Numerical Solution of a Two Dimensional Poisson Equation with Dirichlet Boundary Conditions

Benyam Mebrate, Purnachandra Rao Koya*

School of Mathematical and Statistical Sciences, Hawassa University, Hawassa, Ethiopia

Email address:
benyam134@gmail.com (B. Mebrate), drkpraocecc@yahoo.co.in (P. R. Koya)

To cite this article:
Benyam Mebrate, Purnachandra Rao Koya. Numerical Solution of a Two Dimensional Poisson Equation with Dirichlet Boundary Conditions. American Journal of Applied Mathematics. Vol. 3, No. 6, 2015, pp. 297-304. doi: 10.11648/j.ajam.20150306.19

Abstract: In this paper we have introduced Numerical techniques to solve a two dimensional Poisson equation together with Dirichlet boundary conditions. Specifically two methods are used for the purpose of numerical solution, viz. Finite difference method and Finite element method. The implementation of the solutions is done using Microsoft Office Excel worksheet or spreadsheet. The numerical solutions obtained by these two methods are also compared with each other graphically in two and three dimension.

Keywords: Dirichlet Boundary Conditions, Finite Difference Method, Finite Element Method, Poisson Equation, Spreadsheet

1. Introduction

Differential equations arise in many areas of applications such as in Sciences and Engineering. Most of the problems in almost all the scientific and engineering areas are modeled using partial differential equations. Partial differential equations arise when a dependent variable depends on two or more independent variables. Usually analytic solutions for these partial differential equations are difficult to compute and hence alternative methods are used. Numerical methods are widely used to solve partial differential equations within the given domain together with boundary conditions.

In general the domain is divided in to finite number of elements or the domain is made discretization into small triangles or rectangles. The values of the dependent variable at the nodes or grid points of these triangles or rectangles are computed. In literature finite difference and finite element methods are applied to solve two dimensional Laplace equations with Dirichlet boundary conditions using spreadsheet [8]. This motivates us to use spreadsheet implementation for solving two dimensional Poisson equations with Dirichlet boundary condition.

In Section 2, we introduced and discussed the two dimensional Poisson equation with Dirichlet boundary conditions. In Section 3, the finite difference method is applied for solving Poisson equation with Dirichlet boundary conditions. Also, we have made the use of spreadsheet implementation. In Section 4, the finite element method is applied for solving Poisson equation with Dirichlet boundary conditions. Also, we have made the use of spreadsheet implementation. In Section 5, we have compared the results obtained in Sections 3 and 4. The paper ends with concluding remarks in Section 6.

2. Poisson Equation and the Problem Definition

The Poisson equation is given by \[ 1 – 2, 4, 6, \text{ and } 10 \]

\[ \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) \text{or} \Delta u = f(x, y) \quad (1) \]

Here the Laplace operator denotes \[ \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \]. Equation (1) is elliptic second order linear partial differential equation. In (1), the dependent variable \( u(x, y) \) is a function of its arguments and depends on the independent variables \( x \) and \( y \). The function \( f(x, y) \) is known as a source function. For the purpose of numerical solution of the Poisson equation (1) we consider a plane region defined by \( R = \{ (x, y) : 0 \leq x \leq 1 \text{ and } 0 \leq y \leq 1 \} \). We will also impose Dirichlet boundary conditions for the dependent variable \( u(x, y) \) on the region \( R \) defined as

\[ u(0, y) = 0, u(x, 0) = 0, \]
The region \( R \) together with the Dirichlet boundary conditions (2) is illustrated in Figure 1. The \( x \)-variable increases along the positive horizontal axis while the \( y \)-variable increases along the positive vertical axis.

\[ u(x, 1) = 0, \quad u(1, y) = 0 \quad \text{(2)} \]

Here our problem is to find numerical values of the function \( u(x, y) \) at the interior node points of the region \( R \) provided that the Poisson equation (1) and the Dirichlet conditions given in (2) are satisfied.

### 3. Finite Difference Method

The finite difference techniques are based on the replacement of differential equations with approximately equivalent finite difference equations. Whenever solving differential equations analytically is not easy then the differential equations are replaced by corresponding finite difference equations and those will be solved. Further, the differential equations provide exact values while the difference equations provide approximate values for the variables. Approximation of the solution relates to the number of grids given region is divided into. More are the grids and better is the approximation.

The finite difference equations are algebraic in nature and their approximate solutions are related to number of grid points of the region \( R \). A finite difference solution basically involves the following three steps:

(i) Divide the solution region using grids and nodes into small rectangles

(ii) Approximate the given differential equation by equivalent finite difference equations that relate the solutions to the grid points.

(iii) Solve the difference equations subject to the prescribed boundary and/or initial conditions.

To find the solution of the function \( u(x, y) \) on the region \( R \), we divide the solution region \( R \) of \( xy \) plane into equal rectangles or meshes of sides \( h \) and \( k \) along \( x \) and \( y \) directions respectively, as shown in Figure 2. We now construct a finite difference equation to represent equation (1) and boundary conditions (2). This will help us to calculate the values of \( u(x, y) \) at the nodes of the rectangles in the region \( R \). The detailed derivation of finite difference equation from (1) is given in Appendix 1. Here we directly consider the difference equation as

\[ u_{i,j} = \frac{1}{4}(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - h^2 f_{i,j}), \quad i, j = 2, 3, 4 \quad \text{(3)} \]

Also, the boundary conditions as

\[ u(x_i, y_1) = 0, u(x_i, y_2) = 0, u(x_1, y_j) = 0, u(x_5, y_j) = 0, \quad i, j = 1, 2, 3, 4, 5 \quad \text{(4)} \]

Equation (3) represents a difference equation and (4) represents the boundary conditions equivalent respectively to the differential equation (1) and the boundary conditions (2). Here in (3) we have used the notations as \( u_{i,j} = u(x_i, y_j) \) and \( f_{i,j} = f(x_i, y_j) \). Also, the set \( \{x_1, x_2, x_3, x_4, x_5\} \) is the set of partition points of the interval \([0, 1]\) along \( x \)-axis with step length \( h \). Similarly, the set \( \{y_1, y_2, y_3, y_4, y_5\} \) is the set of partition points of the interval \([0, 1]\) along \( y \)-axis with step length \( k \). Of course in our present study we choose \( h = k \).
numerical solutions for \( u(x, y) \) is computed at all the nine free nodes of the region \( R \) located at the locations \( (i, j); i = 2, 3, 4 \) and \( j = 2, 3, 4 \). These nine free nodes are numbered as shown in Figure 3.

Gauss Seidel method is applied to solve (3) numerically for the specific value of the function \( f(x, y) = 1 \). The function \( f(x, y) \) in fact admits any constant or expression in terms of \( x \) and \( y \). For simplicity and to illustrate the procedure of numerical solution here we have selected that \( f(x, y) = 1 \). Other expressions for \( f(x, y) \) will be considered as and when it demands in our further work.

At the nodal points as shown in Figure 3 we compute the successive approximate solutions for \( u(x, y) \) using spreadsheet. The iterations and the approximations of the solutions are provided in a tabular form in Table 1. The Gauss – Seidel iterations for every node are continued till the approximate solution values converge to a constant.

![Figure 3. Numbered free nodes of the region R.](image)

| Iterations | Node numbers |
|------------|--------------|
| 7          | 0            |
| 8          | -0.0156      |
| 9          | -0.0195      |
| 12         | -0.0205      |
| 13         | -0.0205      |
| 14         | -0.0271      |
| 17         | 0            |
| 18         | 0            |
| 19         | 0            |
| 2          | 0            |
| 3          | -0.0156      |
| 4          | -0.0195      |
| 5          | -0.0205      |
| 6          | -0.0271      |
| 7          | -0.0195      |
| 8          | -0.0195      |
| 9          | -0.0205      |
| 10         | -0.0205      |
| 11         | -0.0271      |
| 12         | -0.0195      |
| 13         | -0.0195      |
| 14         | -0.0205      |
| 15         | -0.0205      |

**Table 1. Solution using Gauss Seidel method.**

4. Finite Element Method

Finite element methods are essentially methods for finding approximate solutions of partial differential equations defined in a finite region or domain. Finite element method involves the following steps in solving a partial differential equation together with boundary conditions:

i. Divide the domain in to a finite number of elements.

ii. Drive the weak formulation corresponding to the given problem.

iii. Calculate the stiffness matrix and load vector for each element in the domain.

iv. Calculate global stiffness matrix and assemble.

v. Solve the resulting system of algebraic linear equations subject to the satisfaction of the boundary conditions.

The given domain \( R \) is divided into 32 congruent right angled and isosceles triangles. The nodes of the triangles are represented using unenclosed numbers. These unenclosed numbers are called global node numbers. Similarly, the triangles are represented by en-rectangle numbers. These enclosed numbers are called element numbers. The en-rectangle number is a number that is enclosed by a rectangle. These details are shown in Figure 4.

![Figure 4. Division of domain R into isosceles and right angled triangles.](image)
The weak formulation of the problem defined in (1) and (2) requires that the function \( u(x, y) \) (i) is a member of Sobolev space \( H^1_0(R) \) i.e., \( u \in H^1_0(R) \) and (ii) has to satisfy the condition 
\[
-\int_R \nabla u \cdot \nabla v \, dR = \int_R f v \, dR, \quad \forall v \in H^1_0(R) .
\]

Here Sobolev space \( H^1_0(R) \) is defined as a set of functions \( v \) satisfying the conditions (i) \( \int_R w^2 \, dR < \infty \) (ii) 
\[
\int_R (w_x)^2 \, dR < \infty \quad \text{(iii)} \quad \int_R (w_y)^2 \, dR < \infty \quad \text{and (iv)} \quad w = 0 \text{ on the boundary of the region } R.
\]

The detailed derivation of weak formulation of equation (1) is given in appendix 2.

The finite element problem for the approximation of Poisson equation (1) together with boundary condition (2) requires finding \( u_d \in V_d \) satisfying the following condition:
\[
-\int_R \nabla u_d \cdot \nabla v_d \, dR = \int_R f v_d \, dR, \quad \forall v_d \in V_d . \tag{5}
\]

Here in (5) the space \( V_d \) approximates \( H^1_0(R) \) and defined as
\[
V_d = \left\{ v_d \in C(\tilde{R}); \quad v_d|_{K_m} \in \mathbb{P}_r, \quad v_d|_{\partial R} = 0, \quad \forall m \in \{1, 2, \ldots, 32\} \right\}
\]

Where \( C(\tilde{R}) \) is the space of continuous function on the closure of \( R, \text{ that is } \tilde{R} \).
\( \partial R \) is the boundary of the region \( R \).
\( \mathbb{P}_r \) is the space of polynomials of degree less than or equal to
\( K_m \) is the \( m \)th triangular element of the region \( R \).
\( d = \max_{m \in \{1, 2, \ldots, 32\}} \text{diameter}(K_m), \text{diameter}(K_m) \)
\[
= \max_{x, y \in K_m} |x - y|.
\]

Furthermore, each function \( v_d \in V_d \) is characterized, uniquely, by the values it takes at the nodes \( N_j, \forall j = 1, 2, 3 \) of the triangular elements. A basis in the space \( V_d \) can be set of the characteristic Lagrangian functions \( \varphi_i \in V_d \), \( i = 1, 2, 3 \), such that
\[
\varphi_i(N_j) = \begin{cases} 0, & i \neq j \in \{1, 2, 3\}, \text{and } j \text{ of } K_m, \\ 1, & j = i. \end{cases}
\]

The function \( \varphi_i \) is called shape function. The function \( v_d \in V_d \) can be expressed through a linear combination of the basis functions of \( V_d \) in the following way
\[
v_d(x, y) = \sum_{j=1}^{3} v_j \varphi_j(x, y), \quad v_j = v_h(N_j) \tag{6}
\]

By expressing the discrete solution \( u_d \) in terms of the basis \( \varphi_i \), we have
\[
u_d(x, y) = \sum_{i=1}^{3} u_i \varphi_i(x, y), \quad v_i = v_h(N_i) \tag{7}
\]

We now assign numbers 1, 2, and 3 to the nodes of each triangle element. Starting with right angled vertex, number the nodes in anti clockwise direction using 1, 2, and 3 respectively. This numbering procedure will help us in having all elements the same stiffness matrix since these elements have the same geometry. The detailed derivation of equation of stiffness matrix and load vector is given in appendix 3. Here we consider only equitation of stiffness matrix and load vector. The stiffness matrix for each triangular element is calculated using the relation
\[
A_{K_m}^{K_m} = -\int_{K_m} \nabla \varphi_i \cdot \nabla \varphi_j \, dx \, dy, \quad i, j = 1, 2, 3
\]

\( K_m \) is the \( m \)th triangular element in the domain \( R \).
Here \( m = 1, \ldots, 32 \).

\( A_{K_m}^{K_m} \) is Stiffness matrix of the \( m \)th triangular element.
\( A_{K_m}^{K_m} \) is (the \( i, j \) entry of the stiffness matrix \( A_{K_m}^{K_m} \)).
\( \varphi_i \) is shape functions at the nodes of the elements \( K_m \).
Here \( i, j = 1, 2, 3 \).

Thus the stiffness matrix for each element is

| Local node numbers | 1 | 2 | 3 |
|-------------------|---|---|---|
| 1                 | -1 | 0.5 | 0.5 |
| 2                 | 0.5 | -0.5 | 0 |
| 3                 | 0.5 | 0 | -0.5 |

The \( (p, q) \) entry of the global matrix of element \( K_m \) takes the \( (i, j) \) entry of the stiffness matrix of element \( K_m \) where \( \text{p, q = 1, 2, ..., 25 are global node numbers, and } i, j = 1, 2, 3 \) are corresponding local node numbers for the global node numbers and \( m = 1, 2, \ldots, 32 \). The sum of the 32 global matrices gives the assembled global matrix and the assembled global matrix is given in Table 3.

| Global node numbers | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|---------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1                   | -1 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| 2                   | -0.5 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 |
| 3                   | 0   | 0.5 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 4                   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 5                   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 6                   | 0.5 | 0   | 0   | 0   | 0   | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 |
| 7                   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 8                   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 9                   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
For the constant \( f(x, y) = 1 \), the load vector for each triangular element \( \mathcal{K}_m \) is denoted by \( F^{K_m} \) and calculated from the relation

\[
F^{K_m} = \int_{\mathcal{K}_m} q_j dx dy, \forall j = 1, 2, 3.
\]

Here \( F^{K_m} \) denotes the \( j \) entry of the load vector \( F^{K_m} \).

Table 4. Load vector corresponds to each triangular element.

| Local node numbers | Load vector |
|--------------------|-------------|
| 1                  | 0.0104      |
| 2                  | 0.0104      |
| 3                  | 0.0104      |

The \( p \) entry of the global load vector of element \( \mathcal{K}_m \) takes the \( i \) entry of the load vector of element \( \mathcal{K}_m \) where \( p = 1, 2, ..., 25 \) are global node numbers, \( i = 1, 2, 3 \) are corresponding local node numbers for the global node numbers and \( m = 1, 2, ..., 32 \). The sum of the 32 global load vectors gives the assembled global load vector and the assembled global load vector is

#### Table 5. Assembled global load vectors.

| Global node numbers | Global node numbers |
|---------------------|---------------------|
| 1                   | 0.0104              |
| 2                   | 0.0417              |
| 3                   | 0.0208              |
| 4                   | 0.0417              |
| 5                   | 0.0104              |
| 6                   | 0.0417              |
| 7                   | 0.0417              |
| 8                   | 0.0833              |
| 9                   | 0.0417              |
| 10                  | 0.0417              |
| 11                  | 0.0208              |
| 12                  | 0.0833              |

Rows through 1 - 12

| Global node numbers | Global node numbers |
|---------------------|---------------------|
| 13                  | 0.0417              |
| 14                  | 0.0833              |
| 15                  | 0.0417              |
| 16                  | 0.0417              |
| 17                  | 0.0833              |
| 18                  | 0.0417              |
| 19                  | 0.0417              |

Rows through 13 - 25

Applying the boundary conditions (2) the assembled global matrix and the load vector becomes respectively

#### Table 6. Reduced assembled global matrix and load vector.

| Global node numbers | Global node numbers |
|---------------------|---------------------|
| 7                   | 0.0417              |
| 8                   | 0.0833              |
| 9                   | 0.0417              |
| 10                  | 0.0417              |
| 11                  | 0.0833              |
| 12                  | 0.0417              |

Using Gauss Seidel method successive approximate solution for \( u(x, y) \) at the free nodes of the region \( R \) is obtained. The results are given in Table 7.
5. Comparison Between the Two Solution

For the differential equation (1) together with boundary condition equations (2) we have found numerical solutions using two methods, viz. Finite difference and Finite element methods. Here in this section we compare these numerical solutions with each other. The solutions of $u(x,y)$ at the free nodes of the region obtained using the two methods are given in Table 8.

| Nodes | Finite difference solution | Finite element solution |
|-------|---------------------------|-------------------------|
| 7     | -0.0430                   | -0.0391                 |
| 8     | -0.0547                   | -0.0573                 |
| 9     | -0.0430                   | -0.0391                 |
| 12    | -0.0547                   | -0.0573                 |
| 13    | -0.0703                   | -0.0677                 |
| 14    | -0.0547                   | -0.0573                 |
| 17    | -0.0430                   | -0.0391                 |
| 18    | -0.0547                   | -0.0573                 |
| 19    | -0.0430                   | -0.0391                 |

As it is seen from Figure 6, the finite difference solution in (a) and finite element solution in (b) are somewhat the same.

6. Conclusions

In this paper we have introduced spreadsheet implementations of numerical methods for solving Poisson equation in two dimensions with Dirichlet boundary conditions. Other types of boundary conditions and right hand side function of the Poisson equation could be considered to solve the same problem using spreadsheet. Application of finite difference method is somewhat complex in comparison with that of the finite element method. Many physical problems have irregularly shaped boundaries and boundary conditions are expressed using derivatives. Boundary conditions which are expressed using differential
equations are difficult to handle using finite difference techniques because each boundary condition involving a derivative must be approximated by a difference quotient at the grid points. Also, if the shape of the boundary is irregular then placing the grid points on the boundary is difficult. The Finite element method alleviates all difficulties appears in finite difference method. It is independent of boundary conditions of the problem.

Appendix 1 Derivation of Finite Difference Equation

To drive finite difference equations (3), we consider Figure 2. Now in Figure 2 we represent node 8, node 12, node 13, node 14 and node 18 respectively by capital letters (x_i,y_j) are respectively corresponds to node 14 and node 18 respectively by capital letters (x_i,y_j), (x_i-1,y_j) and (x_i,y_j+1) are respectively corresponds to U, P, S and V . Expanding u_5 and u_p using Taylor series expansion about the point (x_i,y_j), we get

\[ u_5 = u_0 + \left( \frac{h}{1!} \left( \frac{\partial u}{\partial x} \right)_q \right) + \left( \frac{h^2}{2!} \left( \frac{\partial^2 u}{\partial x^2} \right)_q \right) + \cdots \]  (8)

and

\[ u_p = u_0 - \left( \frac{h}{1!} \left( \frac{\partial u}{\partial x} \right)_q \right) + \left( \frac{h^2}{2!} \left( \frac{\partial^2 u}{\partial x^2} \right)_q \right) + \cdots \]  (9)

respectively. Adding (8) and (9) gives

\[ u_5 + u_p = 2u_0 + h^2 \left( \frac{\partial^2 u}{\partial x^2} \right)_q + \cdots \]

Hence

\[ \frac{\partial^2 u}{\partial x^2} \bigg|_q = \frac{u_5 + u_p - 2u_0}{h^2} + O(h^2) \]  (10)

Similarly

\[ \frac{\partial^2 u}{\partial y^2} \bigg|_q = \frac{u_5 + u_p - 2u_0}{h^2} + O(h^2) \]  (11)

Substituting equations (10) and (11) in equation (1), we obtain

\[ u_0 = \frac{1}{4} \left( u_5 + u_p + u_l + u_r - h^2 f_{i,j} \right) + O(h^2) \]

Hence,

\[ u_{i,j} = \frac{1}{4} \left( u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - h^2 f_{i,j} \right), \]

\[ i,j = 2, 3, 4 \]

Appendix 2 Derivation of Weak Formulation of Equation (1)

Multiply both sides of equation (1) by arbitrary test function \( \nu = \nu(x,y) \in H^2_0(R) \) and integrate over the region \( R \), we get

\[ \int_R \Delta u \nu \, dR = \int_R f \nu \, dR \]  (12)

To reduce the order of derivatives in (12) we use Green's formula. The formula states as follows

\[ \int_R \Delta u \nu \, dR = -\int_R \nabla u \cdot \nabla \nu \, dR + \int_{\partial R} \frac{\partial u}{\partial n} \nu \, ds \]  (13)

Here \( \partial R, dy, \nabla u \) and \( n \) denotes respectively boundary of region \( R \), line segment on the boundary of \( R \), gradient of \( u \) and outward unit normal on the boundary of \( R \).

Substituting equation (13) in (12) we have

\[ -\int_R \nabla u \cdot \nabla \nu \, dR + \int_{\partial R} \frac{\partial u}{\partial n} \nu \, dy = \int_R f \nu \, dR \]  (14)

Since \( \nu \in H^2_0(R) \) the condition \( \int_{\partial R} \frac{\partial u}{\partial n} \nu \, dy = 0 \) is valid. Thus equation (14) becomes

\[ -\int_R \nabla u \cdot \nabla \nu \, dR = \int_R f \nu \, dR \]

Appendix 3 Derivation of Equations of Stiffness Matrix and Load Vector

We now consider the \( m^{th} \) triangular element \( K_m \). Equation (5) can be defined on triangular element \( K_m \) as

\[ -\int_{K_m} \left[ \nabla u_d \cdot \nabla v_d + f u_d \right] \, dx \, dy = 0, \quad m = 1,2, \ldots, 32. \]  (15)

Substituting (6) and (7) in equation (15), we get

\[ -\int_{K_m} \left[ \nabla (\sum_{j=1}^3 u_j \varphi_j(x,y)) \cdot \nabla (\sum_{j=1}^3 v_j \varphi_j(x,y)) + f (\sum_{j=1}^3 v_j \varphi_j(x,y)) \right] \, dx \, dy = 0. \]

Which is equivalent to

\[ \sum_{j=1}^3 v_j \left[ \int_{K_m} \nabla \varphi_i \nabla \varphi_j \, dx \, dy - \int_{K_m} f \varphi_j \, dx \, dy \right] = 0. \]

Since this equation holds true for all \( v_d \in V_d \), we have

\[ \sum_{i=1}^3 u_i \int_{K_m} \nabla \varphi_i \nabla \varphi_j \, dx \, dy = \int_{K_m} f \varphi_j \, dx \, dy, j = 1,2,3. \]

Here for each \( m \in \{1,2, \ldots, 32\} \) we have the stiffness matrix and the load vector respectively are \( A_{ij}^{K_m} = \int_{K_m} \nabla \varphi_i \nabla \varphi_j \, dx \, dy \) and \( F_j^{K_m} = \int_{K_m} f \varphi_j \, dx \, dy, i,j = 1,2,3. \)

References

[1] Alfio Quarteroni, Numerical models for differential problems (2nd edition), Springer-Verlag, Italia, 2014.
[2] Chapra Canal, Numerical methods for engineers (4th edition), The McGraw Hill Companies, 2001.

[3] Erwin Kreyszig, Advanced Engineering Mathematics (9th edition), 2006 John Wiley and Sons, Inc.

[4] G. Evans, J. Blackledge and P. Yardley, Numerical methods for partial differential equations, Springer-Verlag London Limited 2000.

[5] J. David Logan, A first course in DEs, 2006 Springer Science + Business Media. Inc.

[6] Lichard L. Burden and J. Douglas Faires, Numerical analysis (9th edition), 2011, 2005, 2001 Brooks/Cole, Cengage Learning, 20 Channel Center Street Boston, MA02210, USA.

[7] Mark A. Lau and Sastry P. Kuruganty, Spreadsheet implementation for solving boundary value problem in electromagnetic, spreadsheet in education (eJSiE) 4(1), 2001.

[8] Parag V. Patil and Dr. J.S.V.R. Krishna Prasad, Numerical solution for two dimensional Laplace equation with Dirichlet boundary conditions, Volume 6, Issue 4 (May - June 2013), PP 66-75.

[9] Ravi P. Agarwal and Donal O'Regan, Ordinary and Partial DEs, 2006 Springer Science + Business Media, LLC (2009).

[10] Susan C. Brenner, L. Ridgway Scott, The mathematical theory of finite element methods (3rd edition), 2008 Springer Sciences + Business Media, LLC.