

Adaptive Finite Element Method for Solving Poisson Partial Differential Equation

Gokul K C¹, Ram Prasad Dulal¹

¹ Department of Mathematics,
School of Science,
Kathmandu University,
Dhulikhel, Nepal

Correspondence to: Ram Prasad Dulal, Email: ramprasad1983math@gmail.com

Abstract: Poisson equation is an elliptic partial differential equation, a generalization of Laplace equation. Finite element method is a widely used method for numerically solving partial differential equations. Adaptive finite element method distributes more mesh nodes around the area where singularity of the solution happens. In this paper, Poisson equation is solved using finite element method in a rectangular domain with Dirichlet and Neumann boundary conditions. Posteriori error analysis is used to mark the refinement area of the mesh. Dorfler adaptive algorithm is used to refine the marked mesh. The obtained results are compared with exact solutions and displayed graphically.

Keywords: Poisson equation, Finite element method, Posteriori error analysis, Adaptive mesh refinement

DOI: <https://doi.org/10.3126/jnms.v4i1.37107>

1 Introduction

Differential equations arise in many areas of applications such as in science and engineering. Most of the problems in almost all the scientific and engineering areas are modeled using partial differential equations (PDEs). Partial differential equation arise when a dependent variable depends on two or more independent variables. One of the widely known method for solving PDEs is finite element method (FEM). The Finite element method is a numerical analysis technique for obtaining approximate solutions to a wide variety of engineering problems. A finite element model of a problem gives a piecewise approximation to the governing equations. The basic premise of the FEM is that a solution region can be analytically modeled or approximated by replacing it with an assemblage of discrete elements[5].

1.1 Poisson partial differential equation

Poisson equation is a partial differential equation, named after Simeon Denis Poisson. The Poisson equation is derived from the following general elliptic partial differential equation[8]:

$$-\frac{\partial}{\partial x} \left(a_{11} \frac{\partial u}{\partial x} + a_{12} \frac{\partial u}{\partial y} \right) - \frac{\partial}{\partial y} \left(a_{21} \frac{\partial u}{\partial x} + a_{22} \frac{\partial u}{\partial y} \right) + a_{00}u - f = 0 \quad (1)$$

For particular case, setting $a_{11} = a_{22} = k(x, y) = k$ (constant) and $a_{21} = a_{12} = a_{00} = 0$, the equation reduces to:

$$\begin{aligned} -k \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) &= f(x, y) \\ -k \nabla^2 u &= f \\ \therefore -k \Delta u &= f \end{aligned} \quad (2)$$

where, $\Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is called Laplacian operator. The equation (2) is the simplest form of elliptic PDE called Poisson PDE consisting dependent variable u and independent variables x and y . The function $f(x, y)$ is called source function.

1.2 Adaptive finite element method (AFEM)

An adaptive finite element method (AFEM) was developed in the early 1980s. The basic concept of adaptivity developed in the FEM is that, when a physical problem is analyzed using finite elements, there exists some discretization errors caused owing to the use of the finite element model. These errors are calculated in order to assess the accuracy of the solution obtained. If the errors are large, the finite element model is refined through reducing the size of elements or increasing the order of interpolation functions. The new model is re-analyzed and the errors in the new model are re-calculated. This procedure is continued until the calculated errors fall below the specified permissible values. The key features in AFEM are the estimation of discretization errors and the refinement of finite element models[7]. It minimizes the errors by the concept of posteriori error analysis and using adaptive algorithm. It can be shown in the following diagram[7]:

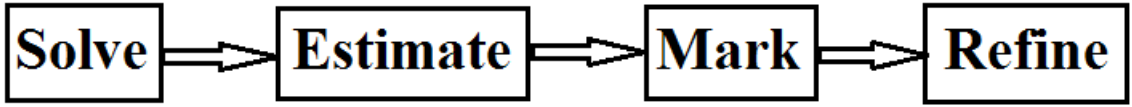


Figure 1: Adaptive finite element process

2 Methodology

2.1 Weak form of poisson equation

In the development of the weak form, we consider a typical triangular element Ω_e of the finite element mesh and develop the finite element model for the Poisson PDE (2). Multiplying (2) by an arbitrary test function $w = w(x, y) \in H_0^1(\Omega_e)$, which is assumed to be differentiable with respect to x and y , we get weak form of (2) as[4]:

$$\int_{\Omega_e} \left[\frac{\partial w}{\partial x} \left(k \frac{\partial u}{\partial x} \right) + \frac{\partial w}{\partial y} \left(k \frac{\partial u}{\partial y} \right) \right] dxdy = \int_{\Omega_e} w f dxdy + \oint_{\Gamma_e} w q_n ds \quad (3)$$

The equation (3) can be written as:

$$B^e(w, u) = L^e(w) \quad (4)$$

where,

$$\int_{\Omega_e} \left[\frac{\partial w}{\partial x} \left(k \frac{\partial u}{\partial x} \right) + \frac{\partial w}{\partial y} \left(k \frac{\partial u}{\partial y} \right) \right] dxdy = B^e(w, u)$$

$$\int_{\Omega_e} w f dxdy + \oint_{\Gamma_e} w q_n ds = L^e(w)$$

2.2 Finite element solution process

2.2.1 Interpolation functions

The governing equation (2) is equivalent to the weak form of equation (4) over the domain Ω_e and also contains natural boundary conditions. The weak form requires the approximation chosen for u should be at least linear in both x and y . There are no terms that are identically zero. Since the primary variable is a function itself, the Lagrange family of interpolation functions is admissible.

Let u is the approximated solution over a typical element Ω_e defined by the expression[8]:

$$u = u(x, y) \approx u_h^e(x, y) = \sum_{j=1}^n u_j^e \psi_j^e(x, y). \quad (5)$$

Where, u_j^e is the value of u_h^e at the j^{th} node (x_j, y_j) of the element and ψ_j^e is the Lagrange interpolation function with the property that:

$$\psi_i^e(x_j, y_j) = \delta_{ij}; \quad i = 1, 2, \dots, n \quad (6)$$

Putting the approximated value of (5) in (4), we get:

$$\sum_{j=1}^n \left\{ \int_{\Omega_e} \left[\frac{\partial w}{\partial x} \left(k \frac{\partial \psi_j^e}{\partial x} \right) + \frac{\partial w}{\partial y} \left(k \frac{\partial \psi_j^e}{\partial y} \right) \right] dx dy \right\} u_j^e = \int_{\Omega_e} w f dx dy + \int_{\Gamma_e} w q_n ds \quad (7)$$

This equation must hold for every admissible choice of weight function w . Since we need n independent algebraic equations to solve for n unknowns $(u_1^e, u_2^e, u_3^e, \dots, u_n^e)$, we choose n linearly independent functions for w . The functions of w are $w = (\psi_1^e, \psi_2^e, \psi_3^e, \dots, \psi_n^e)$. Hence we write:

$$\begin{aligned} \mathbf{u}^e &= [u_1^e \quad u_2^e \quad u_3^e \quad \dots \quad u_n^e] \\ \mathbf{w} &= [\psi_1^e \quad \psi_2^e \quad \psi_3^e \quad \dots \quad \psi_n^e] = \psi^T \end{aligned}$$

This particular choice of weight function is a natural one when the weight function is viewed as a virtual variation of the dependent unknowns ($w = \partial u = \sum_{i=1}^n \partial u_i \psi_i$) and the resulting finite element model is called weak form finite element model[8].

For each choice of w , we obtain an algebraic relation among $(u_1^e, u_2^e, u_3^e, \dots, u_n^e)$. We label the algebraic equation resulting from substituting successively the following in (7):

$$w = \psi_1^e, w = \psi_2^e, w = \psi_3^e, \dots, w = \psi_n^e.$$

to get n algebraic equations. Hence the general i^{th} form of the system can be written as:

$$\begin{aligned} \sum_{j=1}^n \left\{ \int_{\Omega_e} \left[\frac{\partial \psi_i^e}{\partial x} \left(k \frac{\partial \psi_j^e}{\partial x} \right) + \frac{\partial \psi_i^e}{\partial y} \left(k \frac{\partial \psi_j^e}{\partial y} \right) \right] dx dy \right\} u_j^e &= \int_{\Omega_e} f \psi_i^e dx dy + \int_{\Gamma_e} q_n \psi_i^e ds \quad \text{where} \\ K_{ij}^e &= \int_{\Omega_e} \left[\frac{\partial \psi_i^e}{\partial x} \left(k \frac{\partial \psi_j^e}{\partial x} \right) + \frac{\partial \psi_i^e}{\partial y} \left(k \frac{\partial \psi_j^e}{\partial y} \right) \right] dx dy, \quad f_i^e = \int_{\Omega_e} f \psi_i^e dx dy, \quad Q_i^e = \int_{\Gamma_e} q_n \psi_i^e ds \end{aligned}$$

Which reduces to:

$$\sum_{j=1}^n K_{ij}^e u_j^e = f_i^e + Q_i^e; \quad i = 1, 2, 3, \dots, n \quad (8)$$

The term K_{ij}^e is a bilinear form and is denoted as $K_{ij}^e = B^e(\psi_i^e, \psi_j^e)$. Equation 8 abbreviated in vector form as:

$$\mathbf{K}^e \mathbf{u}^e = \mathbf{f}^e + \mathbf{Q}^e \quad (9)$$

The matrix \mathbf{K}^e is called the coefficient matrix and the vector \mathbf{f}^e is called the source vector. The weak form and the finite element matrices in (9) shows that ψ_i^e must be at least linear functions of x and y . The complete linear polynomial in x and y in Ω_e is of the form:

$$u_h^e(x, y) = c_1^e + c_2^e x + c_3^e y \quad (10)$$

where, c_i^e 's are constants for $i = 1, 2, 3$. The set $\{1, x, y\}$ is linearly independent and complete. The equation (10) defines a unique plane for fixed c_i^e . In particular, $u_h^e(x, y)$ is uniquely defined on a triangle. Let $(x_1, y_1), (x_2, y_2)$ and (x_3, y_3) be the vertices of the triangle such that:

$$u_h^e(x_1, y_1) = u_1^e, \quad u_h^e(x_2, y_2) = u_2^e, \quad u_h^e(x_3, y_3) = u_3^e. \quad (11)$$

The constants c_i^e , ($i = 1, 2, 3$) in (10) can be expressed in terms of three nodal values $u_i^e (i = 1, 2, 3)$. The particular nodes of the triangle can be expressed as:

$$\begin{aligned} u_h^e(x_1, y_1) &\equiv u_1^e = c_1^e + c_2^e x_1 + c_3^e y_1 \\ u_h^e(x_2, y_2) &\equiv u_2^e = c_1^e + c_2^e x_2 + c_3^e y_2 \\ u_h^e(x_3, y_3) &\equiv u_3^e = c_1^e + c_2^e x_3 + c_3^e y_3. \end{aligned} \quad (12)$$

The matrix form of this system of equations is:

$$\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} c_1^e \\ c_2^e \\ c_3^e \end{bmatrix} = \begin{bmatrix} u_1^e \\ u_2^e \\ u_3^e \end{bmatrix} \quad (13)$$

where, \mathbf{A} is coefficient matrix, \mathbf{C} is constant matrix and \mathbf{u}^e is solution matrix such that:

$$\mathbf{A} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} c_1^e \\ c_2^e \\ c_3^e \end{bmatrix}, \mathbf{u}^e = \begin{bmatrix} u_1^e \\ u_2^e \\ u_3^e \end{bmatrix}.$$

The inverse matrix of \mathbf{A} is given by:

$$\mathbf{A}^{-1} = \frac{1}{|\mathbf{A}|} \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix} \quad (14)$$

where $\alpha_i, \beta_i, \gamma_i$ are constants depend only on the global coordinates of element nodes (x_i, y_i) s.t.:

$$\alpha_i = x_j y_k - x_k y_j, \quad \beta_i = y_j - y_k, \quad \gamma_i = x_k - x_j \quad (15)$$

for $i \neq j \neq k$ and i, j, k permute in a natural order. From inverse matrix formula. we can set:

$$c_1^e = \frac{1}{|\mathbf{A}|}(\alpha_1 u_1^e + \alpha_2 u_2^e + \alpha_3 u_3^e), c_2^e = \frac{1}{|\mathbf{A}|}(\beta_1 u_1^e + \beta_2 u_2^e + \beta_3 u_3^e), c_3^e = \frac{1}{|\mathbf{A}|}(\gamma_1 u_1^e + \gamma_2 u_2^e + \gamma_3 u_3^e) \quad (16)$$

Substituting the values of (16) in (10), we get:

$$\psi_i^e = \frac{1}{|\mathbf{A}_e|} (\alpha_i^e + \beta_i^e x + \gamma_i^e y).$$

The interpolation function ψ_i^e satisfies the following conditions:

$$\begin{aligned} \psi_i^e(x_j^e, y_j^e) &= \delta_{ij}; \quad (i, j = 1, 2, 3) \\ \sum_{i=1}^3 \psi_i^e &= 1, \quad \sum_{i=1}^3 \frac{\partial \psi_i^e}{\partial x} = 0, \quad \sum_{i=1}^3 \frac{\partial \psi_i^e}{\partial y} = 0, \end{aligned}$$

2.2.2 Element matrices and source vector

When the value k and f of the problem are element-wise constant, i.e. not a function of x and y , we can use the interpolation functions expressed in the local coordinates (\bar{x}, \bar{y}) . It is possible to calculate the integrals exactly over the linear rectangular element. The boundary integral in (Q^e) can be evaluated whenever q_n is known. The coefficient matrix (Taking $\Omega_e = \Delta$) is:

$$K_{ij}^e = \int_{\Delta} \left[\frac{\partial \psi_i^e}{\partial x} \left(k \frac{\partial \psi_j^e}{\partial x} \right) + \frac{\partial \psi_i^e}{\partial y} \left(k \frac{\partial \psi_j^e}{\partial y} \right) \right] dx dy. \quad (17)$$

Also, the interpolation functions for triangular element can be modified as:

$$\psi_i^e(\bar{x}, \bar{y}) = \frac{1}{|\mathbf{A}|} (\alpha_i^e + \beta_i^e x + \gamma_i^e y) \quad [i = 1, 2, 3].$$

Now, we have:

$$K_{11}^e = k \left[\int_{\Delta} \left(\frac{\partial \psi_1^e}{\partial \bar{x}} \cdot \frac{\partial \psi_1^e}{\partial \bar{x}} + \frac{\partial \psi_1^e}{\partial \bar{y}} \cdot \frac{\partial \psi_1^e}{\partial \bar{y}} \right) d\bar{x} d\bar{y} \right] = \frac{k}{4A_e} (\beta_1^e \beta_1^e + \gamma_1^e \gamma_1^e)$$

Similarly, we find all the values of K_{ij}^e . The coefficient matrix for the linear triangular element is:

$$[K^e] = \frac{k}{4A_e} \begin{bmatrix} (\beta_1^e)^2 + (\gamma_1^e)^2 & \beta_1^e\beta_2^e + \gamma_1^e\gamma_2^e & \beta_1^e\beta_3^e + \gamma_1^e\gamma_3^e \\ \beta_2^e\beta_1^e + \gamma_2^e\gamma_1^e & (\beta_2^e)^2 + (\gamma_2^e)^2 & \beta_2^e\beta_3^e + \gamma_2^e\gamma_3^e \\ \beta_3^e\beta_1^e + \gamma_3^e\gamma_1^e & \beta_3^e\beta_2^e + \gamma_3^e\gamma_2^e & (\beta_3^e)^2 + (\gamma_3^e)^2 \end{bmatrix} \quad (18)$$

In the governing equation (2), the source or force term f is considered as element wise constant.

$$f_i^e = \int_{\Omega} f \psi_i^e dx dy = \int_{\Delta} f_e \psi_i^e(x, y) dx dy = \int_{\Delta} f_e \cdot \frac{1}{2A_e} (\alpha_i^e + \beta_i^e x + \gamma_i^e y) dx dy = \frac{f_e A_e}{3}$$

2.2.3 Assembling of finite element equations

The assembling process is carried out in the following discrete problem as shown in figure 2.

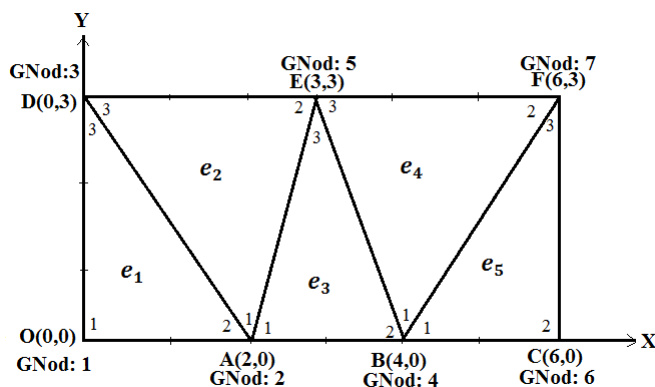


Figure 2: Discretization of rectangular domain into five triangular elements

In figure 2, the symbol 'GNod' stands for global node. A whole domain Ω is partitioned into five triangular elements with each element e_p for $p = 1, 2, 3, 4, 5$. The numbers inside the triangular vertices represent the element nodes rotated in counterclockwise direction. There are seven global nodes represented as 'GNod:N' for $N = 1, 2, 3, 4, 5, 6, 7$. The element equations of the five element mesh are written first by assuming the nodal degrees of freedom (NDF) is 1 per node. We have,

$$[K^e][u^e] = [f^e] + [Q^e] \Rightarrow [K_{ij}^e][u_j^e] = [f_i^e] + [Q_i^e] \quad (i, j = 1, 2, 3)$$

For triangular elements, the matrix form can be expanded as:

$$\begin{bmatrix} K_{11}^e & K_{12}^e & K_{13}^e \\ K_{21}^e & K_{22}^e & K_{23}^e \\ K_{31}^e & K_{32}^e & K_{33}^e \end{bmatrix} \begin{bmatrix} u_1^e \\ u_2^e \\ u_3^e \end{bmatrix} = \begin{bmatrix} f_1^e \\ f_2^e \\ f_3^e \end{bmatrix} + \begin{bmatrix} Q_1^e \\ Q_2^e \\ Q_3^e \end{bmatrix} \quad (19)$$

After assembling the element matrices for each elements, the following assembled matrix is obtained:

$$\begin{bmatrix}
 K_{11}^1 & K_{12}^1 & k_{13}^1 & 0 & 0 & 0 & 0 \\
 K_{21}^1 & K_{22}^1 + K_{11}^2 + K_{11}^3 & K_{23}^1 + K_{13}^2 & K_{12}^3 & K_{12}^2 + K_{13}^3 & 0 & 0 \\
 K_{31}^1 & K_{32}^1 + K_{31}^2 & K_{33}^1 + K_{33}^2 & 0 & K_{32}^2 & 0 & 0 \\
 0 & K_{21}^2 & 0 & K_{22}^2 + K_{11}^4 + K_{11}^5 & K_{23}^2 + K_{13}^4 & K_{12}^5 & K_{12}^4 + K_{13}^5 \\
 0 & K_{21}^2 + K_{31}^3 & K_{23}^2 & K_{32}^3 + K_{31}^4 & K_{22}^2 + K_{33}^3 + K_{33}^4 & 0 & K_{32}^4 \\
 0 & 0 & 0 & K_{21}^5 & 0 & K_{22}^5 & K_{23}^5 \\
 0 & 0 & 0 & K_{21}^4 + K_{31}^5 & K_{23}^4 & K_{32}^5 & K_{22}^4 + K_{33}^5
 \end{bmatrix} \cdot \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \\ U_7 \end{bmatrix} = \begin{bmatrix} F_1^1 \\ F_2^1 + F_1^2 + F_1^3 \\ F_3^1 + F_3^2 \\ F_2^3 + F_1^4 + F_1^5 \\ F_2^2 + F_3^3 + F_3^4 \\ F_2^5 \\ F_2^4 + F_3^5 \end{bmatrix} + \begin{bmatrix} Q_1^1 \\ Q_2^1 + Q_1^2 + Q_1^3 \\ Q_3^1 + Q_3^2 \\ Q_2^3 + Q_1^4 + Q_1^5 \\ Q_2^2 + Q_3^3 + Q_3^4 \\ Q_2^5 \\ Q_2^4 + Q_3^5 \end{bmatrix} \quad (20)$$

After the calculation of assembled matrix (18) using the coordinates as in figure 2, the coefficient matrix is obtained as[1]:

$$\frac{k}{36} \begin{bmatrix}
 39 & -27 & -12 & 0 & 0 & 0 & 0 \\
 -27 & 75 & -6 & -24 & -18 & 0 & 0 \\
 -12 & -6 & 32 & 0 & -14 & 0 & 0 \\
 0 & -24 & 0 & 75 & -18 & -27 & -6 \\
 0 & -18 & -14 & -18 & 64 & 0 & -14 \\
 0 & 0 & 0 & -27 & 0 & 39 & -12 \\
 0 & 0 & 0 & -6 & -14 & -12 & 32
 \end{bmatrix}$$

Similarly, the load vector ($f_e = f_0$) is:

$$[f^e] = \begin{bmatrix} f_1^1 \\ f_2^1 + f_1^2 + f_1^3 \\ f_3^1 + f_3^2 \\ f_2^3 + f_1^4 + f_1^5 \\ f_2^2 + f_3^3 + f_3^4 \\ f_2^5 \\ f_2^4 + f_3^5 \end{bmatrix} = f_0 \begin{bmatrix} 1 \\ 1 + \frac{3}{2} + 1 \\ 1 + \frac{3}{2} \\ 1 + \frac{3}{2} + 1 \\ \frac{3}{2} + 1 + \frac{3}{2} \\ 1 \\ \frac{3}{2} + 1 \end{bmatrix} = f_0 \begin{bmatrix} 1 \\ \frac{7}{2} \\ \frac{5}{2} \\ \frac{7}{2} \\ 4 \\ 1 \\ \frac{5}{2} \end{bmatrix}$$

To incorporate the Dirichlet's boundary condition, we consider,

$$\begin{aligned}
 u(0, y) = u(0, 3) = 0 &\Rightarrow U_1 = U_3 = 300 \\
 u(x, y) = u(6, 3) = 0 &\Rightarrow U_6 = U_7 = 0
 \end{aligned}$$

Also, the constant k and constant source f_0 are considered as:

$$k = 400 \quad f_0 = 3000$$

After imposing these boundary conditions, the assembled matrix for the system reduces to:

$$\begin{bmatrix} 75 & -24 & -18 \\ -24 & 75 & -18 \\ -18 & -18 & 64 \end{bmatrix} \begin{bmatrix} U_2 \\ U_4 \\ U_5 \end{bmatrix} = \begin{bmatrix} 947.25 + 9900 \\ 949.50 \\ 1080 + 4200 \end{bmatrix} = \begin{bmatrix} 10847.25 \\ 949.50 \\ 5280 \end{bmatrix}$$

Solving the matrix, we obtain,

$$U_2 = 230.6217, \quad U_4 = 130.6444, \quad U_5 = 184.1061.$$

Hence the required solution is:

$$U_1 = 300, \quad U_2 = 230.6217, \quad U_3 = 300, \quad U_4 = 130.6444, \quad U_5 = 184.1061, \quad U_6 = 0, \quad U_7 = 0.$$

2.3 Posteriori error estimation

The finite element approximation means to find the function $u_h \in V_h$ such that[2][10]:

$$B(u_h, w_h) = l(w_h) \quad \text{for all } w_h \in V_h \subset V \quad (21)$$

for all test functions $w_h \in V_h \subset V$. The error of the finite element approximation is denoted by

$$e_h = u - u_h \quad (22)$$

which satisfies the error representation

$$B(e_h, w) = B(u - u_h, w) = B(u, w) - B(u_h, w) = l(w) - B(u_h, w) = R(w)$$

where $R(\cdot)$ is called the residual functional or weak residual. Moreover, the standard orthogonality condition for the error in the Galerkin projection holds. That is, the choice of test functions is restricted to the finite element space, the fundamental Galerkin orthogonality condition follows,

$$R(w_h) = B(e, w_h) = 0, \quad \forall w_h \in V_h.$$

Assuming that the bilinear form is positive definite, it follows that the norm of the residual functional R is equal to the energy norm of the error,

$$\|R_h\|_{V'} = \sup_{v \in V(\Omega)} \frac{|R_h(w)|}{\|w\|_w} = \|e\|_E$$

where, $\|R_h\|_{V'}$ denotes the norm of the residual in the dual space $V'(\Omega)$.

2.3.1 Element residual method

Let the error on an element K be denoted by $e = u - U_h$. One finds that the error satisfies:

$$-\nabla e = f + \nabla u_h \quad \text{in } K. \quad (23)$$

There are various cases to consider. Suppose that the element K intersects a portion of the boundary of the domain Ω where an essential boundary condition is imposed. The appropriate boundary condition for the local error residual problem is clearly

$$e = 0 \quad \text{on } \partial K \cap \Gamma_D. \quad (24)$$

It has been assumed that the finite element approximation is constructed so that the essential boundary conditions are satisfied exactly. Suppose that the element intersects a portion of the boundary $\partial\Omega$ where a natural boundary condition is imposed. The local error residual problem is subjected to a natural boundary condition:

$$\frac{\partial e}{\partial \eta_K} = q_n - \frac{\partial u_h}{\partial \eta_K} \quad \text{on } \partial K \cap \Gamma_N$$

Consider the case when the element boundary lies on the interior of the domain. The first decision is whether to impose an essential or a natural boundary condition. The element residual method is based on using a natural boundary condition. Ideally, one would like to impose the condition[2]:

$$\frac{\partial e}{\partial \eta_K} = \frac{\partial u}{\partial \eta_K} - \frac{\partial u_h}{\partial \eta_K} \quad \text{on } K$$

on the edge separating elements K and J , where $u_h|_K$ denotes the restriction of the finite element approximation to an element K .

2.3.2 Elemental error calculation

The element-wise error calculation has been carried out for the mesh as in figure 2. The calculation of elemental gradient, estimated error of common edge, error calculation of Neumann boundary condition and error calculation of source term has been carried out.

The gradient of the triangle is given by the following relation:

$$\nabla u = grad(T) = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix} = \frac{1}{2A_T} \begin{bmatrix} y_j - y_k & y_k - y_i & y_i - y_j \\ x_k - x_j & x_i - x_k & x_j - x_i \end{bmatrix} \cdot \begin{bmatrix} U_i \\ U_j \\ U_k \end{bmatrix} \quad (25)$$

The gradients of each triangular element have been calculated as:

$$\begin{bmatrix} -34.68915 \\ 0 \end{bmatrix}, \begin{bmatrix} -38.6313 \\ -2.6281 \end{bmatrix}, \begin{bmatrix} -49.9886 \\ 1.1577 \end{bmatrix}, \begin{bmatrix} -61.3687 \\ -2.6357 \end{bmatrix}, \begin{bmatrix} -65.3222 \\ 0 \end{bmatrix}$$

$$M = \begin{bmatrix} E_1 \\ E_1 + E_2 \\ E_2 + E_3 \\ E_3 + E_4 \\ E_4 \end{bmatrix} = \begin{bmatrix} 291.8324 \\ 291.8324 + 1433.1920 \\ 1433.1920 + 1438.9650 \\ 1438.9650 + 293.4730 \\ 293.4730 \end{bmatrix} = \begin{bmatrix} 291.8324 \\ 1725.0244 \\ 2872.1570 \\ 1732.4380 \\ 293.4730 \end{bmatrix} \quad (26)$$

Dividing the matrix M by 2, we get a new matrix N such that:

$$N = \begin{bmatrix} 145.9162 \\ 862.5122 \\ 1436.0785 \\ 866.2190 \\ 146.7365 \end{bmatrix}.$$

The error calculation of the Neumann boundary condition has been performed by calculating the jump, unit normal, jump along unit normal and estimated error of the edge where the Neumann boundary condition occurs. After calculation, the following error vectors has been obtained:

$$\begin{bmatrix} 2500 \\ 62.1622 \\ 2373.8221 \\ 62.5222 \\ 2500 \end{bmatrix}$$

Adding the boundary error to the vector N , we get the following new vector P such that:

$$P = \begin{bmatrix} 145.9162 \\ 862.5122 \\ 1436.0785 \\ 865.7524 \\ 146.7365 \end{bmatrix} + \begin{bmatrix} 2500 \\ 62.1622 \\ 2373.8221 \\ 62.5222 \\ 2500 \end{bmatrix} = \begin{bmatrix} 2645.9162 \\ 924.6744 \\ 3809.9006 \\ 928.2746 \\ 2646.7365 \end{bmatrix}.$$

Taking the square root of each entries of P , then the resulting vector Q (say) such that:

$$Q = \begin{bmatrix} 51.4385 \\ 30.4676 \\ 61.7244 \\ 30.4676 \\ 51.4464 \end{bmatrix} \quad (27)$$

Now, the oscillation of f over each triangular element has been calculated. It deals with the calculation of residuals over the triangular element. In general, suppose that $T \in \tau$ be a triangular element on the mesh. Consider three random points p_1, p_2, p_3 and set the relation[9]:

$$\|f - f_h\|_{0,T}^2 \approx \frac{\|T\|}{3} \sum_{i=1}^3 |(f - f_h)(p_i)|^2$$

by the concept of this relation, in this problem, the mid points of sides of the triangle p_1, p_2, p_3 have been taken. The error of f over T is given by:

$$\text{Error of } f \text{ over } T = \sqrt{\left(\frac{Ar(T)}{3}\right) [(f(p_1))^2 + (f(p_2))^2 + (f(p_3))^2]}.$$

Since the value of $f = 3000$, which is a constant, in our case, we have:

$$f(p_1) = f(p_2) = f(p_3) = 3000.$$

Hence the error produced by sources can be presented as in the following matrix form:

$$f = \begin{bmatrix} f_{e_1} \\ f_{e_2} \\ f_{e_3} \\ f_{e_4} \\ f_{e_5} \end{bmatrix} = \begin{bmatrix} 5196.1524 \\ 6363.9610 \\ 5196.1524 \\ 6363.9610 \\ 5196.1524 \end{bmatrix}.$$

Therefore, the estimated global error is obtained by adding the vector Q and f such that:

$$\eta_\tau = Q + f = \begin{bmatrix} 51.4385 \\ 30.4676 \\ 61.7244 \\ 30.4676 \\ 51.4464 \end{bmatrix} + \begin{bmatrix} 5196.1524 \\ 6363.9610 \\ 5196.1524 \\ 6363.9610 \\ 5196.1524 \end{bmatrix} = \begin{bmatrix} 5247.5909 \\ 6394.4286 \\ 5257.8768 \\ 6394.4286 \\ 5247.5988 \end{bmatrix} = \begin{bmatrix} T_{e_1} \\ T_{e_2} \\ T_{e_3} \\ T_{e_4} \\ T_{e_5} \end{bmatrix}.$$

The total error contribution of each element has been displayed. It helps us to refine the mesh by selecting

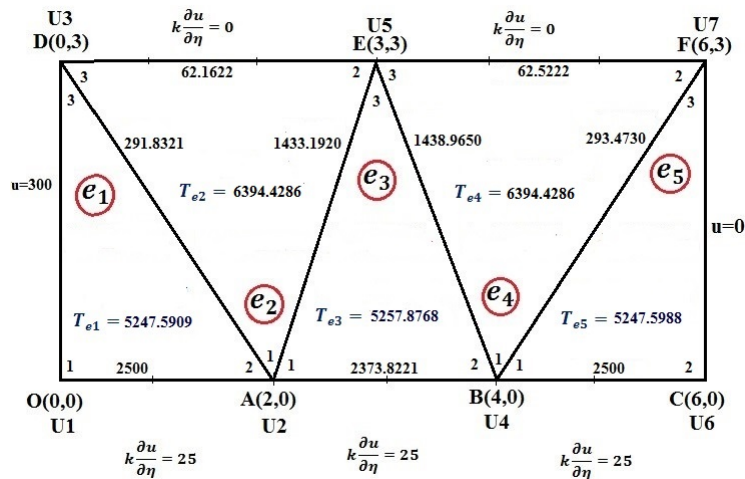


Figure 3: Illustration of errors for triangular element

the largest error element corresponding to the side of largest error. The mesh refinement process has been discussed as in the figure below[3].

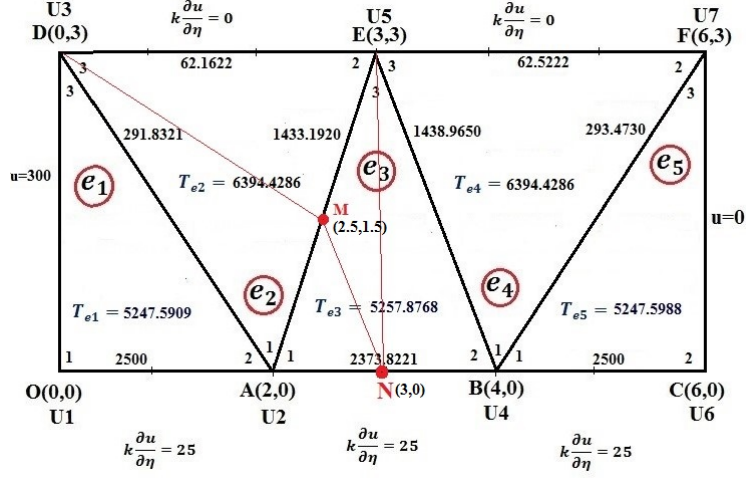


Figure 4: Illustration of adaptive process by bisection approach

3 Results

3.1 AFEM for poisson equation

Some graphical illustrations of adaptive finite element method regarding to Poisson equation have been presented for the analysis of adaptive mesh refinement process. To refine mesh, adaptive algorithm have been applied to get better approximation. The results obtained by mesh refinement process up to tenth iteration have been displayed in Figures (5-17).

3.2 AFEM: Heat equation

A steady state heat equation in two different boundary conditions 1. Neumann and Dirichlet on two adjacent sides and 2. different Dirichlet conditions on two adjacent sides, has been taken for application of AFEM.

3.2.1 Problem-1

Consider steady state heat conduction in an isotropic rectangular region of dimension $3a \times 2a$. The origin of x and y coordinates is taken at the lower left corner such that x is parallel to side $3a$ and y is parallel to side $2a$. The boundaries $x = 0$ and $y = 0$ are insulated, the boundary $x = 3a$ is maintained at zero temperature, and the boundary $y = 2a$ is maintained at a temperature[8]:

$$T = T_0 \cos\left(\frac{\pi x}{6a}\right)$$

Consider a governing equation for steady state heat transfer in plain system:

$$-\frac{\partial}{\partial x} \left(k_x \frac{\partial T}{\partial x} \right) - \frac{\partial}{\partial y} \left(k_y \frac{\partial T}{\partial y} \right) = f(x, y) \quad (28)$$

where, T = Temperature in $^{\circ}C$, k_x and k_y are thermal conductivities in $W/(m \cdot ^{\circ}C)$ along x and y directions respectively, and f is the internal heat generation per unit volume in W/m^3 . Setting, $k_x = k_y = k$, a constant thermal conductivity and $f = 0$,

$$-k \nabla^2 T = -k \Delta T = 0 \quad (29)$$

For $a = 1$, the domain will be a 3×2 rectangle. The exact solution is:

$$T(x, y) = T_0 \frac{\cosh\left(\frac{\pi y}{6a}\right) \cos\left(\frac{\pi x}{6a}\right)}{\cosh\left(\frac{\pi}{3}\right)} \quad (30)$$

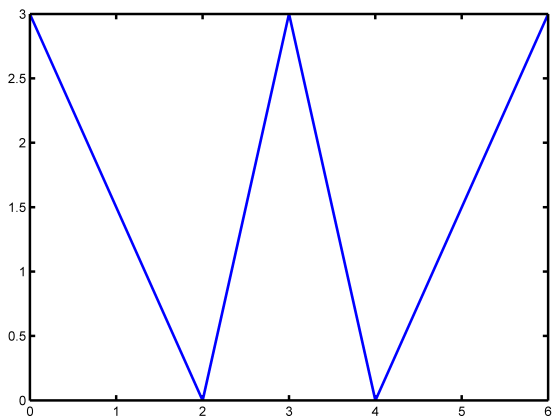


Figure 5: Initial discretization

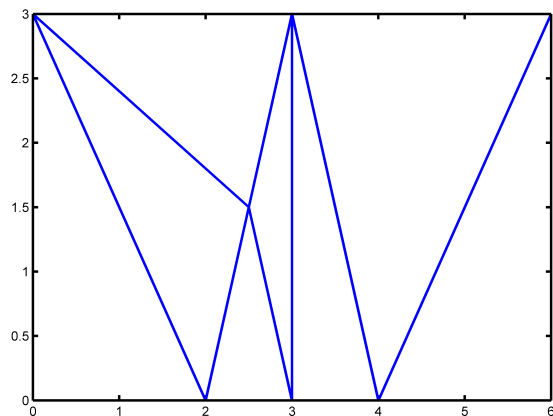


Figure 6: Mesh refinement: Iteration-I

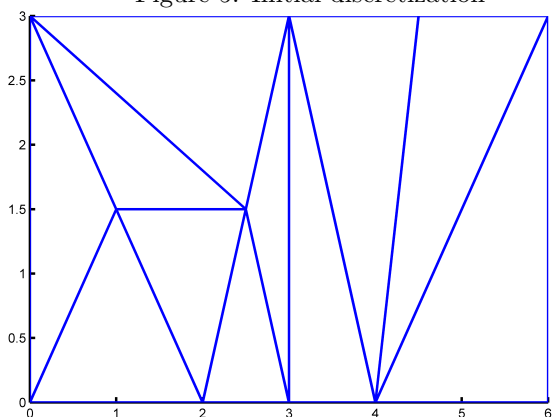


Figure 7: Mesh refinement: Iteration-II

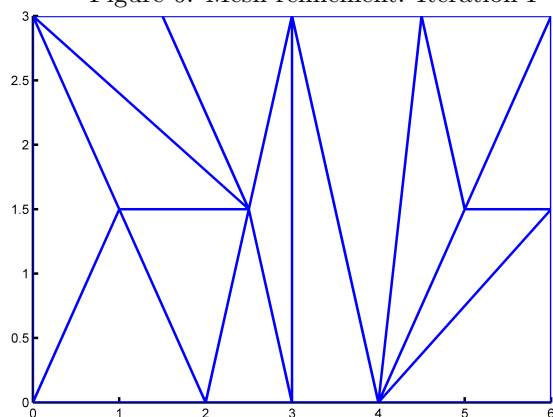


Figure 8: Mesh refinement: Iteration-III

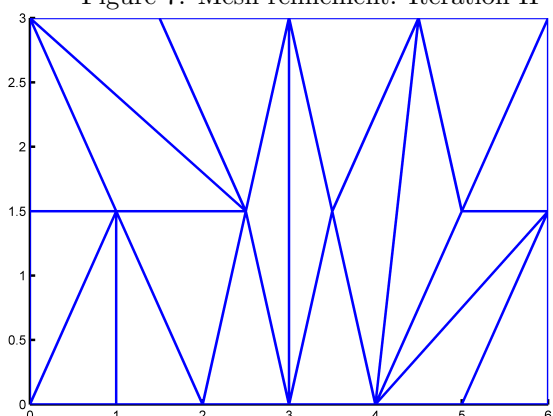


Figure 9: Mesh refinement: Iteration-IV

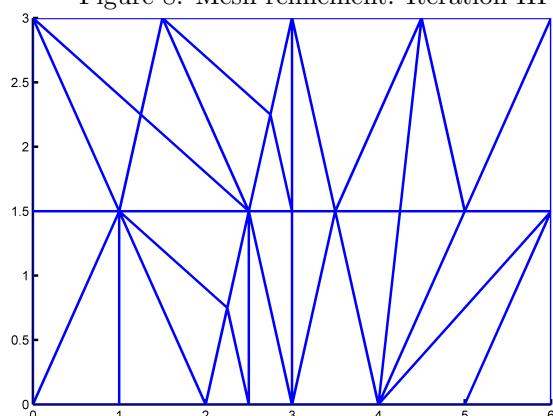


Figure 10: Mesh refinement: Iteration-V

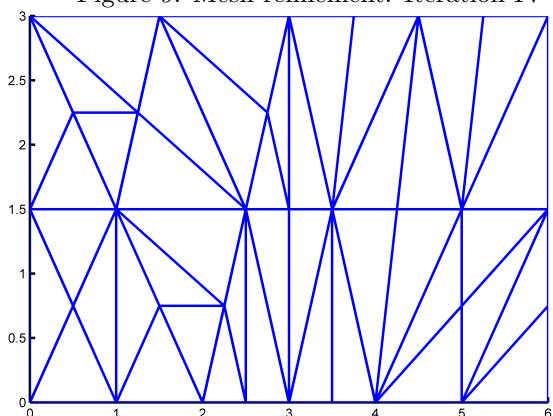


Figure 11: Mesh refinement: Iteration-VI

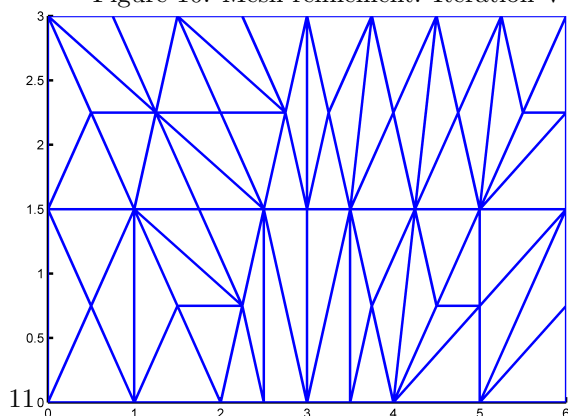


Figure 12: Mesh refinement: Iteration-VII

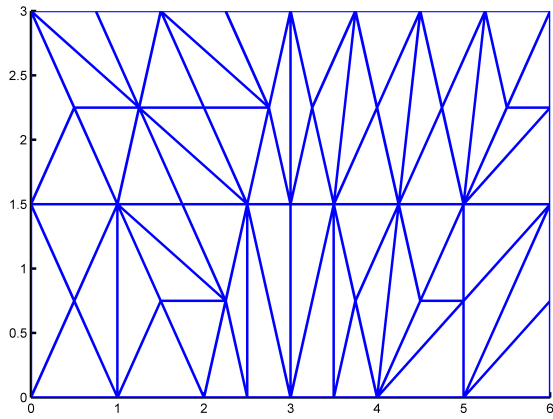


Figure 13: Mesh refinement: Iteration-VIII

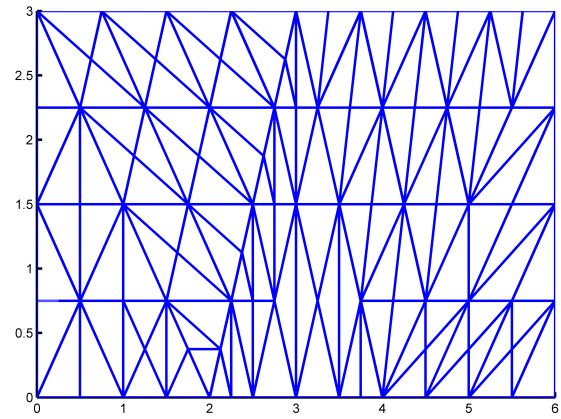


Figure 14: Mesh refinement: Iteration-IX

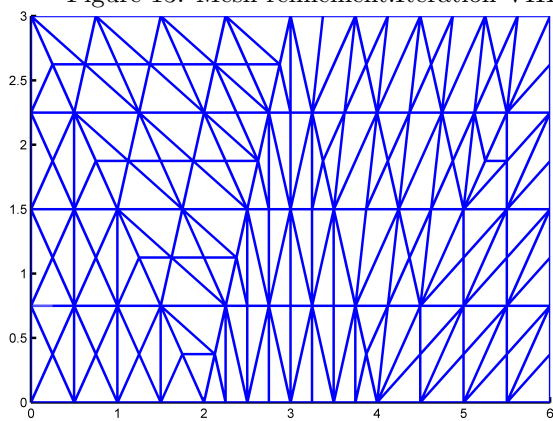


Figure 15: Mesh refinement: Iteration-X

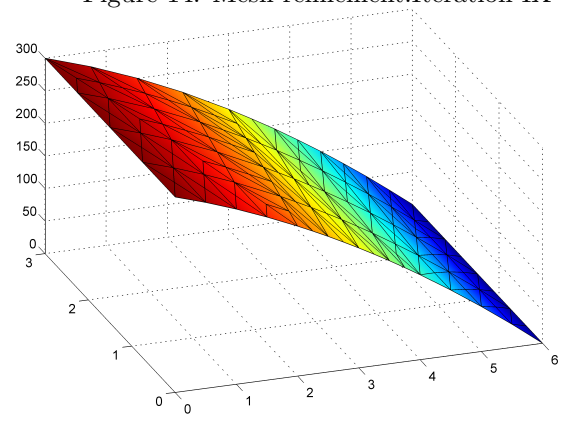


Figure 16: Solution of (2) in space

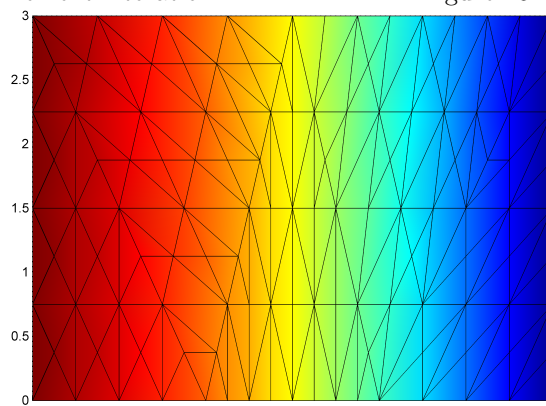


Figure 17: Solution of (2) in xy -plane

Table 1: Comparison between adaptive finite element solution and exact solution

Vertices	Initial Solution	1	2	3	4	5	6
(0,0)	127.3276	127.8006	126.0245	127.4020	126.4231	128.8914	126.1265
(1,0)	110.2354	111.2647	109.3742	110.9760	110.9581	110.2601	110.1882
(2,0)	63.6371	64.2721	63.4234	63.6998	64.3218	64.0525	63.1856
(0,1)	144.3573	144.2740	142.6124	144.1633	141.3133	143.0034	143.2004
(1,1)	124.9886	126.4932	124.0244	123.9808	124.0298	123.9862	123.9058
(2,1)	72.1564	72.9118	72.1598	72.2687	72.5807	71.5961	71.5303

7	8	9	10	Exact Solution
125.6032	125.4859	124.4630	125.4995	124.9776
108.9958	108.9203	108.8791	108.7331	108.2338
63.4587	62.8560	62.8118	62.7012	62.4888
143.0605	142.9089	142.5663	142.7349	142.5042
123.9974	123.9338	123.7362	123.5681	123.4123
71.6943	71.4603	71.5834	71.3307	71.2521

The above table shows that there is big differences between FEM solution and exact solution at nodal points. But when the adaptive process is applied successively up to tenth iteration, we can see that the AFEM solution approaches to the exact solution. The results up to 10th iteration of an adaptive mesh refinement are displayed below: [See fig (18)-(30)]

3.2.2 Problem-2

Consider a heat transfer in a rectangular region of dimensions a by b , subjected to the boundary conditions. We wish to write the finite element algebraic equations for the unknown nodal temperatures and heats. For illustrative purposes, a 4×2 mesh of rectangular elements is chosen. We assume that the medium is orthotropic, with conductivities k_x and k_y in the x and y directions, respectively. The boundaries $x = 0$ and $y = 0$ are insulated such that[8]:

$$-\frac{\partial T}{\partial x} = -\frac{\partial T}{\partial y} = 0$$

Also, the boundary $x = a$ is maintained at zero temperature and the boundary $y = b$ is maintained at the temperature $T = T_0 = 200^\circ$. No internal heat generation is assumed.

The results up to 10th iteration of an adaptive mesh refinement are displayed below: [Fig (31)-(43)]

4 Conclusion

The study has been conducted to get the particular solution of Poisson equation using FEM and AFEM in an arbitrary domain. The results obtained by AFEM are very close to the real solution. The FEM solution of Poisson equation may not represent the exact solution, however, AFEM helps to reduce the error size to get better solution that is approximately approaches to exact solution. Since AFEM helps to decrease the error, we can extend and apply the AFEM technique to find the better numerical solution in various applied problems.

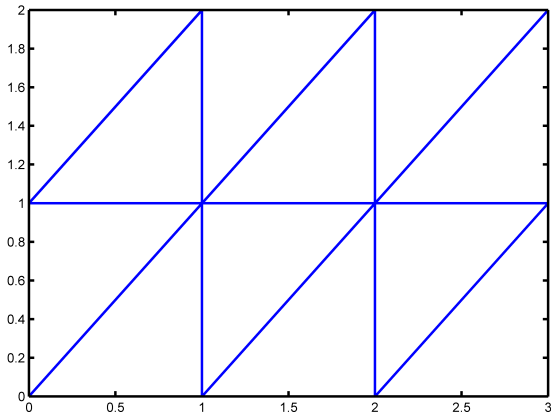


Figure 18: Initial discretization

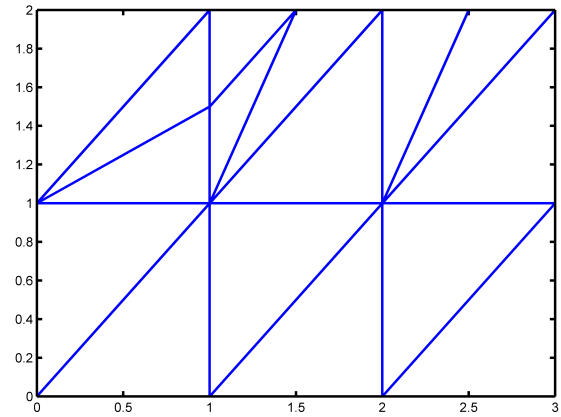


Figure 19: Mesh refinement: Iteration-I

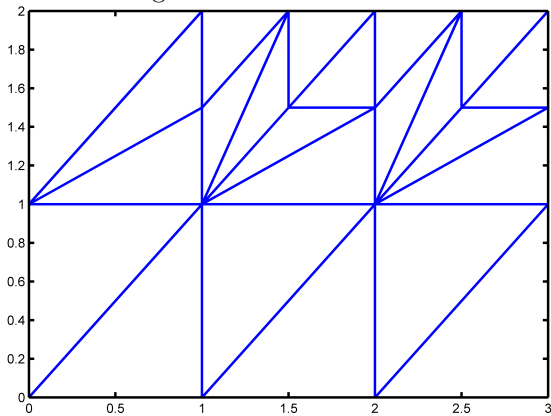


Figure 20: Mesh refinement: Iteration-II

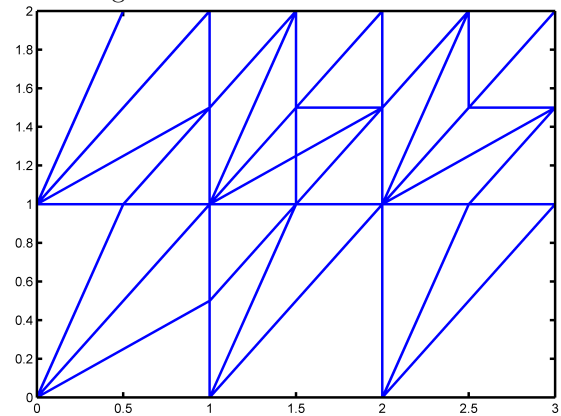


Figure 21: Mesh refinement: Iteration-III

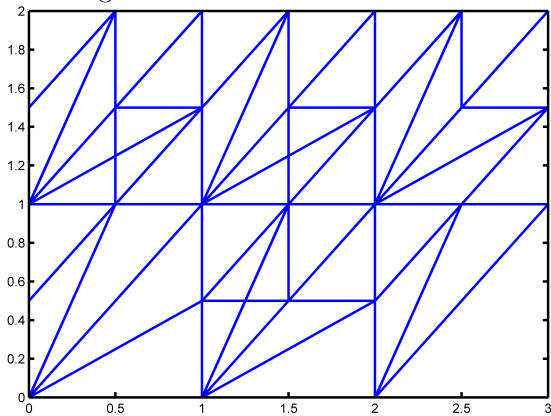


Figure 22: Mesh refinement: Iteration-IV

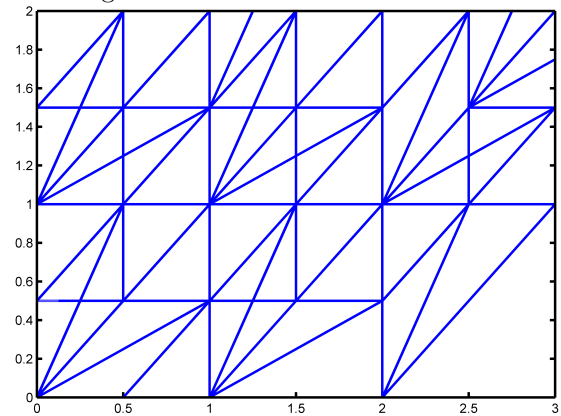


Figure 23: Mesh refinement: Iteration-V

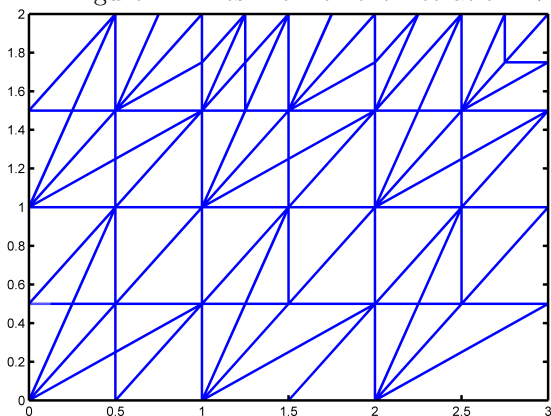


Figure 24: Mesh refinement: Iteration-VI

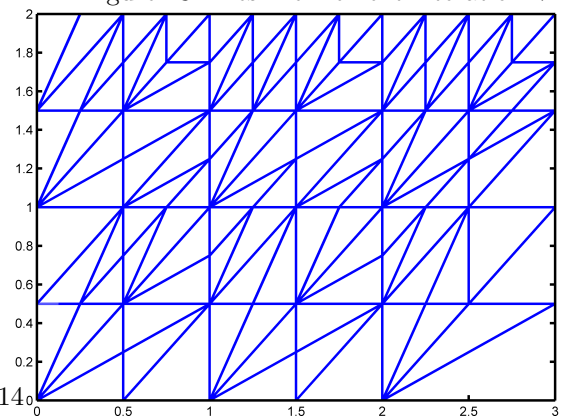


Figure 25: Mesh refinement: Iteration-VII

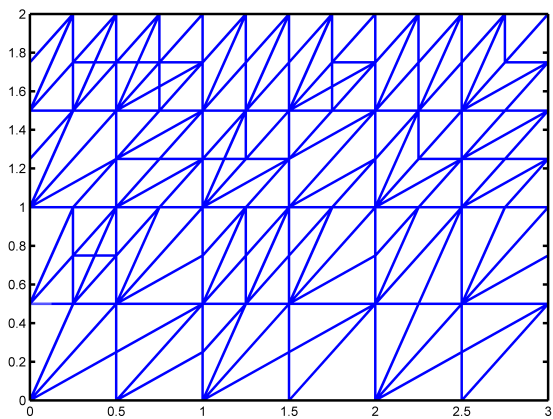


Figure 26: Mesh refinement: Iteration-VIII

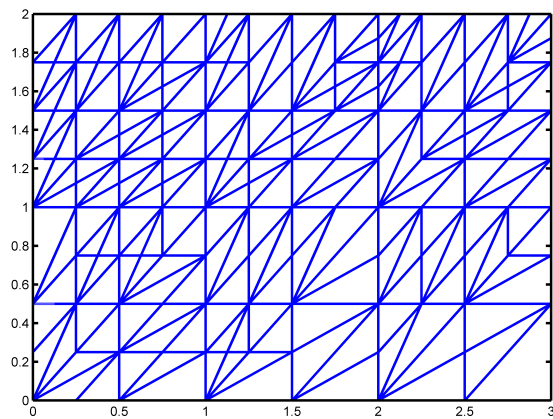


Figure 27: Mesh refinement: Iteration-IX

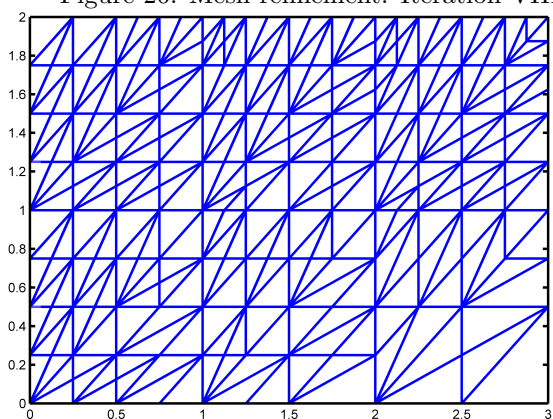


Figure 28: Mesh refinement: Iteration-X

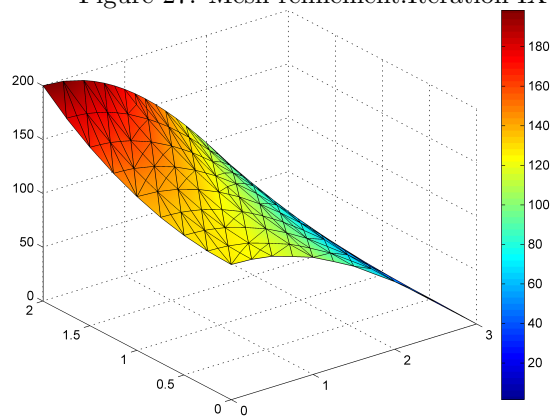


Figure 29: Solution of space

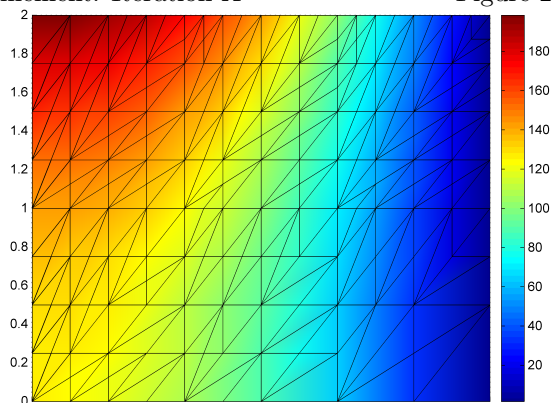


Figure 30: Solution of in xy -plane

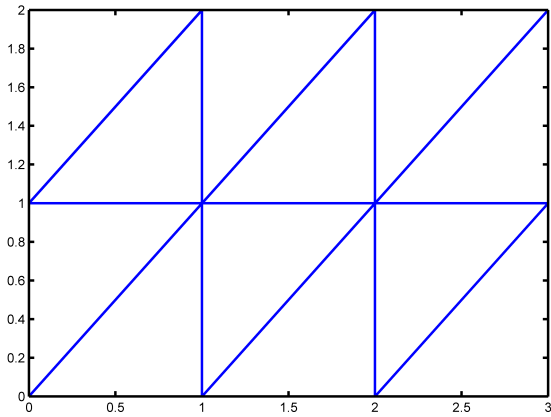


Figure 31: Initial discretization

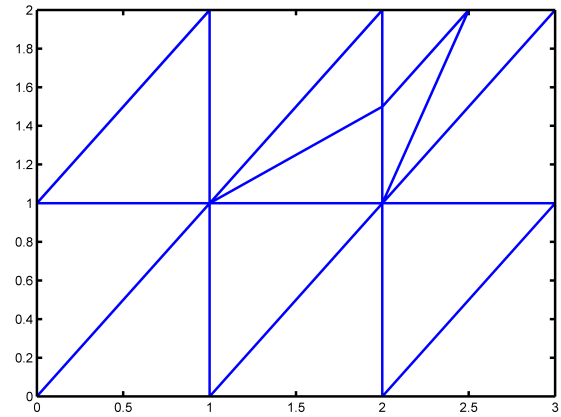


Figure 32: Mesh refinement: Iteration-I

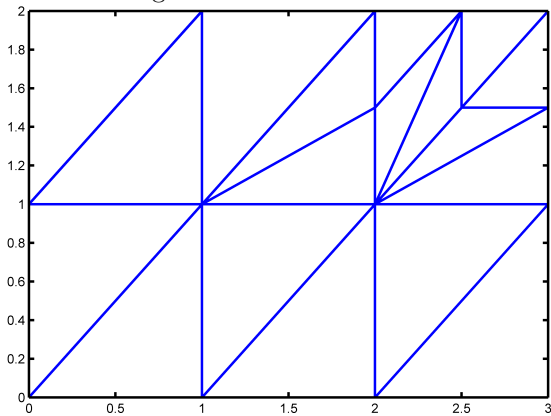


Figure 33: Mesh refinement: Iteration-II

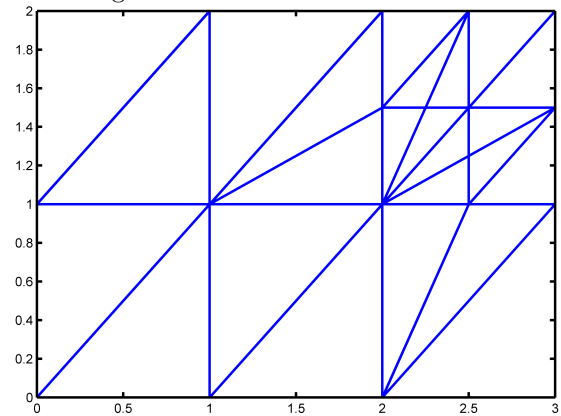


Figure 34: Mesh refinement: Iteration-III

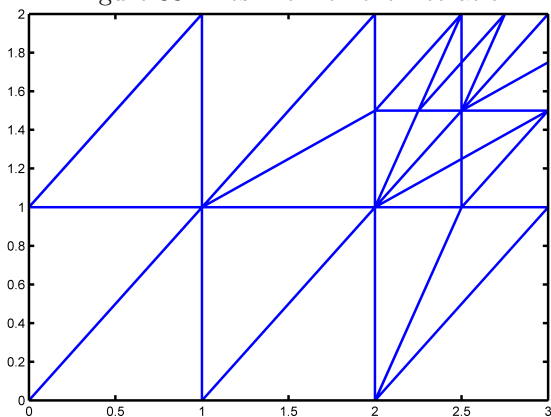


Figure 35: Mesh refinement: Iteration-IV

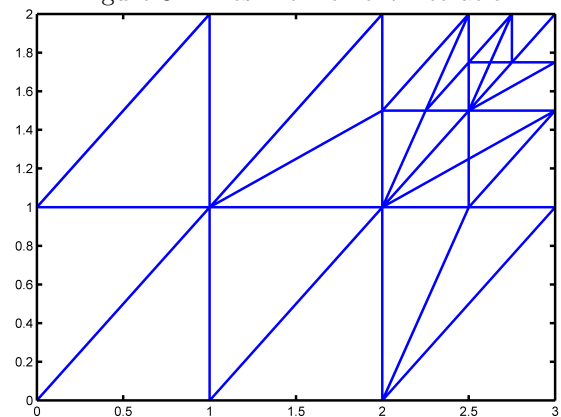


Figure 36: Mesh refinement: Iteration-V

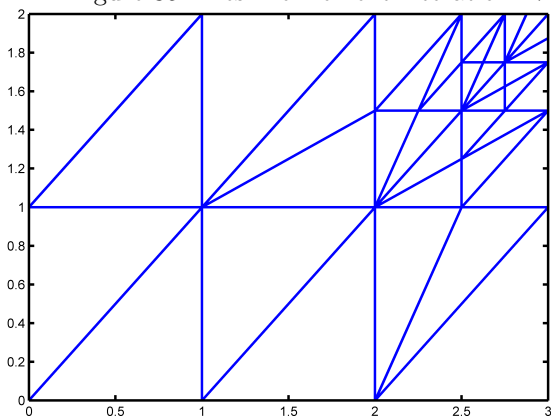


Figure 37: Mesh refinement: Iteration-VI

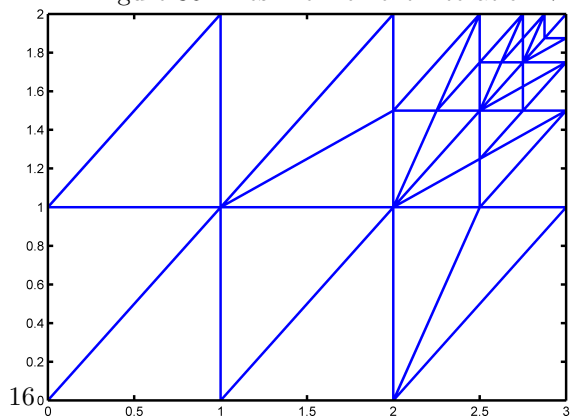


Figure 38: Mesh refinement: Iteration-VII

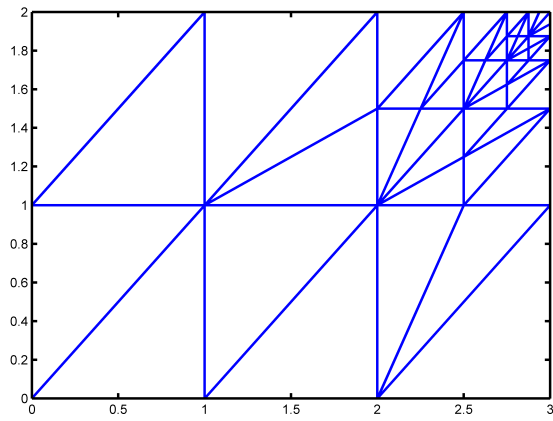


Figure 39: Mesh refinement:Iteration-VIII

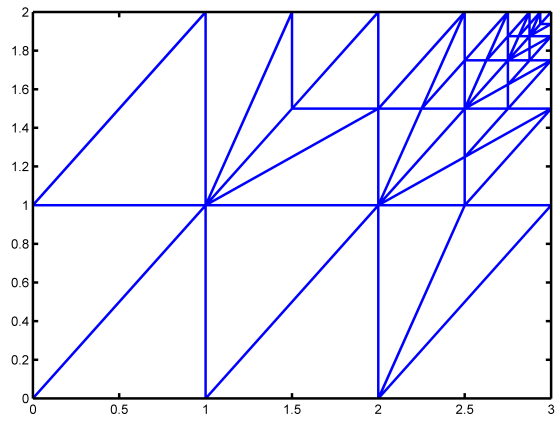


Figure 40: Mesh refinement:Iteration-IX

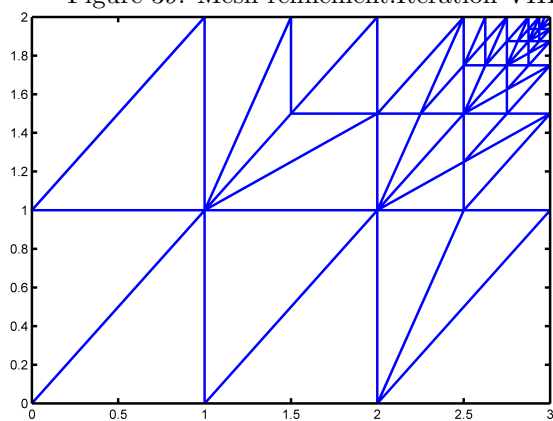


Figure 41: Mesh refinement:Iteration-X

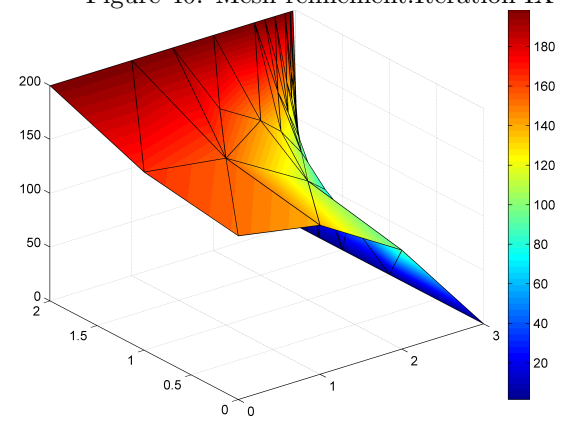


Figure 42: Solution of in space

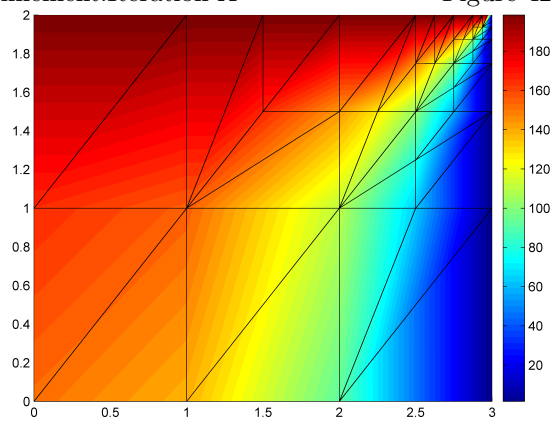


Figure 43: Solution of in xy -plane

References

- [1] Agbezuge, L., 2006, *Finite element solution of the poisson equation with dirichlet boundary conditions in a rectangular domain*, Rochester Institute of Technology, Rochester, NY.
- [2] Ainsworth, M. and Oden, J. T., 1997, A posteriori error estimation in finite element analysis, *Computer Methods in Applied Mechanics and Engineering*, 142, 1-88.
- [3] Dorfler, W., 1996, A convergent adaptive algorithm for poisson's equation, *SIAM Journal on Numerical Analysis*, 33, 1106-1124.
- [4] Gokul, K.C., Gurung, D.B. and Adhikary, P. R., 2012, Fem approach for one dimensional temperature distribution in the human eye, *Proceedings of National Conference on Mathematics*, 35-46.
- [5] Gratsch, T. and Bathe, K., 2005, A posteriori error estimation techniques in practical finite element analysis, *Computers & structures*, 83, 235-265.
- [6] Mebrate, B., Rao, K. P., 2015, Numerical solution of a two dimensional poisson equation with dirichlet boundary conditions, *American Journal of Applied Mathematics*, 3, 297-304.
- [7] Nochetto, R. H., Siebert, K.G. and Veese, A., 2009, Theory of adaptive finite element methods: an introduction, *Multiscale, nonlinear and adaptive approximation*, Springer, 409-542.
- [8] Reddy, J.N., 1993, *An introduction to the finite element method*, Tata McGraw Hill Education, New Delhi, India.
- [9] Verfurth, V., 2013, *A posteriori error estimation techniques for finite element methods*, Oxford University Press, New Delhi, India.
- [10] Wazwaz, M.J., 2017, *A posteriori error estimates for elliptic partial differential equations in the finite element method*. Master's thesis, Hebron University, Hebron, Palestine.