Checking this option will formulate a Lagrangian element. Otherwise, a Serendipity element will be formulated
	Integration quadrature	This option offers two integration quadratures; 1) Gauss Legendre 2) Gauss Lobatto
Tetrahedron element	Element order	
	Number of Sub-Tetrahedral	The number of sub-tetrahedrons used for integration quadrature following the finite cell method. It is active when an element order greater than 4 is used.
	Integration order for reference tetrahedron	The integration order for each sub-tetrahedron. It is active when an element order greater than 4 is used.
Arbitrary or Custom element	Input node coordinates	A table where the user can input the nodal distribution of the desired element and cartesian coordinates (x, y,z) for each node.
	Input interpolation function	A table where the user can input $F_i(x, y, z)$ from Equation 3, where $i = [1\ d]$ and d is the number of nodes.
	Integration quadrature	If the software detects an arbitrary element from the isoparametric hexahedron family, it will determine the selected integration quadrature from one of the selected options: 1) Gauss Legendre; and 2) Gauss Lobatto.
	Integration order for reference tetrahedron	If the element is not from an isoparametric hexahedron family, the integration quadrature is determined using Delaunay triangulation
Check interelement compatibility		Used to illustrate two consecutively generated elements

Figure 2. Overview of the options in Segments 1 and 2: a) Hexahedron or Brick elements, b) Tetrahedron element, c) Arbitrary or Custom elements, and d) Check interelement compatibility.

Segment 2 represents the "Execution" block. It includes the "Generate shape function" button, which executes the code with the input given in Segment 1 and generates shape functions, integration quadrature, and graphical representation of the element properties. The code can automatically detect isoparametric hexahedrons and tetrahedron elements. There are several checkboxes, such as the "Do not guess" command. If this box is checked, the software avoids detecting the element type and assumes it is an arbitrary element. This segment also includes a display window that shows messages and the progress of the execution of the code. If the "Check interelement compatibility" box is selected in Segment 1, this option will be activated in Segment 2. The software will compute the nodes with the same coordinates between two consecutive elements and subtract the corresponding shape functions. If the difference is 0 at an interelement boundary, that boundary satisfies the compatibility condition between two elements, i.e., there are no discontinuities at the boundary.

Segment 3 is a block that outputs results in the Matlab workspace to allow users to implement the elements in other programming languages or through user elements in commercial finite element software, such as ABAQUS and ANSYS. Users can also output shape functions and derivatives of the shape functions evaluated at the integration points by ticking the "Shape matrix" and "Derivatives of Shape matrix" options. A description of the workspace files is presented in Table 3. These files are readily available for FEM calculations.

Finally, segment 4 represents a results and post-processing block that shows the results obtained by the software and allows users to manipulate graphics. There are many components, such as the "Shape function, N" spinner block, which is a control block where the user can select to plot the number of shape functions properties, such as their value and derivatives, to plot. This segment also allows users to plot other element properties, such as node number, integration points, and weights, as shown in the bottom left part of the software. It also includes a display window that shows the shape function property in equation format and other user-controlled options to manipulate the graphics options, such as the view angle, axis, image resolution (Peels), and transparency (Fill). Increasing the "Peels" will increase the overall image resolution. If "Fill" is chosen as 0, the software will not show any 0 value and will represent it as a blank space.

Table 3. Output files to Matlab workspace

Workspace filename	**Description**	**Data structure**
Interpolation_Functions	Stores interpolation in symbolic format	$[F_1 \quad F_2 \quad . \quad F_d]$ F_i: Function I of Equation 1 in symbolic format
Node_Coordinates	Stores node coordinates	$\begin{bmatrix} n_{x_1} & n_{y_1} & n_{z_1} \\ n_{x_2} & n_{y_2} & n_{z_2} \\ . & . & . \\ n_{x_d} & n_{y_d} & n_{z_d} \end{bmatrix}$ $n_j(n_{x_i}, n_{y_i}, n_{z_i})$ where n_{x_i}: x coordinate of i^{th} node
Shape_Functions	Stores shape functions in symbolic format for the corresponding nodes	$\begin{bmatrix} N_1 \\ N_2 \\ . \\ N_d \end{bmatrix}$ N_i: Shape function i in symbolic format
Surface_Nodes_and_vectors	Stores nodes on surface and surface normal vector pointing outward of the domain for all surfaces	$\begin{bmatrix} a & n_{x_a} & n_{y_a} & n_{z_a} \\ b & n_{x_b} & n_{y_b} & n_{z_b} \\ . & . & . & . \\ d & n_{x_d} & n_{y_d} & n_{z_d} \end{bmatrix}$ a, b, d are node numbers and integers. $n_j(n_{x_j}, n_{y_j}, n_{z_j})$ is the normal vector pointing outward of the domain on the surface containing the j node
Integration_Points_Coordinates_Weights	Stores integration points and weights inside the element	$\begin{bmatrix} g_{x_1} & g_{y_1} & g_{z_1} & w_1 \\ g_{x_2} & g_{y_2} & g_{z_2} & w_2 \\ . & . & . & . \\ g_{x_d} & g_{y_d} & g_{z_d} & w_d \end{bmatrix}$ $g_j(g_{x_j}, g_{y_j}, g_{z_j})$ where g_{x_j}: x coordinate of j^{th} integration point. w_j: weight of j^{th} integration point
Integration_Points_Coordinates_Weights_Vectors on_Surface	Stores integration points and weights on surfaces and	$\begin{bmatrix} g_{x_1} & g_{y_1} & g_{z_1} & w_1 \\ g_{x_2} & g_{y_2} & g_{z_2} & w_2 \\ . & . & . & . \\ g_{x_d} & g_{y_d} & g_{z_d} & w_d \end{bmatrix} \begin{bmatrix} n_{x_1} & n_{y_1} & n_{z_1} \\ n_{x_2} & n_{y_2} & n_{z_2} \\ . & . & . \\ n_{x_d} & n_{y_d} & n_{z_d} \end{bmatrix}$

	surface normal vector pointing outward of the domain for all surfaces	$g_j(g_{x_j}, g_{y_j}, g_{z_j})$ where g_{x_j}: x coordinate of j^{th} integration point. w_j: weight of j^{th} integration point. $n_j(n_{x_j}, n_{y_j}, n_{z_j})$ is the normal unit vector pointing outward of the domain on the surface at j^{th} integration point
Parametric_Coordinates	Stores parametric coordinates for each surface of the element	$\begin{bmatrix} n_{x_1} & n_{y_1} \\ n_{x_2} & n_{y_2} \\ n_{x_3} & n_{y_3} \end{bmatrix}_S$ n_{x_j} and n_{y_j} are the parametric x and y coordinates, respectively. S represents the surface number.
Shape_Matrix	Stores the value of each shape function evaluated at each integration point	$\begin{bmatrix} N_1\|_{g_1} & N_2\|_{g_1} & . & N_n\|_{g_d} \\ N_1\|_{g_2} & N_2\|_{g_2} & . & N_n\|_{g_d} \\ . & . & . & . \\ N_1\|_{g_n} & N_2\|_{g_d} & . & N_n\|_{g_d} \end{bmatrix}$ $N_i\|_{g_j}$: i^{th} shape function evaluated at j^{th} integration point
dx_Shape_Matrix	Stores the value of the first derivative of each shape function evaluated at each integration point	$\begin{bmatrix} \left.\frac{\partial N_1}{\partial x}\right\|_{g_1} & \left.\frac{\partial N_2}{\partial x}\right\|_{g_1} & . & \left.\frac{\partial N_n}{\partial x}\right\|_{g_1} \\ \left.\frac{\partial N_1}{\partial x}\right\|_{g_2} & \left.\frac{\partial N_2}{\partial x}\right\|_{g_2} & . & \left.\frac{\partial N_n}{\partial x}\right\|_{g_2} \\ . & . & . & . \\ \left.\frac{\partial N_1}{\partial x}\right\|_{g_n} & \left.\frac{\partial N_n}{\partial x}\right\|_{g_n} & . & \left.\frac{\partial N_n}{\partial x}\right\|_{g_n} \end{bmatrix}$ $\left.\frac{\partial N_i}{\partial x}\right\|_{g_j}$: i^{th} shape function derivative evaluated at j^{th} integration point