Finite Element Mesh Refinement for EM-Field Solvers Modeling using Genetic Algorithm

Kiptoo Lelon
Dedan Kimathi University of Technology
Private Bag – 10143, Dedan Kimathi Nyeri, Kenya

Waweru Njeri
Dedan Kimathi University of Technology
Private Bag – 10143, Dedan Kimathi Nyeri, Kenya

Joseph Muguro
Dedan Kimathi University of Technology
Private Bag – 10143, Dedan Kimathi Nyeri, Kenya

ABSTRACT
Electromagnetic field solvers are pivotal in numerous engineering and scientific applications, ranging from antenna design to medical imaging. Achieving accurate and efficient solutions to complex electromagnetic (EM) field problems often necessitates a refined finite element (FE) mesh. This study explored using genetic algorithms (GA) as a robust optimization tool to enhance the quality of FE meshes for EM field simulations. The proposed approach not only automates the mesh refinement process but also significantly improves the convergence and accuracy of numerical simulations. The initial mesh on a problem domain was generated using the Delaunay triangulation algorithm (DTA), and the developed mesh was then refined using a more flexible GA that could handle regions of the problem domain containing several local extrema. The aspect ratio and the maximum angle at each node of the triangular mesh were used to select the fitness function to be optimized in the GA. The GA was tested and validated for various test cases covering multiple complex geometry applications. The results showed a significant change in the quality of the refined meshes, a shift of fitness value ranges from (0.1-0.50) to (0.60-1.0), and the ability to handle nonconvex regions. Results of mesh refinement and modeling EM field solvers were validated and accomplished through a series of tests and comparisons of the GA and the DTA mesh quality results and by observing the effect of E-fields and H-fields on the computed results.

General Terms
Meshing Algorithms, Magnetostatics.

Keywords
Mesh Generation, Mesh Refinement, Delaunay Triangulation, Genetic Algorithm.

1. INTRODUCTION
Accurate modeling of electromagnetic (EM) fields plays a pivotal role in diverse scientific and engineering applications, particularly in high-frequency simulations such as antenna design, microwave circuits, and broader radio frequency (RF) applications [1]. The pursuit of solving EM field challenges has driven researchers and engineers to explore various numerical techniques for accurately predicting electric and magnetic problems. Notably, among various numerical methods, the Finite Element Method (FEM) has gained significant traction. [2]. This prominence is attributed FEM's exceptional versatility in handling complex geometries and heterogeneous material properties, coupled with its proficiency in tackling a comprehensive array of EM problems. Consequently, FEM has emerged as an indispensable tool in the rigorous modeling of EM phenomena, underscoring its pivotal role in advancing both theoretical foundations and practical applications within the sphere of EM research. [3].

The core principle of FEM is the subdivision of a continuous problem domain into a finite number of subdomains called finite elements (FE) [4]. Mathematically, the discretization process involves transforming a continuous problem described by partial differential equations (PDEs) into a system of algebraic equations, which can be solved more efficiently using computational methods. A notable application of this approach is the solution of EM problems in steady-state [2], [5], [6]. For instance, FEM is employed to solve PDEs that describe the distribution of electric potential (V) within a dielectric medium [7]. This research explored a new approach to mesh refinement for EM field solvers using the properties of GA [8]. Through the application of GAs, the study counters the limitations of conventional mesh refinement approaches in unstructured geometrical domains, thus efficiently adapting to dynamic EM simulation needs. This research also sought to enhance unstructured triangular meshes' solver accuracy and computational efficiency. Besides, its application in solving real FEA problems contributes to developing adaptive refinement approaches in computational electromagnetics [9].

2. THEORETICAL BACKGROUND
Finite element problems commonly manifest through Poisson's and Laplace's equations, as depicted in 1 and 2, depending on the presence of free charge within the domain.

$$\nabla \cdot (\varepsilon \nabla V) = -\rho \quad where\ \rho \neq 0 \qquad (1)$$

$$\nabla \cdot (\varepsilon \nabla V) = 0 \qquad (2)$$

where: ρ - volume charge density; ε - electric permittivity of the dielectric; **V** -scalar electric potential. The Laplace equation describes scenarios where the electric potential is influenced purely by boundary conditions and not by any internal charges.

FEM discretizes a geometric domain (problem domain) into triangles or quadrilaterals in two dimensions (2D) and tetrahedral or hexahedra in three dimensions (3D). This discretization yields two categories of meshes: a structured mesh, characterized by uniform topology among all interior vertices, and an unstructured mesh, where vertices can exhibit arbitrarily varying local neighborhoods [10], [11]. In a 2D domain, each element's potential, **V**, in equation 3 is a weighted sum of a linear combination of the nodal potentials:

$$V(x,y) \approx \sum_{i=1}^{n} N_j(x,y) v_i \qquad (3)$$

Where $N_i(x,y)$ are interpolation functions associated with the element nodes and v_i are the unknown nodal potentials

discretization, the governing PDE (Poisson's or Laplace's equation) is converted into a set of algebraic equations using a suitable weighted residual approach such as the *Galerkin* method [12], [13]. Using the *Galerkin* method, Poisson's equation yields to equation 4.

$$\int_{\Omega}^{\cdot} \nabla \cdot (\varepsilon \nabla V) N_j \, d\Omega = -\int_{\Omega}^{\cdot} \rho N_j d\Omega \qquad (4)$$

Applying the divergence theorem and boundary conditions, this integral equation is transformed into a system of linear algebraic equations represented by equation 5b

$$[K]\{V\} = \{F\} \qquad (5)$$

where: $[K]$ is the global stiffness matrix and encodes the material properties (ε) and geometry of the domain; $\{V\}$ is a vector of unknown nodal potentials; $\{F\}$is the load vector representing charge density ρ contribution to the system.

Developing FE surface meshes is integral for accurately simulating real-world objects, especially within EM field computational models. Achieving a high precision of these meshes directly influences the ability of the model to capture the complex and intricate details of physical geometries [5]. However, as mesh resolution increases to capture finer details, the computational problem also escalates, leading to simulations that can become too expensive in terms of time and computational resources. This creates a critical need to balance accuracy and efficiency, a fundamental concern in the application of finite element methods (FEM) to electromagnetic problems [14], [15].

EM problem computations using FEM can be broken down into three main steps: preprocessing, analysis, and post-processing [2], [15]. Preprocessing involves generating the initial mesh, which is a crucial step as the mesh quality significantly affects the accuracy and convergence of the subsequent analysis. Numerous methods have been developed and presented through mesh generation research, some of which have been incorporated into commercial EM simulation software [11], [13]. These methods include mapping techniques, finite quadtree/octree methods [13], Delaunay Triangulation (DT) [16], [18], and the Advancing Front Technique (AFT) [19], [20], among others. DT and ATF are the most utilized methods. In many simulations, DT is preferred because it avoids skinny triangles, enhancing the overall mesh quality. On the other hand, ATF is known for generating conforming meshes to complex boundaries, making it useful in EM simulations where precise boundary conditions are critical.

In modeling EM fields, Maxwell's equations govern the behavior of electric and magnetic fields [21], [22]. For static EM field problems, these equations are expressed in equation 6.

$$\nabla x E = 0;\ \nabla x H = J_s;\ \nabla . D = \rho_s;\ \nabla . B = 0 \qquad (6)$$

The constitutive equations in 6 link the field quantities to the material properties as in equation 7.

$$D = \varepsilon E, \quad B = \mu H, \quad J = \sigma E \qquad (7)$$

Where ε is the electrical permittivity of the material, μ is the magnetic permeability of the material, and σ is the conductivity tensor of the material. For electrostatic problems, the following Maxwell's equations derived from Gauss's law PDE are solved as in equations 8 and 9

$$-(\varepsilon_r \varepsilon_0 \nabla . V) = \rho_S \qquad (8)$$

$$-\nabla V = E \qquad (9)$$

Where: ρ_S – surface charge density, ε_r – material relative permittivity, ε_0 – material absolute permittivity, V – electric potential. The Dirichlet boundary conditions specify V on the boundary and ρ_s at the conductor medium. For magnetostatic problems, where steady currents generate the magnetic fields, Maxwell's equations derived from Ampere's law PDE are solved in $\nabla \times H = J$ there exists a magnetic vector potential A, such that $B = \nabla \times A$. Substituting this into Ampère's law and applying the vector identity $(1/\mu . \nabla * A) = J$. Using the identity $\nabla \times (\nabla \times A) = \nabla(\nabla \cdot A) - \nabla^2 A$ we equation 10

$$-\nabla\text{^}2 \cdot A = -\nabla \cdot \nabla A = \mu J \qquad (10)$$

Dirichlet boundary conditions specify the magnetic potential on the boundary. Equations (9) & (10) form the basis of FEM for static EM field problems, where the magnetic vector potential A is solved within the mesh to determine the magnetic field distribution and the electric potential V within the dielectric medium.

Mesh refinement is a technique used in numerical simulations, particularly in FEM, to improve the accuracy of the results by increasing the density or quality of the computational mesh. The quality of these mesh elements is often assessed using several geometric criteria, which directly affect the numerical consistency and accuracy of the simulation. The key objective of mesh refinement is to optimize these criteria by iteratively adjusting the positions of mesh vertices [23]. Mathematically, the quality Q_c of a mesh element t can be expressed as a function of its geometric properties, such as edge lengths, angles, and area, as shown in equation 11.

$$Q_c = f(edge\ lengths, angles, area) \qquad (11)$$

For a typical triangular mesh element with vertices (v_1, v_2, v_3) and the corresponding length of edges (l_{12}, l_{23}, l_{31}) the mesh quality criteria include:

i.) Aspect Ratio (AR) is the ratio of the triangle's longest edge to the shortest altitude, as depicted in equation 12. Minimizing AR leads to more equilateral triangles, which are preferred.

$$AR = \frac{l_{max}}{h_{min}} = \frac{(edge\ lengths)}{(altitudes)} \qquad (12)$$

where: $l_{max} = (l_{12}, l_{23}, l_{31})$ *and* $h_{min} = \frac{2*Area}{l_{max}}$

$$Area = \frac{1}{2} |v_1 * v_2 + v_2 * v_3 + v_3 * v_1|$$

ii.) Minimum Angle (θ_{min}). This ensures that the minimum internal angle is maximized to avoid degenerate triangles, i.e., where one angle approaches 0^o as demonstrated in equation 13.

$$\theta_{min} = (\theta_1\ \theta_2\ \theta_3) \qquad (13)$$

The dot product of the vectors formed by the triangle's edges can be used to calculate each angle. Where e_j *and* e_k are vectors representing the edges of the triangle at the vertex opposite θ_i

$$cos\ cos\ \theta_i = \frac{e_j \cdot e_k}{|e_j|\,|e_k|}$$

iii.) Circumradius (R_C) to Shortest Edge Ratio: Minimizing the ratio of the circumradius to the shortest edge prevents

overly thin triangles: $\frac{R_c}{l_{min}} = \frac{circumradious}{shortest\ edge}$. where $R_c = \frac{l_{12}l_{23}l_{31}}{4 \cdot Area}$ and $l_{min} = (l_{12}l_{23}l_{31})$, thus equation 14

$$\frac{R_c}{l_{min}} = \frac{l_{12}l_{23}l_{31}}{4 \cdot Area \cdot l_{min}} \quad (14)$$

This ratio is minimized to prevent the generation of slender triangles, which result in poor numerical performance in FEM. In general, the mesh quality metric Q_c for a triangular element can be presented as a weighted combination of the above criteria, which collectively represent the element's divergence from an ideal shape (an equilateral triangle) and is represented by equation 15.

$$Q_c = w_1 \cdot AR + w_2 \cdot \frac{R_c}{l_{min}} + w_3 \cdot \frac{1}{\theta_{min}} \quad (15)$$

where $w_1, w_2\ and\ w_3$ are weights assigned based on the importance of each criterion in a specific application. The ultimate goal in mesh refinement is to minimize the distortion metric D_t, which is the sum of the quality metrics for all elements t in the mesh as in equation 16.

$$D_t = \sum_{t \in mesh} Q_c \quad (16)$$

Minimizing D_t is constrained by the geometric domain and physical boundaries of the problem. The process of iterative refinement aims to optimize vertex positions to minimize D_t, which improves the mesh quality and enhances the accuracy of FEM simulations. The majority of refinement techniques employ iterative processes to adjust individual vertices to enhance the quality of the triangular mesh [23]. Typically, triangular meshes are evaluated based on criteria such as maintaining a small aspect ratio and avoiding angles that are extremely close to 0 or 180 degrees as in equation 17 for $0^o < \theta_{min} < \theta_i < \theta_{max} < 180^o$.

$$minimize\ AR = \frac{l_{max}}{h_{min}} \quad (17)$$

However, a range of other optimization objectives exist, including minimizing the maximum angle, maximizing the minimum angle, minimizing the maximum circumradius, reducing the maximum aspect ratio, and minimizing the radius of the enclosing circle for the constituent triangles [24]. Techniques such as refinement, averaging, and optimization-based methods are also utilized to enhance the quality of triangular mesh. Mesh refinement involves adjusting the positions of the mesh vertices to enhance the mesh elements' shapes and the mesh's overall quality. During this process, the topology of the mesh remains unchanged. The most commonly utilized technique for mesh refinement is the Laplacian refinement method [24], where vertex positions $(v_i^{new})\ are$ adjusted and updated based on the average of a set of neighboring vertices $(N(i))$ as in equation 18

$$v_i^{new} = \frac{1}{|N(i)|} \sum_{j \in N(i)} v_j \quad (18)$$

Though this technique is computationally efficient and reasonably effective, it does not guarantee an improvement in mesh quality, especially if the initial mesh elements are of poor quality. Several other averaging and refinement methods involve formulating and solving nonlinear optimization problems. The objective is to minimize the distortion metric Dt, subject to constraints that ensure the physical validity of the mesh in the feasible region, as in equation 19 [25].

$$minimize\ D(v) | v_i \in \Omega \quad (19)$$

Advancements in this field of mesh refinement and optimization have been explored in several works. Hwang et al. [23] presented an optimization-based strategy for maximizing the aspect ratio by solving a nonlinear and constrained global optimization problem confined by the mesh topology. Canann et al. [26] combined Laplacian smoothing with optimization-based refinement algorithms to improve mesh quality. Lori et al. [27] pursued an approach that maximizes the minimum angle in triangular meshes using the steepest descent method as the refinement function. Amenta et al. [28] presented theoretical findings demonstrating how extended linear programming can optimize triangle shapes locally, showing that this framework allows for systematic local refinement. Numerous efficient algorithms, distortion metrics, and optimization techniques are discussed and compared. Additional optimization-based methods encompass the works that rely on posterior error estimates and center around distortion metrics [29]. The selection of appropriate distortion metrics is also a topic explored.

Over time, there has been a growing adoption of artificial intelligence principles in the realm of problem-domain discretization and refinement. In one instance, the application of Neural network concepts was explored for generating finite element meshes [30]. Also, Salama presented a binary genetic algorithm (GA) that uses a distortion metric as the fitness function to refine quadrilateral mesh elements utilized in finite element analysis [31]. The GA technique iteratively evolves the mesh by selecting, crossing over, and mutating vertices to achieve a mesh configuration that minimizes the distortion metric. The fitness function (F) is employed in GA as in equation 20.

$$F(x) = f(x) |\ x \in \Omega \quad (23)$$

Where $f(x)$ is the objective function to be optimized; x is a solution vector within the feasible region Ω; $F(x)$ the function evaluates the fitness of each candidate solution x in a population.

The performance of GAs is highly dependent on the fitness function, which relates to the aspect ratio, element size, and the global smoothness of the refined mesh [32]. Also, there are implementation difficulties due to the dynamic tuning of the exploration-exploitation trade-off in the GA process to allow the algorithm to fine-tune its specific mesh local irregularities while preserving the overall solution quality [33]. These problems call for fine-tuning GA parameters, mutation rates, and population sizes to achieve higher performance without considerable computational costs. As shown in Figure 1, the schedule-free transit problem was transformed into a high-level system global optimization problem and a low-level dispatch optimization problem to improve GA performance [34] effectively.

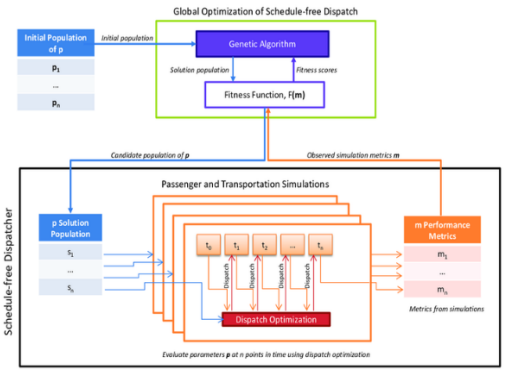

Fig. 1 Parametric refinement through the use of a genetic algorithm [34]

3. METHODOLOGY

3.1 Problem domain definition

The initial stage of the finite element mesh refinement procedure was directed towards defining the geometry of the problem domain, which served as a basis for practical EM simulation. The domain for this research was thus developed based on a patch antenna made from copper and dielectric material. The geometry was mathematically defined and made compatible with finite element analysis (FEA) by defining the physical dimensions of the copper patch and dielectric substrate, length (l), and width (w). The overall domain was represented as a combination of the copper (Ω_{Cu}) and dielectric ($\Omega_{dielectric}$) regions as in equation 21.

$$\Omega_{antenna} = \Omega_{Cu} \cup \Omega_{dielectric} \quad (21)$$

The material characteristics were important in determining the appropriate model of the antenna. For the copper patch, the properties incorporated were conductivity (σ_{Cu}), relative permittivity (ϵ_r), and the relative permeability (μ_r) with the following values $\sigma_{cu} \approx 5.8 \times 10^7 S/\text{m}$, $\epsilon_r \approx 1$ and $\mu_r \approx 1$, respectively. The dielectric substrate conductivity was considerably lower $\sigma_{dielectric} \approx 10^{-12}\ S/m$, and permittivity and permeability were almost constant ($\epsilon_r \approx 0$, $\mu_r \approx 1$). The boundary between the copper and dielectric areas represented by Γ was essential to simulate the EM fields appropriately. It is defined as in equation 22.

$$\Gamma = \partial\Omega_{Cu} \cap \partial\Omega_{dielectric} \quad (22)$$

It also clearly separated areas with different electromagnetic characteristics to facilitate the proper modeling of field interactions.

3.2 Initial Mesh Generation for Patch Antennas

3.2.1 Critical Steps in DTA Implementation for Patch Antennas

i.) Initial Points Distribution

The geometry was divided into small subdomains by specifying that the point density is higher near critical boundaries Γ. This also meant high-resolution meshing where copper and dielectric materials overlapped.

ii.) Boundaries and limitations

Triangle limits were placed on points on Γ to retain the boundary during the triangulation process. To ensure that the boundary remained intact during mesh generation, constrained Delaunay Triangulation (CDT), an extension of DTA, was used. Region

iii.) Constrained Discretization

Constrained Delaunay Triangulation discretized the interior region of both Ω_{Cu} and $\Omega_{dielectric}$. The algorithm also made it possible to avoid the formation of triangles that crossed over the boundary Γ, thus preserving the homogeneity of the two areas with different material properties. Material Property Assignment: The material properties of each triangle of the triangles were then determined according to their position. For triangles within the copper region $T \subset \Omega_{Cu}$, other parameters, including σCu and ϵCu, have been employed. For dielectric triangles $T \subset \Omega_{dielectric}$, such properties as $\sigma_{dielectric}$ and $\epsilon_{dielectric}$ were used

3.2.2 DTA Workflow

Figure 3 illustrates the pseudocode diagram of the DTA process for creating Initial mesh generation- In the first step of the triangulation, when the points belonging to the boundary of the geometry are given, super triangles (Δ_{super}) containing all of those points, were built. As the points were developed, these were integrated into the triangulation to complement the data.

Begin
Construct super triangle $\Delta_{super} | \Delta_{super} \supseteq P$
Initialize triangulation T with Δ_{super} i.e. $T = \{\Delta_{super}\}$
For each point $p_i \in P$:
Identify bad triangles:
Let $B = \emptyset$ be the set of bad triangles
For each triangle $\Delta \in T$ check:
If p_i lies inside circumcircle of Δ:
Add Δ to B i.e. $B \leftarrow B \cup \{\Delta\}$
Remove bad triangles:
Remove all Δ in B from T i.e. $T \leftarrow T \setminus B$
Determine polygonal hole boundary:
Determine holes formed by the removed Δ in B
Let $E = \{e \in edges\ of\ B | e$ is not shared by two Δ in $B\}$
Re-triangulate:
For $\forall e \in E$: create Δs by connecting p_i to endpoints of e
Add the new Δs to T
Remove Super triangles:
Remove any Δ in T that shares node with Δ_{super} i.e. $T = T \setminus \{\Delta \in T | \Delta \cap \Delta_{super} \neq \emptyset\}$
Return Delaunay Triangulation:
Resulting set T is Delaunay triangulation of P
End

Fig. 2 Delaunay Triangulation pseudocode

For every point, all the triangles with their circumcircle containing the new point of interest were marked as 'bad,' and all these bad triangles were erased from the current mesh; they left polygonal holes behind. The polygonal boundaries were re-triangulated by joining the new point to the vertices of the holes. As a final touch, any triangles situated at the nodes of Δ_{super} were stripped off, which means all the graph triangulation was done here and to come was restricted to the particular problem area of interest. A high-quality mesh that conforms to the Delaunay triangle requirements of the patch antenna geometry of the patch antenna was generated.

3.3 Genetic Algorithm for Mesh Refinement

3.3.1 Mesh refinement

The first meshes constructed using the Delaunay Triangulation Algorithm (DTA) usually comprise inferior triangles, implying that meshes can perform suboptimal numerical computations,

mainly when applied in finite element analysis (FEA). The refinement stage can be done using a Genetic Algorithm (GA) on the initial mesh used in the research. GA was identified as a valuable tool for refining the first DT mesh for the patch antenna model. This method is aimed at raising the density in the region where a high field gradient is expected to obtain a higher mesh quality while at the same time not compromising on computational costs.

3.3.2 Algorithm Framework

Starting with a set of chromosomes, the genetic algorithm (GA) proceeds to refinement using meshes represented in the chromosomes. It continues to evaluate and create other sets of chromosomes related to new generations of possible solutions, as shown in the block diagram in Figure 3. In each iteration, the fitness of each chromosome is measured using calculated criteria known as the objective function. In light of this fitness, specific chromosomes are permitted in reproduction. The number of copies a given parent GA produces is directly proportional to its 'fitness'—this is done in a process analogous to natural selection. This process further enhances the choice of better-fit solutions as the least-fitted solutions are ejected.

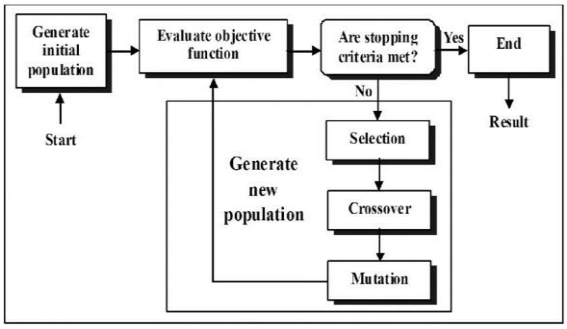

Fig. 3 Genetic Algorithm (GA) Block Diagram

The mesh refinement process applied the selection, crossover, and mutation operators using the GA framework. Selection guaranteed that fitter entities were selected for reproduction following the mesh quality metrics, which included aspect ratios and uniformity of elements used in meshing. During the optimization phase, there were different selections, including a tournament based on roulette wheel selection, to enhance exploration and exploitation [8]. In every run of the GA, set at time t, denoted as in equation 23.

$$P(t) = \{x_{0t}, x_{1t}, \dots, x_{mt}\} \qquad (23)$$

These solutions are examined to assess how 'fit' they are. Then, using the newly generated population for the subsequent iteration, denoted by $t + 1$, introduces new solutions. Forming the new population $(t + 1)$ involves two key genetic operations: crossover and mutation. The crossover operation involves features from two parent chromosomes, giving the offspring similar features to those of the parents. It is done by swapping equivalent sections of the parent chromosomes. On the other hand, in mutation operation, one or more genes or loci in a defined chromosome will be changed randomly with a specified mutation rate to generate variation in the population. The genetic algorithm used in this study is proportional selection, an elitist model, one-point crossover, and uniform mutation as shown in pseudocode in Figure 4. First, to define an objective function for the optimization a geometrical approach is used. This assuredly applies and develops potential genetic solutions by optimizing genetic fitness and using genetic operators, namely crossover and mutation.

```
Start t ← 0
    Initialize P(t)
    Evaluate P(t)
    While (not termination-condition) DO
    Start t ← t + 1
        select P(t) from P(t-1)
        Crossover
        Mutate
        Evaluate P(t)
    end
end
```

Fig. 4 Genetic Algorithm Pseudocode

3.3.3 Genetic Algorithm Parameters

i.) Population Size

In the problem domain mesh refinement, the population size N influences the simulation of the solution space and the convergence speed. Each internal vertex v_i in the mesh has a population of chromosomes, with each potential vertex position represented by a chromosome $X_i^{(j)} = \left(x_i^{(j)}, y_i^{(j)}\right)$. The coordinates are initialized, as shown in equation 24, where $r_x, r_y \sim U(0,1)$.

$$\left.\begin{matrix} x_i^{(j)} = x_{min} + r_x \cdot (x_{max} - x_{min}) \\ y_i^{(j)} = y_{min} + r_x \cdot (y_{max} - y_{min}) \end{matrix}\right\} \qquad (24)$$

The fitness function $f(X_i)$ evaluates each chromosome based on the finite element error, e_i as in equation 25, with a more considerable N solution diversity, is enhanced, but convergence is slowed, improving the chances of reaching a global optimum

$$f(X_i) = \sum_{i=1}^{N} e_i \qquad (25)$$

ii.) Evaluation Function

In the case of each randomly generated node (chromosome), the algorithm computes the sum of areas for all triangles surrounding that node. This sum must match the summation of areas $N = \sum_1^n t$ for all triangles (t) around the initial node N, which is intended to be relocated to the optimal location.

iii.) Selection Process

A roulette wheel is employed to perform the selection process, which involves choosing a new population based on a probability distribution derived from fitness values. This roulette wheel is constructed with slots sized by individual fitness levels, ensuring a proportional selection process. It entailed the following steps

a.) The fitness value $f(N_i)$ for each vertex (chromosome) was calculated using the expression N_i where $\{i = 1,2,3 \dots m\ (population\ size)\}$

b.) The total fitness of the population was determined using equation 26.

$$Fp = \sum_{i=1}^{m} f(N_i) \qquad (26)$$

c.) The probability p_i of selection for each chromosome (node N_i) (i = 1,2,3 … m) was computed as in equation 27.

$$p_i = \frac{f(N_i)}{Fp} \quad (27)$$

d.) The cumulative probability q_i for each chromosome (node N_i (i = 1,2,3 . . . m)) was computed using equation 28, as shown below.

$$q_i = \sum_{j=1}^{i} p_j \quad (28)$$

e.) The cumulative fitness probability was used to generate the elitist model used to choose survivors based on cumulative fitness probability: Let b be a randomized number selected from the interval (0,1), and let q_i represent the cumulative fitness probability of the i^{th} chromosome, where $q_1 \leq q_2 \leq \cdots \leq q_m$ and m is the total number of chromosomes. The selection rule for the i^{th}chromosome N_i as in equation 29 where $2 \leq \ i \leq \ m$

$$N_i = \begin{Bmatrix} N_1, & if\ b < q_1 \\ N_i, if & q_{i-1} < b < q_i \end{Bmatrix} \quad (29)$$

iv.) Crossover Probability

The crossover probability typically falls within the range of 0.01 to 1.0. Crossover determines the likelihood that the upcoming node population will contain a blend of information inherited from the previous node generation. A rate of 0.5 indicates that a child node will inherit approximately 50% of its characteristics from one parent node and the remaining 50% from the other. A rate of 1.0 signifies that no crossover will occur, resulting in evaluations that are essentially clones of the parents. The current algorithm employed a crossover probability of 0.85.

v.) Mutation Rate

The mutation rate can vary between 0.0 and 1.0. A higher mutation rate increases the likelihood that future node chromosomes will incorporate some random values. As mutation occurs after crossover, an excessively high mutation rate can diminish the impact of crossover. The present algorithm utilized random uniform mutation with a probability of 0.2.

vi.) Evaluation (Fitness) Function

Three different approaches were considered for selecting the function to be optimized within the genetic algorithm. These approaches encompass the least square error related to angle variations, the average aspect ratio of triangles at each node within the triangular mesh, and a linear combination of both methodologies. In this method, the evaluation function takes on a composite form, combining the least square error of angles and the average aspect ratio for triangles. Empirical evidence suggests that the third approach, involving the linear combination, is the most effective criterion for selecting the fitness function. The fitness function is calculated for each selected node. In every iteration, the absolute change in the fitness function is denoted as F, and its maximum value is computed. This process was iterated until convergence was achieved. A fixed value of 0.015 serves as the convergence criterion for differences in fitness.

4. RESULTS AND DISCUSSION

4.1 Test cases

4.1.1 Initial and Refined Meshes for Circular Case

In the case of a circular domain, figure 5 shows an initial mesh using the Delaunay triangulation. A red-edged circle encloses the entire figure, while the interior comprises triangular elements joined by blue lines. These triangles are formed to maximize the mesh quality, with no point inside the circumcircle of any triangle. The mesh includes 144 nodes and 254 elements, and this structured approach also guarantees effective numerical computation used in finite element modeling, especially in applying partial differential equations on circular domains. It is also observed that the triangulation is fairly uniform and consistent, which indicates a relatively balanced distribution of computational precision and performance.

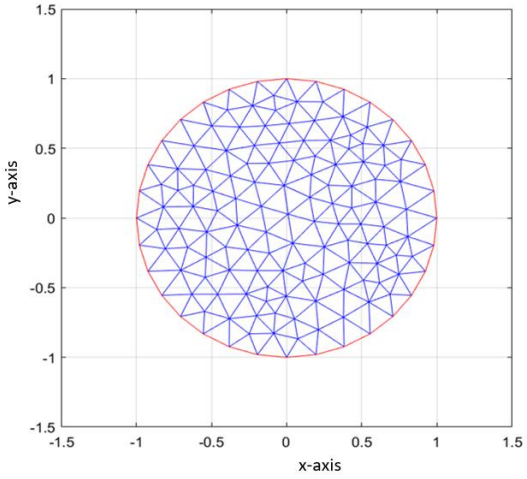

Fig. 5 An initial mesh using the Delaunay triangulation for the regular circular domain

Figure 6 represents a refined Delaunay triangulation of a circular domain obtained using genetic algorithm-based refinement. Compared to the initial mesh, this refined mesh is denser and has a finer triangulation to address precision issues in computational. More precisely, the refined mesh includes 541 nodes and 1016 triangles; their quantity is much higher in the circle region, thereby improving the resolution in that area. This refinement enhances element quality and distribution to simulate the exactness needs of finite element analysis in electromagnetic structures. GA-based refinement involves repeating the optimization process to score better triangle shapes and guarantee adherence to the Delaunay criteria to avoid forming poorly shaped elements. This makes the process robust in applications, particularly where there is a need for higher-resolution results regarding the boundary condition or gradients within the domain.

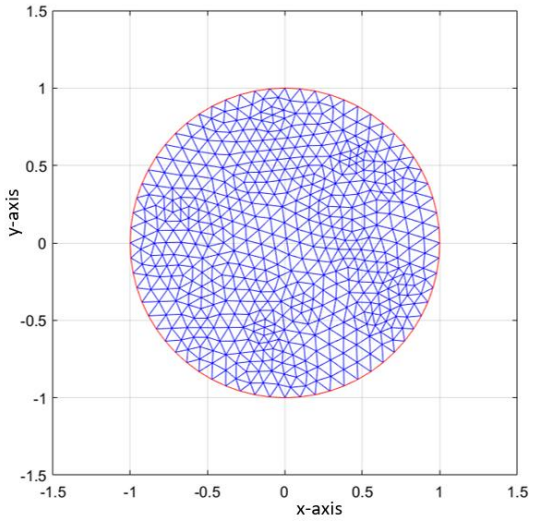

Fig. 6 A refined mesh using the GA for the regular circular domain

4.1.2 Electric Potential Solution for the Circular Domain

In electrical influences, Figure 7 shows the electric potential (V) and the vector field of electric potential (E) inside a circular region. The potential is displayed with a high-to-low gradient color, while the color bar on the right contains quantitative reference values. The electric field vectors are represented by red arrows going out of the plain from the region inside the circle. These arrows point to the direction and approximate intensity of the electric field. A relatively smooth electric potential variation was observed within the circular domain, where the concentric circles represent equipotential lines in the present electric field distribution.

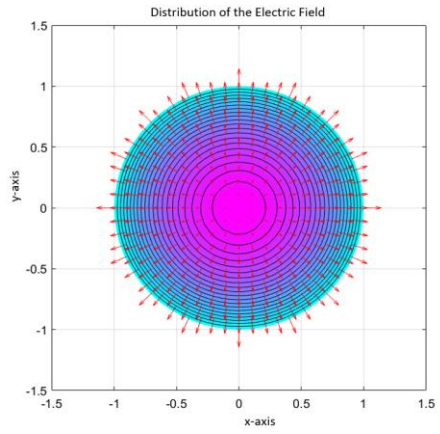

Fig. 7 Distribution of the electric potential on a circular potential

4.2 Patch Antenna

Figure 8 shows the patch antenna, which is T-shaped, on a dielectric element before the meshing was built in MATLAB.

The patch was modeled by loading the CSV Excel file containing the nodes and edges of the patch antenna in MATLAB to define its domain.

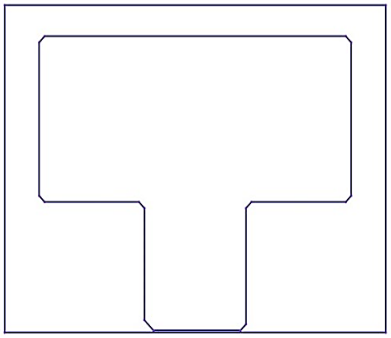

Fig. 8 Patch domain of the antenna element geometry

The initial mesh generation patch antenna problem domain was applied to the domain both inside and outside, as depicted in Figure 9. A total of 277 Delaunay triangles were obtained on the outer dielectric domain and 101 Delaunay triangles in the inner patch of the element. The mean aspect ratio for the antenna patch domain Delaunay triangle was 0.2439, while that of the dielectric was 0.4568.

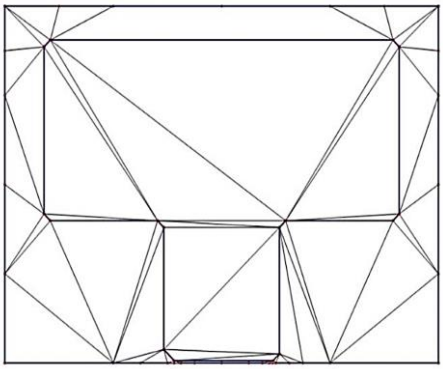

Fig. 9 Initial mesh generation of Delaunay triangles for the antenna and dielectric domains

The Genetic Algorithm (GA) refined the initial Delaunay mesh. Using the fitness function, the GA was utilized in the discretization and optimization process to adjust the positions of nodes within poorly formed Delaunay triangles. The enhancement in the mesh quality depended on the triangles' aspect ratio. Figures 10 and 11 show the refined mesh obtained for the patch antenna and dielectric element domains. The number of triangles for the patch domain increased to 403, while that of the dielectric domain increased to 407.

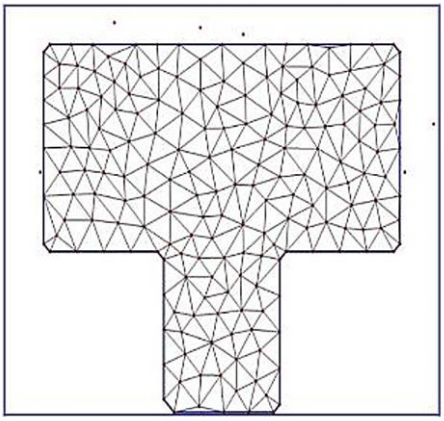

Fig. 10 Refined mesh for the antenna patch domain

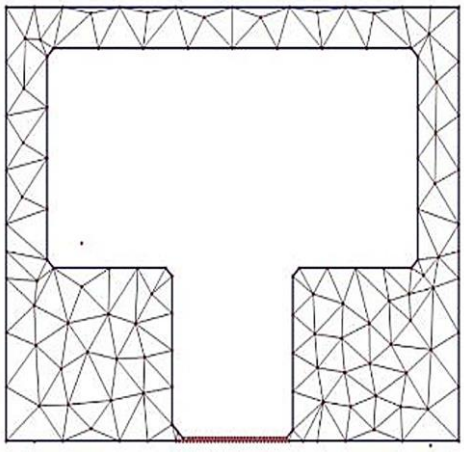

Fig. 11 Refined mesh for the dielectric domain

The application of genetic algorithms led to an increase in the number of triangles. Figures 12 and 13 show more triangles generated through refinement than the initial Delaunay triangles. Subsequently, there are more refined triangles with higher aspect ratios than those with lower aspect ratios. Following the refinement process, the fitness values for most triangles have shifted from a range of 0.1 to 0.50 to an improved range of 0.60 to 1.0.

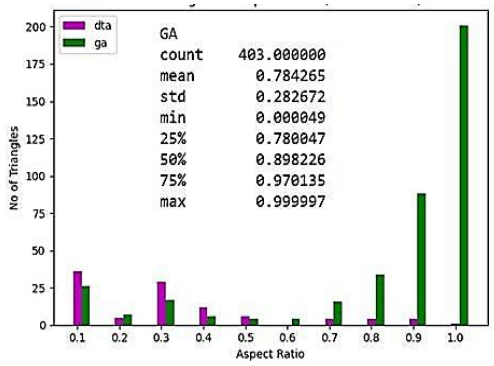

Fig. 12 The number of triangles against the aspect ratio for antenna patch domain

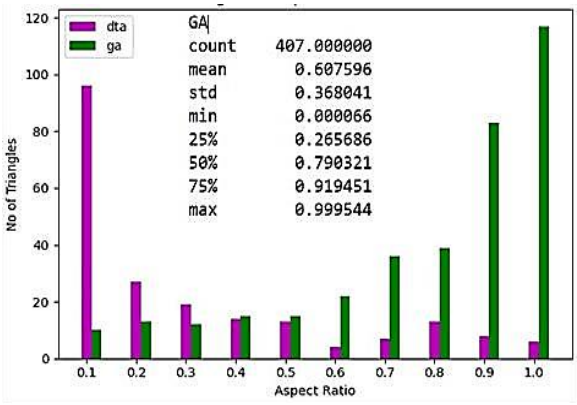

Fig. 13 The number of triangles against the aspect ratio for the dielectric domain

During the refinement process, the nodes were shifted to improve the fitness value of the Delaunay triangles. Figure 14 shows the improved aspect ratio of triangles, with the peak increasing from approximately 0.34 to 0.7. These peaks correspond to equilateral triangles or those whose node angles approach 60 degrees.

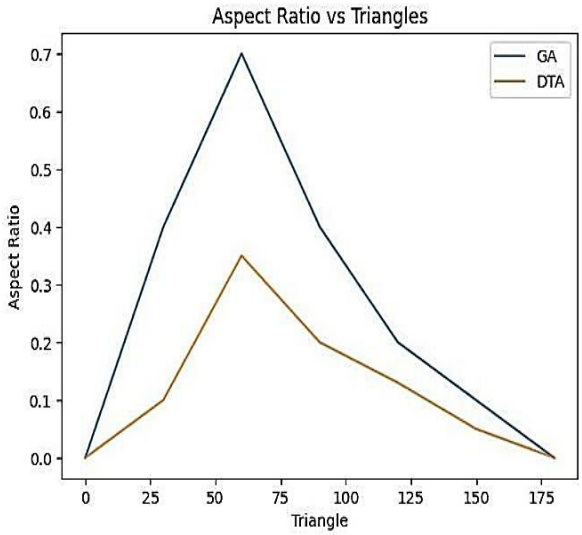

Fig. 14 Plot of aspect ratio against the

the nodal angle of the triangles in the refined mesh

The graph in Figure 15 displays the solution to a PDE of the antenna patch domain. The color represents the scalar field V, with a color bar indicating the value range from 0 to 3.5×10^7 volts per meter. The contour lines show the distribution of scalar field V, while red arrows indicate the vector field E, demonstrating the direction and magnitude of the electric field. The plot shows how V and E vary within the defined geometry, highlighting the areas of high intensity at the excitation point and the vector field's flow direction.

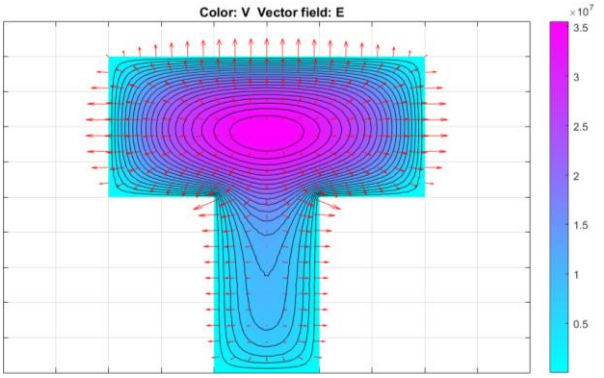

Fig. 15 Electric field excitation

This graph in Figure 16 illustrates the magnetic field distribution following the Delaunay GA triangulation. The refined mesh ensured accurate solutions for the magnetic field analysis. The color gradient on the graph represents the scalar potential A of the magnetic field, ranging from 0 to 1000 T. Contour lines illustrate the distribution and intensity of this scalar potential, with denser lines indicating regions of higher potential. Additionally, red arrows depict the magnetic flux density B, showing the direction and magnitude of the magnetic field within the geometry. This visualization effectively demonstrates how the magnetic field is distributed and oriented, highlighting the impact of the shapes and boundaries on the magnetic field's behavior.

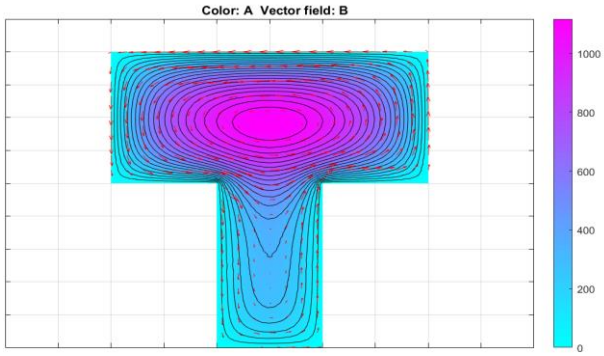

Fig. 16 Magnetic field distribution

5. CONCLUSION

This paper introduced an unstructured triangular grid generation and refinement algorithm for arbitrary two-dimensional regions. The mesh generation method was rooted in a modified Delaunay triangulation procedure complemented by a set of heuristic rules, ensuring the efficient triangulation of intricate areas. During triangulation, emphasis is placed on prioritizing the smallest angle between adjacent segments. These heuristics are strategically devised to mitigate common triangulation challenges, including front overlapping, intersection, and proximity of neighboring triangles. Furthermore, the algorithm employs an efficient branching strategy for triangle construction and selection. A fast permutation algorithm is incorporated to establish connectivity within the triangular grid, which is necessary for the Delaunay triangulation algorithm and post-processing. The results confirm the algorithm's capability to generate triangles for complex two-dimensional regions.

The paper also details a refinement algorithm for optimizing the mesh quality. This optimization process consists of two steps: first, diagonal swapping based on triangle aspect ratios, and second, the utilization of a floating-point genetic algorithm, offering greater flexibility than traditional binary genetic algorithms. Three approaches are explored for function selection in the genetic algorithm, encompassing the least square error of angle variation, the average aspect ratio of triangles at each node in the grid, and a linear combination of both methods. The results unequivocally demonstrate substantial enhancements in the quality of the optimized grids and the effectiveness of the floating-point genetic algorithm, even in the context of non-convex regions.

Future research may extend this methodology to the more intricate domain of three dimensions, aiming to establish smooth tetrahedral grids for 3D regions. Combining machine learning, artificial intelligence, and data science techniques can also be explored with mesh optimization to automate grid generation and refinement. Machine learning and artificial intelligence can help identify patterns and optimize grids more efficiently. Data science applications can facilitate data analysis, including dimensionality reduction, visualization, and clustering.

6. ACKNOWLEDGMENTS

The author would like to acknowledge the invaluable comments and guidance from Dr. Waweru Njeri and Dr. Joseph Muguro, who are from the Dedan Kimathi University of Technology, throughout the research study. Special thanks to them for their technical insights and encouragement during the course of this work. Additionally, he is highly indebted to the Dedan Kimathi University of Technology for granting access to their facilities, which significantly contributed to the success of this study.

7. REFERENCES

[1] M. A. AL-Amoudi, “STUDY, DESIGN, AND SIMULATION FOR MICROSTRIP PATCH ANTENNA,” International Journal of Applied Science and Engineering Review, vol. 02, no. 02, pp. 01–29, 2021, Doi: 10.52267/ijaser.2021.2201.

[2] M. KURYLA, “Introduction to Finite Element Analysis (FEA) or finite element method (FEM),” Freak Weather, 2018.

[3] L. B. W. and R. E. White, “An Introduction to the Finite Element Method with Application to Nonlinear Problems.,” Math Comput, vol. 50, no. 181, 1988, doi: 10.2307/2007936.

[4] M. BERN and D. EPPSTEIN, “MESH GENERATION AND OPTIMAL TRIANGULATION,” 1992. doi: 10.1142/9789814355858_0002.

[5] D. Schillinger and M. Ruess, “The Finite Cell Method: A Review in the Context of Higher-Order Structural Analysis of CAD and Image-Based Geometric Models,” Archives of Computational Methods in Engineering, vol. 22, no. 3, 2015, doi: 10.1007/s11831-014-9115-y.

[6] J. C. Heinrich, D. R. Poirier, and D. F. Nagelhout, “Mesh generation and flow calculations in highly contorted geometries,” Comput Methods Appl Mech Eng, vol. 133, no. 1–2, 1996, doi: 10.1016/0045-7825(96)01021-3.

[7] K. Key and J. Ovall, “A parallel goal-oriented adaptive finite element method for 2.5-D electromagnetic modeling,” Geophys J Int, vol. 186, no. 1, 2011, doi: 10.1111/j.1365-246X.2011.05025.x.

[8] Alhijawi, B., & Awajan, A. (2023). Genetic algorithms: theory, genetic operators, solutions, and applications. Evolutionary Intelligence. https://doi.org/10.1007/s12065-023-00822-6

[9] Chong, C. S., Lee, H. P., & Kumar, A. S. (2006). Genetic algorithms in mesh optimization for visualization and finite element models. Neural Computing and Applications, 15(3–4), 366–372. https://doi.org/10.1007/s00521-006-0041-2

[10] M. J. Aftosmis, J. E. Melton, and M. J. Berger, "Adaptation and surface modeling for cartesian mesh methods," in 12th Computational Fluid Dynamics Conference, 1995. doi: 10.2514/6.1995-1725.

[11] D. J. Mavriplis, "Unstructured Mesh Generation and Adapativity," 1995.

[12] X.-H. Wang, "Automatic Mesh Generation," in Finite Element Methods for Nonlinear Optical Waveguides, 2019. doi: 10.1201/9780203756027-4.Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullender.

[13] A. C. Polycarpou, "Introduction to the finite element method in electromagnetics," Synthesis Lectures on Computational Electromagnetics, vol. 4, 2006, doi: 10.2200/S00019ED1V01Y200604CEM004.

[14] D. J. Mavriplis, "Unstructured mesh generation and adaptivity. 26th Computational Fluid Dynamics Lecture Series, Rhode-Saint-Genese, March 1995," 1995.

[15] G. Meunier, The Finite Element Method for Electromagnetic Modeling. 2010. doi: 10.1002/9780470611173.

[16] D. J. Mavriplis, "An advancing front Delaunay triangulation algorithm designed for robustness," J Comput Phys, vol. 117, no. 1, 1995, doi: 10.1006/jcph.1995.1047.

[17] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The Quickhull Algorithm for Convex Hulls," ACM Transactions on Mathematical Software, vol. 22, no. 4, 1996, doi: 10.1145/235815.235821.

[18] D. T. Lee and B. J. Schachter, "Two algorithms for constructing a Delaunay triangulation," International Journal of Computer & Information Sciences, vol. 9, no. 3, 1980, doi: 10.1007/BF00977785.

[19] A. Rassineux, "Generation and optimization of tetrahedral meshes by advancing front technique," Int J Numer Methods Eng, vol. 41, no. 4, 1998, doi: 10.1002/(SICI)1097-0207(19980228)41:4<651::AID-NME304>3.0.CO;2-P.

[20] S. H. Chen, "Geometrical Description and Discretization of Hydraulic Structures," in Springer Tracts in Civil Engineering, 2019. doi: 10.1007/978-981-10-8135-4_3.

[21] T. Wan, B. Tang, and M. Li, "An Iteration-Free Domain Decomposition Method for the Fast Finite Element Analysis of Electromagnetic Problems," IEEE Trans Antennas Propag, vol. 68, no. 1, 2020, doi: 10.1109/TAP.2019.2943352.

[22] M. N. O. Sadiku, "Simple introduction to finite element analysis of electromagnetic problems," IEEE Transactions on Education, vol. v, no. n, 1992.

[23] J. T. Hwang and J. R. R. A. Martins, "An unstructured quadrilateral mesh generation algorithm for aircraft structures," Aerosp Sci Technol, vol. 59, 2016, doi: 10.1016/j.ast.2016.10.010.

[24] D. A. Field, "Laplacian smoothing and Delaunay triangulations," Communications in Applied Numerical Methods, vol. 4, no. 6, 1988, doi: 10.1002/cnm.1630040603.

[25] J. Y. Kang and B. S. Lee, "Optimisation of pipeline route in the presence of obstacles based on a least cost path algorithm and laplacian smoothing," International Journal of Naval Architecture and Ocean Engineering, vol. 9, no. 5, 2017, doi: 10.1016/j.ijnaoe.2017.02.001.

[26] S. A. Canann, M. B. Stephenson, and T. Blacker, "Optismoothing: An optimization-driven approach to mesh smoothing," Finite Elements in Analysis and Design, vol. 13, no. 2–3, 1993, doi: 10.1016/0168-874X(93)90056-V.

[27] C. O. Lori A. Freitag, "A Comparison of Tetrahedral Mesh Improvement Techniques," Fifth International Meshing Roundtable, 1996.

[28] N. Amenta, M. Bern, and D. Eppstein, "Optimal point placement for mesh smoothing," in Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, 1997.

[29] S. A. Canann, J. J. R. Tristano, and M. M. L. Staten, "An Approach to Combined Laplacian and Optimization-Based Smoothing for Triangular, Quadrilateral, and Quad-Dominant Meshes," in 7th International Meshing Roundtable, 1998.

[30] C. Cervellera and M. Muselli, "Deterministic Learning and an Application in Optimal Control," 2006. doi: 10.1016/S1076-5670(05)40002-6.

[31] M. Salama, A. Abdo Ali, and E. H. Atta, "A Genetic Algorithm Approach for Optimizing Unstructured Meshes."

[32] Prieto, F. G., Lahtinen, J., Samavaki, M., & Pursiainen, S. (2023). Multi-compartment head modeling in EEG: Unstructured boundary-fitted tetra meshing with subcortical structures. PLoS ONE, 18(9), e0290715. https://doi.org/10.1371/journal.pone.0290715

[33] Li, H., Kondoh, T., Jolivet, P., Nakayama, N., Furuta, K., Zhang, H., Zhu, B., Izui, K., & Nishiwaki, S. (2022). Topology optimization for lift–drag problems incorporated with distributed unstructured mesh adaptation. Structural and Multidisciplinary Optimization, 65(8). https://doi.org/10.1007/s00158-022-03314-w

[34] Lam, S. (2016). A Methodology for the Optimization of Autonomous Public Transportation [Thesis]. In UNSW Sydney, The School of Mechanical and Manufacturing Engineering, The University of New South Wales. https://doi.org/10.13140/RG.2.2.33767.91046