Engineered Science
DOI: https://dx.doi.org/10.30919/es1413

An Automated Two-Dimensional Octic Order Triangular Mesh Generator in MATLAB for Sustainable Solutions through Finite Element Scheme

S. Deivanai,[1] P. Charan Sai,[1] T. V. Smitha[2,*] and K. V. Nagaraja[2,*]

Abstract

This work proposes an automated mesh generator using the octic order triangular elements for various two-dimensional (2D) geometries. The developed code produces high-quality eighth-order triangular meshes by providing nodal connectivity, node position, and boundary details. This method has a major influence on various engineering applications to provide sustainable solutions. Discretizing 2D domains into octic meshes can be advantageous in waveguide problems to minimise energy loss, leading to sustainable energy, aeroscope engineering, sustainable structural designs, fluid dynamics, manufacturing industries, crocheting, knitting, etc. The proposed automated octic triangular mesh generator using distmesh in MATLAB can be utilized for finite element simulations in various applications effectively. This scheme confirms that, in comparison to lower-order elements, octic elements dramatically reduce the count of degrees of freedom and elements needed to attain a desired level of precision. The numerical results show that the octic mesh elements perform better than the lower-order elements in terms of efficiency and accuracy. It provides reductions of over 96% in the number of elements and up to 90% in the degrees of freedom compared to linear, quadratic, and cubic orders, for the desired cutoff frequencies of the transverse electric and transverse magnetic modes in waveguides. This demonstrates the accuracy and efficiency of the octic order, making it ideal for various engineering applications.

Keywords: Octic order elements; Finite element method; Sustainable structural design; Sustainable energy.
Received: 29 November 2024; Revised: 21 January 2025; Accepted: 06 February 2024.
Article type: Research article.

1. Introduction

An essential initial phase in several numerical techniques, including the finite volume method, the finite difference method, and the finite element method (FEM), is the generation of meshes.[1] Mesh generation is a prerequisite for finding solutions to complex mathematical representations modeled as integral equation formulas or partial differential equations (PDEs).[2] This technique creates a set of discrete points over the region under consideration, which are called nodes. The adjacent nodes in the considered area are joined to form smaller elements, such as triangular or quadrilateral elements, in two-dimensional space. The domain that is being considered is decomposed into smaller elements called mesh elements. By breaking down the two-dimensional (2D) domain into numerous smaller triangular mesh components, one can easily analyze the data in that particular domain for a better understanding. Mesh generation also provides valuable information about the coordinates of the nodes, boundary nodes, and elemental connectivity of the discretized meshes, which is of great importance for solving several numerical problems represented as PDEs.

Engineering problems often involve two-dimensional geometries, which include regular and irregular shapes. Achieving accurate numerical solutions for such geometries using traditional meshes with higher-order elements remains a challenge. Although finite element analysis (FEA) offers a powerful solution, its effectiveness is very dependent on the quality of the initial mesh. FEA has gained widespread acceptance among modern engineers as a result of progress in computational capabilities that allow the resolution of complex problems.[3] Mesh generation has a major influence

[1] *Department of Computer Science Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Bengaluru, 560035, India*

[2] *Computational Science Lab, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Bengaluru, 560035, India*

**Email:* tv_smitha@blr.amrita.edu (T. V. Smitha);
kv_nagaraja@blr.amrita.edu (K. V. Nagaraja)

on various engineering applications, such as providing physical simulations, such as computer-aided design (CAD) geometries, computational fluid dynamics, mechanical engineering, aerospace engineering, and biomechanics, and is frequently used in FEA.[4–7] Recent advances in FEA and mesh generation enabled the automated creation of higher-order triangular meshes around airfoil designs using MATLAB, enhancing the accuracy of aerodynamic performance evaluations.[6,8] The effectiveness of mesh generation is essential for computational systems used in handicrafts such as knitting, crocheting, and many others, in addition to engineering applications.[9] Enormous research is done on numerical applications of finite element meshing in regular and irregular domains for waveguide problems,[10] dynamic fluid flow problems,[11] elliptical PDEs,[12] *etc*. In all of these applications, meshing is a crucial step. Moreover, engineering problems often involve two-dimensional geometries, which include regular and irregular shapes. As an example, the multiscale FEM has been introduced to address the shortcomings of traditional finite element techniques in designing intricate electrical equipment.[13] The FEM has been successfully employed to accurately calculate and optimize stray losses in power transformers, incorporating the temperature factor into the analysis.[14]

In the proposed method, the decomposition of domains into triangular elements is implemented with the help of distmesh2d based on the work presented by Per-Olof Person and Gilbert Strang.[15,16] But Person and Strang developed meshes for various domains only in the form of linear triangular elements. Later, Koko improvised the work and generated meshes up to quadratic triangle elements,[17] *i.e.*, second-order triangle elements for many domains by using distmesh2d. Quadratic triangular elements have three nodes, one at each edge of the triangle. Numerous investigations have been conducted about the subparametric transformation of triangle elements of quadratic (6 nodes), cubic (10 nodes), quartic (15 nodes), quintic (21 nodes), sextic (28 nodes), and septic (36 nodes). Unique meshing schemes were presented in the works published in Refs. [18,19] using the transformations to generate meshes with higher-order curved triangular elements of up to septic order for various two-dimensional regular and curved domains using subparametric transformations. After reviewing various studies, it is clear that there have been numerous attempts to perform meshing using linear triangular, quadrilateral, and polygonal elements.[20,21] However, these approaches have limitations in capturing complex geometries and may require a vast quantity of elements to achieve the desired accuracy, as their focus typically lies on lower-order elements.

Thus, an automatic generation of meshes utilizing octic order triangular elements in MATLAB is proposed in this study for both regular and irregular two-dimensional domains. The FEM implementation of the developed meshes is performed for engineering applications to demonstrate the efficiency of the proposed method. Octic order triangular elements provide an advantage in improving the solution accuracy in numerical methods with fewer elements compared to linear elements. This approach helps in providing faster simulations and potentially lower computational costs as fewer elements are required compared to lower-order FEM. Thus, the proposed code discretizes domains, with or without ducts, into meshes with structured and unstructured octic order triangular elements by using distmesh2d as the base mesh. From the literature, it is evident that automated octic order mesh generators are not available to users in MATLAB.

The organization of the remaining sections is as follows: Section 2 explains the methodology, which includes subsection 2.1, providing a brief overview of the mathematical formulation of the octic order triangular mesh. A detailed explanation of MATLAB code for generating an octic order coarse triangular mesh for some of the irregular and regular domains is demonstrated, depicting the octic order finite elements in subsections of subsection 2.2. Validation of the octic order meshing scheme using the finite element implementation is provided in section 3, with two applications explained in subsections 3.1 and 3.2. In section 4, the results obtained from the proposed methodology are described. Finally, the article is concluded in section 5.

2. Methodology

2.1 Mathematical formulation of octic order triangular meshes

Mesh generation is a fundamental step in many numerical methods. It involves discretizing complex geometries into smaller elements for numerical analysis. This section describes a method for generating two-dimensional octic order triangular meshes in MATLAB. The proposed method makes use of the existing distmesh2d.m function in MATLAB.[15] This function creates an initial mesh with linear elements. This initial mesh serves as a foundation upon which the octic order mesh is subsequently built.

In the field of FEA, a key objective is to generate a mesh for arbitrary domains and to solve the PDEs defined over these domains. Among the key challenges in this process is the complicated numerical integration that occurs in these irregular geometries. To make these challenges manageable, each element in the cartesian coordinate system (x, y) is mapped to a conventional isosceles right-angled triangle in the natural coordinate system (ϵ, μ), as illustrated in Fig. 1. This transformation provides a better approach to numerical integration evaluation within the mesh elements. The process for mapping nodes for any domain using higher-order finite elements is outlined by a key point transformation formula, as described by Eq. (1):[22]

$$p = \sum_{k=1}^{\frac{(n+1)(n+2)}{2}} N_k^{(n)}(\epsilon, \mu) p_k \, , (p = x, y) \qquad (1)$$

where $N_k^{(n)}(\epsilon, \mu)$ denotes the usual triangular element shape functions of order n at node k, and n = 8 for octic order triangular elements.[19] For octic order triangular elements, the

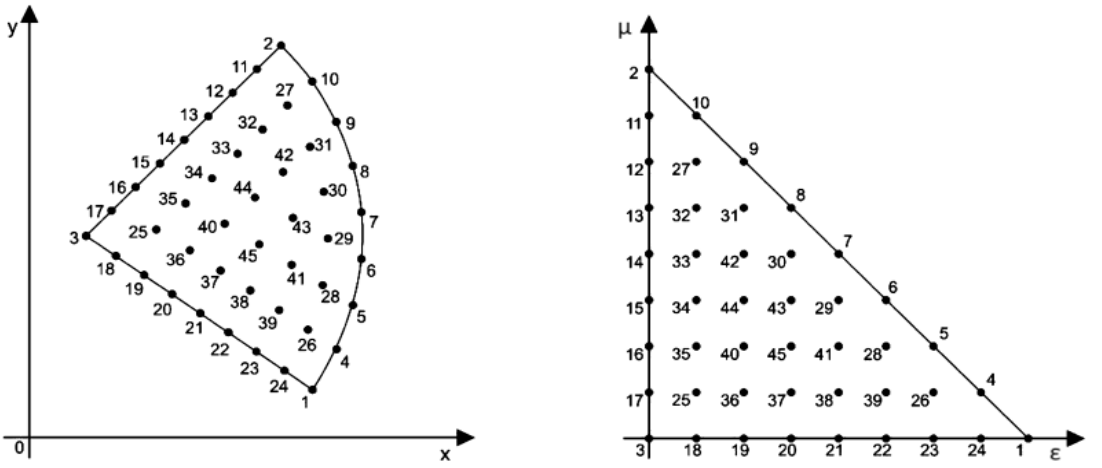

Fig. 1: Node mapping from cartesian coordinates (x, y) to natural coordinates (ϵ, μ).

meshing process involves representing the nodal points inside and on the boundary of the domain using the point transformation. The transformation for a triangular element of octic order can be represented by Eq. (2):

$$(x_k\ , y_k\ )\ =\ \sum_{i=1}^{45} N_i\ (\epsilon_k, \mu_k)(x_i, y_i) \tag{2}$$

The physical (global/cartesian) coordinates are (x_k, y_k), while the reference (local/natural) coordinates are (ϵ_k, μ_k) for each node k. The Lagrange shape functions for the octic order triangular element at node *i* and the physical coordinates of the nodes in the reference element are denoted by the expressions $N_i\ (\epsilon_k, \mu_k)$ and (x_i, y_i), respectively.[22] For any higher-order triangle element, the transformation formula Eq. (2) can be expressed as Eq. (3):

$$p = p_3 + (p_1\ -\ p_3\ )\epsilon\ +\ (p_2\ -\ p_3\ )\mu \tag{3}$$

In the case of an octic order triangular mesh, 45 nodes are required per triangle, with seven additional nodes added to each edge of the triangle. These additional nodes are aligned evenly along the straight edge.

A theoretical study has been carried out to develop the transformations for octic order triangular elements, as documented in the literature.[22] The study indicates that the coordinates $p_1, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}$ & p_2 span the boundary side of the triangle.

The following equations (Eq. (4)) determine the inner nodes p_{25}, p_{26}, p_{27}, p_{28}, p_{29}, p_{30}, p_{31}, p_{32}, p_{33}, p_{34}, p_{35}, p_{36}, p_{37}, p_{38}, p_{39}, p_{40}, p_{41}, p_{42}, p_{43}, p_{44}, p_{45} of the octic ordered triangle:

$$p_{25} = \frac{1}{32}(3p_1 + p_2 + 24p_3 + 2p_7)$$

$$p_{26} = \frac{1}{16}(9p_1 - p_2 + 2p_3 + 6p_7)$$

$$p_{27} = \tfrac{1}{16}(-p_1 + 9p_2 + 2p_3 + 6p_7) \tag{4}$$

$$p_{28} = \frac{1}{16}(5p_1 - p_2 + 2p_3 + 10p_7)$$

$$p_{29} = \frac{1}{8}(p_1 + p_3 + 6p_7)$$

$$p_{30} = \frac{1}{8}(p_2 + p_3 + 6p_7)$$

$$p_{31} = \frac{1}{16}(-p_1 + 5p_2 + 2p_3 + 10p_7)$$

$$p_{32} = \frac{1}{32}(-p_1 + 15p_2 + 8p_3 + 10p_7)$$

$$p_{33} = \frac{1}{8}(3p_2 + 3p_3 + 2p_7)$$

$$p_{34} = \frac{1}{32}(p_1 + 9p_2 + 16p_3 + 2p_7)$$

$$p_{35} = \frac{1}{16}(p_1 + 3p_2 + 10p_3 + 2p_7)$$

$$p_{36} = \frac{1}{16}(3p_1 + p_2 + 10p_3 + 2p_7)$$

$$p_{37} = \frac{1}{32}(9p_1 + p_2 + 16p_3 + 6p_7)$$

$$p_{38} = \frac{1}{8}(3p_1 + 3p_3 + 2p_7)$$

$$p_{39} = \frac{1}{32}(15p_1 - p_2 + 8p_3 + 10p_7)$$

$$p_{40} = \frac{1}{8}(p_1 + p_2 + 4p_3 + 2p_7)$$

$$p_{41} = \frac{1}{4}(p_1 + p_3 + 2p_7)$$

$$p_{42} = \frac{1}{4}(p_2 + p_3 + 2p_7)$$

$$p_{43} = \frac{1}{32}(3p_1 + 3p_2 + 8p_3 + 18p_7)$$

$$p_{44} = \frac{1}{16}(p_1 + 9p_2 + 6p_3 + 6p_7)$$

$$p_{45} = \frac{1}{16}(3p_1 + p_2 + 6p_3 + 6p_7)$$

Thus, these equations are used to define internal nodes distributed within octic order triangular elements. By discretizing the 2D domain into higher-order octic triangular mesh elements, this approach improves the precision and convergence rates of numerical simulations,[22] making it suitable for various engineering applications.

2.2 Octic order meshing for different 2D domains in MATLAB

This section contains the explanation of the proposed code OcticMesh2d for octic order ($n = 8$) mesh using eighth-order triangular elements in MATLAB. Both regular and irregular geometries are considered for the physical domains which will help the FEM utilizers. The meshing technique proposed in this section is based on the scheme adopted by Persson and Gilbert Strang.[15] They offered a short and simple description of the MATLAB code distmesh2d for linear triangular meshes that create good-quality meshes. However, there is a lack of easily accessible automated generators for octic order triangular meshes. Consequently, the suggested mesh generator OcticMesh2d provides a solution to this gap. The geometry is described implicitly by a few terms, such as mesh size function (fh), signed distance (fd), boundary box (bbox), fixed nodes that must appear in the mesh (pfix), and reference edge length, or the original characteristic size of the elements (h0). Along with the meshes generated, detailed information about the domain and the meshes, like nodal positions (p), indices of triangular elements (t), boundary edges (be), and boundary nodes (bn), is given as output of the code. The MATLAB code creates initial octic meshes with element size h0: [Nl, p, t, b] = OcticMesh2d(fd, fh, h0, bbox, pfix).

Algorithm 1: OcticMesh2d.

Input: fd, fh, h0, bbox, pfix
Output: N, p, t, b
1: Display "Octic mesh generation"
2: $(p,t) \leftarrow$ distmesh2d$(fd, fh, h0, bbox, pfix)$
3: $be \leftarrow$ boundedges(p,t)
4: $b \leftarrow$ unique(be)
5: $N \leftarrow$ size$(p,1)$
6: $T \leftarrow$ size$(t,1)$
7: Initialize sparse matrices $M1, M2, M3, M4, M5, M6, M7$ of size $N \times N$
8: $cnt \leftarrow N+1$
9: Initialize *lnodes* as a 9×2 matrix of zeros
10: **for** each edge e in T **do**
11: $nodes \leftarrow t(e,:)$
12: **if** $M1(nodes[1], nodes[2]) == 0$ **then**
13: Generate 9 linearly spaced points between *nodes*[1] and *nodes*[2]
14: Update sparse matrices $M1, M7, M2, M6, M3, M5, M4$ with new nodes
15: Update p with new points & increment *cnt*
16: **end if**
17: **if** $M1(nodes[2], nodes[3]) == 0$ **then**
18: Generate 9 linearly spaced points between *nodes*[2] and *nodes*[3]
19: Update sparse matrices $M1, M7, M2, M6, M3, M5, M4$ with new nodes
20: Update p with new points & increment *cnt*
21: **end if**
22: **if** $M1(nodes[1], nodes[3]) == 0$ **then**
23: Generate 9 linearly spaced points between *nodes*[1] and *nodes*[3]
24: Update sparse matrices $M1, M7, M2, M6, M3, M5, M4$ with new nodes
25: Update p with new points & increment *cnt*
26: **end if**
27: Update edge e with newly generated node indices in t
28: Generate additional internal points for the current element using equations given in Eq.(4)
29: Update p with these points and increment *cnt* accordingly
30: **end for**
31: **for** each node i in p **do**
32: Display '*' at coordinates $(p[i,1], p[i,2])$
33: **end for**
34: $Np \leftarrow$ size$(p,1)$
35: $T \leftarrow$ size$(t,1)$
36: **for** each node i in p **do**
37: Print coordinates of node i
38: **end for**
39: **return** N, p, t, b

Initially, different mesh domains are generated using the MATLAB code provided, which begins by defining the geometric constraints of the domain and using distance functions based on the definitions of the terms mentioned above in Ref. [15]. The MATLAB algorithmic function OcticMesh2d code lines (2 to 9) allocate memory and declare explicit variables for the octic order mesh generation, as demonstrated in Algorithm 1. The process of meshing the domains into octic triangular elements is shown in Fig. 2. The MATLAB algorithm OcticMesh2d generates equispaced nodes on each edge and interior using Eq. (4) for each triangle element. A thorough description of Eq. (4) is provided in Ref. [22].

The meshing approach provided yields the following results: (i) number of nodes N in the domain, (ii) an array p of $N \times 2$ elements that represent (x, y) coordinates for each of these nodes, (iii) a matrix t with each row representing an octic order triangular element node numbers, where each row includes $(n+1)(n+2)/2$ node numbers for that triangle and (iv) a list b of boundary nodes that are necessary for creating boundary conditions in FEA that are situated on the edge of the meshing domain.

Here, uniform and non-uniform meshing outputs from the proposed octic order mesh generator using regular straight-sided octic order triangular elements for various two-dimensional domains are provided. Recent studies demonstrated the effectiveness of Physics Informed Neural

Networks (PINNs) in solving complex heat transfer problems in fins and in accurately capturing temperature distributions and boundary conditions in different states.[23,24] Mesh generation is used to create training data for PINNs by discretizing the unit disk and sampling points within this domain.[25] The retrieved nodal details help train the neural network to correctly solve the Poisson equation. The proposed code discretizes the geometry into octic order meshes with edge length $h_0 = 0.2$ with each * symbolizing a node as shown in Fig. 3. The following subsections provide a detailed explanation and illustration of meshing schemes for various domains, both with and without ducts, using octic order triangular elements, which are relevant to the engineering applications discussed in the introduction.

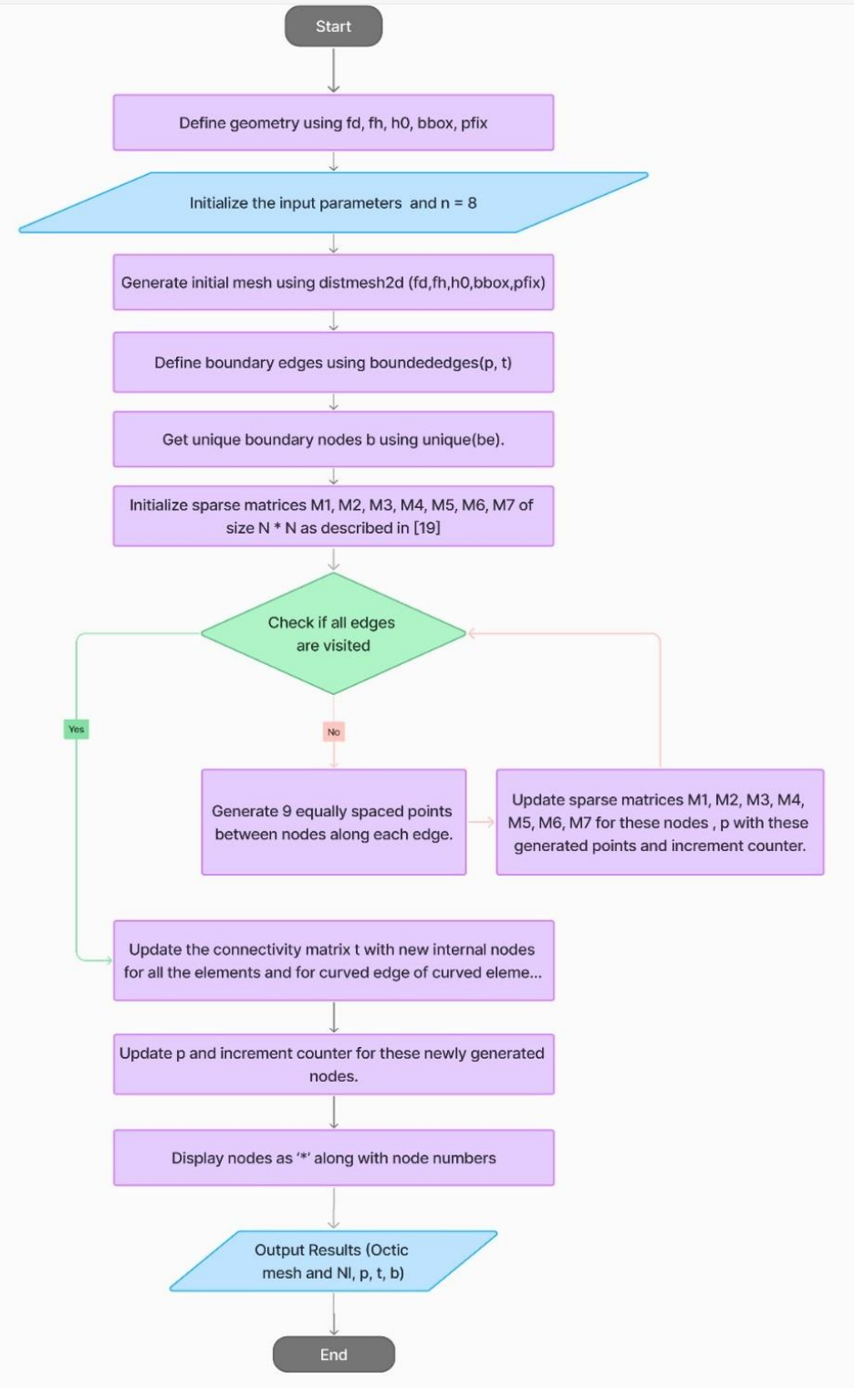

Fig. 2: Flow chart of automated octic order meshing.

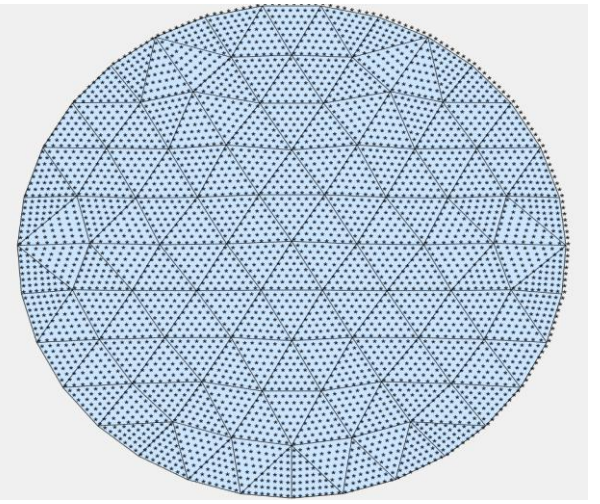

Fig. 3: Uniform octic mesh in a circular domain ($h_0 = 0.2$).

Fig. 4: Uniform Octic mesh for square domain ($h_0 = 2$).

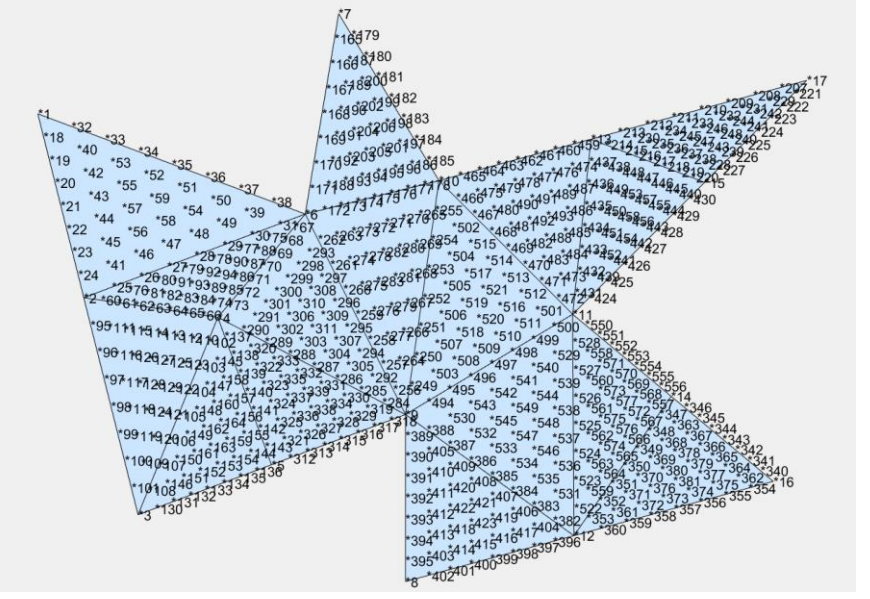

Fig. 5: Uniform Octic mesh for irregular polygon domain ($h_0 = 0.5$).

2.2.1 Uniform meshing for regular square domain

Beginning with a regular square domain that contains four equally sized straight edges can provide an easy and clear understanding of the mesh generation technique proposed. Initially, the mesh is generated by altering the signed distance function (fd) to the distance function of the rectangle (drectangle()) and later fixing the boundary points (pfix) to obtain a square domain based on the definitions given in Ref. [16, 26]. The huge benefit of using this scheme is that the mesh sizes can be modified according to the requirements by changing the value of the term h0. An octic order mesh with h_0 set as 2 units is automated for a square domain, as shown in Fig. 4. The following lines of code explain the generated mesh:

```
fd = @(x) drectangle(x, -2, 2, -2, 2); p_fix = [-2, -2; -2, 2; 2, -2; 2, 2];
h0 = 2; bbox = [-2, -2; 2, 2];
[Nl, p, t, b] = OcticMesh2d(fd, @huniform, h0, bbox, p_fix);
```

2.2.2 Uniform meshing for irregular star shaped polygon

The irregular domain, as the name itself explains, represents geometries with uneven or unbalanced boundaries. The boundaries in this domain are straight edges. The geometry of this irregular polygon is described by assigning a distance function to (dpoly).

The dpoly calculates the signed distance from any point in the 2D space to the closest edge of the polygon defined by the vertices provided. The discretized mesh with specified node numberings for this star-shaped irregular polygon with octic order triangular elements can be seen in Fig. 5. The following two lines of code can generate an initial triangular mesh for an irregular star-shaped polygon:

```
pv = [-0.4 -0.5; 0.4 -0.2; 0.4 -0.7; 1.5 -0.4; 0.9 0.1; 1.6 0.8; 0.5 0.5; 0.2 1; 0.1 0.4; -0.7 0.7; -0.4 -0.5];
h0 = 0.5; fd = @(p) dpoly(p, pv); bbox = [-1,-1; 2,1];
[Nl, p, t, b] = OcticMesh2d(fd, @huniform, h0, bbox, pv);
```

2.2.3 Uniform meshing for a square domain with a hexagon duct

A hexagonal duct inscribed on a square has straight side edges both internally and externally. The discretization of this geometry can be done by defining the distance functions of both polygons, and the overall signed distance function of the domain can be obtained by the difference (diff()) of these polygons. It then generates an initial mesh with specified edge lengths and then discretizes into uniform octic order triangular elements with each dot depicting a node as shown in Fig. 6. The code begins by defining the geometric constraints of the domain and the internal hole using the distance as given below:

```
fd_sq = @(p) drectangle(p, -2, 2, -2, 2);
fd_hex = @(p) dpoly(p, [cos((0:5)'*2*pi/6), sin((0:5)'*2*pi/6)]);
fd = @(p) ddiff(fd_sq(p), fd_hex(p));sq_corners = [-2, -2; -2, 2; 2, -2; 2, 2];
phi = (0:5)' * 2 * pi / 6; hex_points = [cos(phi), sin(phi)];
pfix = [sq_corners; hex_points]; h0 = 0.7; bbox = [-2, -2; 2, 2];
[Nl, p, t, b]=OcticMesh2d(fd, @huniform, h0, bbox, pfix);
```

2.2.4 Uniform meshing for hexagon with rotated hexagon duct

Fig. 7 shows an excellent, high-quality structured mesh within a domain defined by an outer hexagon and an inner hexagon,utilizing fixed points and a specified mesh density. It

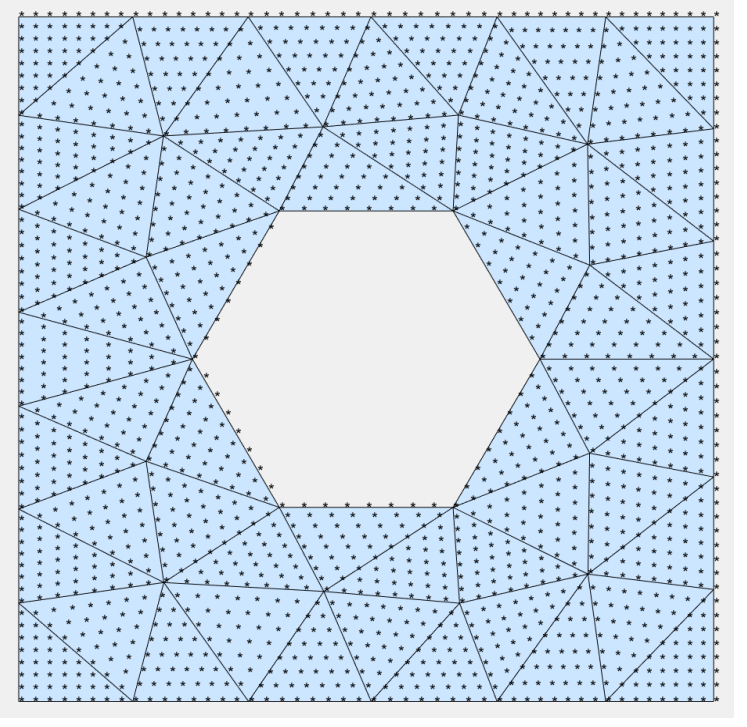

Fig. 6: Uniform Octic order triangular mesh in a square domain with a hexagon duct ($h_0 = 0.7$).

begins by defining distance functions for the outer and inner hexagons, combining these using set operations to define the domain boundaries. The inner hexagon is rotated by rotating the internal hexagon point to 30 degrees with the help of the function protate() defined in Ref. [16]. The following code explains the formation of this domain:

```
dhexagon = @(p, r) dpoly(p, r * [cos((0:5)'*2*pi/6), sin((0:5)'*2*pi/6)]);
out_hex_r = 2; in_hex_r = 1;
fd_outer = @(p) dhexagon(p, out_hex_r); fd_inner = @(p) dhexagon(p, in_hex_r);
fd = @(p) ddiff(fd_outer(p), fd_inner(p));
phi_out = (0:5)' * 2 * pi / 6; phi_in = (0:5)' * 2 * pi / 6;
out_hex_points = out_hex_r * [cos(phi_out), sin(phi_out)];
in_hex_points = in_hex_r * [cos(phi_in), sin(phi_in)];
rotation_angle = pi / 6; in_hex_points = protate(in_hex_points, rotation_angle);
pfix = [out_hex_points; in_hex_points]; h0 = 0.5;
bbox = [-out_hex_r, -out_hex_r; out_hex_r, out_hex_r];
[Nl, p, t, b] = OcticMesh2d(fd, @huniform, h0, bbox, pfix);
```

2.2.5 Non-uniform meshing for rectangle with circular duct

One common shape in engineering is a rectangular plate with a circular cutout, often used in fluid flow analysis, mechanics, aerospace engineering, and waveguide problems. A circle inscribed inside a rectangle has a central circular hole with curved boundaries and straight-sided boundaries on the outside.

In this research, the code provided is customized to automate higher-order octic triangular meshes. Fig. 8 shows the non-uniform mesh produced in a domain where a circle is inscribed inside a rectangle, consisting of discretized unstructured octic triangular elements of the octic order ($n = 8$) with an edge length of $h_0 = 0.3$. In a non-uniform meshing scheme, the mesh elements are of different sizes. The automated triangular mesh elements are unstructured to ensure that the generated geometry is accurate. OcticMesh2d can be utilized with a minor change to the input arguments, as shown:

```
fd_rect = @(p) drectangle(p, -3, 3, -2, 2); fd_circle = @(p)sqrt(sum(p.^2, 2))-1;
fd = @(p) ddiff(fd_rect(p), fd_circle(p)); pfix = [-3, -2; -3, 2; 3, -2; 3, 2];
fh = @(p) 0.1 + 0.5 * sqrt(sum(p.^2, 2)); h0 = 0.3; bbox = [-3, -2; 3, 2];
[Nl, p, t, b] = OcticMesh2d(fd, fh, h0, bbox, pfix);
```

2.2.6 Non-uniform meshing for NACA0012 airfoil

The cross-sectional profile of a turbine, rotor, propeller blade, or wing is called an airfoil. Before the National Advisory Committee for Aeronautics (NACA) established a methodical approach to airfoil design, airfoil designs were purely arbitrary. Finally, new designs of airfoils were produced as a result of their expertise.[27] When applied to an airfoil for 2D analysis, the octic order meshing technique described in Algorithm 1 can improve several aerodynamic parameters that are important for aerospace applications. In Ref. [28], MATLAB code is implemented to discretize the region surrounding the airfoil design utilizing higher-order triangular elements up to sextic order. Numerous aerospace applications, including pressure gradient computation, atmospheric conditions research, and laminar viscous compressible flow evaluation around airfoil designs, can benefit from the establishedmeshing method.[29-31] Fig. 9 shows a comprehensive view of the discretized airfoil design of NACA0012. The airfoil is surrounded by a circle, and then the

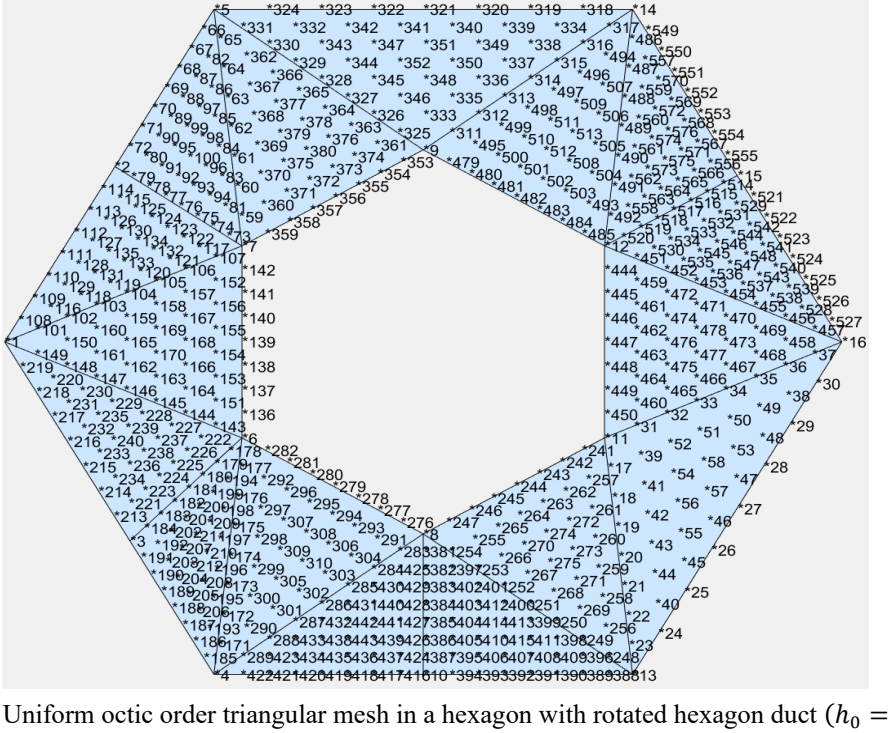

Fig. 7: Uniform octic order triangular mesh in a hexagon with rotated hexagon duct ($h_0 = 0.5$).

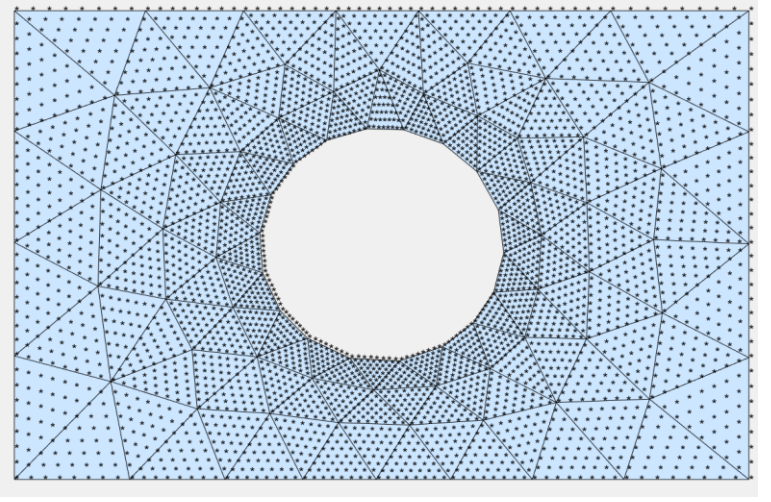

Fig. 8: Non-uniform octic order triangular mesh in a rectangle with a circular duct ($h_0 = 0.3$).

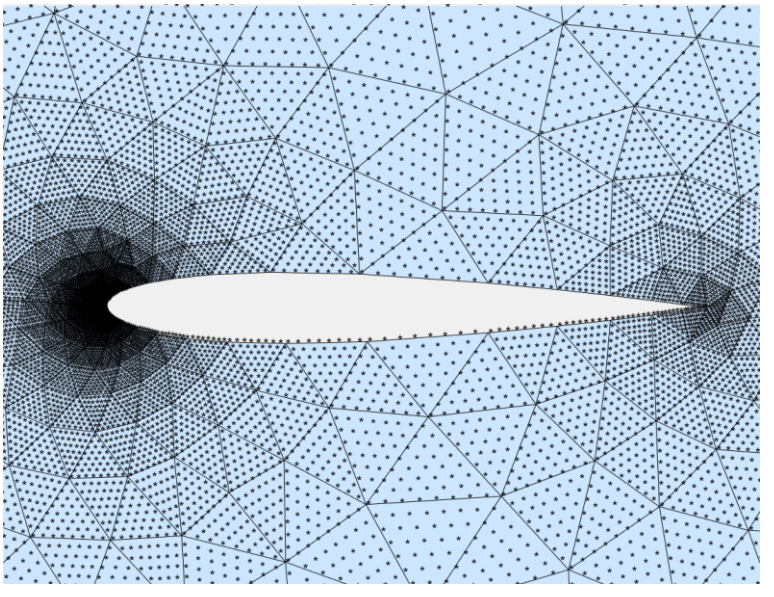

Fig. 9: Non-uniform octic order triangular mesh in NACA0012.

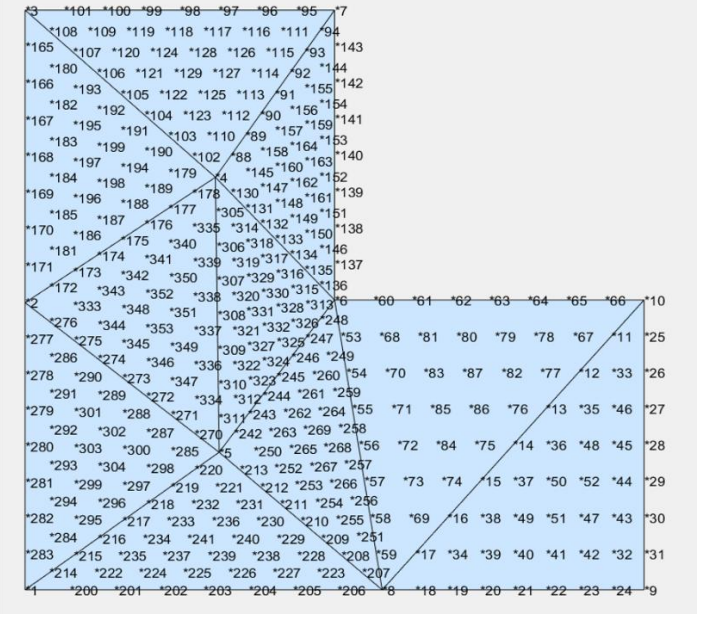

Fig. 10: Octic order triangular mesh in a L-shaped waveguide structure.

airfoil design area is discretized utilizing octic order elements using a bespoke algorithm, as explained below:

```
h_lead = 0.01; h_trail = 0.04; h_max = 2; cx = 2; cr = 4;
coeff = .12/.2 * [0.2969, -0.1260, -0.3516, 0.2843, -0.1036];
fd = @(p) ddiff(dcircle(p, cx, 0, cr), (abs(p(:,2)) - ...
   polyval([coeff(5:-1:2), 0], p(:,1))).^2 - coeff(1)^2 * p(:,1));
fh = @(p) min(min(h_lead + 0.3 * dcircle(p, 0, 0, 0), h_trail + 0.3
* dcircle(p, 1, 0, 0)), h_max);
fx = 1 - h_trail * cumsum(1.3 .^ (0:4)');
fy = coeff(1) * sqrt(fx) + polyval([coeff(5:-1:2), 0], fx);
fix_pts = [[cx + [-1, 1, 0, 0] * cr; 0, 0, cr * [-1, 1]]'; 0, 0; 1, 0; fx,
fy; fx, -fy];
bbox = [cx - cr, -cr; cx + cr, cr];
h0 = min([h_lead, h_trail, h_max]);
[Nl, p, t, b] = OcticMesh2d(fd, fh, h0, bbox, fix_pts);
```

3. Octic order finite element method implementation for engineering applications

FEM is a computational technique for approximating solutions to differential and integral equations. FEM predicts how a product will react to factors like external forces, temperature changes, vibrations, fluid dynamics, and other physical impacts. It is widely used in consumer goods, automotive, industrial machinery, electronics, aviation, bio-medical applications, marine industries, *etc*. Using FEM is advantageous in the quest for providing solutions to sustainable energy-efficient and structural designs.[32-34] Consequently, FEM will significantly impact a variety of engineering applications to offer sustainable solutions.[35, 36] By simulating and analysing design performance before production, FEA helps reduce expensive mistakes while saving engineers and designers valuable time and resources. Higher-order FEA provides accurate and effective solutions to most of these applications in comparison to the existing popular software like COMSOL, ABAQUS, ANSYS, NASTRAN, DIANA, PAFEC, *etc*., which use mostly up to quadratic elements to solve these problems. The following examples illustrate the effectiveness of the proposed octic order meshes and FEM solution in comparison with the lower-order elements. In this section, the effectiveness of the proposed meshing scheme is demonstrated by solving Maxwell's equations and Poisson's equation using octic order FEM.

3.1 Octic order FEM implementation for 2D Helmholtz equation

The FEM is utilized in solving and analyzing a variety of sets of PDEs including Maxwell's equations governing electromagnetic (EM) radiation. EM waves can be guided and transferred with minimal energy loss using hollow metal structures called waveguides.[31] As a result, it is discovered to be extremely important for making sustainable energy-efficient optical devices and helpful in a variety of electrical applications.[32] Apart from these applications, the generators, transformers, and motors transfer mechanical energy to electrical energy and vice versa via electromagnetic processes, which are essential for everyday activities. These physical problems can be mathematically formulated as PDEs, requiring analytical or numerical solutions. Predicting electromagnetic system behavior needs accurate solutions. Thus, solving eigenvalue problems is vital in this field. A considerable demand for efficient numerical techniques exists. Therefore, there is a high requirement to get accurate solutions to many electromagnetic research problems like single-phase permanent magnet generators, acoustics, switched reluctance motor optimization, performance annular thermoelectric couple analysis, micropower generators, solar thermoelectric, and piezoelectric energy harvesting. Here, the developed octic order meshes are utilized for FEM implementation to solve an EM problem.

Using the FEM, first, the domain discretization of the Helmholtz equation for EM wave propagation in waveguides using the proposed octic meshing approach is performed. The numerical results are presented in Ref. [29]. Waveguides are categorized as L-shaped, ridge-shaped, rectangular, coaxial line, circular, *etc*., depending on the types of waves and the geometric configurations of the structures. Therefore, in microwave applications, where strong computational techniques are required, the waveguide cutoff frequencies must be determined. Computing EM is therefore essential for obtaining effective approximations of these values. Various methods have been used to obtain cutoff wavenumbers of the transverse electric (TE) and transverse magnetic (TM) modes in previous studies in waveguides with arbitrary cross-sections.[10,30,31] The Helmholtz equation for any waveguide is given by Eq. (5):

$$-\Delta^2 (u) + k_c^2 u = 0 \tag{5}$$

where k^2 is the unknown cutoff frequency and u is the wave amplitude. For TM modes, the wave amplitude becomes zero at the boundary, whereas for TE modes, the normal derivative equals zero.

To solve the Helmholtz equation using the octic order FEM scheme, the following steps are involved:

Step 1: Generate an octic order mesh for the waveguide using OcticMesh2D.

Step 2: Express the waveguide domain element geometry in terms of Lagrange shape functions using the Galerkin weighted residual, finite element technique, as described by Eqs. (6)-(11):[32]

$$[K + G]_{45\times45} \cdot \{U\}_{45\times1} = \{0\}_{45\times1} \tag{6}$$

where U is the displacement vector, G and K, stiffness matrices are expressed as:

$$K_{q,r} = \iint_{\Omega_e} \left(\frac{\partial N_q}{\partial x}\frac{\partial N_r}{\partial x} + \frac{\partial N_q}{\partial y}\frac{\partial N_r}{\partial y}\right) dxdy = K_{x,x}^{q,r} + K_{y,y}^{q,r} \tag{7}$$

where N_q and N_r are the Lagrange shape functions, and r is the region of one element, and further, for $(q, r =$

$1, 2, 3, \ldots, 45$), $K_{x,x}^{q,r}$ and $K_{y,y}^{q,r}$ are calculated as follows:

$$K_{x,x}^{q,r} = \int_0^1 \int_0^{1-\epsilon} \frac{1}{J}\left(\frac{\partial N_q}{\partial \epsilon}\frac{\partial x}{\partial \mu} - \frac{\partial N_q}{\partial \mu}\frac{\partial x}{\partial \epsilon}\right)\left(\frac{\partial N_r}{\partial \epsilon}\frac{\partial x}{\partial \mu} - \frac{\partial N_r}{\partial \mu}\frac{\partial x}{\partial \epsilon}\right) d\mu d\epsilon \quad (8)$$

$$K_{y,y}^{q,r} = \int_0^1 \int_0^{1-\epsilon} \frac{1}{J}\left(-\frac{\partial N_q}{\partial \epsilon}\frac{\partial y}{\partial \mu} + \frac{\partial N_q}{\partial \mu}\frac{\partial y}{\partial \epsilon}\right)\left(-\frac{\partial N_r}{\partial \epsilon}\frac{\partial y}{\partial \mu} + \frac{\partial N_r}{\partial \mu}\frac{\partial y}{\partial \epsilon}\right) d\mu d\epsilon \quad (9)$$

where the Jacobian J is given by:

$$J = \frac{\partial(x,y)}{\partial(\epsilon,\mu)} = \frac{\partial x}{\partial \epsilon}\frac{\partial y}{\partial \mu} - \frac{\partial x}{\partial \mu}\frac{\partial y}{\partial \epsilon} \quad (10)$$

The expression for $G_{q,r}$ is given as:

$$G_{q,r} = \int_{\Omega_e} k_c^2 N_q N_r dx dy \quad (11)$$

Step 3: Use Gauss quadrature rules on triangles to calculate each element's matrix components (K & G).

Step 4: Assemble each triangular element to utilize the effect of all octic order finite mesh elements with respect to the global node numberings to construct a global matrix equation as shown by Eq. (12):

$$[K + G]_{N_l \times N_l} \times \{U\}_{N_l \times 1} = \{0\}_{N_l \times 1} \quad (12)$$

Step 5: Eq. (5) is reduced to an eigenvalue problem by computing k_c as $\sqrt{eigenvalue}$, where k_c is the wavenumber. The obtained k_c is referred to as TE modes.

Step 6: Apply the boundary conditions.

Step 7: After the boundary conditions are applied, the modified Helmholtz equation reduces to an eigenvalue problem in N_w algebraic equations by Eq. (13).

$$[K + G]_{N_w \times N_w} \times \{U\}_{N_w \times 1} = \{0\}_{N_w \times 1} \quad (13)$$

Step 8: To determine the TM modes, compute the eigenvalues. The lowest wavenumber that is obtained represents the cutoff wavenumber.

To show the effectiveness of the suggested technique, the first four cutoff wave numbers for an L-shaped waveguide construction are computed.[32] The first and crucial step in this process is to generate a mesh for the given 2D domain. Higher precision in solution approximations can be achieved by using a FEM that employs triangular elements of higher-order, as opposed to elements of lower order such as linear, quadratic, or cubic, as implemented in previous work.[32] The proposed octic meshing technique is used to discretize the mesh, setting the foundation for the subsequent computations. Fig. 10 displays the triangular mesh of octic order for the L-shaped waveguide. The 2D Helmholtz equation is formulated using an octic order finite element scheme. The formulation of the equation for elements of lower order is clearly explained in Ref. [32]. The proposed mesh generator is implemented for the octic order FEM by solving the Helmholtz equation as described in Fig. 11 and hence validated by comparing it with the published results.[32] The data in Table 1 show the cutoff wave numbers computed for the TE and TM modes within the L-shaped waveguide, demonstrating the performance of the proposed scheme.

3.2 Octic order FEM implementation for 2D Poisson equation

Poisson's equation is a key model for elliptic PDEs, which appears in engineering problems like magnetic fields, elastic membranes, and complex scenarios such as the Navier-Stokes equation,[33] This second-order partial differential equation is essential in applied mathematics, characterizing potential fields in space, with applications in astronomy, heat transfer, fluid dynamics, electromagnetism, electrostatics, gravitation, aviation industry, biomedical fields and thermal analysis. Therefore, it is crucial to employ efficient methods for solving such equations in real-world scenarios. To demonstrate the efficacy of the suggested technique, a Poisson equation is solved over a square domain. Consequently, the proposed

Table 1: Comparing the cutoff frequencies of TM and TE modes over a L-shaped domain.

Polynomial order	Mesh multi domain DQ [29] n, h_0	Linear [32] (n = 1) (h_0 = 0.02)	Quadratic [32] (n = 2) (h_0 = 0.05)	Cubic [32] (n = 3) (h_0 = 0.1)	Octic (n = 8) (h_0 = 0.5)
	1.9123	1.9148	1.9141	1.9141	1.9139
Cutoff	2.9605	2.9608	2.9605	2.9605	2.9605
Wavenumbers of TE modes	4.9474	4.9489	4.9474	4.9474	4.9474
	4.9474	4.9490	4.9474	4.9474	4.9474
	4.8902	4.8952	4.8921	4.8922	4.8918
Cutoff wavenumbers	6.1350	6.1421	6.1392	6.1392	6.1392
of TM modes	6.9921	7.0011	6.9967	6.9967	6.9967
	8.5565	8.5645	8.5566	8.5566	8.5565
Degrees of freedom		3559	2228	1243	353
Number of elements		6843	1059	258	10

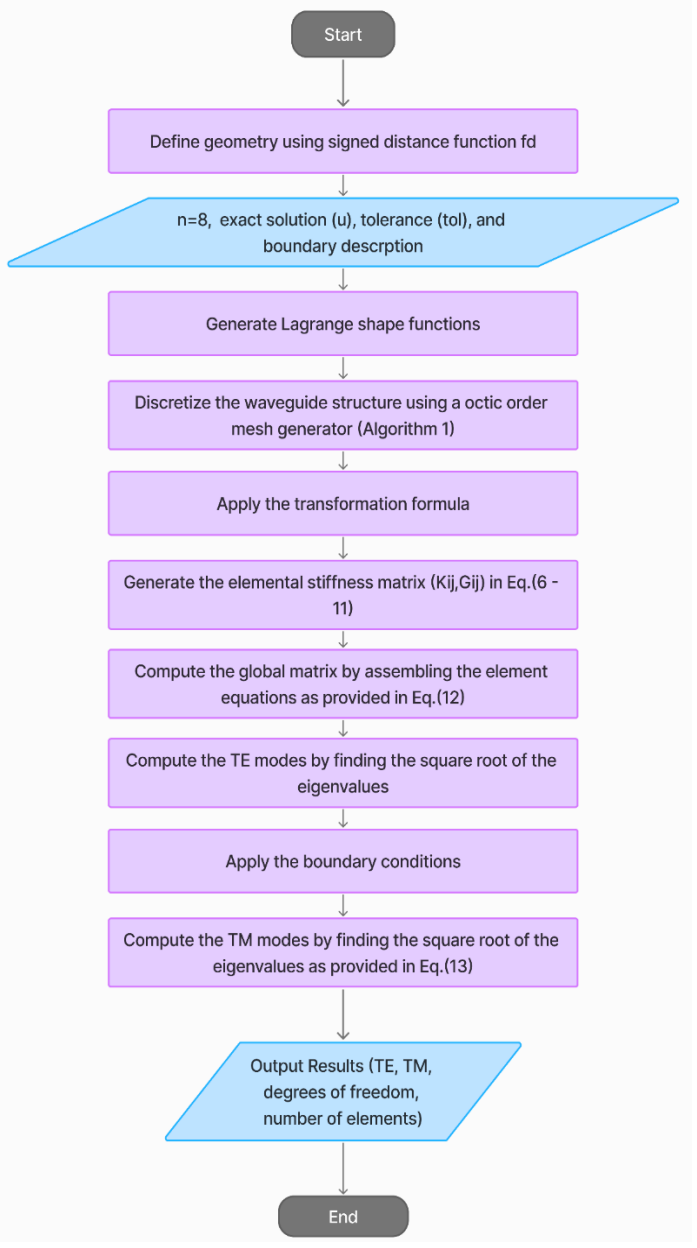

Fig. 11: Octic order FEM for L-shaped waveguide structure.

octic order FEM is employed to address an elliptic problem and compare the outcome with the exact solution. Themethodology to solve any PDE with the octic order FEM scheme is similar to the procedure explained in subsection 3.1. Finite elements necessitate a mesh for the two-dimensional region of arbitrary configuration. To do this, we utilize the automated octic order mesh generating MATLAB code built as detailed in section 2.2 and the resulting mesh is shown in Fig. 12. The subsequent Poisson equation within a two-dimensional square domain is addressed with the FEM in MATLAB, incorporating linear, quadratic, cubic, and octic order triangular elements.

$$-\nabla^2(v) = s\,sin(rs) + r^2 sin(rs) \quad (14)$$

With the boundary conditions $v = 0$ on $r = 0$ and $s = 0$,

$$v(2,s) = \sin(2s)\,; u(r,2) = \sin(2r) \quad (15)$$

Over the square $[0,0] \times [2,2]$, and the analytical solution of Eqs. (14) and (15) are documented in the literature as $v(r,s) = sin(rs)$.[33]

The resulting solutions are evaluated against the exact solution by calculating the L_2 Norm. The L_2 Norm is computed as, $L_2 = (\int\int (V - v)^2\, dA)^{\frac{1}{2}}$, where V is the FEM solution and v is the analytical solution. The outcomes of this problem are presented in Table 2. Table 2 demonstrates that octic order elements yield the most precise approximation to the exact solution in terms of the L_2 Norm.

4. Results and discussion

The effectiveness of the proposed meshing scheme can be verified by finding details of the output data allowing comparison of mesh characteristics and computational requirements across different geometries and mesh configurations, which is useful for optimizing numerical simulations by FEA. For each domain, different mesh sizes and orders of elements are considered, showing how these parameters affect the number of nodes, elements, and computational time.

In this article, a meshing scheme for two-dimensional geometries using uniform and nonuniform octic order triangular elements is implemented in MATLAB using the external distmesh library. The output data obtained from this octic meshing approach includes information about nodal connectivity, nodal placements, border nodes, and edges for all elements in the domain. The advanced meshing approach provided in this article is evaluated for various domains.

Table 1 shows that triangular elements of octic order perform better in electromagnetic simulations than triangular elements of lower order. Clearly, the automated octic order meshing method significantly reduces computational time, degrees of freedom, and element count compared to quadratic or linear meshes. The suggested octic order FEM provides an optimal and intuitive approach to calculate cutoff wavenumbers. It also offers an excellent method for determining the TE and TM modes in any waveguide configuration. This technique is quite accurate for configuration. This technique is quite accurate forcomputing the cutoff wave numbers of the TM and TE modes in the L-shaped waveguide, as shown in Table 1.

Octic order meshing gives an excellent balance between the number of terms in the polynomials and computational time. It reduces the number of elements by 99.85% from linear to octic (6843 to 10 elements), 99.06% from quadratic to octic (1059 to 10 elements), and 96.12% from cubic to octic (258 to 10 elements) and is therefore computationally less expensive while producing very accurate results. Moreover, the degrees of freedom involved, which directly impact computational cost, are also drastically reduced by 90.08% from linear (3559) to octic (353), by 84.14% from quadratic (2228) to octic, and by 71.60% from cubic (1243) to octic. The large reduction in elements and degrees of freedom with the octic order points to its potential to provide accurate cutoff frequencies with much

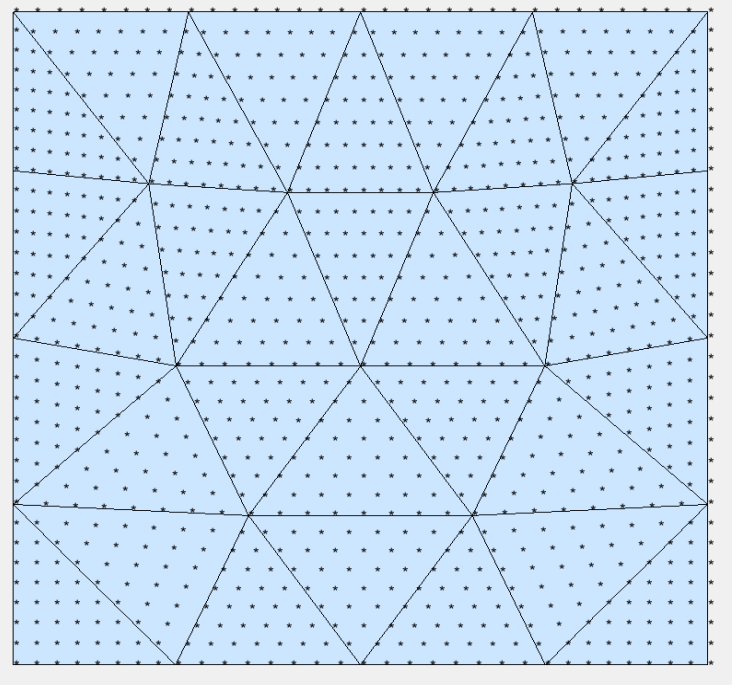

Fig. 12: Octic order FEM for a square domain.

Table 2: Lower-orders and octic order FEM solution comparison of the Poisson equation over a square domain.

Polynomial order	Linear (n = 1) (h_0 =0.5)	Quadratic (n = 2) (h_0 =0.5)	Cubic (n = 3) (h_0 =0.5)	Octic (n = 8) (h_0 =0.5)
L_2 Norm	0.021	0.0054	9.118e-04	4.028e-06
Degrees of freedom	25	81	169	1089
Number of elements	32	32	32	32

smaller computational resources, resulting in low energy loss. Also, from Table 2, it is very apparent that the octic order scheme delivers the most accurate approximation to the exact solution for the two-dimensional Poisson equation with respect to the L_2 Norm. Therefore, octic order is highly desired in applications and real situations where a high degree of accuracy is required. Thus, it should be one of the best choices for finite element techniques. As a result, it is one of the effective ways to solve many engineering and science applications.

5. Conclusion

Finite element meshes are very crucial to providing structurally optimized numerical solutions. This octic order meshing technique is useful in enormous engineering applications where complex geometries with internal voids need to be meshed for FEA, such as in the development of energy-efficient electromagnetic devices, structural engineering for the construction of energy-efficient and sustainable buildings, aerospace design, and computational fluid dynamics. Engineers and scientists can perform detailed simulations to study stress distributions, heat transfer, or fluid flow around complex structures by accurately meshing two-dimensional domains of arbitrary geometry. The majority of engineering applications have sharp edges or curved geometries, which make current techniques computationally difficult to handle. It is visible from the tabulated results that the octic order mesh elements provide better results than the lower order elements. Meshing such geometries into high-quality octic order triangular elements can result in a refined and accurate output. Thus, future research will focus on further developing this octic order meshing method, specifically for curved domains, to enhance the quality and accuracy of the output. This refinement will provide even more detailed and precise simulations, providing engineers and scientists with advanced tools.

Acknowledgements

The authors would like to express their profound gratitude to the management of Amrita Vishwa Vidyapeetham for their invaluable support and encouragement towards this research.

Conflict of Interest

There is no conflict of interest.

Supporting Information

Not applicable.

References

[1] V. Papadopoulos, D. G. Giovanis, Stochastic finite element method, Stochastic Finite Element Analysis: An Introduction, Springer Cham, New York, 2018, 47-70, ISBN: 9783319645285.

[2] A. Bawin, A. Garon, J. F. Remacle, Optimally convergent isoparametric P^2 mesh generation, SIAM International Meshing Roundtable Workshop 2023, Amsterdam, Netherlands, March 6-9, 2023, Cham: Springer Nature Switzerland, 2024, 373-395, doi: 10.1007/978-3-031-40594-5_17.

[3] N. Lei, Z. Li, Z. Xu, Y. Li, X. Gu, What's the situation with intelligent mesh generation: a survey and perspectives, *IEEE Transactions on Visualization and Computer Graphics*, 2023, **30**, 4997-5017, doi: 10.1109/tvcg.2023.3281781.

[4] B. G. Pantoja-Rosero, R. Achanta, K. Beyer, Automated image-based generation of finite element models for masonry buildings, *Bulletin of Earthquake Engineering*, 2024, **22**, 3441-3469, doi: 10.1007/s10518-023-01726-7.

[5] N. R. Secco, G. K. W. Kenway, P. He, C. Mader, J. R. R. A. Martins, Efficient mesh generation and deformation for aerodynamic shape optimization, *AIAA Journal*, 2021, **59**, 1151-1168, doi: 10.2514/1.j059491.

[6] M. Abobaker, S. Addeep, L. O. Afolabi, A. M. Elfaghi, Effect of mesh type on numerical computation of aerodynamic coefficients of NACA 0012 airfoil, *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences*, 2021, **87**, 31-39, doi: 10.37934/arfmts.87.3.3139.

[7] R. Phellan, B. Hachem, J. Clin, J.-M. Mac-Thiong, L. Duong, Real-time biomechanics using the finite element method and machine learning: Review and perspective, *Medical Physics*, 2021, **48**, 7-18, doi: 10.1002/mp.14602.

[8] Z. Liu, H. Liu, Y. Chen, W. Zhang, W. Song, L. Zhou, Q. Wei, J. Xu, Evaluating airfoil mesh quality with transformer, *Aerospace*, 2023, **10**, 110, doi: 10.3390/aerospace10020110.

[9] V. Narayanan, K. Wu, C. Yuksel, J. McCann, Visual knitting machine programming, *ACM Transactions on Graphics*, 2019, **38**, 1-13, doi: 10.1145/3306346.3322995.

[10] S. I. Piltyay, A. V. Bulashenko, Y. I. Kalinichenko, Parametric optimization of waveguide polarizer by equivalent network and fem models, *Telecommunications and Radio Engineering*, 2021, **80**, 49-74, doi: 10.1615/telecomradeng.2021037160.

[11] J. Tu, G. H.Yeoh, C. Liu, CFD mesh generation: A practical guideline, Computational Fluid Dynamics, Butterworth-Heinemann, 2018, 125-154, ISBN: 9780081011270.

[12] T. Li, Y. Zou, S. Zou, X. Chang, L. Zhang, X. Deng, A fully differentiable gnn-based pde solver: With applications to poisson and navier-stokes equations, arxiv preprint arxiv: 2405.04466, 2024, doi: 10.48550/arXiv.2405.04466.

[13] X. Ma, N. Duan, W. Xu, S. Wang, Multi-scale finite element method applied in 3D nonlinear problem, 2024 IEEE 21st Biennial Conference on Electromagnetic Field Computation, June 2-5, Jeju, Korea, IEEE, 2024, 1-2, doi: 10.1109/CEFC61729.2024.10585808.

[14] W. Chang, C. Kuo, W. Lin, M. Hsieh, Simulation of stray and core shielding loss for power transformer based on 2D/3D FEM, *IEEE Access*, 2023, **11**, 16943-16950, doi: 10.1109/ACCESS.2023.3242975.

[15] P. O. Persson, G. Strang, A simple mesh generator in MATLAB, *SIAM Review*, 2004, **46**, 329-345, doi: 10.1137/s0036144503429121.

[16] P. O. Persson, Mesh generation for implicit geometries, Ph.D. dissertation, Massachusetts Institute of Technology, 2005.

[17] J. Koko, A Matlab mesh generator for the two-dimensional finite element method, *Applied Mathematics and Computation*, 2015, **250**, 650-664, doi: 10.1016/j.amc.2014.11.009.

[18] S. A. Funken, A. Schmidt, Adaptive Mesh Refinement in 2D - An Efficient Implementation in Matlab, *Computational Methods in Applied Mathematics*, 2020, **20**, 459-479, doi: 10.1515/cmam-2018-0220.

[19] T. V. Smitha, K. V. Nagaraja, S. Jayan, MATLAB 2D higher-order triangle mesh generator with finite element applications using subparametric transformations, *Advances in Engineering Software*, 2018, **115**, 327-356, doi: 10.1016/j.advengsoft.2017.10.012.

[20] Y. Kang, E. J. Kubatko, An automatic mesh generator for coupled 1D-2D hydrodynamic models, *Geoscientific Model Development*, 2024, **17**, 1603-1625, doi: 10.5194/gmd-17-1603-2024.

[21] S. Soman, N. Mehendale, Faster and efficient tetrahedral mesh generation using generator neural networks for 2D and 3D geometries, *Neural Computing and Applications*, 2024, **36**, 1805-1813, doi: 10.1007/s00521-023-09119-2.

[22] J. Sasikala, K. Naidu V., B. Venkatesh, S. M. Mallikarjunaiah, On an efficient octic order sub-parametric finite element method on curved domains, *Computers & Mathematics with Applications*, 2023, **143**, 249-268, doi: 10.1016/j.camwa.2023.05.006.

[23] S. Riasat, M. R. Ali, S. Kanwal, A. S. Hendy, Assessment of temperature distribution on inclined porous rod with a convective and insulated tip, *Case Studies in Thermal Engineering*, 2024, **60**, 104786, doi: 10.1016/j.csite.2024.104786.

[24] K. Chandan, R. Saadeh, A. Qazza, K. Karthik, R. S. Varun Kumar, R. N. Kumar, U. Khan, A. Masmoudi, M. M. M. Abdou, W. Ojok, R. Kumar, Predicting the thermal distribution in a convective wavy fin using a novel training physics-informed neural network method, *Scientific Reports*, 2024, **14**, 7045, doi: 10.1038/s41598-024-57772-x.

[25] C. Kumar, P. Srilatha, K. Karthik, C. Somashekar, K. V. Nagaraja, R. S. Varun Kumar, N. Ali Shah, A physics-informed machine learning prediction for thermal analysis in a convective-radiative concave fin with periodic boundary conditions, *ZAMM - Journal of Applied Mathematics and Mechanics*, 2024, **104**, e202300712, doi:

10.1002/zamm.202300712.
[26] T. V. Smitha, T. Aishwarya, V. Nataraj, A simple and flexible higher-order 2D triangle mesh generator with MATLAB, *AIP Conference Proceedings*, 2023, **2764**, 10, doi: 10.1063/5.0144291.
[27] J. Julian, W. Iskandar, F. Wahyuni, Effect of mesh shape and turbulence model on aerodynamic performance at naca 4415, *Journal of Applied Fluid Mechanics*, 2023, **16**, 2504-2517, 2023. doi: 10.47176/jafm.16.12.1983.
[28] D. Supriya, K. V. Nagaraja, T. V. Smitha, S. Jayan, Accurate higher order automated unstructured triangular meshes for airfoil designs in aerospace applications using parabolic arcs, *Aerospace Science and Technology*, 2019, **88**, 405-420, doi: 10.1016/j.ast.2019.03.034.
[29] S. Chang, Differential quadrature and its application in engineering, Springer-Verlag London, Berlin Heidelberg, 2000, ISBN: 978-1-4471-1132-0.
[30] T. Yan, S. Hu, L. Ye, Y. Shi, X. Meng, Analysis of the cutoff wavenumber for arbitrarily shaped waveguides using method of dipole representations with an excitation source, *IEEE Transactions on Antennas and Propagation*, 2024, **72**, 3848-3853, doi: 10.1109/TAP.2024.3374551.
[31] E. Vinogradova, P. Smith, Y. Shestopalov, High-precision calculation using the method of analytical regularization for the cut-off wave numbers for waveguides of arbitrary cross sections with inner conductors, *Applied Sciences*, 2024, **14**, 2265, doi: 10.3390/app14062265.
[32] T. V. Smitha, K. V. Nagaraja, Application of automated cubic-order mesh generation for efficient energy transfer using parabolic arcs for microwave problems, *Energy*, 2019, **168**, 1104-1118, doi: 10.1016/j.energy.2018.11.138.
[33] J. Coady, Finite element experiments in MATLAB, 2012.
[34] F. Tahmasebinia, A. A. Jabbari, K. Skrzypkowski, The application of finite element simulation and 3D printing in structural design within construction industry 4.0, *Applied Sciences*, 2023, **13**, 3929, doi: 10.3390/app13063929.
[35] G. Sidhu, S. Srinivasan, Integration of ethics, sustainability, and social responsibility components in an undergraduate engineering course on finite element analysis, *International Journal of Engineering Education*, 2022, **38**, 656-662.
[36] A. Y. Patil, T. Kundu, R. Kumar, Finite element analysis megatrends: A road less traveled, *Computer Applications in Engineering Education*, 2024, **32**, e22721, doi: 10.1002/cae.22721.