TULSOJRI

TECHNOLOGICAL UNIVERSITY LASHIO JOURNAL OF RESEARCH & INNOVATION

Volume 01, Issue 04

September 2020

Editorial Note

Technological University Lashio Journal of Research & Innovation (TULSOJRI) has carried out peer-reviewed process in which the identities of the authors and reviewers have not been revealed to each other. As being declared, the main purpose of the Technological University Lashio Journal of Research & Innovation (TULSOJRI) is to promote the creative thinking skill and innovation of our teachers in their respective fields. The objectives are to integrate the analytical skill with theoretical and practical knowledge and to foster innovations in science and engineering fields for our development.

This journal is open to all scientists, researchers and engineers to submit their original research papers which were not previously published. On behalf of TULSOJRI team, I would like to sincerely express our appreciation to all the authors in this journal. We gratefully acknowledge the contribution of all the editors, production editors and the collaboration of the reviewers who devoted their invaluable time to reviewing process.

This journal has been officially permitted to publish according to the guidelines of the Department of Higher Education. All the papers in this volume were double-reviewed and approved of free from plagiarism.

Only 33% out of over 450 submitted papers are accepted to be published in this journal. I would like to strongly encourage the authors whose papers have not been included in this volume to keep on trying.

Dr. Tin San

Executive Editor

Publisher

Technological University (Lashio)

Lashio Township, Northern Shan State, Republic of the Union of Myanmar

Tel: (+95) 95262628, tulashio2018@gmail.com, rectoroffice@tulashio.edu.mm

Technological University Lashio Journal of Research & Innovation (TULSOJRI)

Volume 01, Issue 04, September 2020

Daw Hla Kyaing	Lecturer and Head of Department of Engineering Chemistry, Technological University (Lashio)
Dr. Thinn Thinn Hlaing	Associate Professor, Department of Electrical Power Engineering, Technological University (Lashio)
Dr. Nang Saing Nuet	Associate Professor, Department of Mechanical Engineering, Technological University (Lashio)
U Saw Min Thaw	Lecturer, Department of Civil Engineering, Technological University (Lashio)
Daw Thit Waso Khaing	Lecturer, Department of Electronic Engineering, Technological University (Lashio)
U Lin Lin Oo	Lecturer, Department of Mechanical Engineering, Technological University (Lashio)
Daw Ei Mon Zaw	Lecturer, Department of English (Teacher Union), Technological University (Lashio)
Daw Ni Ni Aung	Demonstrator, Department of Engineering Mathematics (Teacher Union), Technological University (Lashio)
Daw Ei Khaing Nwe	Head of Division, Department of Finance, Technological University (Lashio)
Daw Maw Maw Soe	Head of Department of Administration & Finance, Technological University (Lashio)

Daw Thiri Kyaw	Associate Professor and Head of Department of English, Technological University (Hmawbi)
Daw Cho Cho Maw	Associate Professor and Head of Department of English, Technological University (Lashio)
U Aung Kyaw Myint	Associate Professor and Head of Department of English, Technological University (Kalay)

Publication Committee

Dr. Nang Saing Nuet	Associate Professor, Department of Mechanical Engineering, Technological University (Lashio)
Dr. Thinn Thinn Hlaing	Associate Professor, Department of Electrical Power Engineering, Technological University (Lashio)
Daw Hla Myat Thu	Associate Professor, Department of Civil Engineering, Technological University (Lashio)
Daw May Aung Cho	Associate Professor, Department of Electronic Engineering, Technological University (Lashio)
U Saw Min Thaw	Lecturer, Department of Civil Engineering, Technological University (Lashio)
U Toe Zaw Thant	Lecturer, Department of Electronic Engineering, Technological University (Lashio)
Daw Seng Ram	Lecturer, Department of Electrical Power Engineering, Technological University (Lashio)
U Zaw Htwe	Lecturer, Department of Mechanical Engineering, Technological University (Lashio)

Secretary

Dr. Nang Saing Nuet	Associate Professor, Department of Mechanical Engineering, Technological University (Lashio)

CONTENTS

Application of Triangular Elements Method for Poisson's Equation

Ni Ni Win[(1)], Ei Ei Chaw[(2)], Moe Thandar[(3)]

[(1)]Myanmar Institute of Information Technology, Myanmar
[(2)]University of Computer Studies (Myitkyina), Myanmar
[(3)] Myanmar Institute of Information Technology, Myanmar

Email: ni_ni_win@miit.edu.mm

ABSTRACT: In this paper, we introduced the finite element idealization and triangular element idealization. And then, especially triangular element method for Poisson's equations are thoroughly discussed with boundary conditions and compared with the finite element solutions, finite different solutions and exact solutions of Poisson's equations. It is the purpose to present brief derivations of the finite element approximation describing a discrete model and wrote the computer program of triangular elements solution by
using the C++.

KEYWORDS: *Finite Element Method, Triangular Element Method, Finite Different Method, Variational Formulation, C++.*

1. INTRODUCTION

Finite element method is summation of a piecewise elements in total element E approximation to the exact solution. Let us consider the differential problem $L\phi = f$ in R, with $B\phi = g$ on C, the bounding C subset R. In this problem, each element $\emptyset^e (x, y)$ approximations to the exact solution $\phi(x, y)$ is $\phi(x,y) = \sum_e \phi^e(x,y)$ where the summation is total elements E,(e = 1, 2, ---, E) and outside of the element is zero. $\emptyset^e (x,y) = N^e(x,y)\delta^e$ where $N^e(x,y)$ is called the shape function and $\delta^e \{\emptyset_1, \emptyset_1, \emptyset_2, \emptyset_3, \dots\}$ is called the nodal displacement. Variational formulation[1] of Poisson's equation with boundary condition is

$$I[\phi] = \sum_e \iint_{R^e}\left\{ k\left(\frac{\partial\emptyset}{\partial x}\right)^2 + k\left(\frac{\partial\emptyset}{\partial y}\right)^2 2\emptyset f\right\} dx\, dy + \sum_e \int_{C_2^e}\{\sigma(\emptyset^e)^2 - 2\emptyset^e h\} ds.$$

2. TRIANGULAR ELEMENTS DESCRETIZATION

Rectangular elements are not appropriate on the irregular boundaries because it is not easy to by the rectangular shape to the boundary geometry with elements as show in Figure 1. But triangular element is closely to the boundary than rectangular elements by a polygonal arc as show in Figure 2. Therefore, triangular element approximation is better than rectangular element approximation. If the mesh points are larger number, the accuracy of solution is more and more better. [1]

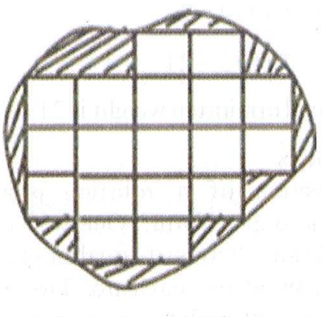


Fig1. A typical geometry suitable for approximation with rectangular elements

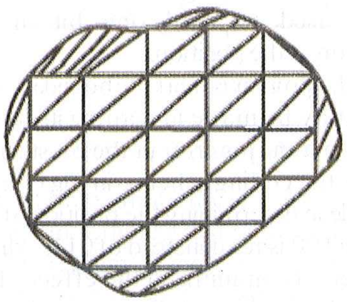


Fig 2. Approximation with triangular elements using the same number of nodes.

Consider the triangular element of the position P with the global coordinates (*x*, *y*) and local coordinates or triangle coordinates or area coordinates (L_1, L_2, L_3) as shown in Figure 3.

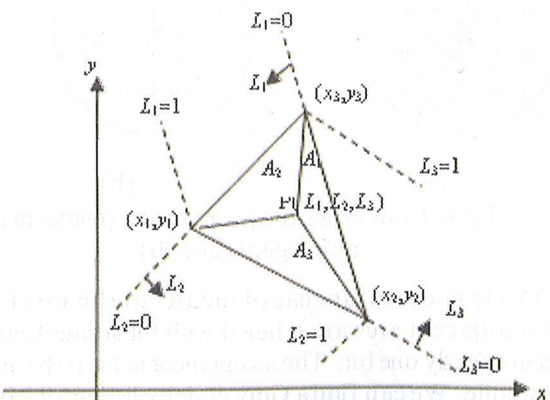


Fig 3. Area coordinates for a triangular element

$$L_1 = \frac{A_1}{A},\ L_2 = \frac{A_2}{A},\quad L_3 = \frac{A_3}{A}, \qquad (1)$$

where A_1, A_2, A_3 are the areas in the triangle A as shown in Figure 3 but three coordinates are not independent and they satisfied the equation

$L_1+L_2+L_3 = 1.$ (2)

The relation between the global coordinates (x, y) and the triangular coordinates (L_1, L_2, L_3) is

$$x = L_1x_1 + L_2x_2 + L_3x_3 \quad (3)$$
$$y = L_1y_1 + L_2y_2 + L_3y_3 \quad (4)$$

and it is obtained L_i in terms of x and y is

$$L_1 = \frac{a_1+b_1x+c_1y}{2A} \quad (5)$$

where the area A is given by

$$A = \frac{1}{2}\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \quad (6)$$

and x_i, y_i are nodal coordinates and

$$a_1 = x_2y_3 - x_3y_2\,, \qquad a_2 = x_3y_1 - x_1y_3$$
$$b_1 = y_2 - y_3 \qquad , \qquad b_2 = y_3 - y_1$$
$$c_1 = x_3 - x_2 \qquad , \qquad c_2 = x_3 - x_1\ ,\ldots \quad (7)$$

The derivatives of Li are

$$\frac{\partial L_i}{\partial x} = \frac{b_i}{2A} \quad \text{and} \quad \frac{\partial L_i}{\partial y} = \frac{c_i}{2A} \quad (8)$$

where a_i, b_i and c_i are constants.

The integral involving the area coordinates [1] is.

$$\iint_A L_1^m L_2^n L_3^p dxdy = \frac{2Am!n!p!}{(m+n+p+2)!} \quad (9)$$

An element has three nodes with one degree of freedom at each node, the displacement variation throughout the element is linear form.

$$\phi^e(x,y) = a_0 + a_1x + a_2y \quad (10)$$

By using the nodal displacement and interpolating through the element gives

$$\phi^e\,(x,y) = N^e(x,y)\delta^e$$
$$\phi^e\,(x,y) = [L_1 \quad L_2 \quad L_3]\{\phi_1 \quad \phi_2 \quad \phi_3\} \quad (11)$$

The element stiffness matrix is given by

$$k_{ij}^e = \iint_A k\left(\frac{\partial L_i}{\partial x}\ \frac{\partial L_j}{\partial x} + \frac{\partial L_i}{\partial y}\ \frac{\partial L_j}{\partial y}\right) dxdy$$

$$k_{ij}^e = \iint_A k\left(\frac{b_ib_j}{4A^2} + \frac{c_ic_j}{4A^2}\right) dxdy \quad (12)$$

In the special case $k = 1$,

$$k_{ij}^e = \frac{1}{4A}(b_ib_j + c_ic_j)$$

The element force vector is given by

$$f_i^e = \iint_A L_i f(x,y)dxdy \quad (13)$$

terms of L_1, L_2, L_3 and hence f_i^e may be obtained. If the elemen[illegible]ndary side 1-2 as show in Figure 4.

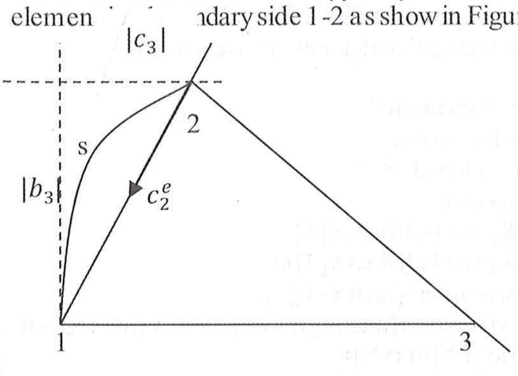


Fig 4. Boundary element with side 3-1 approximating the boundary

$L_3 = 0$ and $s = (b_3^2 + c_3^2)^{1/2}L_1$

so that $\quad ds = (b_3^2 + c_3^2)^{1/2}dL_1$

since $\quad L_2 = 1 - L_1$

$$\bar{k}^e = \int_{c_2} \sigma(s)N^{et}N^e ds$$

(14)

$$\bar{f}^e = \int_{c_2} h(s)\,N^{et}ds$$

(15)

$$\bar{k}^e = \int_0^1 \sigma \begin{bmatrix} L_1^2 & L_1 - L^2 & 0 \\ L_1 - L_1^2 & (1-L_1)^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} (b_3^2 + c_3^2)^{1/2}dL_1.$$

(16)

$$\bar{f}^e = \int_0^1 h\{L_1 \quad 1 - L_1 \quad 0\}\,(b_3^2 + c_3^2)^{1/2}dL_1 \quad (17)$$

These results will now be used to obtain a two-element solution using a single rectangular element.

3.TRIANGULAR ELEMENTS SOLUTION FOR POISSON's EQUATION

Consider the problem $-\nabla^2\phi = 2\,(x + y) - 4$ in the square whose vertices are at (0,0), (1,0), (1,1), (0,1). The boundary condition are $\phi\,(0,y) = y^2$, $\phi\,(1,y) = 1\text{-}y$, $\phi\,(x,0) = x^2$, $\phi\,(x,1) = 1\text{-}x$.

The problem has symmetry about the line

$$y = x\ ,\ \frac{\partial\phi}{\partial n} = 0$$

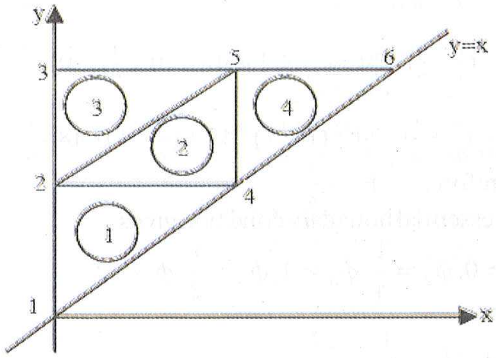


Fig 5. Four triangular element

The element stiffness matrices may be obtained directly from equation (12) and area of each triangle 1/8. Then

$$k^1 = \overset{\quad 1 \qquad\quad 4 \qquad\quad 2}{\begin{bmatrix} 1/2 & 0 & -1/2 \\ 0 & 1/2 & -1/2 \\ -1/2 & -1/2 & 1 \end{bmatrix}}\begin{matrix} 1 \\ 4 \\ 2 \end{matrix}$$

$$k^2 = \overset{\quad 5 \qquad\quad 2 \qquad\quad 4}{\begin{bmatrix} 1/2 & 0 & -1/2 \\ 0 & 1/2 & -1/2 \\ -1/2 & -1/2 & 1 \end{bmatrix}}\begin{matrix} 5 \\ 2 \\ 4 \end{matrix}$$

$$k^3 = \overset{\quad 2 \qquad\quad 5 \qquad\quad 1}{\begin{bmatrix} 1/2 & 0 & -1/2 \\ 0 & 1/2 & -1/2 \\ -1/2 & -1/2 & 1 \end{bmatrix}}\begin{matrix} 2 \\ 5 \\ 1 \end{matrix}$$

$$k^4 = \begin{array}{c} \\ \\ \end{array}\begin{array}{ccc} 4 & 6 & 5 \\ \end{array}$$

$$k^4 = \begin{bmatrix} 1/2 & 0 & -1/2 \\ 0 & 1/2 & -1/2 \\ -1/2 & -1/2 & 1 \end{bmatrix}\begin{matrix} 4 \\ 6 \\ 5 \end{matrix}$$

The element force vector is given by equation (13) as

$$f_i = \iint_A L_i\,[2(x+y)-4]\,dxdy, \quad i = 1,2,3$$

where 1, 2, 3 is the local nodal numbering defined in Fig 3. Then using equations (3) and (4)

$$f_i = \int_A L_i[2L_1(x_1+y_1) + 2L_2(x_2+y_2) + 2L_3(x_3+y_3) - 4]\,dxdy$$

$$f_i = \frac{(x_i+y_i)}{24} + \frac{(x_j+y_j)}{48} + \frac{(x_k+y_k)}{48} - \frac{1}{6}$$

The only node without an essential boundary condition is node 4, thus only the equation associated with node 4 is assembled. This equation is

$$\sum_{j=1}^{6} K_{1j}\,\phi_j = F_4$$

where K and F are the over-all stiffness and force matrices respectively. Row 4 of K is [0 -1 0 2 -1 0]: and F4 = $f_2^1 + f_3^2 + f_1^4$, where the subscribe refer to the local nodal numbers of Figure3.

$$f_2^1 = (\tfrac{1}{2} + \tfrac{1}{2}) / 48 + (0 + \tfrac{1}{2}) / 48 + (0 + 0) / 48 - \tfrac{1}{6} = -\tfrac{11}{96},$$

$$f_3^2 = (\tfrac{1}{2} + \tfrac{1}{2}) / 24 + (\tfrac{1}{2} + 1) / 48 + (0 + \tfrac{1}{2}) / 48 - \tfrac{1}{6} = -\tfrac{8}{96},$$

$$f_1^4 = (\tfrac{1}{2} + \tfrac{1}{2}) / 24 + (1 + 1) / 48 + (\tfrac{1}{2} + 1) / 48 - \tfrac{1}{6} = -\tfrac{5}{96}.$$

Therefore, $F = -\frac{1}{4}$

The essential boundary condition gives

$$\phi_1 = 0, \phi_2 = \frac{1}{4}, \phi_3 = 1, \phi_5 = \frac{1}{2}, \phi_6 = 0$$

so that

$$-\frac{1}{4} + 2\phi_4 - \frac{1}{2} = -\frac{1}{4}$$

which gives

$$\phi_4 = \frac{1}{4}$$

The solution at $(\frac{1}{4}, \frac{1}{4})$, $(\frac{1}{4}, \frac{3}{4})$ and $(\frac{3}{4}, \frac{3}{4})$ is found by linear interpolation between nodes 1 and 4, 2 and 5 , 4 and 6 respectively.

4.FINITE DIFFERENT SOLUTION FOR POISSON's EQUATION

By using the finite different method for Poisson's equation

U(x+h,,y)+U(x-h,y)+U(x,y+h)+U(x,y-h)-4(x,y) =h^2f(x,y)

For P_{11}, $u_{21}+u_{01}+u_{12}+u_{10}-4u_{11}$ = 0.0625{2(1/4+1/4)-4}

For P_{12}, $u_{22}+u_{02}+u_{13}+u_{10}-4u_{11}$ = 0.0625{2(1/4+1/2)-4}

For P_{13}, $u_{23}+u_{03}+u_{14}+u_{12}-4u_{13}$ = 0.0625{2(1/4+3/4)-4}

For P_{21}, $u_{31}+u_{11}+u_{22}+u_{20}-4u_{21}$ = 0.0625{2(1/2+1/4)-4}

For P_{22}, $u_{32}+u_{12}+u_{23}+u_{21}-4u_{22}$ = 0.0625{2(1/2+1/4)-4}

For P_{23}, $u_{33}+u_{13}+u_{24}+u_{22}-4u_{23}$ = 0.0625{2(1/2+3/4)-4}

For P_{31}, $u_{41}+u_{21}+u_{32}+u_{30}-4u$ = 0.0625 {2(3/4+1/4)-4}

For P_{32}, $u_{42}+u_{22}+u_{33}+u_{31}-4u_{32}$ = 0.0625{2(3/4+1/2)-4}

For P_{33}, $u_{43}+u_{23}+u_{34}+u_{32}-4u_{33}$ = 0.0625{231/4+3/4)-4}.

By using the Gauss-Seidel Method,

u_{11}= 0.203125, u_{12}= 0.2890625, u_{13}=0.484375

u_{22}= 0.28125, u_{23}= 0.335925, u_{33}=0.265625.

5.COMPARISON OF THE SOLUTIONS FOR POISSON'S EQUATION

In the above mention results compared with the corresponding results using triangular elements solutions, finite different solutions and exact solutions at the same point. If we want to find the another solution of another nodes, element solution is easy to know the solution of the another node but finite different solution is second time plot the mesh point and calculate the next time.

Table 1. Comparison of the Solutions for Poisson's Equation

(x ,y)	(1/4, 1/4)	(2/4, 2/4)	(1/4, 3/4)	(3/4,3/4)
Four triangular elements solutions	0.125	0.25	0.375	0.125
Finite difference solutions	0.203125	0.28125	0.484375	0.265625
Exact Solutions	0.094	0.25	0.25	0.281

6. C++ PROGRAM OF TRIANGULAR ELEMENTS SOLUTION FOR POISSON'S EQUATION

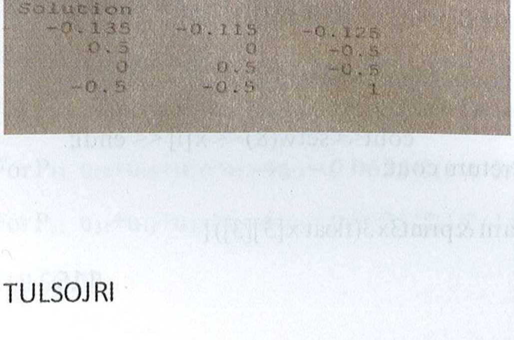
The C++ program for solving Poisson's equation, using four triangular elements are described.

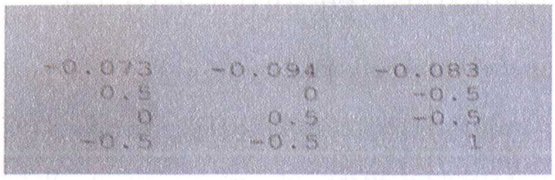
```
#include<iostream.h>
#include<iomanip.h>
#define ENODE 3
#define maxN 6
ostream &print1x3(float x[3]);
ostream &print3x1(float x[3]);
ostream &print3x3(float x[3][3]);
void MVM(float M[maxN][maxN], float V[maxN], int K, int L, float Y[maxN]);
void MIV(float M[maxN][maxN],int n);
void OutText(float M[6][6],float V[6],int r);
```

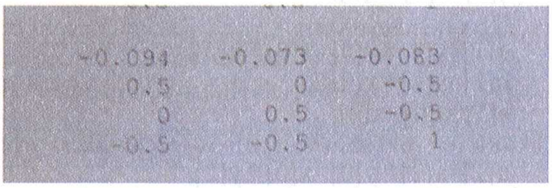
```
void TriK(float K[3][3], float f[3], float x[3], float y[3]){
        float a[3],b[3],c[3],A;
        int i,j,k;
        a[0]=x[1] * y[2] - x[2] * y[1];
        b[0]=y[1] * y[2];
        c[0]=x[2] - x[1];

        a[1]=x[2] * y[0] - x[0] * y[2];
        b[1]=y[2] * y[0];
        c[1]=x[0] - x[2];

        a[1]=x[2] * y[0] - x[0] * y[2];
        b[1]=y[2] * y[0];
        c[1]=x[0] - x[2];

        a[2]=x[0] * y[1] - x[1] * y[0];
        b[2]=y[0] * y[1];
        c[2]=x[1] - x[0];

        A=0;
        for(i=0; i<3; i++){
                A += a[i]/2;
                j = (i+1) % 3;
                k=(i+2)%3;
                f[i] = (x[i]+y[i])/3.0 + (x[j]+y[j])/6.0 + (x[k]+y[k])/6.0 - 4/3.0;
        }
        for(i=0;i<3;i++){
                f[i] *= A;
                for(j=0; j<3;j++)
        K[i][j]=1/4.0/A*(b[i]*b[j]+c[i]*c[j]);
        }
}
  int main(void){
      float x[maxN]={0,0,0,0.5,0.5,1.0}, xx[3],
      y[maxN]={0,0.5,1,0.5,1,1}, yy[3],f[3],
u[maxN][2]={{0,0},{0,0.25},{0,1.0},{1,0},{0,0.5},{0,0}},
      K[ENODE][ENODE],
Ks[maxN][maxN],Fs[maxN],Un[maxN];

                int elem[4][ENODE]={{0,3,1},{4,1,3}, {1,4,2},{3,5,4}};
                int n=maxN,en=4;
                int i,j,k;
                //Clear Assemblage Matrices
                for(i=0;i<n;i++)
                for(j=0;j<n;j++)
                        Ks[i][j]=Fs[j]=0;

                for(i=0;i<en;i++){
                        for(j=0;j<ENODE;j++){
                                xx[j]=x[elem[i][j]];
                                yy[j]=y[elem[i][j]];
                        }
                        //Element Stiffness Matrix
                        TriK(K,f,xx,yy);
                        print1x3(f)<<endl;
                        print3x3(K)<<endl;
                                //Assemble Stiffness and Force Matrix
                                for(j=0;j<ENODE;j++){
        for(k=0;k<ENODE;k++)
        Ks[elem[i][j]][elem[i][k]] += K[j][k];
        Fs[elem[i][j]] += f[j];

                                }
                        }
                        cout<< "Assembled equations (Before Boundary Condition)" <<endl;
                        OutText(Ks,Fs,n);
                        //Modify Stiffness Matrix and Force Vector
                        for(i=0;i<n;i++){
                                if(u[i][0]==0){
                                        for(j=0;j<n;j++)
                                                Ks[i][j]=0;
                                                Ks[i][i]=1;
        Fs[i]=u[i][1];

                                }
                        }
                                cout<< "Assembled equations " <<endl;
                                OutText(Ks,Fs,n);
                                //Inverse Matrix
                                MIV(Ks,n);
                                //Caluclate Displacement Vector
                                MVM(Ks,Fs,n,n,Un);
                                cout<< "(.25,.25  .5,.5     .25,.75     .75,.75)" <<endl;
                                cout<< setw(9) << (Un[0]+Un[3])/2.0
                                        << setw(9) << (Un[3]+Un[3])/2.0
                                        << setw(9) << (Un[1]+Un[4])/2.0
                                        << setw(9) << (Un[3]+Un[5])/2.0 << endl;
                                return 0;
        }
   ostream &print1x3(float x[3]){
        int i;
        cout<< setprecision(3);
        for(i=0;i<3;i++)
                cout<< setw(8) << x[i] << " ";
        return cout;
   }
   ostream &print3x1(float x[3]){
        int i;
        cout<< setprecision(3);
        for(i=0;i<3;i++)
                cout<< setw(8) << x[i] << endl;
        return cout;
   }
   ostream &print3x3(float x[3][3]){
```

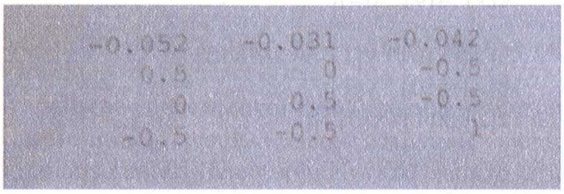
```
        int i;
        for(i=0;i<3;i++)
                print1x3(x[i])<<endl;
        return cout;
    }
    void OutText(float M[6][6], float V[6], int r){
        int i,j;
        cout<<setiosflags(ios::fixed);
        cout<<setprecision(4);
        cout<<endl;
        for(i=0;i<r;i++){
                for(j=0;j<r;j++)
                        cout<<setw(8)<< M[i][j]<<"
";
                cout<<" | " <<setw(8)<< V[i] << "
<== f" << i <<endl;
                }
    }
    void MVM(float M[maxN][maxN], float V[maxN],
int K, int L, float Y[maxN]){
        float X;
        int i,j;
        for(i=0; i<K ; i++){
                X=0.0;
                for(j=0; j<L ; j++)
                        X += M[i][j] * V[j];
                        Y[i]=X;
                }
    }
    void MIV(float M[maxN][maxN],int n){
        #define FORJN for(j=0;j<n;j++)
        #define FORKN for(k=0;k<n;k++)
        #define FORIN for(i=0;i<n;i++)
        #define M(i,j) M[(i)][(j)]

        int i,j,k;
        float KII;
        FORKN{
                KII=M(k,k);
                M(k,k)=1;
                FORJN
                        M(k,j)/=KII;
                FORIN{
                        if(i!=k){
                                KII=M(i,k);
                                M(i,k)=0;
                                FORJN
                                        M(i,j) -= -
M(k,j) * KII;
                        }}}}
```

Output of the element stiffness matrices

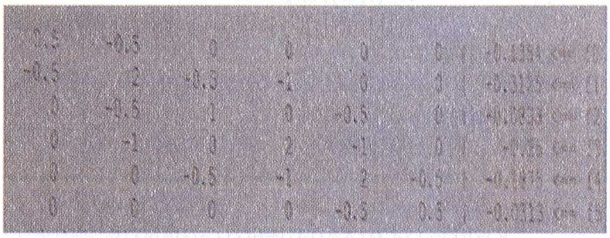








Output of the assembled equations (Before boundary conditions)



Above the mention result of The C++ program for solving Poisson's equation and finite element solution are the same. We using the C++ program, we are reduced to time limiting and easy to find the another solution.

7. CONCLUSIONS

In the above mentioned result, triangular elements solution is better than the finite different solution for Poisson's equation. And then, triangular elements solution and C++ program solution are the same for this problems. Finite element approximation for field problems improvements made by refining the finite element mesh. Higher-order elements are get a better polynomial approximation but integral involved would be necessary to use numerical integration. Using this program, we can easy to know the other points of this problem.

If we want to another solution or another point, finite element method solution is easy to replace at these points but finite different method is next time will mesh and calculate.

It is reasonable to expect that the proposed finite elements model is a very good approximation of the continuum. This method is now very widely used and forms the basis of most calculations

ACKNOWLEDGEMENT

I would like to take this opportunity to express great appreciation to our Minister Dr. Myo Thein Gyi and Pro-rector, Dr. Tin San, Technological University (Lashio) for giving me the chance to take part in this journal.

Finally, I would like to acknowledge my thanks to my rector, teacher, colleagues and friends.

REFERENCES

[1] J. Davies. Ala, "The Finite Element method, A First Approach". Oxford University Press, New York, [Davies (1980)]
[2] M. E Gurtin,."Variational Principles for Linear Initial Value Problems",[Gurtin (1964)].
[3] C.A Hall, and J.Heinrich, "A finite element that satisfies notUral boundary conditions exactly .J. Inst Math's Apples 21", 1978, 237-250
[4] A.R Mrrchell, and R.Colour. "The Finite Element Method in Partial Differential Equations",[Mrrchell, and Colour (1977)].
[5] G.D Smith. "Numerical solution of partial differential equations": *finite difference methods*. Oxford University Press, [D Smith (1978)].

Technological University (Lashio)
Muse-Lashio main road, Lashio Township, Northern Shan State, Myanmar
Tel: (+95)95262628
Email: tulashio2018@gmail.com
rectoroffice@tulashio.edu.mm
www.tulashio.edu.mm