ELECTRIC FIELD CALCULATIONS BY NUMERICAL TECHNIQUES

A THESIS SUBMITTED IN FUFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

ELECTRICAL ENGINEERING

BY

BISWANATH MALIK

ROLL NO-10502034

DEPARTMENT OF ELECTRICAL ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA-769008

2009

ELECTRIC FIELD CALCULATIONS BY NUMERICAL TECHNIQUES

A THESIS SUBMITTED IN FUFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

ELECTRICAL ENGINEERING

BY

BISWANATH MALIK

ROLL NO-10502034

UNDER THE GUIDANCE OF

PROF. SARADINDU GHOSH

& PROF. SANDIP GHOSH

DEPARTMENT OF ELECTRICAL ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA-769008

2009

Certificate

This is to certify that the project progress report entitled “Electric field calculation by numerical techniques” submitted by Biswanath malik in partial fulfillment of the requirements for the award of bachelor of technology degree in electrical engineering at national institute of technology, Rourkela is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge the matter embodies in the project work has not submitted to any other university/institute for the award of any degree & diploma.

Date: Prof. Sandip Ghosh

Department of Electrical engineering

Place: National institute of technology

Rourkela-769008

ACKNOWLEDGMENT

I would like to articulate our deep gratitude to our project guide Prof.Saradindu Ghosh & Prof. Sandip Ghosh who has always been our motivation for carrying out the project. I am thanking to C programmer Susanta Kumar Rout for his sincere help to write programs. It is our pleasure to refer Microsoft Word exclusive of which the compilation of this report would have been impossible. An assemblage of this nature could never have been attempted with out reference to and inspiration from the works of others whose details are mentioned in reference section. We acknowledge out indebtedness to all of them. Last but not the least, our sincere thanks to all of our friends who have patiently extended all sorts of help for accomplishing this undertaking.

Biswanath Malik

Roll no.-10502034

Contents

ABSTRACT

Objective of the study of electric field calculations by numerical techniques is to use different numerical techniques to find electric field distributions, which are inevitable tool in various electricity-concerned technologies, in particular, for analyzing discharge phenomenon and designing high voltage equipment. In this thesis, two numerical methods are discussed; e.g. finite difference method and finite element method. Both methods are used to find two dimensional electric field distributions with given boundary conditions using MATLAB. Electric field distributions in more practical three dimensional cases with non-uniformly distributed dielectric of a capacitor in a DC busbar has found using C-programming. Also, electromagnetic field calculations of electric motor have been done in ANSYS.

Chapter 1

Introduction

Calculation of electric fields with the aid of an computer is now a inevitable tool in various electricity-concerned technology, in particular, for analyzing discharge phenomenon and designing high voltage equipments .Electric and magnetic fields comprises two components dealt with in one of the classical physics, electromagnetism. Calculation of electric fields is usually considered easier than that of magnetic ones from two reasons. First, the electric field is expressed with a scalar potential at least in simple low frequency problems. Secondly, non linear characteristics are more often involved in magnetic fields. Compared with magnetic field, however, the calculation electric fields generally require higher accuracy, because the highest electric field stress on insulator is usually the most important and decisive value in insulation design or discharge study. This is one of reason why the boundary-dividing methods are preferred to the region-dividing ones, such as finite difference method (FDM) or finite element method (FEM). Usually former method does not need numerical differentiation to obtain field values.

A fundamental equation for the electric field is Laplace’s equation or Poisson’s equation; perhaps the simplest among many partial differential equations that express physical phenomena among various numerical calculation methods, FDM and FEM is very unique as it is applied exclusively to electric field calculations. Fundamental difference between FDM and FEM is that, FDM can be used for calculation of potential at nodes only but FEM can be used for calculation of potential at nodes as well as within the elements. Calculation of electric field in 3D arrangement poses no essential problem by of the numerical methods if the field is given by Laplace’s equation .The difficulty is that it usually required the tedious work preparing the input of a large amount of errorless data associate with 3D conditions.

Numerical solution of EM problems started in the mid-1960s with the availability of modern high-speed digital computers. Since then, considerable effort has been expended on solving practical, complex EM-related problems for which closed form analytic solutions are either intractable or do not exist. The numerical approach has the advantage of allowing the actual work to be carried out by operates without a knowledge of higher mathematics, with a resulting economy of labor on the part of the highly trained personnel.

Chapter 2

THE Finite difference method

2.1 fundamentals of FDM

The finite difference method is a powerful numerical method for solving partial differential equations. In applying the method of finite differences a problem is defined by:

- A partial differential equation such as Poisson's equation
- A solution region
- Boundary and/or initial conditions.

An FDM method divides the solution domain into finite discrete points and replaces the partial differential equations with a set of difference equations. Thus the solutions obtained by FDM are not exact but approximate. However, if the discretization is made very fine, the error in the solution can be minimized to an acceptable level.

The Poisson's equation in 3-D is given by

$$\nabla^2 V = -\frac{\rho_v}{\epsilon} \quad \text{..} \quad (2.1)$$

For 2-D case, Poisson's equation simplifies to

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = -\frac{\rho_v}{\epsilon} \quad \text{..} \quad (2.2)$$

In applying the methods of finite differences, we define the solution region into a finite number of meshes as shown in Fig2.1.

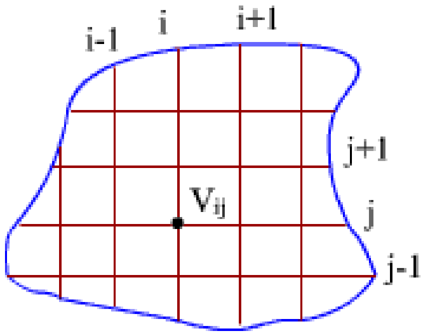


Fig2.1: Division of solution region into grid points

The meshes can be various shapes; we shall only consider the rectangular and square meshes only. First we consider a mesh configuration having five nodes and unequal arms as the Fig2.2.

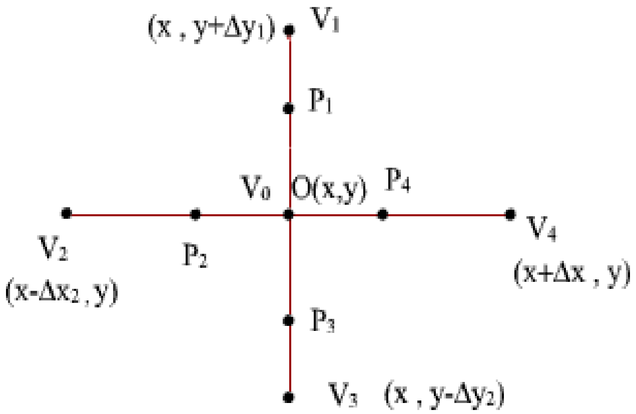


Fig2.2: A mesh with unequal arms

With reference to Fig2.1, V_o corresponds to the voltage V_{ij}. For the five node mesh configuration of Fig2.2, the voltages are defined as:

$$V_0 = V(x,y) \qquad (2.3a)$$

$$V_1 = V(x,y+\Delta y_1) \qquad (2.3b)$$

$$V_2 = V(x-\Delta x_2,y) \qquad (2.3c)$$

$$V_3 = V(x,y-\Delta y_2) \qquad (2.3d)$$

$$V_4 = V\left(x + \Delta x_1, y\right) \quad \text{...} \quad (2.3e)$$

Let, P1, P2, P3 andP4 represent the midpoint of the arms as shown in Fig 2.2. In order to replace the Poisson equation (2.2) by difference equations, we obtain the approximate first derivatives at the points P1 to P2 and use these first derivatives to approximate the second derivative.

The first derivatives at P1and P2 are

$$\left.\frac{\partial V}{\partial y}\right|_{P_1} = \frac{V_1 - V_0}{\Delta y_1} \quad \text{...} \quad (2.4a)$$

$$\left.\frac{\partial V}{\partial y}\right|_{P_3} = \frac{V_0 - V_3}{\Delta y_2} \quad \text{...} \quad (2.4b)$$

$$\therefore \left.\frac{\partial^2 V}{\partial y^2}\right|_0 = \frac{\frac{V_1 - V_0}{\Delta y_1} - \frac{V_0 - V_3}{\Delta y_2}}{\frac{1}{2}\left(\Delta y_1 + \Delta y_2\right)} = 2\frac{\Delta y_2\left(V_1 - V_0\right) - \Delta y_1\left(V_1 - V_0\right)}{\Delta y_1 \Delta y_2\left(\Delta y_1 + \Delta y_2\right)} \quad \text{...} \quad (2.5)$$

In the same manner,

$$\left.\frac{\partial^2 V}{\partial x^2}\right|_0 = \frac{2\left(V_4 - V_0\right)\Delta x_2 - \left(V_4 - V_0\right)\Delta x_1}{\Delta x_1 \Delta x_2\left(\Delta x_1 + \Delta x_2\right)} \quad \text{...} \quad (2.6)$$

The first derivative at P1and P2 is

$$\left.\frac{\partial V}{\partial y}\right|_{P_1} = \frac{V_1 - V_0}{\Delta y_1} \quad \text{...} \quad (2.7a)$$

$$\left.\frac{\partial V}{\partial y}\right|_{P_3} = \frac{V_0 - V_3}{\Delta y_2} \quad \text{...} \quad (2.7b)$$

$$\therefore \left.\frac{\partial^2 V}{\partial y^2}\right|_0 = \frac{\frac{V_1 - V_0}{\Delta y_1} - \frac{V_0 - V_3}{\Delta y_2}}{\frac{1}{2}\left(\Delta y_1 + \Delta y_2\right)} = 2\frac{\Delta y_2\left(V_1 - V_0\right) - \Delta y_1\left(V_1 - V_0\right)}{\Delta y_1 \Delta y_2\left(\Delta y_1 + \Delta y_2\right)} \quad \text{...} \quad (2.8a)$$

In the same manner,

$$\left.\frac{\partial^2 V}{\partial x^2}\right|_0 = \frac{2\left(V_4 - V_0\right)\Delta x_2 - \left(V_4 - V_0\right)\Delta x_1}{\Delta x_1 \Delta x_2 \left(\Delta x_1 + \Delta x_2\right)}$$
.... (2.8b)

Further, for Laplace equation, ρ_s and equation (2-8) simplifies to

$$V_1 + V_2 + V_3 + V_4 = 4V_0$$
.... (2.9)

Thus we see that voltage at the central node is the mean of the voltages at the other four nodes.

With reference to Fig2-1, equation (2-8) can be written as

$$V_{ij} = \frac{1}{4}\left(V_{i,j+1} + V_{i-1,j} + V_{i,j-1} + V_{i+1,j} + \frac{h^2 \rho_s}{\epsilon}\right)$$
.... (2.10)

Equation (2.8) & equation (2.9) can be used to solve Poisson's and Laplace's equation respectively when uniform grids are used. These equations, along with the specified boundary conditions can be used to solve a problem.

2.2 Two dimensional electric field calculations by finite difference method

The solution region is divided into square meshes. Here boundary is regular. Total 21 nodes has unknown potentials. We have marked that and 24 nodes are known potential. Here using difference elements we have found the potential at each nodes whose potential are not known, up to 17th iteration.

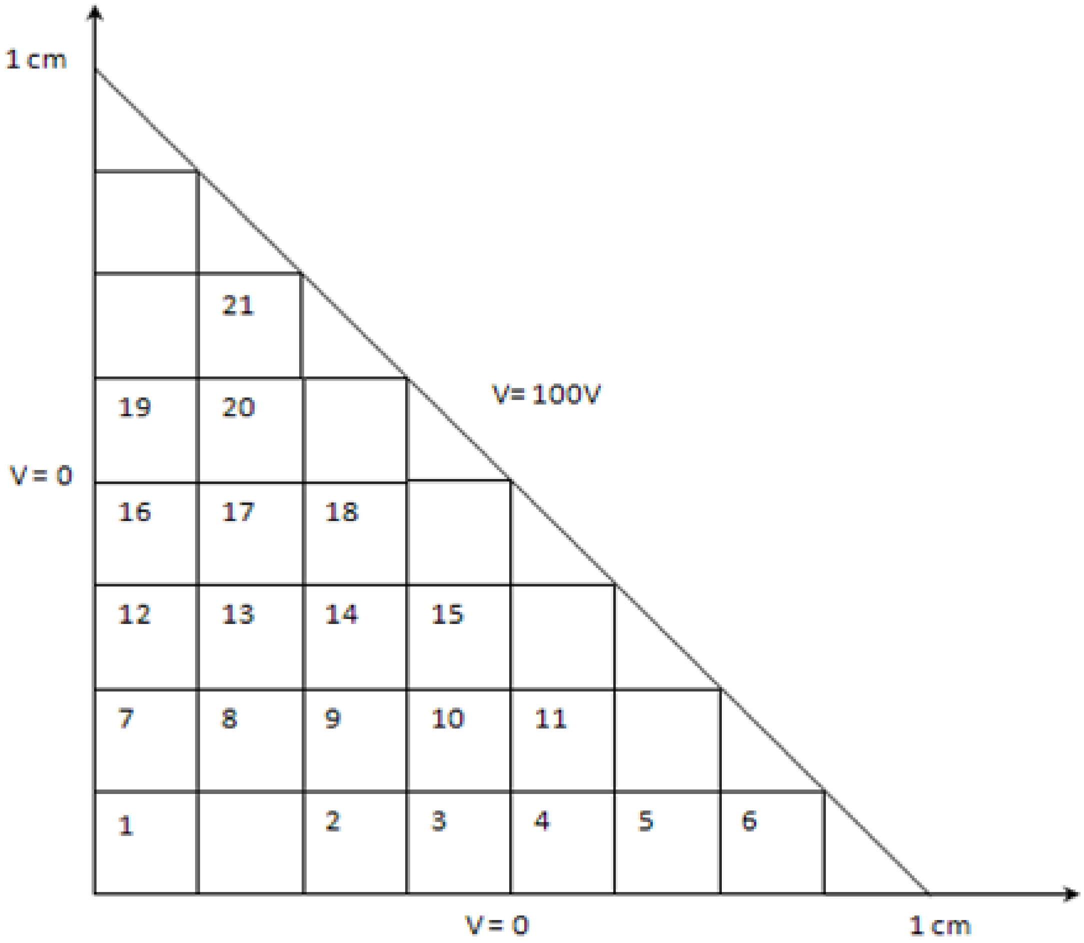


Figure (2.3)

RESULT: - No. of iteration ⟶

	1	2	3	4	5	6	7	8	9
V1	0	0	0	0	0	1.861	3.38	4.51	5.312
V2	0	0	0	0	3.125	6.645	9	10.6	11.68
V3	0	0	0	6.25	11.23	14.66	16.86	18.3	19.24
V4	0	0	12.5	19.14	23.044	25.3487	26.82	27.73	28.34
V5	0	25	32.81	36.22	38.08	39.1381	39.8	40.21	40.51
V6	50	56.25	58.2	59.055	59.52	59.79	59.95	60.05	60.127
V7	0	0	0	0	4.321	6.87	9.063	10.63	11.69
V8	0	0	0	6.25	13.49	17.95	21.1	23.15	24.64
V9	0	0	12.5	22.656	28.96	33.064	35.63	37.625	38.8
V10	0	25	37.5	44.726	48.655	51.25	52.85	53.92	54.625
V11	50	0	67.58	70.2365	71.684	72.6	73.1625	73.531	73.784
V12	0	0	0	11.035	12.1271	14.92	17	18.833	19.164
V13	0	0	12.5	23.8525	29.3303	33.23	35.72	37.625	38.82
V14	0	25	40.625	46.39	50.998	53.32	55.97	57.411	58.33
V15	50	62.5	69.53	72.78	74.9137	76.33	77.205	77.833	78.239
V16	0	0	12.5	20.3368	23.49	25.565	26.98	27.853	28.43
V17	0	25	37.5	45.3151	49.05	51.447	53.05	54.058	54.71
V18	50	62.5	69.535	72.926	75	76.2	77.255	77.867	78.185
V19	0	25	32.815	36.52	38.27	39.46	40.0325	40.5	40.7
V20	50	62.5	67.57	70.4587	72	73.3	73.5	73.73	73.85
V21	50	56.25	58.203	59.13	59.5675	59.85	60	60.125	60.18

	10	11	12	13	14	15	16	17
V1	5.92	6.22	6.47	6.644	6.76	6.84	6.885	6.91
V2	12.46	12.94	13.27	13.52	13.7	13.77	13.81	13.9
V3	19.89	20.33	20.62	20.82	20.97	21.02	21.1	21.13
V4	28.76	29	29.21	29.357	29.4	29.46	29.5	29.52
V5	40.72	40.79	40.91	40.94	40.96	40.98	41	41
V6	60.17	60.2	60.22	60.23	60.24	60.245	60.25	60.25
V7	12.44	12.95	13.3	13.51	13.65	13.77	13.82	13.88
V8	25.63	26.3	26.73	27	27.24	27.35	27.43	27.5
V9	39.62	40.16	40.55	40.8	40.89	41	41.11	41.16
V10	55.1	55.43	55.7	55.79	55.86	55.92	55.96	56
V11	73.955	74	74.15	74.2	74.22	74.23	74.23	74.25
V12	19.94	20.36	20.66	20.82	20.95	21	21.1	21.12
V13	39.65	40.2	40.56	40.8	40.96	41	41.1	41.16
V14	58.93	59.42	59.67	59.85	59.9	60	60.05	60.08
V15	78.6	78.7	78.84	78.91	78.95	78.98	79	79.02
V16	28.84	29	29.227	29.34	29.42	29.47	29.5	29.523
V17	55.13	55.44	55.67	55.78	55.87	55.91	55.97	56
V18	78.51	78.83	78.841	78.91	78.95	79	79	79.02
V19	40.72	40.81	40.87	40.93	40.96	40.98	41	41
V20	74	74.06	74.13	74.18	74.2	74.25	74.24	74.25
V21	60.19	60.2	60.22	60.23	60.24	60.25	60.25	60.25

MATLAB PROGRAM:

APPENDIX A: A.1 MATLAB program for 2D problems using finite difference method**.**

Result:

Voltages =

```
                                                                                   x
0        0        0        0        0        0        0        0   50.0000

0    6.9652   13.9304   21.1778   29.5534   41.0167   60.2542  100.0000        0

0   13.9304   27.5788   41.2271   56.0194   74.2590  100.0000        0        0

0   21.1778   41.2271   60.1326   79.0380  100.0000        0        0        0

0   29.5534   56.0194   79.0380  100.0000        0        0        0        0

0   41.0167   74.2590  100.0000        0        0        0        0        0

0   60.2542  100.0000        0        0        0        0        0        0

0  100.0000        0        0        0        0        0        0        0

50.0000        0        0        0        0        0        0        0        0
y
```

Chapter 3

The finite element method

3.1 fundamentals of FEM

The finite element method has its origin in the field of structural analysis. The method was not applied to EM problems until 1968.Like the finite difference method, the finite element method is useful in solving differential equations. As finite difference method represents the solution region by array of grid points; its application becomes difficult with problems having irregularly shaped boundaries .Such problems can be handled more easily by using the finite element method. The finite elements analysis of any problem involves basically four steps: (A) discretizing the solution region into a finite number of sub regions or elements ,(B) deriving governing equations for a typical element ,(C) assembling all the elements in the solution region, and(D) solving the system of equations obtained.

A. FINITE ELEMENTS DISCRETIZATION

We divide the solution region into a number of finite elements as illustrated in figure3.1.

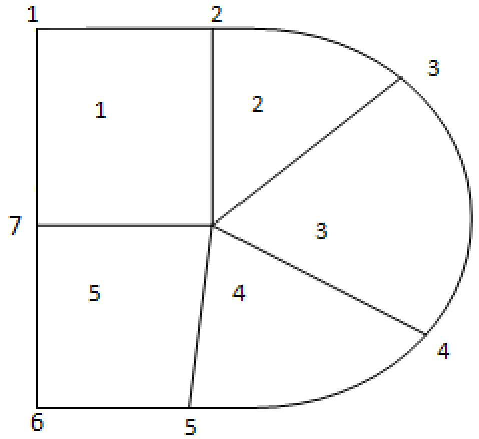


Fig3.1: A typical finite element subdivision of an irregular domain.

Where the region is subdivided into four non overlapping elements and seven nodes .We seek an approximation for the potential Ve within an element ‘e’ and then interrelate the potential distributions in various elements such that the potential is continuous across inter elements boundaries. The approximate solutions for the whole region is

$$\mathrm{V(x, y)} = \sum_{e=0}^{N} V_e(x, y) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3.1)}$$

Where N is the number of triangular elements into which the solution region is divided.

The most common from of approximation for $V_e$within an element is polynomial approximation, namely,

$$V_e\mathrm{(x,y)=a+bx+cy} \ldots\ldots\ldots\ldots\ldots\ldots.. \text{(3.2)}$$

For a triangular element and

$$V_e\mathrm{(x,y)=a+bx+cy+dx\ y} \ldots\ldots\ldots\ldots\ldots\ldots\ldots. \text{(3.3)}$$

for a quadrilateral element .The potential V_e in general is nonzero within element 'e" but zero outside "e" .It is difficult to approximate the boundary of the solution region with quadrilateral elements; such elements are useful for problems whose boundaries are sufficiently regular .As assumption of linear variation of potential within the triangular elements is same as assuming that the electric field is uniform within the element; that is,

$$E_e \bar{=} - \mathrm{V}\ V_e = -(\mathrm{b}a_x + \mathrm{c}a_y) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3.3)}$$

B. GOVERNING EQUATIONS OF EACH FINITE ELEMENT

Consider a typical triangular element, as shown in figure 3.2.The potentialV_{e1}, V_{e2} and V_{e3}at nodes 1, 2, 3, respectively, are obtained by using eq. (3.3); that is

$$\begin{bmatrix} \mathrm{Ve1} \\ \mathrm{Ve2} \\ \mathrm{Ve3} \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} \mathrm{a} \\ \mathrm{b} \\ \mathrm{c} \end{bmatrix}$$

The coefficients a, b, c is determined from above equation as

$$\begin{bmatrix} \mathrm{a} \\ \mathrm{b} \\ c \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} \mathrm{Ve1} \\ \mathrm{Ve2} \\ \mathrm{Ve3} \end{bmatrix}$$

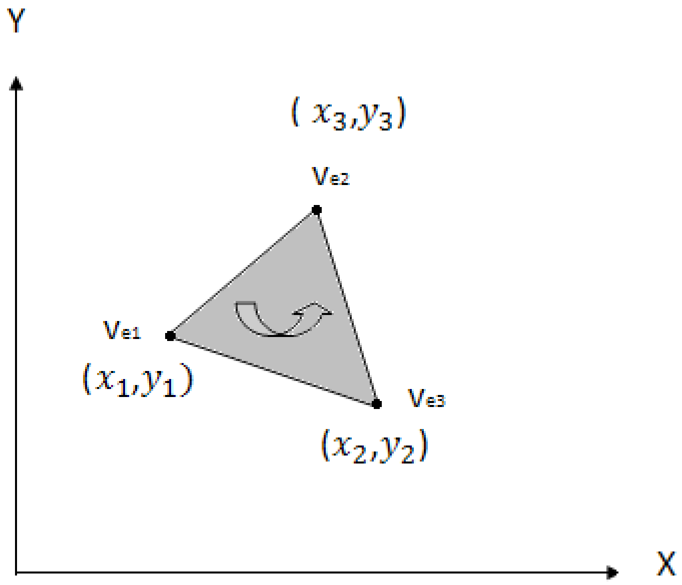


Fig3.2, typical triangular element.

Substituting this into above equation gives

$$\text{Ve}= [1 \quad x \quad y] \ \frac{1}{2A} \begin{bmatrix} x2y3 - x3y3 & x3y3 - x1y3 & x1y3 - x2 \\ y3 - y3 & y3 - y1 & y1 - y2 \\ x3 - x2 & x1 - x3 & x2 - x1 \end{bmatrix}$$

$$\text{Ve} = \sum_{i=1}^{3} \alpha_i(x,y)V_{ei} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3.6)}$$

$$\alpha_i\,(x_j,y_j) = \begin{cases} 1, \ i=j \\ 0\ , \ i\neq j \end{cases}$$

And A is the area of the element “e”; that is,

$$2A=\begin{vmatrix} 1 & x1 & y1 \\ 1 & x2 & y2 \\ 1 & x1 & y1 \end{vmatrix}$$

The value of A is positive if the nodes are numbered counterclockwise. Above equation gives the potential at any point within the element, provided the potentials at the vertices are known. This is unlike the in finite difference analysis, where the potential is known at the grid points only. Also note that α_i are linear interpolation functions, and they have the following properties.

$$\sum_{i=1}^{3} \alpha_i\,(x,y)=1 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3.7)}$$

Then we have to calculate the element coefficient matrix for each node separately, which is given below.

$$\begin{bmatrix} c_{11}^{e} & c_{12}^{e} & c_{13}^{e} \\ c_{21}^{e} & c_{22}^{e} & c_{23}^{e} \\ c_{31}^{e} & c_{32}^{e} & c_{33}^{e} \end{bmatrix} = [C^{e}]$$

p_1= y2-y3, Q_1 = x3-x2

p_2= y3-y1, Q_2 = x1-x3

p_3= y1-y2, Q_3= x2-x1

$C_{ij}^{(e)} = \frac{1}{4A}(p_i p_j + Q_i Q_j)$

A = $\frac{1}{2}(p_2 Q_3$-$p_3 Q_2)$

C. ASSEMBLING ALL THE ELEMENTS

Having considered a typical element, the next step is to assemble all such elements in the solution region. The energy associated with the assemblage of all elements in the mesh is

$$W=\sum_{e=1}^{N} W_e=\frac{1}{2}\varepsilon\,[V]'[C][V] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.7)$$

Where

$$[V] = \begin{bmatrix} V_1 \\ \cdot \\ \cdot \\ \cdot \\ V_n \end{bmatrix}$$

and n is the number of nodes, N is the number of elements, and [C] is called the overall or global coefficient matrix, which is the assemblage of individual element coefficient matrices.

D. SOLVING THE RESULTING EQUATIONS

BAND MATRIX METHOD:-

If all free nodes are numbered first and the fixed nodes last, eq. (3.7) can be written such that

$$W=\frac{1}{2}\varepsilon[V_f \quad V_p] \begin{bmatrix} c_{ff} & c_{fp} \\ c_{pf} & c_{pp} \end{bmatrix} \begin{bmatrix} V_f \\ V_p \end{bmatrix} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.8)$$

Where subscripts f and p, respectively, refer to nodes with free and fixed (or prescribed) potentials. Since V_p is constant, we differentiate only

With respect to V_f, yields

[C_{ff}][C_p]= -[C_{fp}][V_p] …………………………………… (3.9)

This can be written as

$$[A][V] = [B]$$

$$[V] = [A]^{-1}[B]$$

Where [V] = [V_f], [A] = [C_{ff}] and [B] = - [C_{fp}][V_p]

Since [A] is, in general, nonsingular, the potential at the free nodes can be found by using eq. (3.9).

3.2 Typical two dimensional electric field calculations by finite elements method

Here we have find the potential at the free nodes in the potential system using the finite elements method.

The solution region is divided into 25 three-node triangular elements with the total number of nodes being 21, shown in figure 3.3.

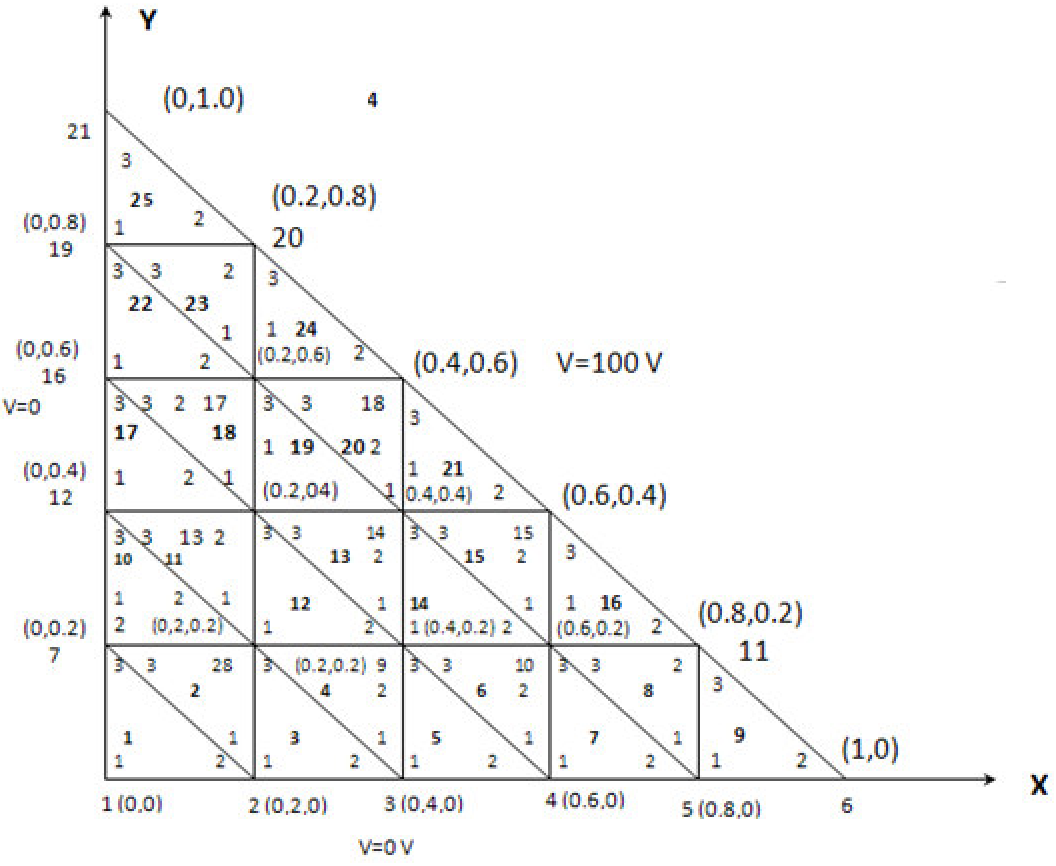
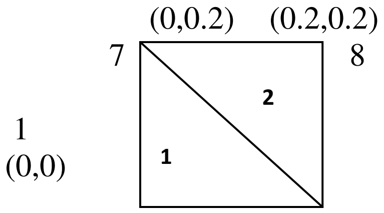
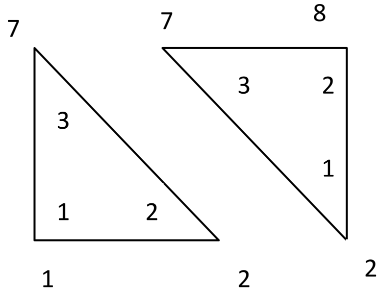


Figure 3.3, solution region divided into 25 triangular elements.

SOLUTION:

Determination of different elements coefficient matrix:





Node	(x,y)
1	(0,0)
2	(0.2,0)
7	(0,0.2)
8	(0.2,0.2)

P_1 = (y_2-y_3), P_2= (y_3-y_1)

P_3= (y_1-y_2)

Q1= (x_3-x_2),Q_2 =(x_1-x_3)

Q_3 = (x_2-x_1)

For element 1

1-2-7→1-2-3

P_1 = 0-0.2 = -0.2 $\quad$ Q_1 = 0-0.2 = -0.2

P_2 = 0.2-0 = 0.2 $\quad$ Q_2 = 0-0 = 0

P_3 = 0-0 = 0 $\quad$ Q_3 = 0.2 -0 = 0.2

A = ½ {0.2 x 0.2 – 0x 0} = $\frac{0.2^2}{2} = \frac{0.04}{2}$ = 0.02

A= 0.02

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$$C^{(1)} = \begin{bmatrix} 1 - 0.5 - 0.5 \\ -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$$

For element 2

2-8-7→1-2-3

P_1 = 0.2-0.2 = 0 $\quad$ Q_1 = 0-0.2 = -0.2

P_2 = 0.2-0 = 0.2 $\quad$ Q_2 = 0.2-0 = 0.2

P_3 = 0-0 =- 0.2 $\quad$ Q_3 = 0.2 -0.2 = 0

A = ½ {0.2 x 0.2 – 0x 0} = $\frac{0.2^2}{2} = \frac{0.04}{2}$ = 0.02

A= 0.02

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$C^{(2)} = \begin{bmatrix} 0.5 & -0.5 & 0 \\ -0.5 & 1 & -0.5 \\ 0 & -0.5 & 0.5 \end{bmatrix}$

For element 3

2-3-8→1-2-3

P_1 = 0-0.2 = -0.2 Q_1 = 0-0.2 = -0.2

P_2 = 0.2-0 = 0.2 Q_2 = 0-0 = 0

P_3 = 0-0 = 0 Q_3 = 0.2 -0 = 0.2

A = ½ {0.2 x 0.2 – 0x 0} = $\frac{0.2^2}{2} = \frac{0.04}{2}$ = 0.02

A= 0.02

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$C^{(3)} = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$

For element 4

3-9-8→1-2-3

P_1 = 0-0.2 = 0 Q_1 = 0.2-0.4 = -0.2

P_2 = 0.2-0 = 0.2 Q_2 = 0.4-0 .2= 0.2

P_3 = 0-0 = -0.2 Q_3 = 0.4 -0.4 = 0

A = ½ {0.2 x 0.2 – 0x 0} = $\frac{0.2^2}{2} = \frac{0.04}{2}$ = 0.02

A= 0.02

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$C^{(4)} = \begin{bmatrix} 0.5 & -0.5 & 0 \\ -0.5 & 1 & -0.5 \\ 0 & -0.5 & 0.5 \end{bmatrix}$

For element 5

3-4-9→1-2-3

P_1 = 0-0.2 = -0.2 Q_1 = 0.4-0.6 = -0.2

P_2 = 0.2-0 = 0.2 Q_2 = 0.4-0.4 = 0

P_3 = 0-0 = 0 Q_3 = 0.6 -0.4 = 0.2

A = ½ {0.2 x 0.2 – 0x 0} = $\frac{0.2^2}{2} = \frac{0.04}{2}$ = 0.02

A= 0.02

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$$C^{(5)} = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$$

For element 6

4-10-9→1-2-3

P_1 = 0.2-0.2 = 0 Q_1 = 0.4-0.6 = -0.2

P_2 = 0.2-0 = 0.2 Q_2 = 0.6-0.4 = 0.2

P_3 = 0-0.2 = -0.2 Q_3 = 0.6 -0.6 = 0

A = ½ {0.2 x 0 + 0.2x 0.2} = $\frac{0.2^2}{2} = \frac{0.04}{2}$ = 0.02

A= 0.02

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$$C^{(6)} = \begin{bmatrix} 0.5 & -0.5 & 0 \\ -0.5 & 1 & -0.5 \\ 0 & -0.5 & 0.5 \end{bmatrix}$$

For element 7

4-5-10→1-2-3

P_1 = 0-0.2 = -0.2 Q_1 = 0.6-0.8 = -0.2

P_2 = 0.2-0 = 0.2 Q_2 = 0.6-0.6 = 0

P_3 = 0-0 = 0 Q_3 = 0.8 -0.6 = 0.2

A = ½ {0.2 x 0.2 – 0x 0} = $\frac{0.2^2}{2} = \frac{0.04}{2}$ = 0.02

A= 0.02

Cij$^{(e)}$ = $\frac{1}{4A}$ ($P_iP_j + Q_iQ_j$)

C$^{(7)}$ =$\begin{bmatrix} 1-0.5-0.5 \\ -0.5 \quad 0.5 \quad\; 0 \\ -0.5 \quad 0 \;\; 0.5 \end{bmatrix}$

For element 8

5-11-10→1-2-3

P_1 = 0-0 = 0 Q_1 = 0.6-0.8 = -0.2

P_2 = 0.2-0 = 0.2 Q_2 = 0.8-0.6 = 0.2

P_3 = 0-0.2 = 0.2 Q_3 = 0.8 -0.8 = 0

A = ½ {0.2 x 0.2 + 0x 0} = $\frac{0.2^2}{2}$ = $\frac{0.04}{2}$ = 0.02

A= 0.02

Cij$^{(e)}$ = $\frac{1}{4A}$ ($P_iP_j + Q_iQ_j$)

C$^{(8)}$ =$\begin{bmatrix} 0.5-0.5 \quad 0 \\ -0.5 \quad 1 \quad -0.5 \\ 0 \quad -0.5 \quad 0.5 \end{bmatrix}$

For element 9

5-6-11→1-2-3

P_1 = 0-0.2 = -0.2 Q_1 = 0.8-1 = -0.2

P_2 = 0.2-0 = 0.2 Q_2 = 0.8-0.8 = 0

P_3 = 0-0 = 0 Q_3 = 1 -0.8 = 0.2

A = ½ {0.2 x 0.2 – 0x 0} = $\frac{0.2^2}{2}$ = $\frac{0.04}{2}$ = 0.02

A= 0.02

Cij$^{(e)}$ = $\frac{1}{4A}$ ($P_iP_j + Q_iQ_j$)

C$^{(9)}$ =$\begin{bmatrix} 1-0.5-0.5 \\ -0.5 \quad 0.5 \quad\; 0 \\ -0.5 \quad 0 \;\; 0.5 \end{bmatrix}$

For element 10

7-8-12→1-2-3

$P_1 = 0.2-0.4 = -0.2$ $Q_1 = 0-0.2 = -0.2$

$P_2 = 0.4-0.2 = 0.2$ $Q_2 = 0-0 = 0$

$P_3 = 0.2-0.2 = 0$ $Q_3 = 0.2 -0 = 0.2$

$A = ½ \{0.2 \text{ x } 0.2 – 0\text{x } 0\} = \frac{0.2^2}{2} = \frac{0.04}{2} = 0.02$

$A= 0.02$

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$$C^{(10)} = \begin{bmatrix} 1 - 0.5 - 0.5 \\ -0.5 \quad 0 \quad 0.5 \\ -0.5 \quad 0.5 \quad 0 \end{bmatrix}$$

For element 11

8-13-12→1-2-3

$P_1 = 0.4-0.4 = 0$ $Q_1 = 0-0.2 = -0.2$

$P_2 = 0.4-0.2 = 0.2$ $Q_2 = 0.2-0 = 0.2$

$P_3 = 0.2-0.4 =- 0.2$ $Q_3 = 0$

$A = ½ \{0.2 \text{ x } 0.2 – 0\text{x } 0\} = \frac{0.2^2}{2} = \frac{0.04}{2} = 0.02$

$A= 0.02$

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$$C^{(11)} = \begin{bmatrix} 0.5 - 0.5 \quad 0 \\ -0.5 \quad 0.5 \quad 0 \\ -0.5 \quad 0 \quad 0.5 \end{bmatrix}$$

For element 12

8-9-13→1-2-3

$P_1 = 0.2-0.4 = -0.2$ $Q_1 = 0.2-0.4 = -0.2$

$P_2 = 0.4-0.2 = 0.2$ $Q_2 = 0.2-0.2 = 0$

$P_3 = 0.2\text{-}0.2 = 0$ $\qquad\qquad$ $Q_3 = 0.4 \text{ -}0.2 = 0.2$

$A = ½ \{0.2 \text{ x } 0.2 – 0\text{x } 0\} = \frac{0.2^2}{2} = \frac{0.04}{2} = 0.02$

$A= 0.02$

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$$C^{(12)} = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$$

For element 13

9-14-13→1-2-3

$P_1 = 0.4\text{-}0.4 = 0$ $\qquad\qquad$ $Q_1 = 0.2\text{-}0.4 = \text{-}0.2$

$P_2 = 0.4\text{-}0.2 = 0.2$ $\qquad\qquad$ $Q_2 = 0.4\text{-}0.2 = 0.2$

$P_3 = 0.2\text{-}0.4 = 0$ $\qquad\qquad$ $Q_3 = 0$

$A = ½ \{0.2 \text{ x } 0.2 – 0\text{x } 0\} = \frac{0.2^2}{2} = \frac{0.04}{2} = 0.02$

$A= 0.02$

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$$C^{(13)} = \begin{bmatrix} 0.5 & -0.5 & 0 \\ -0.5 & 1 & 0.5 \\ 0 & -0.5 & 0.5 \end{bmatrix}$$

For element 14

9-10-14→1-2-3

$P_1 = 0.2\text{-}0.4 = \text{-}0.2$ $\qquad\qquad$ $Q_1 = 0.4\text{-}0.6 = \text{-}0.2$

$P_2 = 0.4\text{-}0.2 = 0.2$ $\qquad\qquad$ $Q_2 = 0.4\text{-}0.4 = 0$

$P_3 = 0\text{-}0 = 0$ $\qquad\qquad$ $Q_3 = 0.6 \text{ -}0.4 = 0.2$

$A = ½ \{0.2 \text{ x } 0.2 – 0\text{x } 0\} = \frac{0.2^2}{2} = \frac{0.04}{2} = 0.02$

$A= 0.02$

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$C^{(14)} = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$

For element 15

10-15-14→1-2-3

$P_1 = 0$ $\quad Q_1 = 0\text{-}0.2 = -0.2$

$P_2 = 0.4\text{-}0.2 = 0.2$ $\quad Q_2 = 0.6\text{-}0.4= 0$

$P_3 = 0.2\text{-}0 .4= -0.2$ $\quad Q_3 = 0.6 \text{-}0 .6= 0$

$A = ½ \{0.2 \text{ x } 0.2 – 0\text{x } 0\} = \frac{0.2^2}{2} = \frac{0.04}{2} = 0.02$

A= 0.02

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$C^{(15)} = \begin{bmatrix} 0.5 & -0.5 & 0 \\ -0.5 & 1 & -0.5 \\ 0 & -0.5 & 0.5 \end{bmatrix}$

For element 16

10-11-15→1-2-3

$P_1 = 0.2\text{-}0.4 = -0.2$ $\quad Q_1 = 0.6\text{-}0.8 = -0.2$

$P_2 = 0.4\text{-}0 .2= 0.2$ $\quad Q_2 = 0.6\text{-}0.6 = 0$

$P_3 = 0\text{-}0 = 0$ $\quad Q_3 = 0.8 \text{-}0.6 = 0.2$

$A = ½ \{0.2 \text{ x } 0.2 – 0\text{x } 0\} = \frac{0.2^2}{2} = \frac{0.04}{2} = 0.02$

A= 0.02

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$C^{(16)} = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$

For element 17

12-13-16→1-2-3

P_1 = 0.4-0.6 = -0.2 Q_1 = 0-0.2 = -0.2

P_2 = 0.6-0.4 = 0.2 Q_2 = 0-0 = 0

P_3 = 0-0 = 0 Q_3 = 0.2 -0 = 0.2

A = ½ {0.2 x 0.2 – 0x 0} = $\frac{0.2^2}{2}$ = $\frac{0.04}{2}$ = 0.02

A= 0.02

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$$C^{(17)} = \begin{bmatrix} 1 - 0.5 - 0.5 \\ -0.5 \quad 0.5 \quad 0 \\ -0.5 \quad 0 \quad 0.5 \end{bmatrix}$$

For element 18

13-17-16→1-2-3

P_1 = 0.6-0.6 = 0 Q_1 = 0-0.2 = -0.2

P_2 = 0.6-0.4 = 0.2 Q_2 = 0.2-0 = 0.2

P_3 = 0.4-0.6 =- 0.2 Q_3 = 0.2 -0.2 = 0

A = ½ {0.2 x 0.2 – 0x 0} = $\frac{0.2^2}{2}$ = $\frac{0.04}{2}$ = 0.02

A= 0.02

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$$C^{(18)} = \begin{bmatrix} 0.5 - 0.5 \quad 0 \\ -0.5 \quad 1 \quad 0.5 \\ 0 \;-\; 0.5 \quad 0.5 \end{bmatrix}$$

For element 19

13-14-17→1-2-3

P_1 = 0.4-0.6 = -0.2 Q_1 = 0.2-0.4 = -0.2

P_2 = 0.6-0.4= 0.2 Q_2 = 0.2-0.2 = 0

P_3 = 0-0 = 0 Q_3 = 0.4 -0.2 = 0.2

A = ½ {0.2 x 0.2 – 0x 0} = $\frac{0.2^2}{2} = \frac{0.04}{2}$ = 0.02

A= 0.02

Cij$^{(e)}$ = $\frac{1}{4A}$ (P_iP_j + Q_iQ_j)

C$^{(19)}$ =$\begin{bmatrix} 1 - 0.5 - 0.5 \\ -0.5 \quad 0.5 \quad\quad 0 \\ -0.5 \quad 0 \quad 0.5 \end{bmatrix}$

For element 20

14-18-17→1-2-3

P_1 = 0.6-0.6 = 0 Q_1 = 0.2-0.4 = -0.2

P_2 = 0.4-0.6 = -0.2 Q_2 = 0.4-0.2= 0.2

P_3 = 0.4-0.6 = -0.2 Q_3 = 0.4 -0.4 = 0

A = ½ {0.2 x 0.2 – 0x 0} = $\frac{0.2^2}{2} = \frac{0.04}{2}$ = 0.02

A= 0.02

Cij$^{(e)}$ = $\frac{1}{4A}$ (P_iP_j + Q_iQ_j)

C$^{(20)}$ =$\begin{bmatrix} 0.5 - 0.5 \quad 0 \\ -0.5 \quad 1 \quad - 0.5 \\ 0 \quad - 0.5 \quad 0.5 \end{bmatrix}$

For element 21

14-15-18→1-2-3

P_1 = 0.4-0.6 = -0.2 Q_1 = 0.4-0.6 = -0.2

P_2 = 0.6-0.4= 0.2 Q_2 = 0.4-0.4 = 0

P_3 = 0.4-0.4 = 0 Q_3 = 0.2

A = ½ {0.2 x 0.2 – 0x 0} = $\frac{0.2^2}{2} = \frac{0.04}{2}$ = 0.02

A= 0.02

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$$C^{(21)} = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$$

For element 22

16-17-19→1-2-3

$P_1 = 0.6-0.8 = -0.2$ $\quad$ $Q_1 = 0-0.2 = -0.2$

$P_2 = 0.8-0.6 = 0.2$ $\quad$ $Q_2 = 0-0 = 0$

$P_3 = 0.6-0.6 = 0$ $\quad$ $Q_3 = 0.2 -0 = 0.2$

$A = ½ \{0.2 \text{ x } 0.2 – 0\text{x } 0\} = \frac{0.2^2}{2} = \frac{0.04}{2} = 0.02$

$A= 0.02$

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$$C^{(22)} = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$$

For element 23

17-20-19→1-2-3

$P_1 = 0$ $\quad$ $Q_1 = 0-0.2 = -0.2$

$P_2 = 0.8-0.6= 0.2$ $\quad$ $Q_2 = 0-0 = 0$

$P_3 = 0.6-0.8 =- 0.2$ $\quad$ $Q_3 = 0.2 -0 = 0.2$

$A = ½ \{0.2 \text{ x } 0.2 – 0\text{x } 0\} = \frac{0.2^2}{2} = \frac{0.04}{2} = 0.02$

$A= 0.02$

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$$C^{(23)} = \begin{bmatrix} 0.5 & -0.5 & 0 \\ -0.5 & 1 & -0.5 \\ 0 & -0\ .5 & 0.5 \end{bmatrix}$$

For element 24

17-18-20→1-2-3

$P_1 = 0.6-0.8 = -0.2$ $\qquad$ $Q_1 = 0.2-0.4 = -0.2$

$P_2 = 0.8-0.6 = 0.2$ $\qquad$ $Q_2 = 0.2-0.2 = 0$

$P_3 = 0$ $\qquad$ $Q_3 = 0.4 -0.2 = 0.2$

$A = ½ \{0.2 \text{ x } 0.2 – 0\text{x } 0\} = \frac{0.2^2}{2} = \frac{0.04}{2} = 0.02$

A= 0.02

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$$C^{(24)} = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$$

For element 25

19-20-21→1-2-3

$P_1 = 0.8-1 = -0.2$ $\qquad$ $Q_1 = 0-0.2=- 0.2$

$P_2 = 1-0.8 = 0.2$ $\qquad$ $Q_2 = 0-0 = 0$

$P_3 = 0-0 = 0$ $\qquad$ $Q_3 = 0.2 -0 = 0.2$

$A = ½ \{0.2 \text{ x } 0.2 – 0\text{x } 0\} = \frac{0.2^2}{2} = \frac{0.04}{2} = 0.02$

A= 0.02

$Cij^{(e)} = \frac{1}{4A} (P_iP_j + Q_iQ_j)$

$$C^{(25)} = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$$

$$[Cff] = \begin{bmatrix} C_{88} & C_{89} & C_{810} & C_{813} & C_{814} & C_{817} \\ C_{98} & C_{99} & C_{910} & C_{913} & C_{914} & C_{917} \\ C_{108} & C_{109} & C_{1010} & C_{1013} & C_{1014} & C_{1017} \\ C_{138} & C_{139} & C_{1310} & C_{1313} & C_{1314} & C_{1317} \\ C_{148} & C_{149} & C_{1410} & C_{1413} & C_{1414} & C_{1417} \\ C_{178} & C_{179} & C_{1710} & C_{1713} & C_{1714} & C_{1717} \end{bmatrix}$$

$$[\mathrm{Cff}] = \begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & -1 & 4 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & & -1 \\ 0 & -1 & 0 & -1 & 4 & 0 \\ 0 & 0 & 0 & -1 & 0 & 4 \end{bmatrix}$$

$$[\mathrm{B}] = \begin{bmatrix} 0 \\ 0 \\ -200 \\ 0 \\ -200 \\ -200 \end{bmatrix}$$

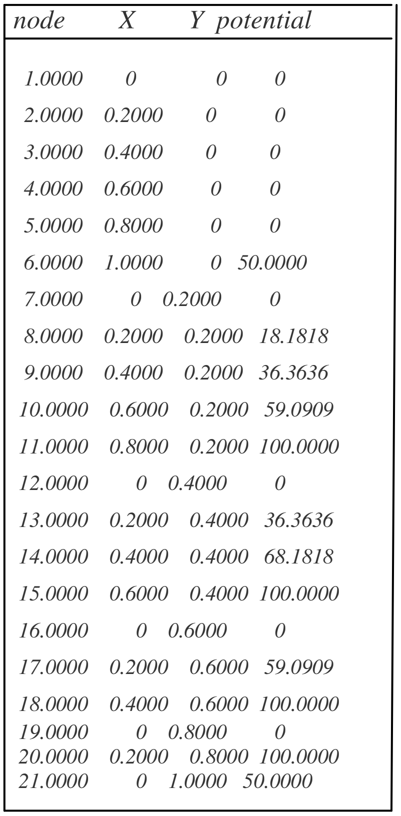
$$[\mathrm{V}] = [A]^{-1}[\mathrm{B}]$$

$$[V]^{T} = \begin{bmatrix} 18.1818 \\ 36.3636 \\ 59.0909 \\ 36.3636 \\ 68.1818 \\ 59.0909 \end{bmatrix} \qquad \text{(ANSWER)}$$

MATLAB PROGRAM:

APPENDIX A: A.2 MATLAB program for 2D problem using finite element method.

Result:



```
node        X         Y  potential

  1.0000       0            0        0
  2.0000   0.2000       0         0
  3.0000   0.4000       0        0
  4.0000   0.6000        0        0
  5.0000   0.8000        0        0
  6.0000   1.0000        0   50.0000
  7.0000        0    0.2000        0
  8.0000   0.2000    0.2000   18.1818
  9.0000   0.4000    0.2000   36.3636
 10.0000    0.6000    0.2000   59.0909
 11.0000    0.8000    0.2000  100.0000
 12.0000        0    0.4000        0
 13.0000    0.2000    0.4000   36.3636
 14.0000    0.4000    0.4000   68.1818
 15.0000    0.6000    0.4000  100.0000
 16.0000        0    0.6000        0
 17.0000    0.2000    0.6000   59.0909
 18.0000    0.4000    0.6000  100.0000
 19.0000        0    0.8000        0
 20.0000    0.2000    0.8000  100.0000
 21.0000        0    1.0000   50.0000
```

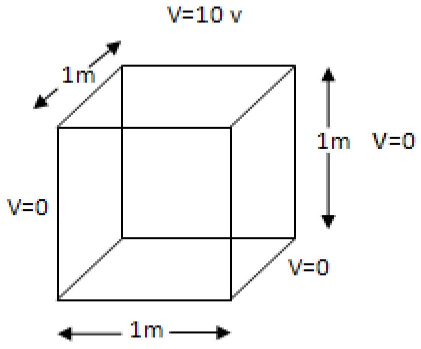
Chapter 4

Three dimensional electric field calculations

4.1 Three-dimensional Laplace's equation

The three –dimensional box is given below. The upper plate is connected to a 10V potential and five plates are grounded. We have found the potential and electric field intensity everywhere inside box.

Here finite difference method uses a rectangular grid. A three-dimensional grid, with five divisions in each directional is defined over the mesh as shown figure (4.1b).the mesh has total of 64 internal nodes which must be evaluated.



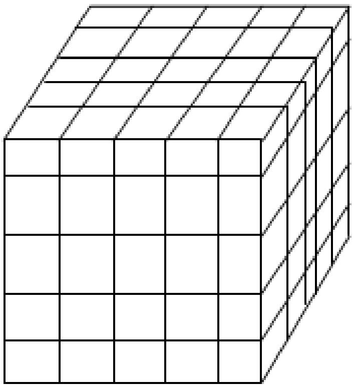


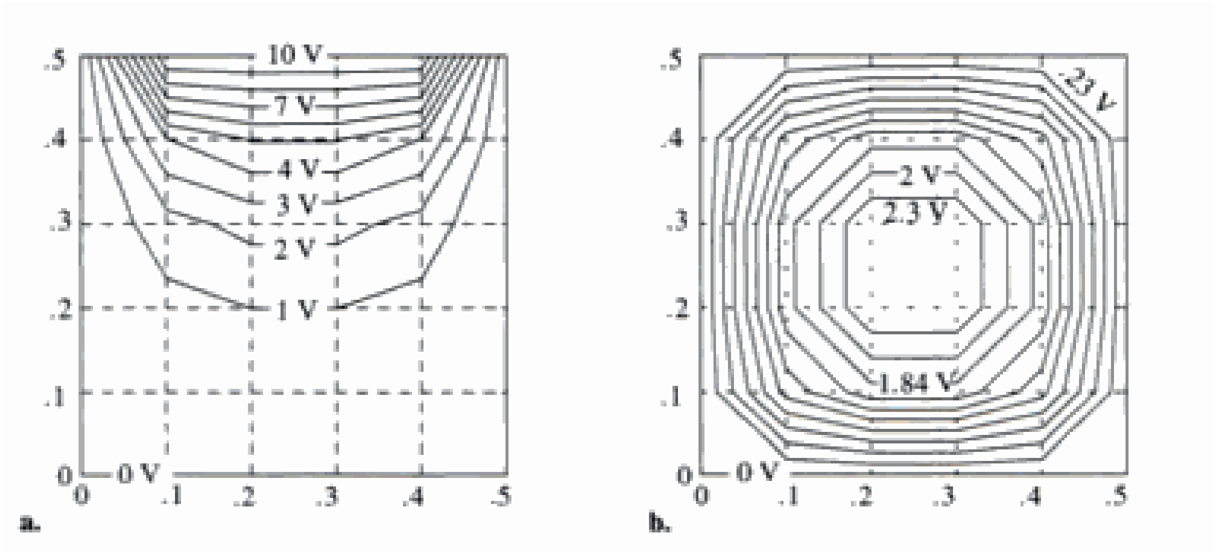
Fig 4.1a, conducting box with given boundary potentials. Fig 4.1b, 5×5×5 finite difference grid over the box.

Program in C:

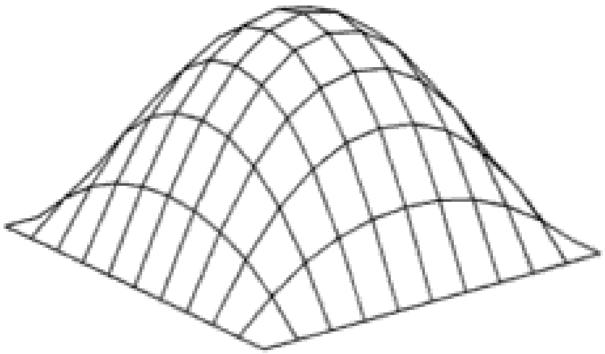
APPENDIX B: B.1 C programming for Three-dimensional Laplace's equation.

Results:-

Below two contour plots of potential distributions and equipotential lines and 3D potential distribution in box are shown.



Figure,(4.2a)contour plot on cross section cut vertically,Fig(4.2b)contour on a cross section cut horizontally.both plots are for the $5 \times 5 \times 5$ grids.



Figure(4.2c), three dimentional plot of the potential distribution in box.

4.2 Non-uniformly distributed dielectric of a capacitor

A parallel plate capacitor is given. The insulator between the plates is mica with relative permittivity ε_r = 4. Because of problems in production, there is a fault in the mica in the form a rectangular vein, as shown. The vain may be considered to be air. We have to calculate the potential everywhere inside the capacitor if a potential V=100 Volts is connected across the plates, the electric field intensity at the centre of the fault for the condition in above and what the maximum potential difference allowable with and without fault.

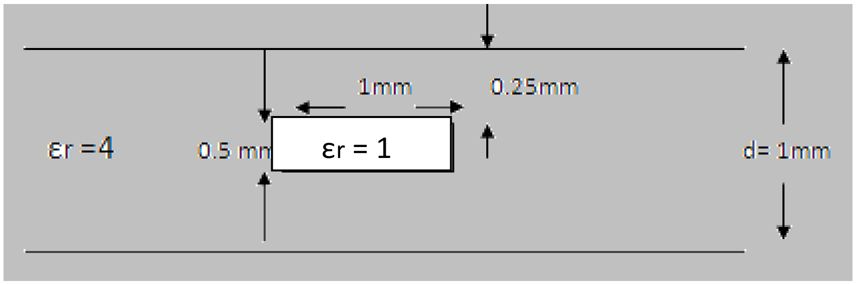


Figure (4.3a), shows mica insulated, parallel plate capacitor with a small flaw in the dielectric.

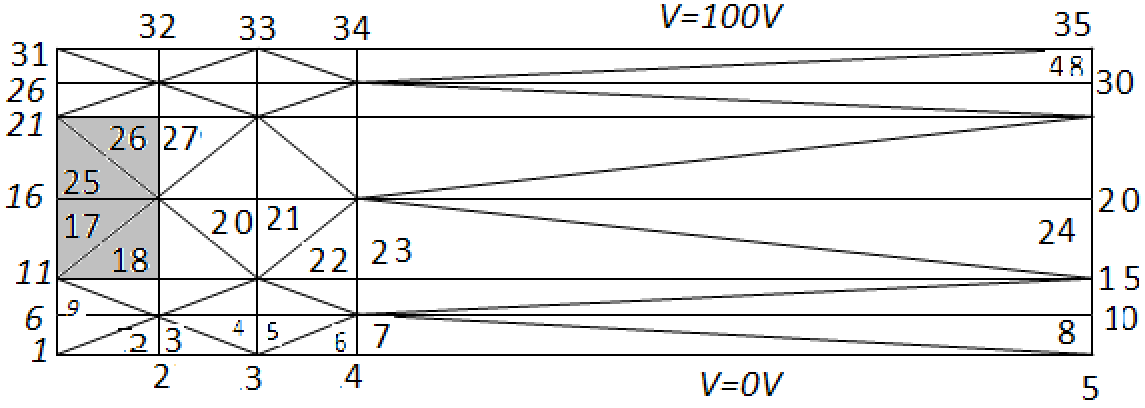


Figure (4.3b) shows a finite elements mesh and boundary conditions for the half of the capacitor.

Program:

APPENDIX B: B.2 C programming for non-uniformly distributed dielectric of capacitor.

Result:

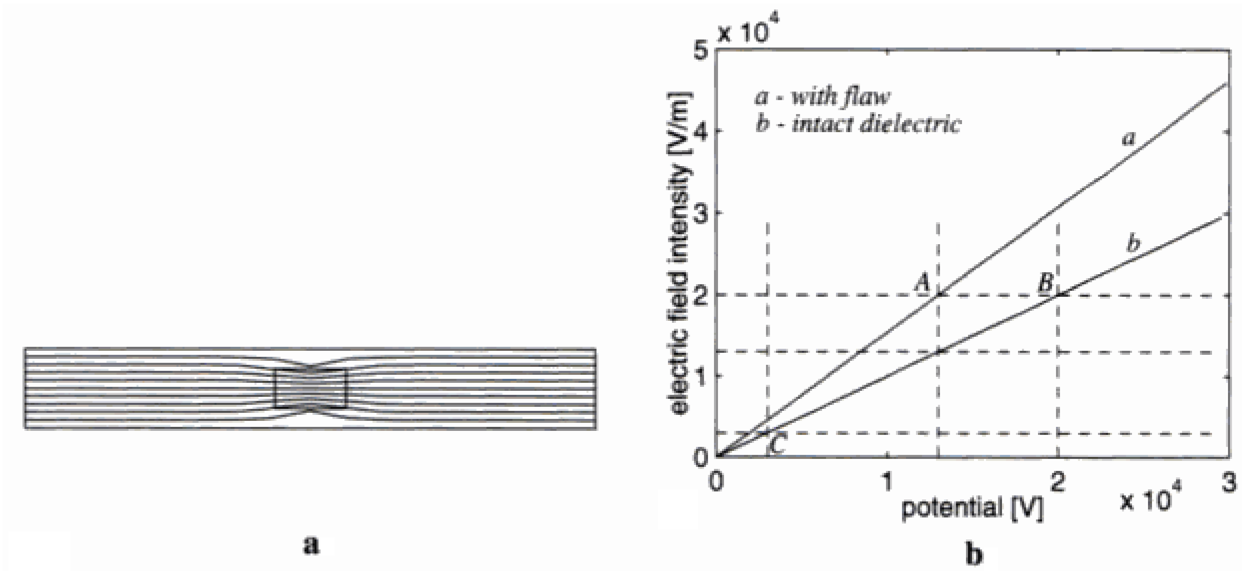
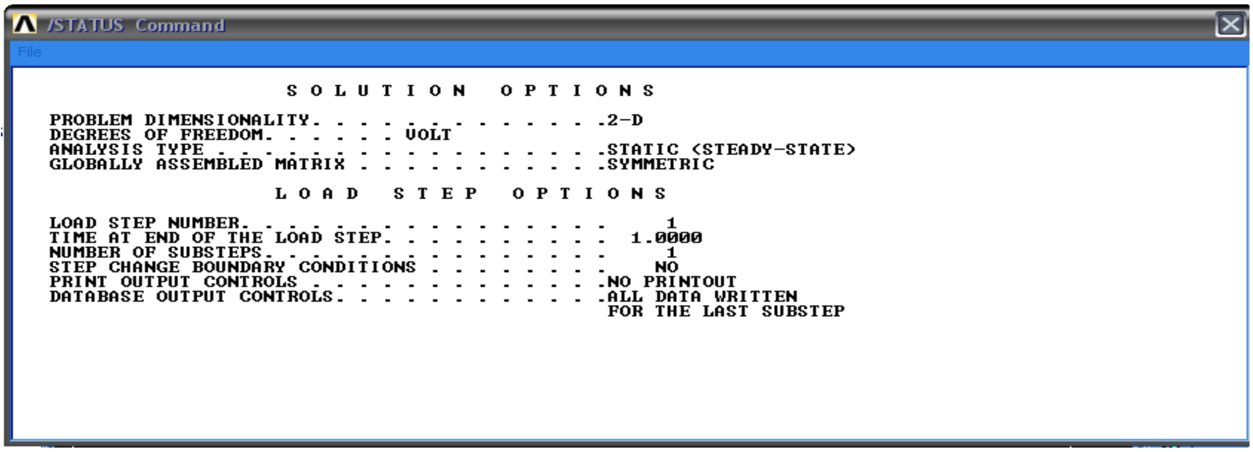


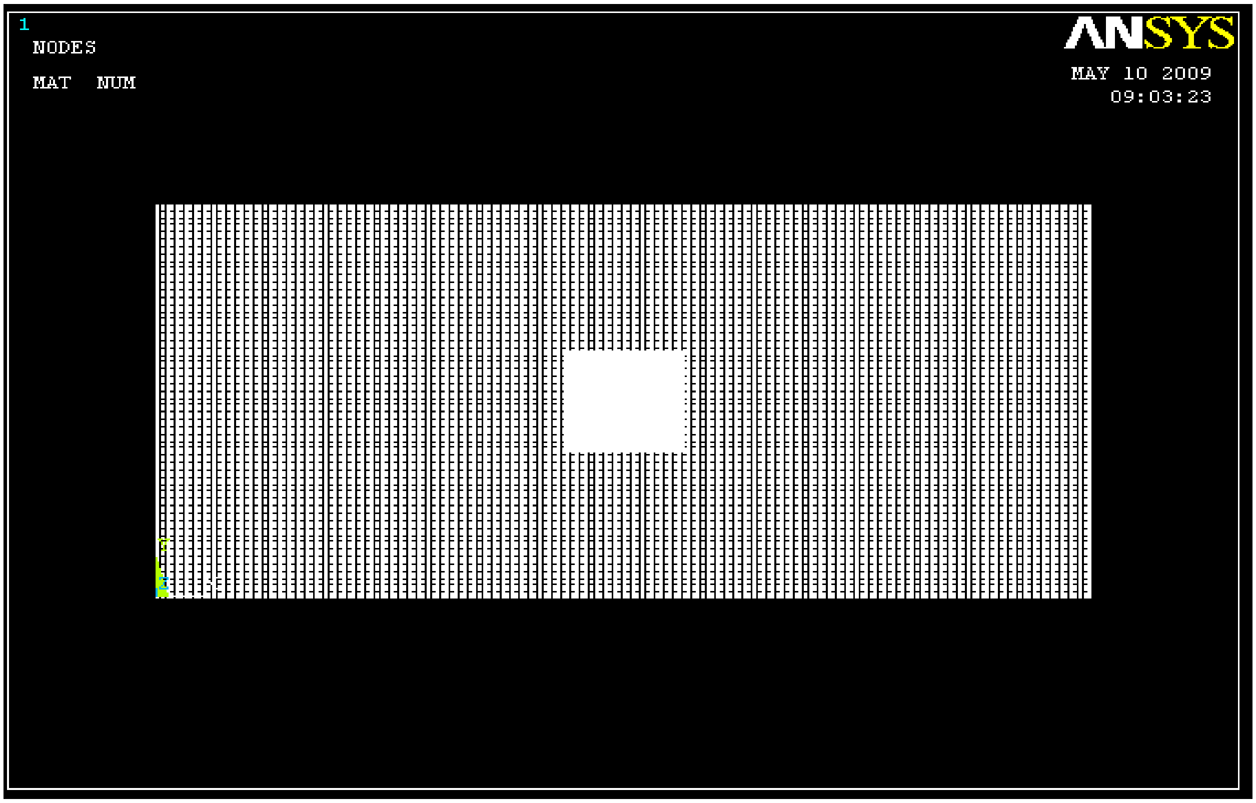
Figure (4.4a), Contours of constant potential in the capacitor. (4.4b) Magnitude of electric field intensity in the flaw as a function of applied potential.

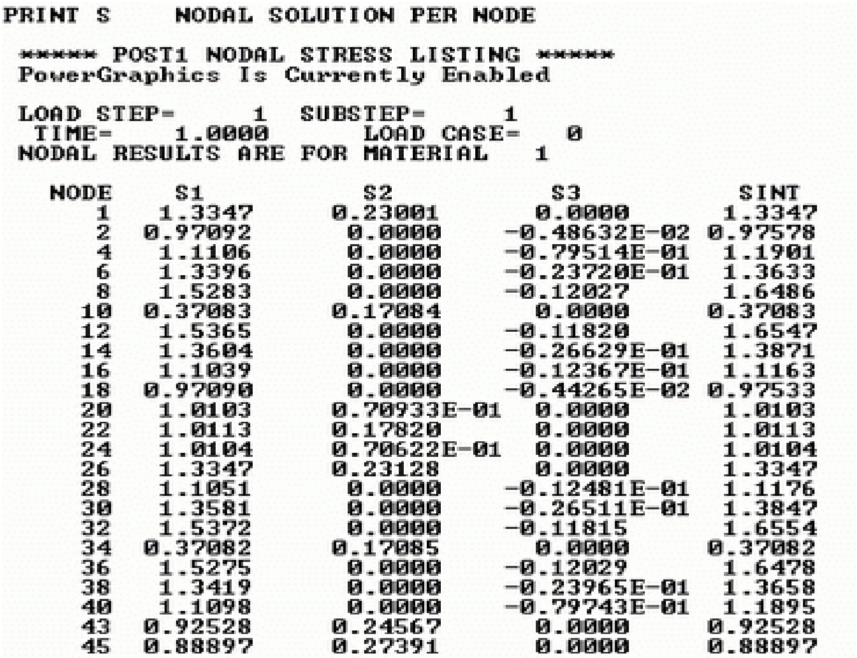
According to the output of program, when the flaw is present, the potential is not uniformly distributed and we must calculate the potential difference numerically. However, there is a slight difficulty here: we are in effect trying to find the boundary conditions that will provide the maximum electric field intensity allowable in the flaw. The finite elements method requires known boundary conditions to calculate the field intensity. The way we approach this problems to start with a known potential difference and increment the potential on the boundary until the electric field intensity in the flaw will equal the breakdown electric field intensity. The potential difference we obtained is the maximum allowable potential difference. In other words we run the finite element program with known trial potentials and choose the potential which provides the required result.

Analysis of above problem in ANSYS:-



```
                        S O L U T I O N   O P T I O N S

  PROBLEM DIMENSIONALITY. . . . . . . . . . . . .2-D
  DEGREES OF FREEDOM. . . . . . VOLT
  ANALYSIS TYPE . . . . . . . . . . . . . . . . .STATIC (STEADY-STATE)
  GLOBALLY ASSEMBLED MATRIX . . . . . . . . . . .SYMMETRIC

                        L O A D   S T E P   O P T I O N S

  LOAD STEP NUMBER. . . . . . . . . . . . . . . .     1
  TIME AT END OF THE LOAD STEP. . . . . . . . . .  1.0000
  NUMBER OF SUBSTEPS. . . . . . . . . . . . . . .     1
  STEP CHANGE BOUNDARY CONDITIONS . . . . . . . .    NO
  PRINT OUTPUT CONTROLS . . . . . . . . . . . . .NO PRINTOUT
  DATABASE OUTPUT CONTROLS. . . . . . . . . . . .ALL DATA WRITTEN
                                                  FOR THE LAST SUBSTEP
```





```
PRINT S     NODAL SOLUTION PER NODE

 ***** POST1 NODAL STRESS LISTING *****
 PowerGraphics Is Currently Enabled

 LOAD STEP=     1  SUBSTEP=     1
  TIME=    1.0000      LOAD CASE=   0
 NODAL RESULTS ARE FOR MATERIAL    1

   NODE     S1          S2          S3          SINT
      1   1.3347     0.23001     0.0000       1.3347
      2  0.97092     0.0000    -0.48632E-02  0.97578
      4   1.1106     0.0000    -0.79514E-01   1.1901
      6   1.3396     0.0000    -0.23720E-01   1.3633
      8   1.5283     0.0000    -0.12027       1.6486
     10  0.37083     0.17084     0.0000      0.37083
     12   1.5365     0.0000    -0.11820       1.6547
     14   1.3604     0.0000    -0.26629E-01   1.3871
     16   1.1039     0.0000    -0.12367E-01   1.1163
     18  0.97090     0.0000    -0.44265E-02  0.97533
     20   1.0103    0.70933E-01  0.0000       1.0103
     22   1.0113    0.17820      0.0000       1.0113
     24   1.0104    0.70622E-01  0.0000       1.0104
     26   1.3347    0.23128      0.0000       1.3347
     28   1.1051     0.0000    -0.12481E-01   1.1176
     30   1.3581     0.0000    -0.26511E-01   1.3847
     32   1.5372     0.0000    -0.11815       1.6554
     34  0.37082    0.17085      0.0000      0.37082
     36   1.5275     0.0000    -0.12029       1.6478
     38   1.3419     0.0000    -0.23965E-01   1.3658
     40   1.1098     0.0000    -0.79743E-01   1.1895
     43  0.92528    0.24567      0.0000      0.92528
     45  0.88897    0.27391      0.0000      0.88897
```

4.3 **Electric fields near a dc busbar**

A busbar used in the distribution of electric power in a distribution box is at potential 220 volts. The busbar is a rectangular conductor as shown. The ground and the busbar may be considered to be perfect conductors. We have found electric potential everywhere in space and location and magnitude of the maximum electric fields intensity.

Solution:

We again start by defining the geometry and the boundary conditions. Using a symmetry line vertically through the center of the bar, we eliminate half the geometry. Next, we must place artificial boundaries at some distant from the bar. By placing these boundaries at reasonable distance from the source, the solution can be an accurate while the mesh required is reasonably small. In this case, the boundaries are placed at 0.1 m from the symmetry line and 0.12m from the ground plane, as shown. A total of 52 elements and 39 nodes are used. The left boundary is left unspecified with the exception of the bar which is held at 100volts.

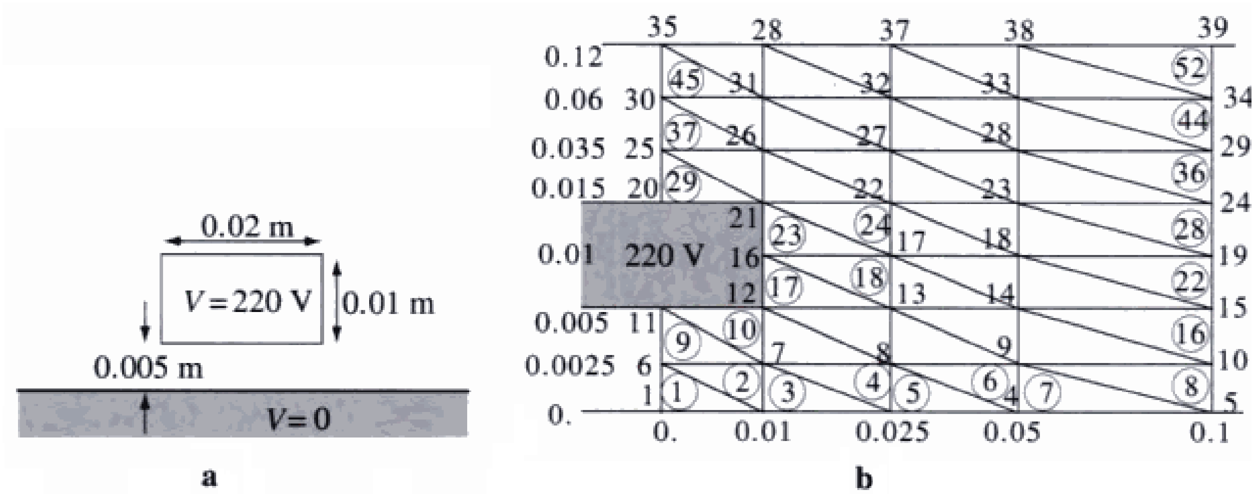


Fig 4.5a busbar at 220 V over conducting ground, fig 4.5b placement of artificial boundary and symmetry for geometry in fig. (4.5a), mesh is shown.

Program in C:

Program for this problem is same as that of program for flaw in the dielectric of a capacitor; because that program is general program for finite elements methods. Only we have changed the input data.

Input data:

APPENDIX B: B.3 input data for DC busbar.

Results:

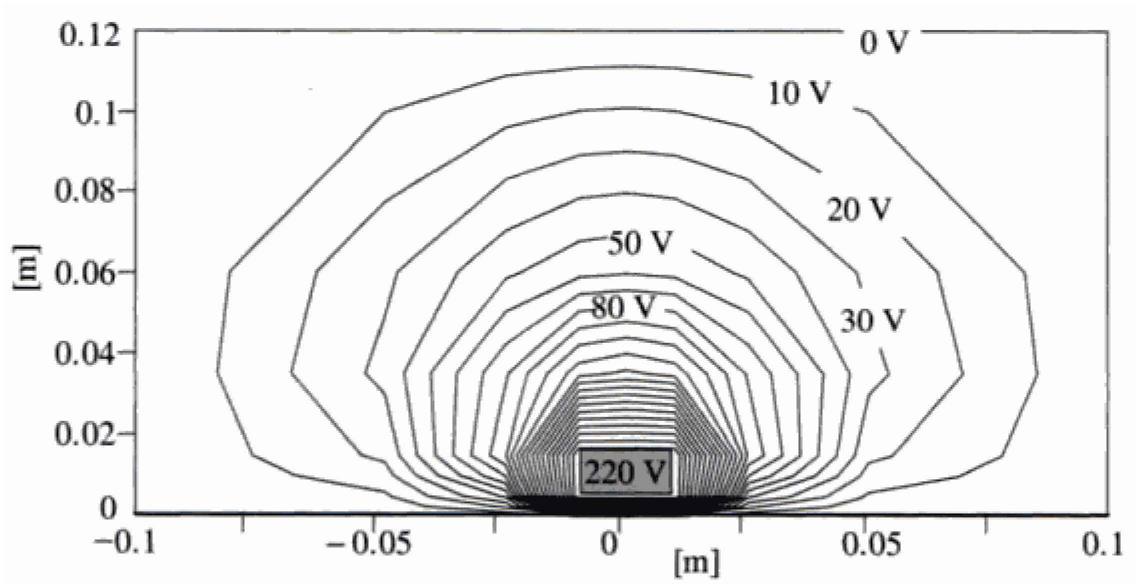


Figure 4.6, shows the Contour plot of the solutions.

Above plot shows the equipotential lines. The magnitude of the electric field intensity is highest in element No.11 and equals 4.496× 10^4 v/m and corresponds to a distance of about 1.7 mm below and about 10 mm to the right of the corner of the busbar. That the maximum electric field must be around the corner is expected, but the exact location and magnitude depends on the dimensions and on the mesh used.

Chapter 5

Finite Element Analysis using ANSYS

The finite element method has become a staple for predicting and simulating the physical behavior of complex engineering systems. The commercial finite element analysis programs have gained common acceptance among engineers in industry and researchers at universities and governments laboratories. Most widely used commercial finite element analysis program is the ANSYS.

5.1 Fundamental of ANSYS:

Jobname: a specific name to be used for the files created during an ANSYS session. This name can be assigned either before or after starting the ANSYS program.

Working Directory: a specific folder for ANSYS to store all of the files created during a session. It is possible to specify the Working Directory before or after starting ANSYS.

Interactive Mode: This is the most common mode of interaction between the user and the ANSYS program. It involves activation of a platform called Graphical User Interface {GUI), which is composed of menus, dialog boxes, push-buttons, and different windows. Interactive Mode is the recommended mode for beginner ANSYS users as it provides an excellent platform for learning. It is also highly effective for postprocessing.

Batch Mode: This is a method to use the ANSYS program without activating the GUI It involves an Input File written in ANSYS Parametric Design Language {APDL), which allows the use of parameters and common programming features such as DO loops and IF statements. These capabilities make the Batch Mode a very powerful analysis tool. Another distinct advantage of the Batch Mode is realized when there is an error/mistake in the model generation. This type of problem can be fixed by modifying a small portion of the Input File and reading it again, saving the user a great deal of time.

Combined Mode: This is a combination of the Interactive and Batch Modes in which the user activates the GUI and reads the Input File. Typically, this method allows the user to generate the model and obtain the solution using the Input File while reviewing the results using the Postprocessor within the GUL This method combines the salient advantages of the Interactive and Batch Modes.

Before an ANSYS Session

The construction of solutions to engineering problems using FEA requires either the development of a computer program based on the FEA formulation or the use of a commercially available general-purpose FEA program such as ANSYS. The ANSYS program is a powerful, multi-purpose analysis tool that can be used in a wide variety of engineering disciplines. Before using ANSYS to generate an FEA model of a physical system, the following questions should be answered based on engineering judgment and observations:

- What are the objectives of this analysis?
- Should the entire physical system be modeled, or just a portion?
- How much detail should be included in the model?
- How refined should the finite element mesh be?

In answering such questions, the computational expense should be balanced against the accuracy of the results. Therefore, the ANSYS finite element program can be employed in a correct and efficient way after considering the following:

- Type of problem.
- Time dependence.
- Nonlinearity.
- Modeling idealizations/simplifications.

Analysis Discipline

The ANSYS program is capable of simulating problems in a wide range of engineering disciplines. Example- structural analysis, thermal analysis, electromagnetic analysis etc.

Degrees of Freedom

The ANSYS solution for each of these analysis disciplines provides nodal values of the field variable. This primary unknown is called a degree of freedom (DOF). The degrees of freedom for these disciplines are presented in Table 5.1. The analysis discipline should be chosen based on the quantities of interest.

Discipline	quantity	DOF
Electrostatic	electric potential, Charge density	electric potential
magnetostatics	magnetic potential, Magnetic potential	magnetic potential

Table 5.1

Time Dependence:

The analysis with ANSYS should be time-dependent if:

- The solid body is subjected to time varying loads.
- The solid body has an initially specified temperature distribution.
- The body changes phase.

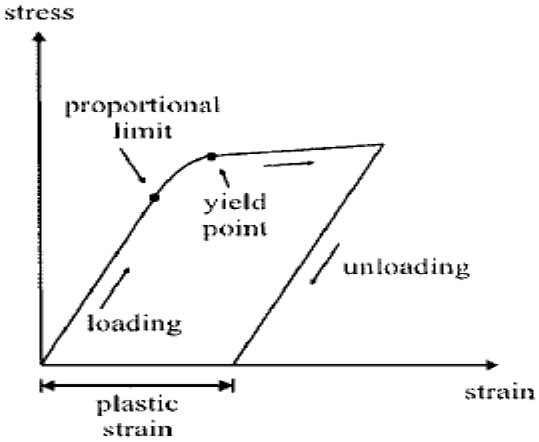


Fig.5.1 Non-linear material response.

Nonlinearity:

Most real-world physical phenomena exhibit nonlinear behavior. There are many situations in which assuming a linear behavior for the physical system might provide satisfactory results. On the other hand, there are circumstances or phenomena that might require a nonlinear solution. A nonlinear structural behavior may arise because of geometric and material nonlinearities, as well as a change in the boundary conditions and structural integrity. These nonlinearities are discussed briefly in the following subsections.

Geometric Nonlinearity:

There are two main types of geometric nonlinearity:

Large deflection and rotation: If the structure undergoes large displacements compared to its smallest dimension and rotations to such an extent that it's original dimensions and position, as well as the loading direction, change significantly, the large deflection and rotation analysis becomes necessary. For example, a fishing rod with a low lateral stiffness under a lateral load experiences large deflections and rotations.

Stress stiffening: When the stress in one direction affects the stiffness in another direction, stress stiffening occurs. Typically, a structure that has little or no stiffness in compression while having considerable stiffness in tension exhibits this behavior. Cables, membranes, or spinning structures exhibit stress stiffening.

Material Nonlinearity:

A typical nonlinear stress-strain curve is given in Fig. A linear material response is a good approximation if the material exhibits a nearly linear stress-strain curve up to a proportional limit and the loading is in a manner that does not create stresses higher than the yield stress anywhere in the body.

Nonlinear material behavior in ANSYS is characterized as:

Plasticity: Permanent, time-independent deformation.

Creep: Permanent, time-dependent deformation.

Nonlinear Elastic: Nonlinear stress-strain curve; upon unloading, the structure returns back to its original state—no permanent deformations.

Viscoelasticity: Time-dependent deformation under constant load. Full recovery upon unloading.

Hyperelasticity: Rubber-like materials

Practical Modeling Considerations

In order to reduce computational time, minor details that do not influence the results should not be included in the FE model. Minor details can also be ignored in order to render the geometry symmetric, which leads to a reduced FE model. However, in certain structures, "small'* details such as fillets or holes may be the areas of maximum stress, which might prove to be extremely important in the analysis and design. Engineering judgment is essential to balance the possible gain in computational cost against the loss of accuracy.

Mesh Density:

In general, a large number of elements provide a better approximation of the solution However, in some cases, an excessive number of elements may increase the round-off error. Therefore, it is important that the mesh is adequately fine or coarse in the appropriate regions. How fine or coarse the mesh should be in such regions is another important question. Unfortunately, definitive answers to the questions about mesh refinement are not available since it is completely dependent on the specific physical system considered. However, there are some techniques that might be helpful in answering these questions:

Adaptive Meshing: The generated mesh is required to meet acceptable energy error estimate criteria. The user provides the "acceptable" error level information. This type of meshing is available only for linear static structural analysis and steady-state thermal analysis.

Mesh Refinement Test Within ANSYS: An analysis with an initial mesh is performed first and then reanalyzed by using twice as many elements. The two solutions are compared. If the results are close to each other, the initial mesh configuration is considered to be adequate. If there are substantial differences between the two, the analysis should continue with a more-refined mesh and a subsequent comparison until convergence is established.

Submodeling: If the mesh refinement test yields nearly identical results for most regions and substantial differences in only a portion of the model, the built-in "submodeling" feature of ANSYS should be employed for localized mesh refinement.

Organization of ANSYS Software

There are two primary levels in the ANSYS program, as shown in Fig.

Begin Level: Gateway into and out of ANSYS and platform to utilize some global controls such as changing the jobname, etc.

Processor Level: This level contains the processors (preprocessor, solution, postprocessor, etc.) that are used to conduct finite element analyses.

The user is in the Begin Level upon entering the ANSYS program. One can proceed to the Processor Level by clicking the mouse on one of the processor selections in the ANSYS Main Menu.

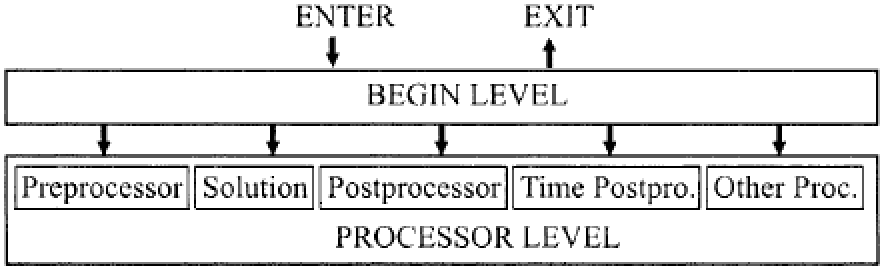


Fig.5.2 Schematic of ANSYS levels

ANSYS Analysis Approach

There are three main steps in a typical ANSYS analysis:

- Model generation:
  - Simplifications, idealizations.
  - Define materials/material properties.
  - Generate finite element model (mesh).
- Solution:
  - Specify boundary conditions.
  - Obtain the solution.
- Review results:
  - Plot/list results.
  - Check for validity.

Each of these steps corresponds to a specific processor or processors within the Processor Level in ANSYS. In particular, model generation is done in the Preprocessor and application of loads and the solution is performed in the Solution Processor, Finally, the results are viewed in the General Postprocessor and Time History Postprocessor for steady-state (static) and transient (time-dependent) problems, respectively. There are several other processors within the ANSYS program. These mostly concern optimization- and probabilistic-type problems. The most commonly used processors are described in the following subsections.

ANSYS Preprocessor :

Model generation is conducted in this processor, which involves material definition, creation of a solid model, and, finally, meshing. Important tasks within this processor are:

- Specify element type.

- Define real constants (if required by the element type).

- Define material properties,

- Create the model geometry.

- Generate the mesh.

Although the boundary conditions can also be specified in this processor, it is usually done in the Solution Processor,

ANSYS Solution Processor:

This processor is used for obtaining the solution for the finite element model that is generated within the Preprocessor, Important tasks within this processor are:

- Define analysis type and analysis options,

- Specify boundary conditions.

- Obtain solution.

ANSYS General Postprocessor :

In this processor, the results at a specific time (if the analysis type is transient) over the entire or a portion of the model are reviewed. This includes the plotting of contours, vector displays, deformed shapes, and listings of the results in tabular format.

ANSYS Time History Postprocessor :

This processor is used to review results at specific points in time (if the analysis type is transient). Similar to the General Postprocessor, it provides graphical variations and tabular listings of results data as functions of time.

ANSYS File Structure

Several files are created during a typical ANSYS analysis. Some of these files are in ASCII format while the others are binary. Brief descriptions of common file types are given below.

Database File

During a typical ANSYS analysis, input and output data reside in memory until they are saved in a Database File, which is saved in the Working Directory. The syntax for the name of the Database File is jobname, db. This binary file includes the element type, material properties, geometry (solid model), mesh (nodal coordinates and element connectivity), and the results if a solution is obtained. Once the Database File is saved, the user can resume from this file at any time. There are three distinct ways to save and resume the Database File:

- Use the Utility Menu.

- Click on SAVE_DB or RESUM_DB button on the ANSYS Toolbar.

- Issue the command SAVE or RESUME in the Input Field.

Log File

The Log File is an ASCII file, which is created (or resumed) immediately upon entering ANSYS. Every action taken by the user is stored sequentially in this file in command format (ANSYS Parametric Design Language (APDL)). The syntax for the name of the Log File, which is also saved in the Working Directory, isjobname.log. If jobname.log already exists in the Working Directory, ANSYS appends the newly executed actions instead of overwriting the file. The Log File can be utilized to:

- Understand how an analysis was performed by another user.

- Learn the command equivalents of the actions taken within ANSYS.

Error File

Similar to the Log File, the Error File is an ASCII file, which is created (or resumed) immediately upon entering ANSYS. This file captures all warning and error messages issued by ANSYS during a session. It is saved in the Working Directory with the following syntax for the name: jobname.err. If jobname.err already exists in the Working Directory, ANSYS appends the newly issued warning and error messages instead of overwriting the file. This file is particularly important when ANSYS issues several warning and error messages too quickly during an interactive session. The user can then consult the Error File to discover the exact cause(s) of each of the warnings or errors.

Results Files

The results of an ANSYS analysis are stored in a separate Results File. This file is a binary file and, depending upon the Analysis Type, the file's extension takes a different form. The following syntax applies to the Results File name for the selected Analysis Type:

Structural analysis: jobname.rst

Thermal analysis: jobname.rth

Fluids analysis: jobname,rfl

Description of ANSYS Menus and Windows:

When using the ANSYS program in Interactive Mode, the Graphical User Interface (GUT) is activated. The GUI has six distinct components:

Utility Menu: Contains functions that are available throughout the ANSYS session, such as file controls, selecting, graphic controls, and parameters. The ANSYS Help System is also accessible through this menu.

Main Menu: Contains the primary ANSYS functions organized by processors (Preprocessor, Solution, General Postprocessor, etc.).

Toolbar: Contains push-buttons for executing commonly used ANSYS commands and functions. Customized buttons can be created.

Input Field: Displays a text field for typing commands. All previously typed commands are stored in a pull-down menu for easy reference and access.

Graphics Window: Displays the graphical representation of the models/ meshes created within ANSYS. Also, the related results are reviewed in this window.

Output Window: Receives text output from the program. This window is usually positioned behind other windows and can be raised to the front when necessary.

Figure shows a typical ANSYS GUI with each of the preceding components identified:-

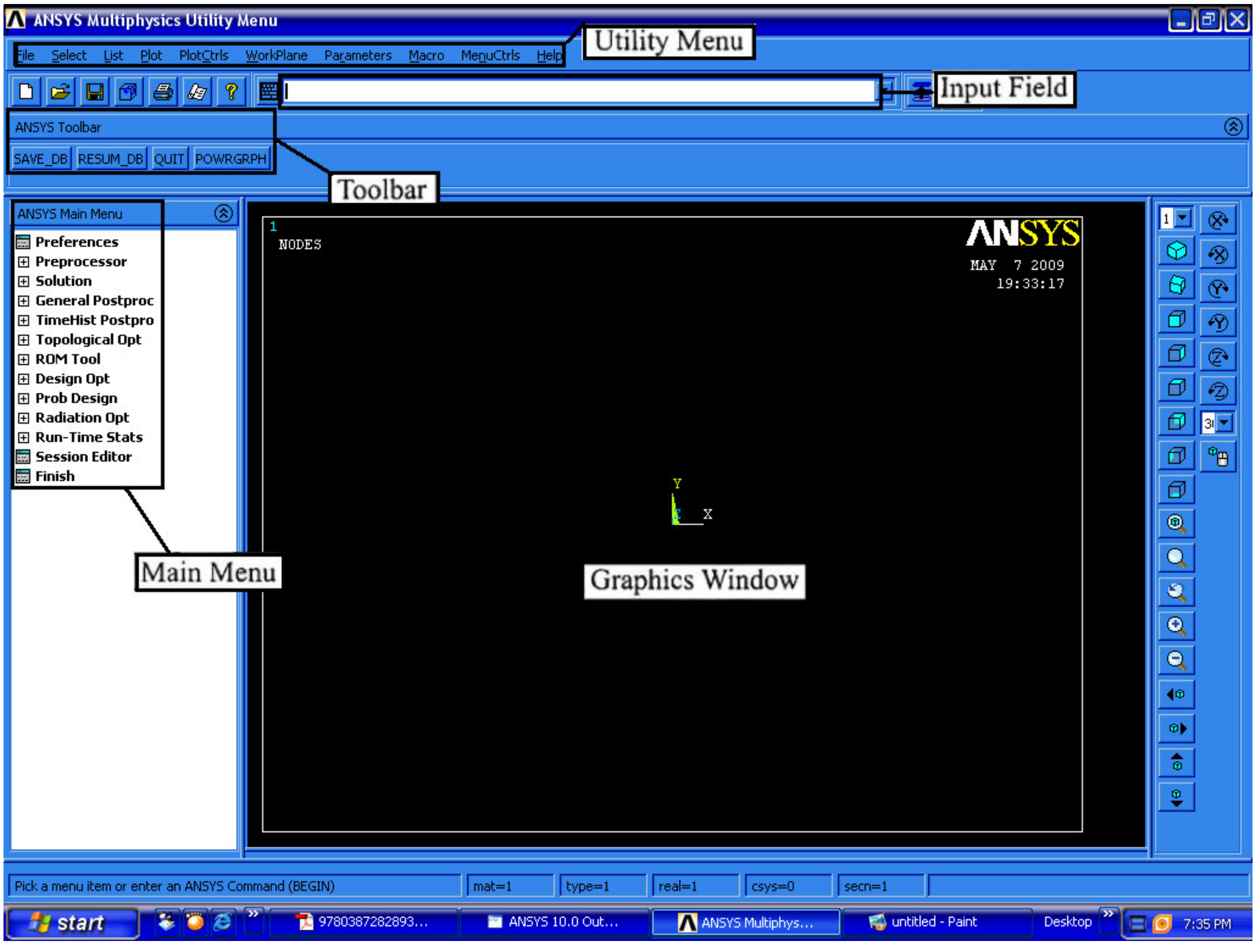


Fig 5.3 ANSYS GUI

Utility Menu

The Utility Menu contains utility functions that are independent of ANSYS Levels (i.e., begin and processor levels), with some exceptions. The Utility Menu contains ten items, each of which brings up a pull-down menu of subitems. Clicking the left mouse button on these subitems will result in one of the following:

- Bring up a submenu, indicated by the icon • .
- Immediately execute a function.
- Bring up a dialog box, indicated by the icon... .
- Bring up a picking menu, indicated by the icon + .

Brief descriptions of each of the menu items under the Utility Menu are given below.

File item under Utility Menu: Contains file- and database-related functions, such as clearing the database, reading an input file, saving the database to a file, or resuming a database from a file. This menu item can be used to exit the program.

Select item under Utility Menu: Includes functions that allow the user to select a subset of data and to create Components.

List item under Utility Menu: This menu item allows the user to list any data stored in the ANSYS database. Also, status information about different areas of the program and contents of files in the system are available.

Plot item under Utility Menu: This menu item allows the user to plot ANSYS entities such as keypoints, lines, areas, volumes, nodes, and elements. If a solution is obtained, results can also be plotted through this menu item.

PlotCtrls item under Utility Menu: Contains functions that control the view, style, and other characteristics of graphic displays.

WorkPlane item under Utility Menu: Use of WorkPlane offers great convenience for Solid Model generation. This menu item enables the user to toggle the Working Plane on and off, and to move, rotate, and maneuver it. Coordinate system operations are also performed under this menu item.

Parameters item under Utility Menu: Contains functions to define, edit, and delete scalar and array parameters.

Macro item under Utility Menu: This menu item allows the user to execute Macros and data blocks. Under this menu item, the user can also manipulate the push-buttons on the Toolbar.

MenuCtrls item under Utility Menu: Allows the user to format the menus, as well as manipulate the Toolbar.

Help item under Utility Menu: Brings up the ANSYS Help System.

Main Menu

The Main Menu contains main ANSYS functions and processors, such as the preprocessor, solution, and postprocessor. It has a tree structure, where menus and submenus can be expanded and collapsed. Similar to the Utility Menu, clicking the left mouse button on the Main Menu items results in one of the following:

- Expand or collapse the submenus attached to the menu item, indicated by icons ⊞ and ⊟ , respectively.

- Bring up a dialog box, indicated by the icon 🗔 .

- Bring up a picking menu, indicated by the icon ↗ .

Toolbar

The Toolbar contains a set of push-buttons that execute frequently used ANSYS functions. When the user starts ANSYS, predefined push-buttons such as QUIT, SAVE_DB, and RESUM_DB appear in the toolbar. The user can create customized push-buttons and delete or edit the existing ones.

Input Field

This field allows the user to type in commands directly as opposed to the use of menu items. The Input Field consists of two main regions:

- Command entry box.

- History buffer.

Graphics Window

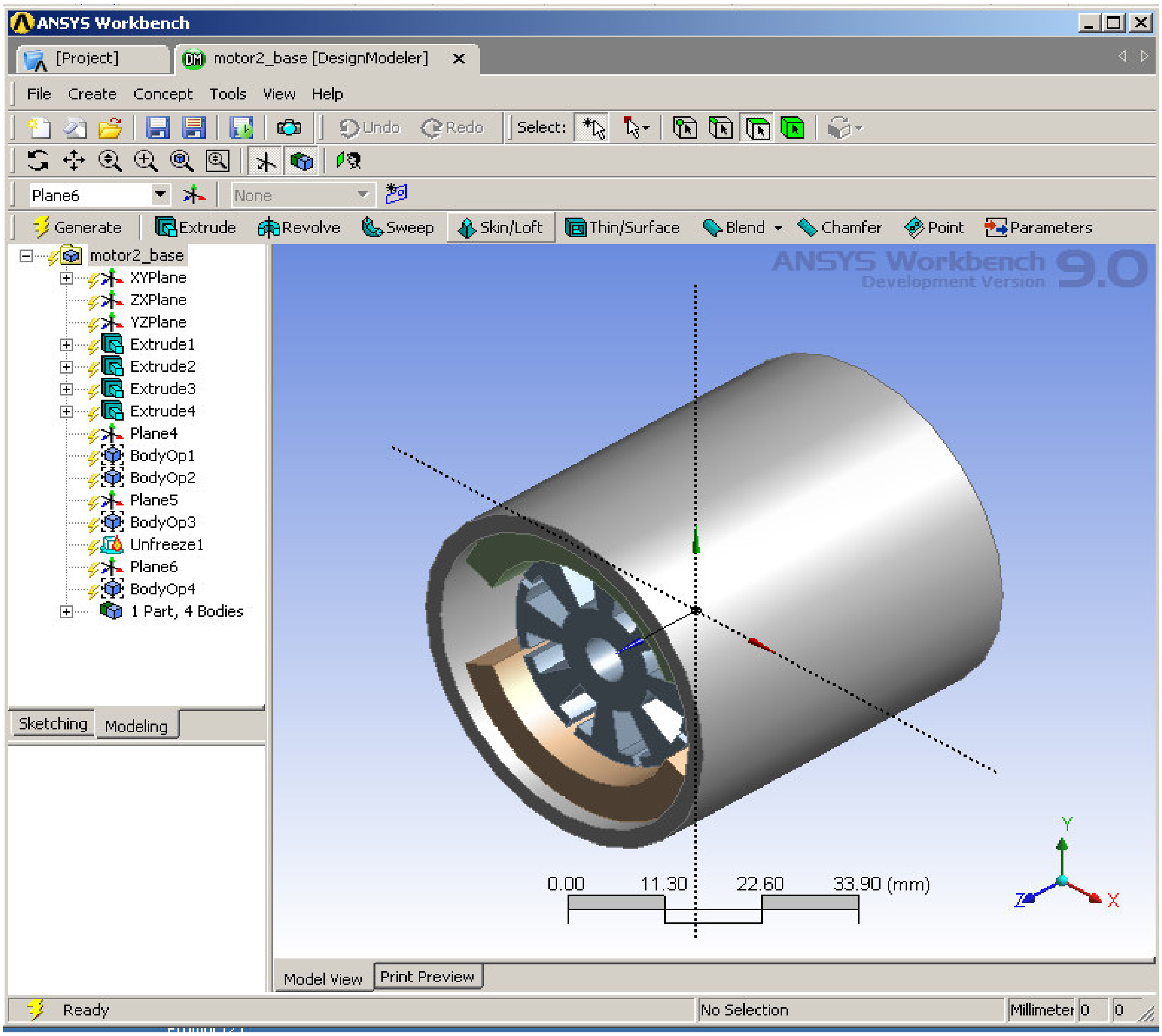
All ANSYS graphics are displayed in the Graphics Window. Also, the user performs all of the graphical "picking" in this window.

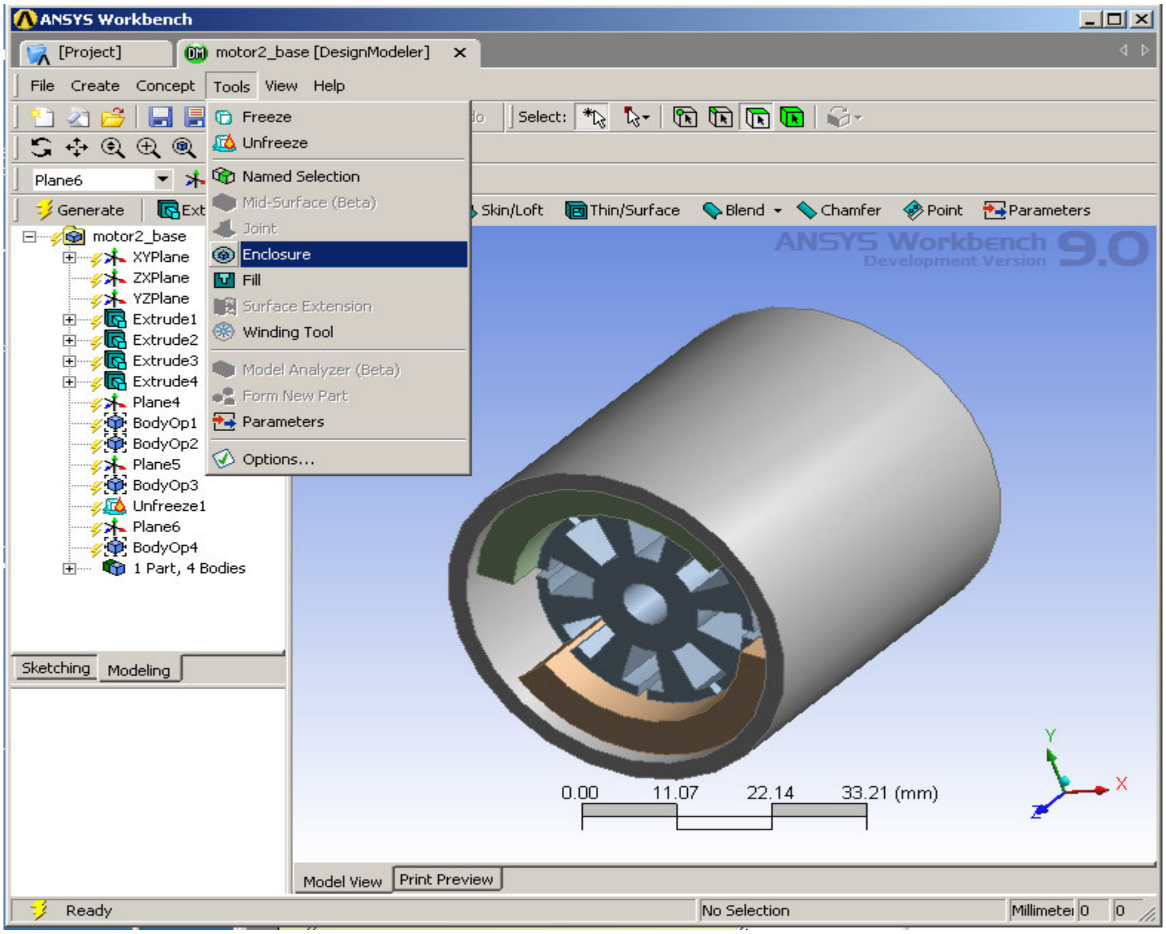
Output Window

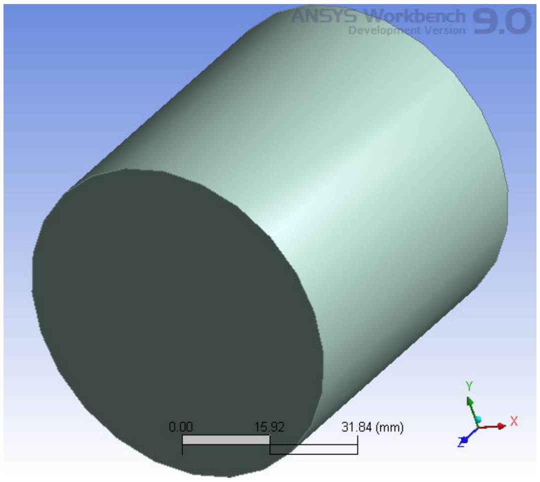
All of the text output generated as a result of command responses, warnings, and errors appear in the Output Window. It is positioned behind the main ANSYS window, but can be raised to the front when necessary.

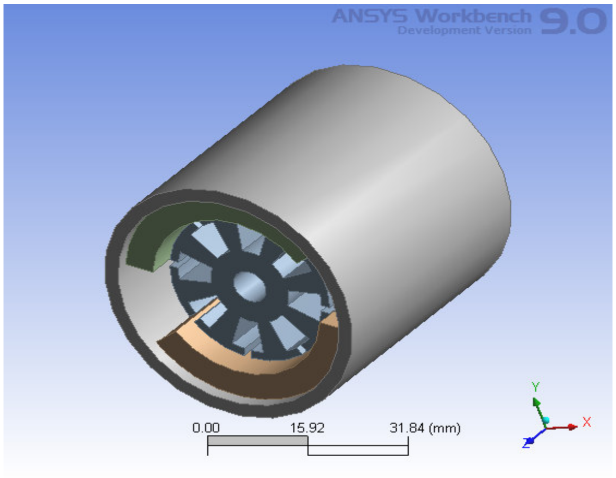
5.2MOTOR ANALYSIS USING FINITE ELEMENTS METHODS IN ANSYS WORKBENCH:

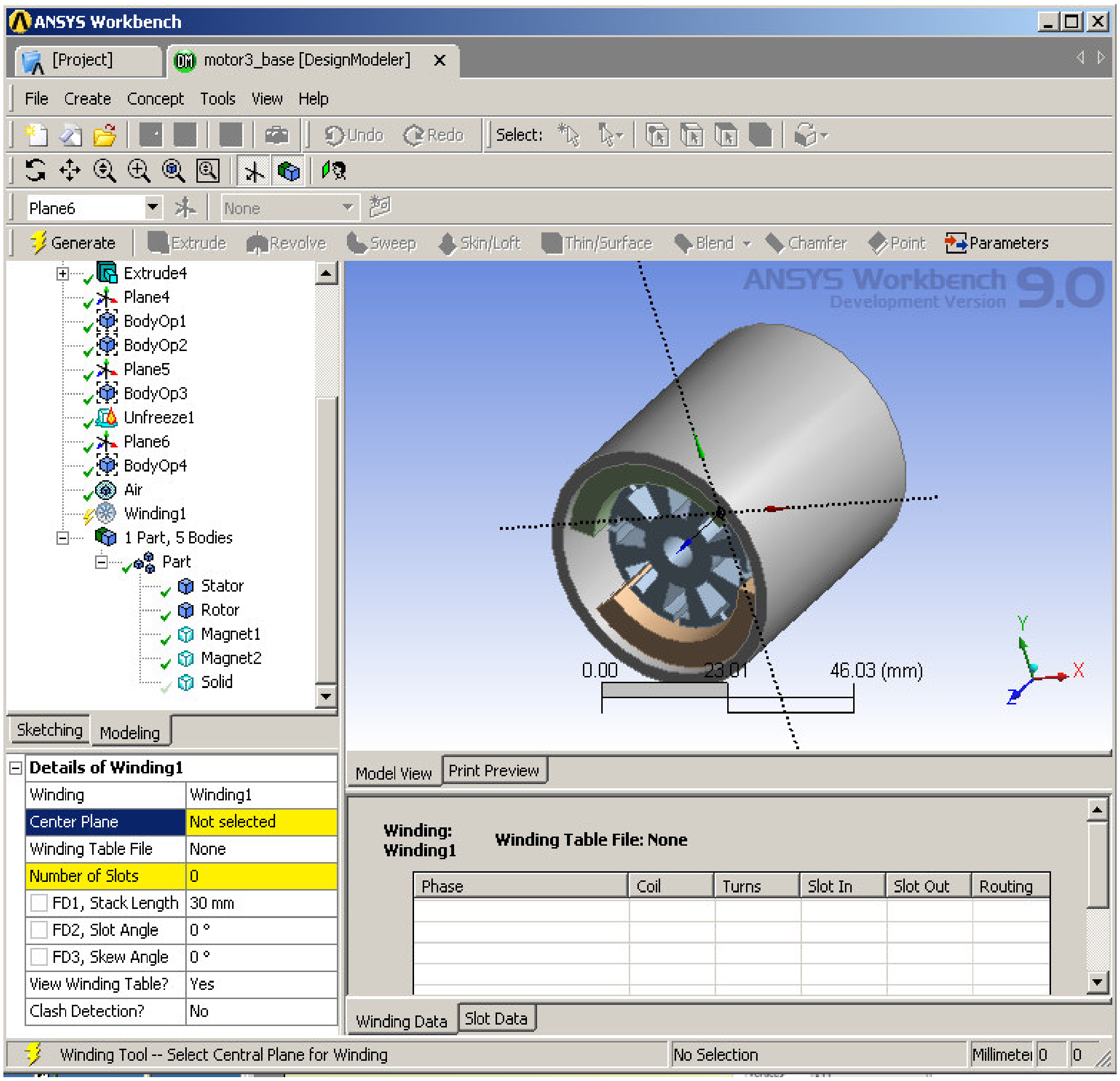
Figures below show step by step field analysis of electric motor.




ANSYS Workbench
[Project]
motor2_base [DesignModeler]
File
Create
Concept
Tools
View
Help
Select:
Plane6
Generate
Ext
Skin/Loft
Thin/Surface
Blend
Chamfer
Point
Parameters
Freeze
Unfreeze
Named Selection
Mid-Surface (Beta)
Joint
Enclosure
Fill
Surface Extension
Winding Tool
Model Analyzer (Beta)
Form New Part
Parameters
Options…
motor2_base
XYPlane
ZXPlane
YZPlane
Extrude1
Extrude2
Extrude3
Extrude4
Plane4
BodyOp1
BodyOp2
Plane5
BodyOp3
Unfreeze1
Plane6
BodyOp4
1 Part, 4 Bodies
Sketching
Modeling
ANSYS Workbench 9.0
Development Version
0.00
11.07
22.14
33.21 (mm)
Y
X
Z
Model View
Print Preview
Ready
No Selection
Millimeter
0
0


ANSYS Workbench 9.0
Development Version
0.00
15.92
31.84 (mm)
Y
X
Z


ANSYS Workbench 9.0
Development Version
0.00
15.92
31.84 (mm)
Y
X
Z



Details of Winding1

Winding	Winding1
Center Plane	Not selected
Winding Table File	None
Number of Slots	0
□ FD1, Stack Length	30 mm
□ FD2, Slot Angle	0 °
□ FD3, Skew Angle	0 °
View Winding Table?	Yes
Clash Detection?	No

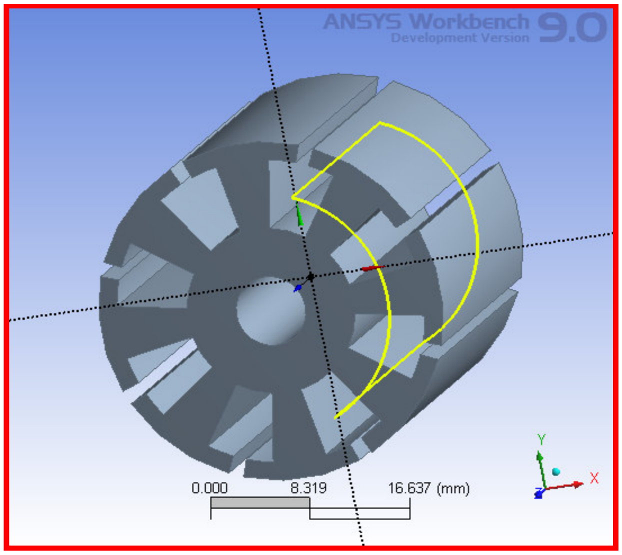
Winding: Winding1 **Winding Table File: None**

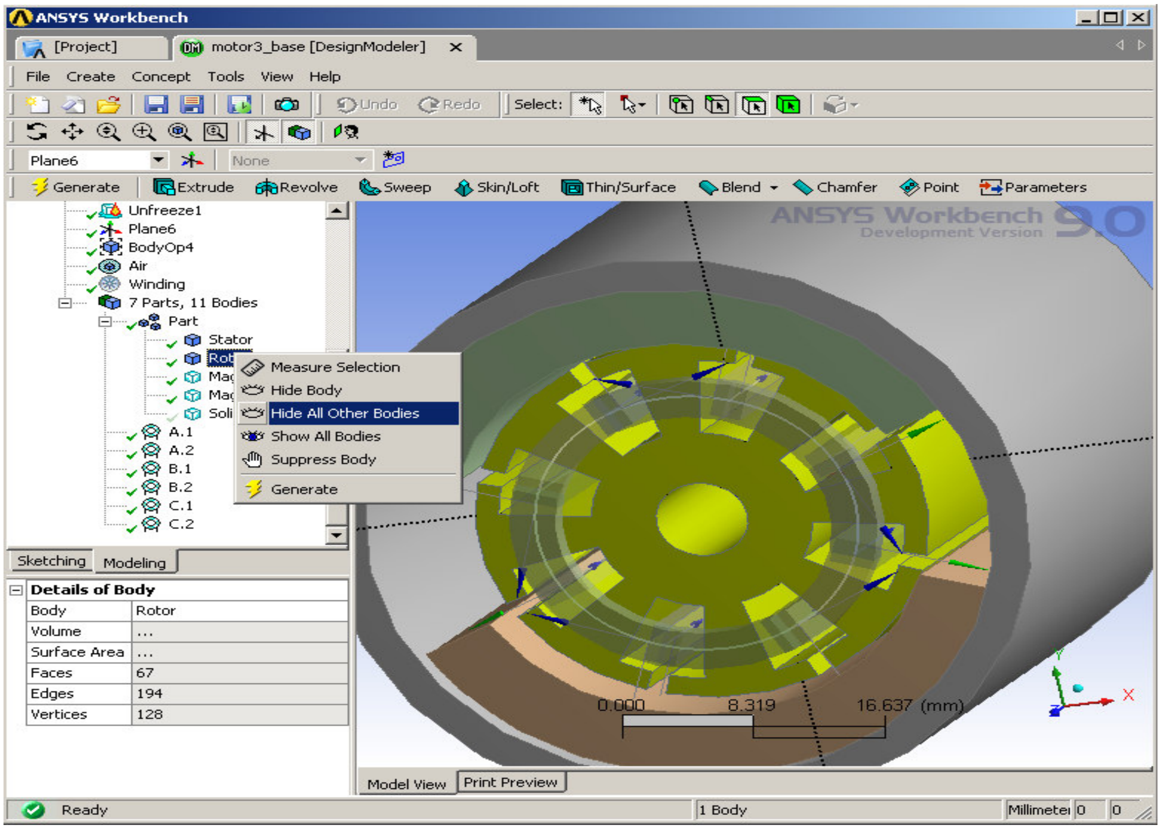
Phase	Coil	Turns	Slot In	Slot Out	Routing

Winding Tool -- Select Central Plane for Winding

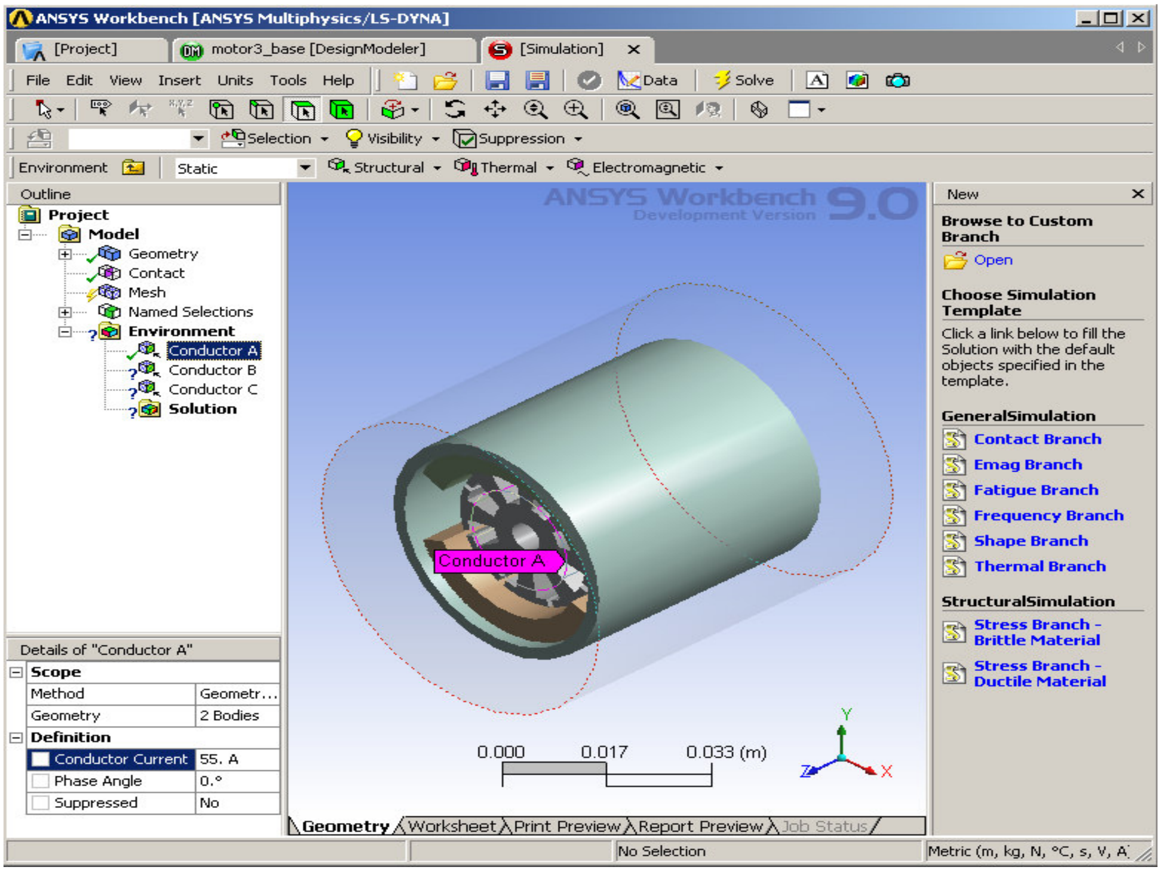
No Selection

Millimeter 0 0





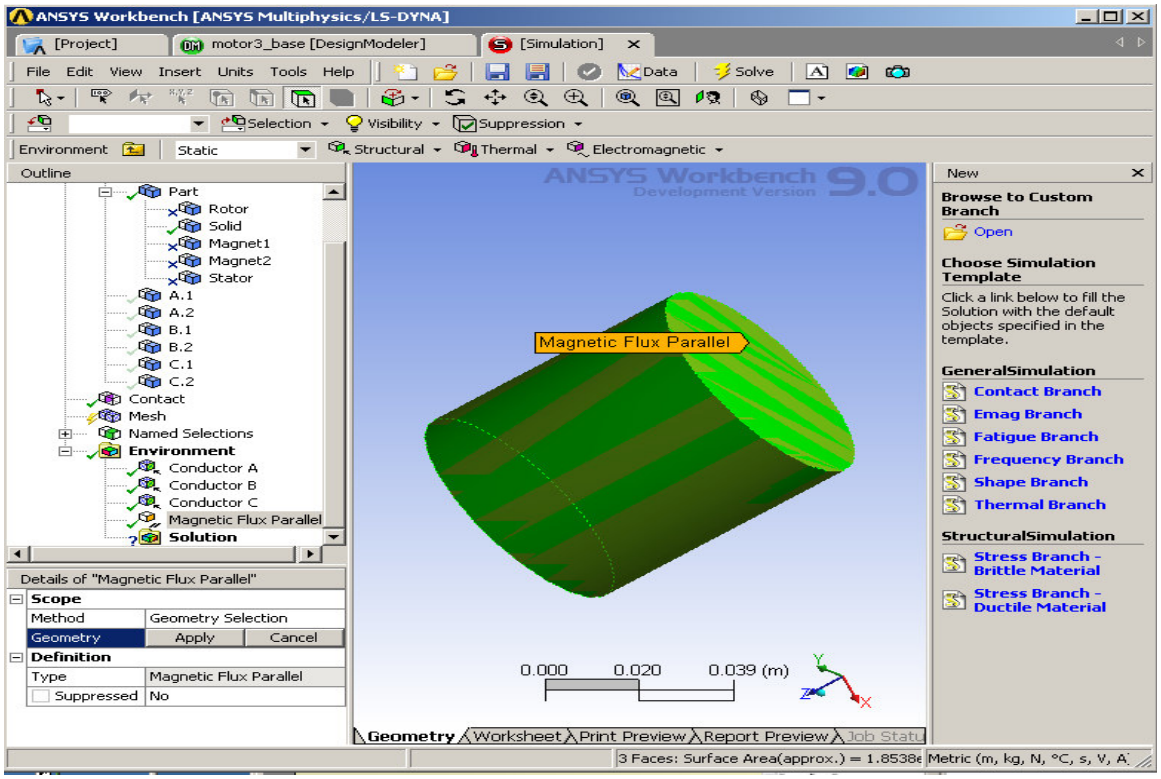
Details of Body	
Body	Rotor
Volume	...
Surface Area	...
Faces	67
Edges	194
Vertices	128



Click a link below to fill the Solution with the default objects specified in the template.

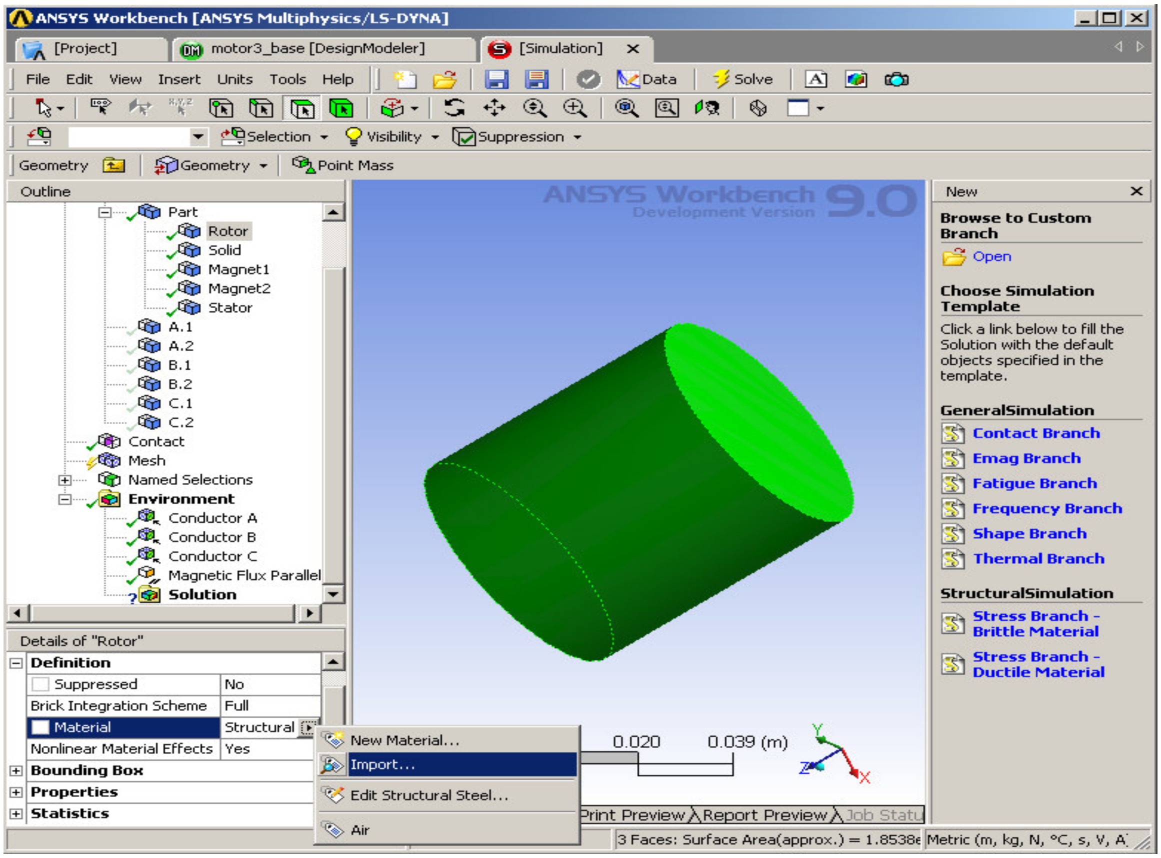
Details of "Conductor A"

Scope	
Method	Geometr...
Geometry	2 Bodies
Definition	
Conductor Current	55. A
Phase Angle	0.°
Suppressed	No



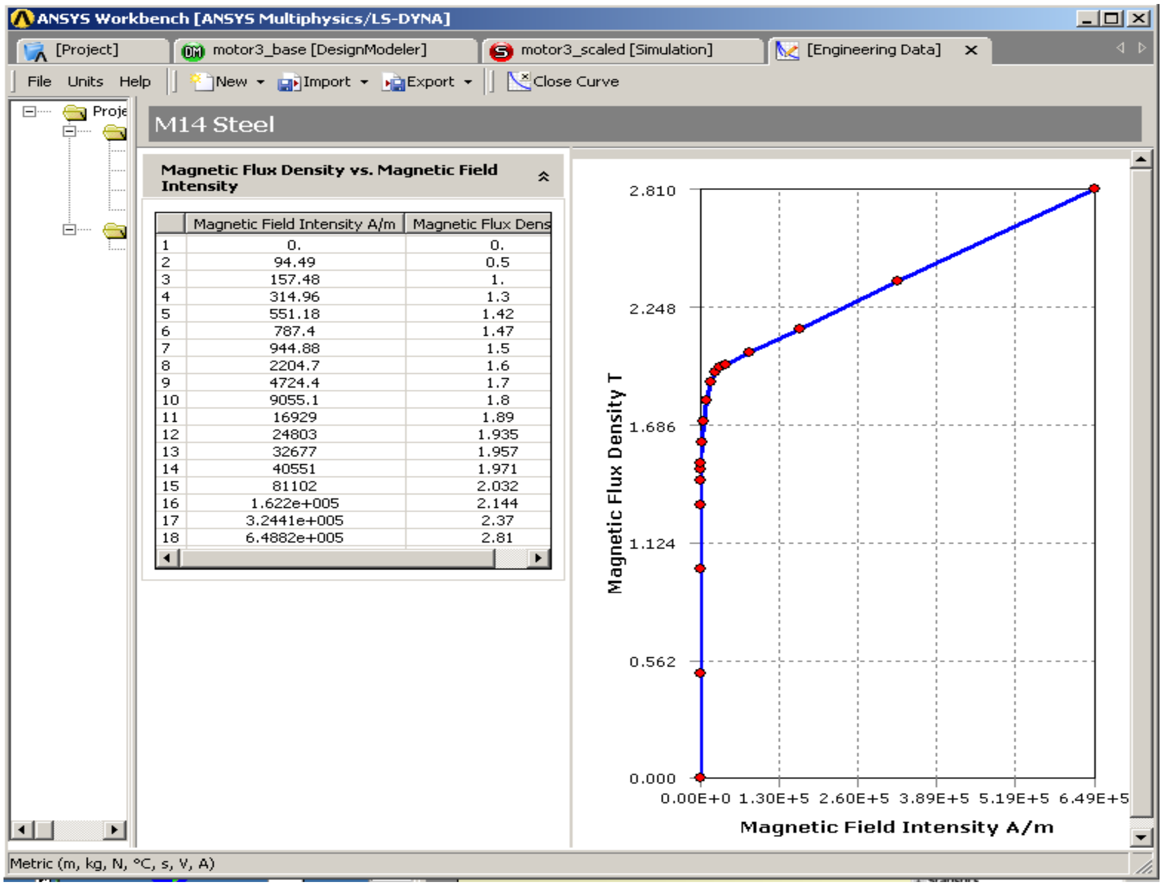
Details of "Magnetic Flux Parallel"

Scope	
Method	Geometry Selection
Geometry	Apply / Cancel
Definition	
Type	Magnetic Flux Parallel
Suppressed	No



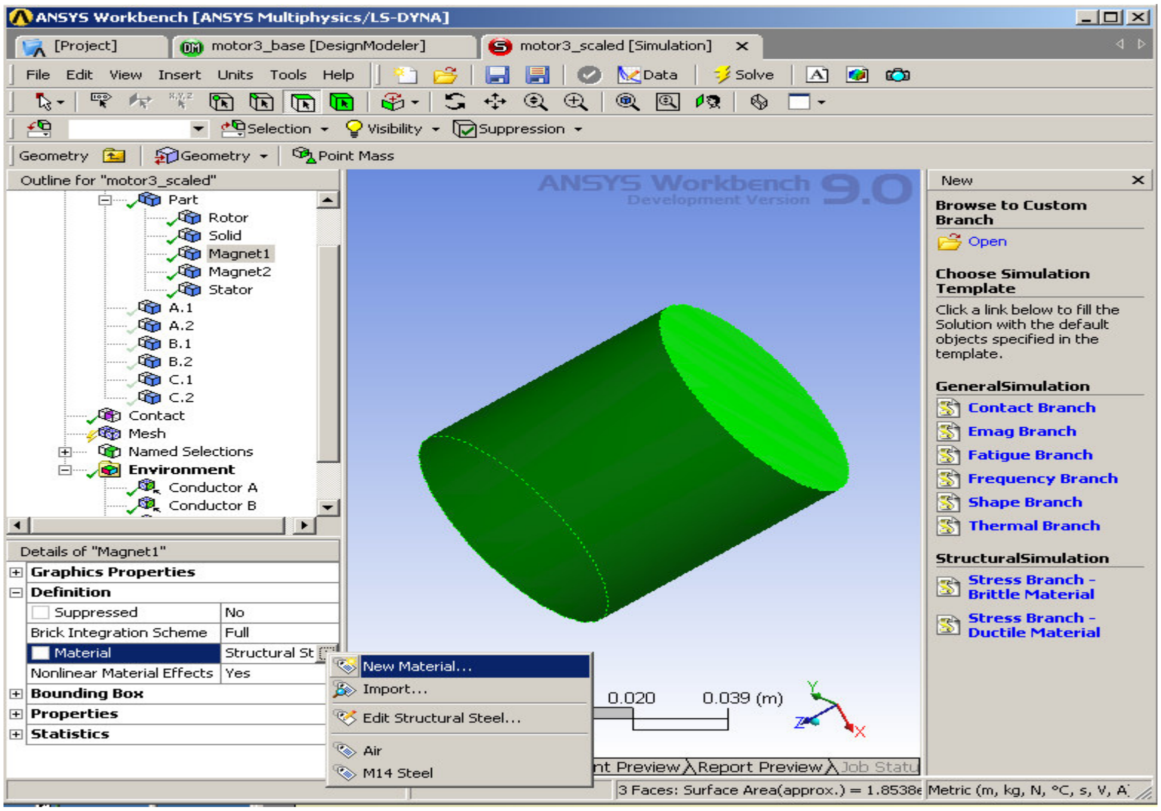
Details of "Rotor"

Definition	
Suppressed	No
Brick Integration Scheme	Full
Material	Structural
Nonlinear Material Effects	Yes
Bounding Box	
Properties	
Statistics	


ANSYS Workbench [ANSYS Multiphysics/LS-DYNA]
[Project]
motor3_base [DesignModeler]
motor3_scaled [Simulation]
[Engineering Data]
File
Units
Help
New
Import
Export
Close Curve
Proje
M14 Steel
Magnetic Flux Density vs. Magnetic Field Intensity
2.810
2.248
1.686
1.124
0.562
0.000
0.00E+0
1.30E+5
2.60E+5
3.89E+5
5.19E+5
6.49E+5
Magnetic Flux Density T
Magnetic Field Intensity A/m

	Magnetic Field Intensity A/m	Magnetic Flux Dens
1	0.	0.
2	94.49	0.5
3	157.48	1.
4	314.96	1.3
5	551.18	1.42
6	787.4	1.47
7	944.88	1.5
8	2204.7	1.6
9	4724.4	1.7
10	9055.1	1.8
11	16929	1.89
12	24803	1.935
13	32677	1.957
14	40551	1.971
15	81102	2.032
16	1.622e+005	2.144
17	3.2441e+005	2.37
18	6.4882e+005	2.81

Metric (m, kg, N, °C, s, V, A)



Details of "Magnet1"

Graphics Properties

Definition

Suppressed	No
Brick Integration Scheme	Full
Material	Structural St
Nonlinear Material Effects	Yes

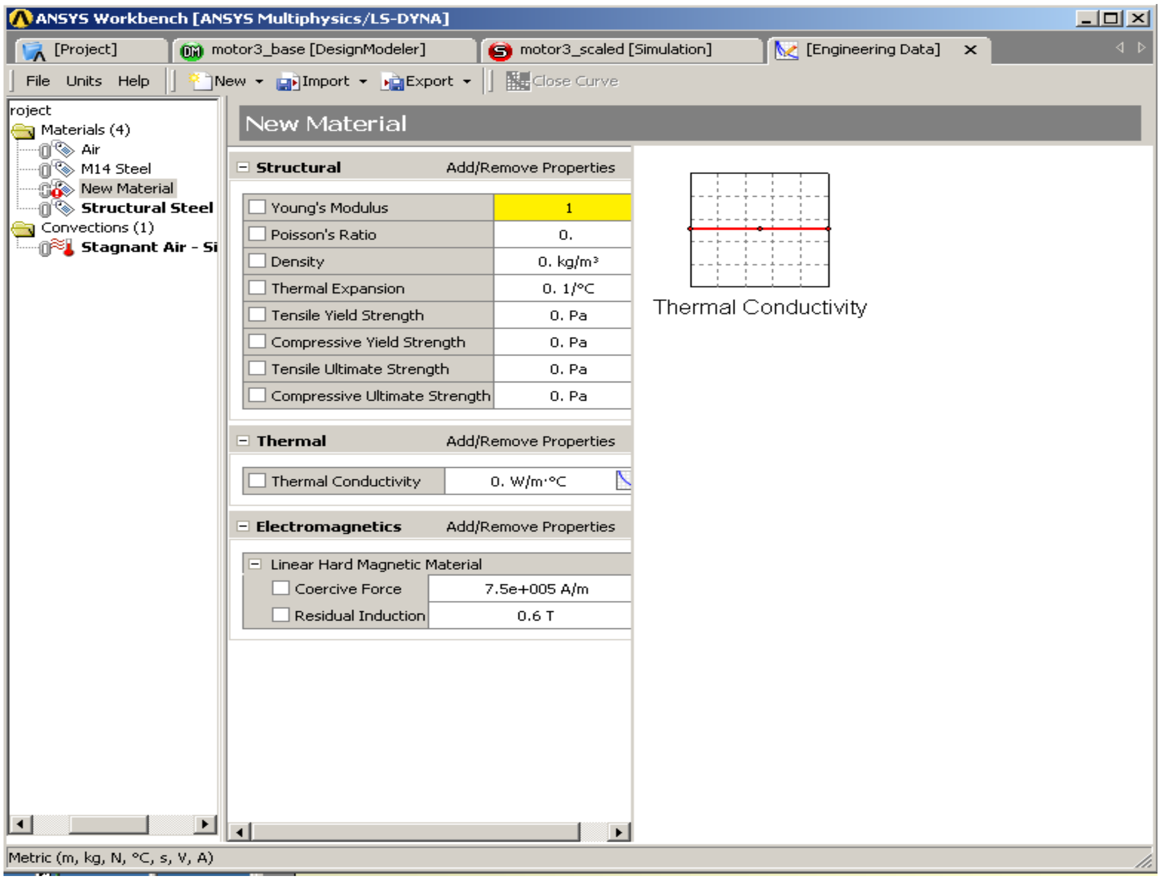
Bounding Box

Properties

Statistics

3 Faces: Surface Area(approx.) = 1.8538e

Metric (m, kg, N, °C, s, V, A)



New Material

Structural — Add/Remove Properties

□ Young's Modulus	1
□ Poisson's Ratio	0.
□ Density	0. kg/m³
□ Thermal Expansion	0. 1/°C
□ Tensile Yield Strength	0. Pa
□ Compressive Yield Strength	0. Pa
□ Tensile Ultimate Strength	0. Pa
□ Compressive Ultimate Strength	0. Pa

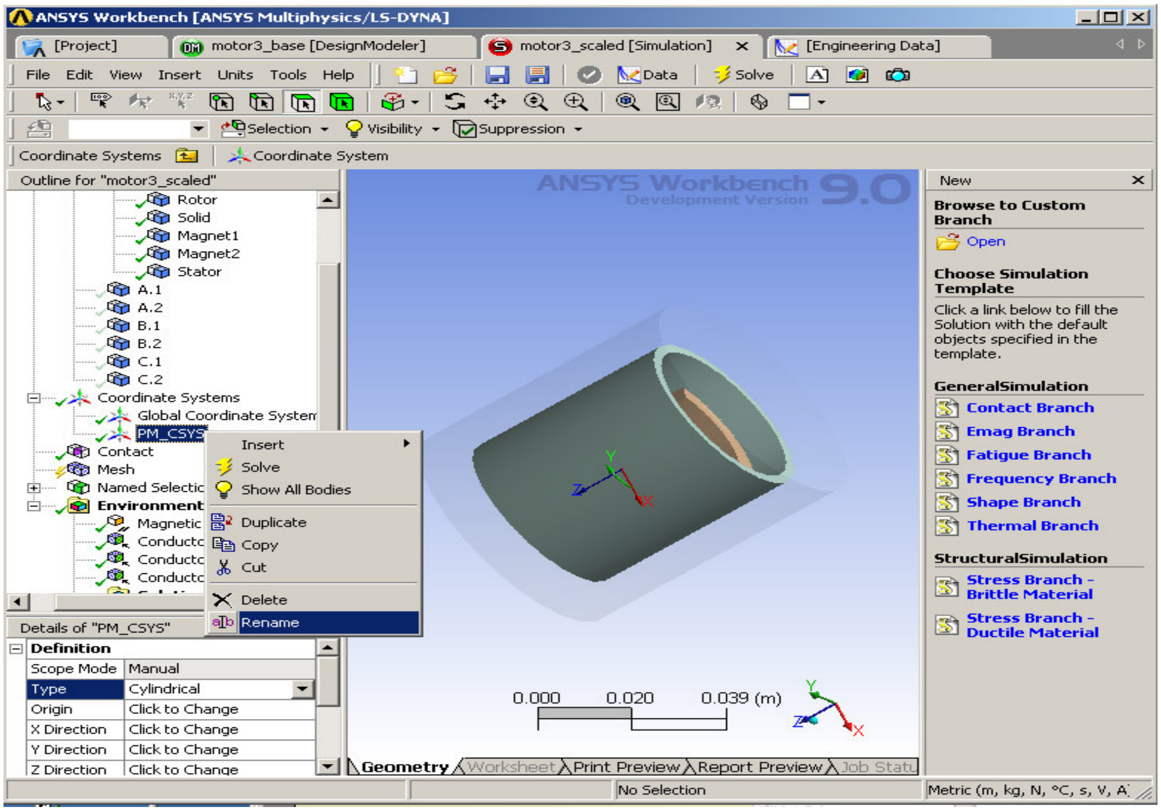
Thermal — Add/Remove Properties

□ Thermal Conductivity	0. W/m·°C

Electromagnetics — Add/Remove Properties

Linear Hard Magnetic Material

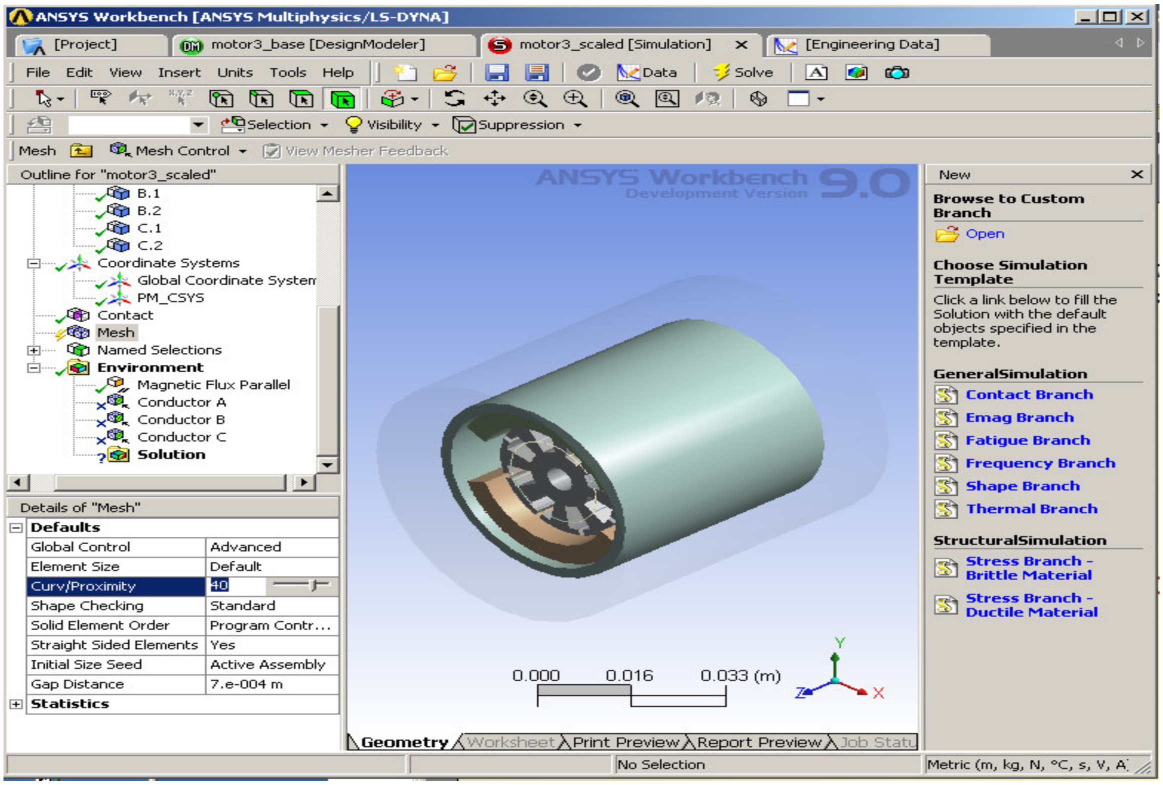
□ Coercive Force	7.5e+005 A/m
□ Residual Induction	0.6 T



Details of "PM_CSYS"

Definition

Scope Mode	Manual
Type	Cylindrical
Origin	Click to Change
X Direction	Click to Change
Y Direction	Click to Change
Z Direction	Click to Change



Outline for "motor3_scaled"

- B.1
- B.2
- C.1
- C.2
- Coordinate Systems
  - Global Coordinate System
  - PM_CSYS
- Contact
- Mesh
- Named Selections
- **Environment**
  - Magnetic Flux Parallel
  - Conductor A
  - Conductor B
  - Conductor C
  - **Solution**

Details of "Mesh"

Defaults	
Global Control	Advanced
Element Size	Default
Curv/Proximity	40
Shape Checking	Standard
Solid Element Order	Program Contr…
Straight Sided Elements	Yes
Initial Size Seed	Active Assembly
Gap Distance	7.e-004 m
Statistics	

New

Browse to Custom Branch

Open

Choose Simulation Template

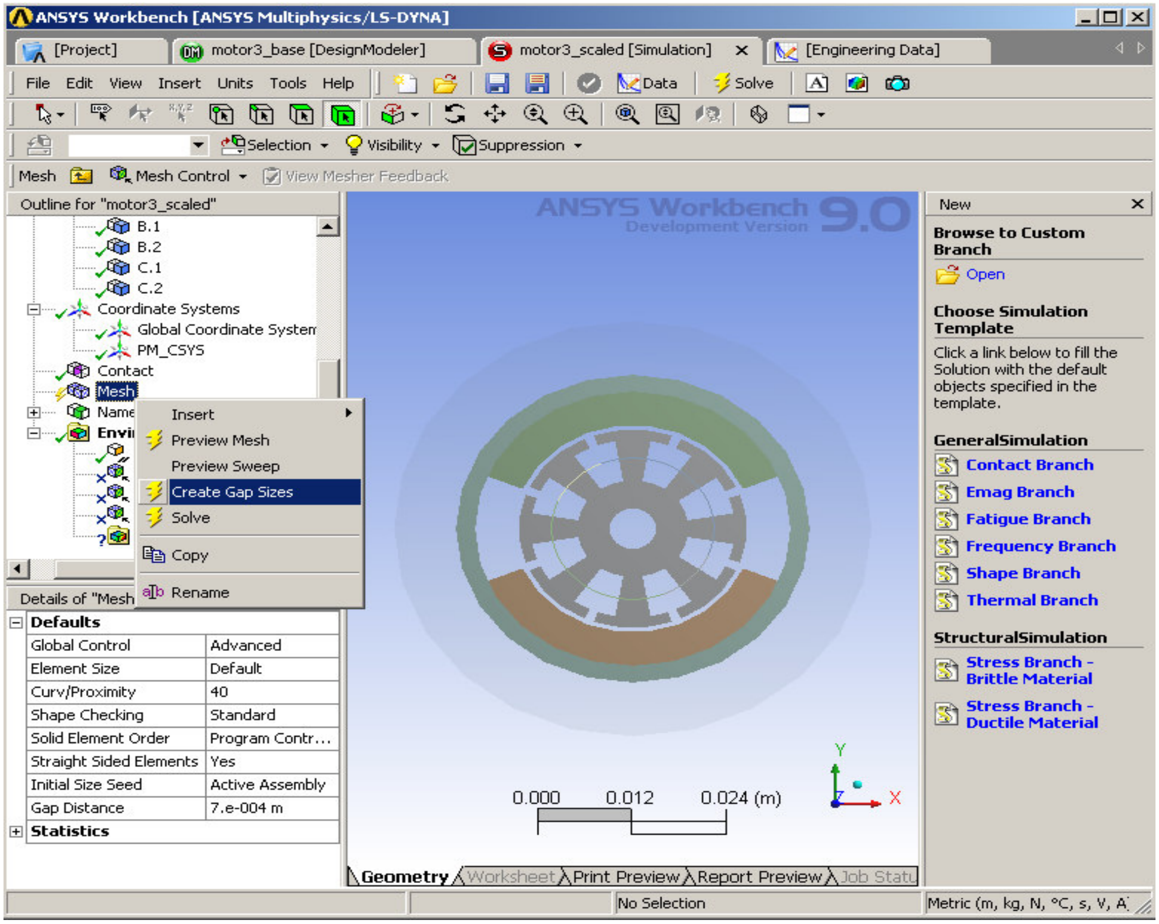
Click a link below to fill the Solution with the default objects specified in the template.

GeneralSimulation

- Contact Branch
- Emag Branch
- Fatigue Branch
- Frequency Branch
- Shape Branch
- Thermal Branch

StructuralSimulation

- Stress Branch - Brittle Material
- Stress Branch - Ductile Material



Outline for "motor3_scaled"

- B.1
- B.2
- C.1
- C.2
- Coordinate Systems
  - Global Coordinate System
  - PM_CSYS
- Contact
- Mesh
- Name
- **Envi**

Details of "Mesh

Defaults	
Global Control	Advanced
Element Size	Default
Curv/Proximity	40
Shape Checking	Standard
Solid Element Order	Program Contr…
Straight Sided Elements	Yes
Initial Size Seed	Active Assembly
Gap Distance	7.e-004 m
Statistics	

New

Browse to Custom Branch

Open

Choose Simulation Template

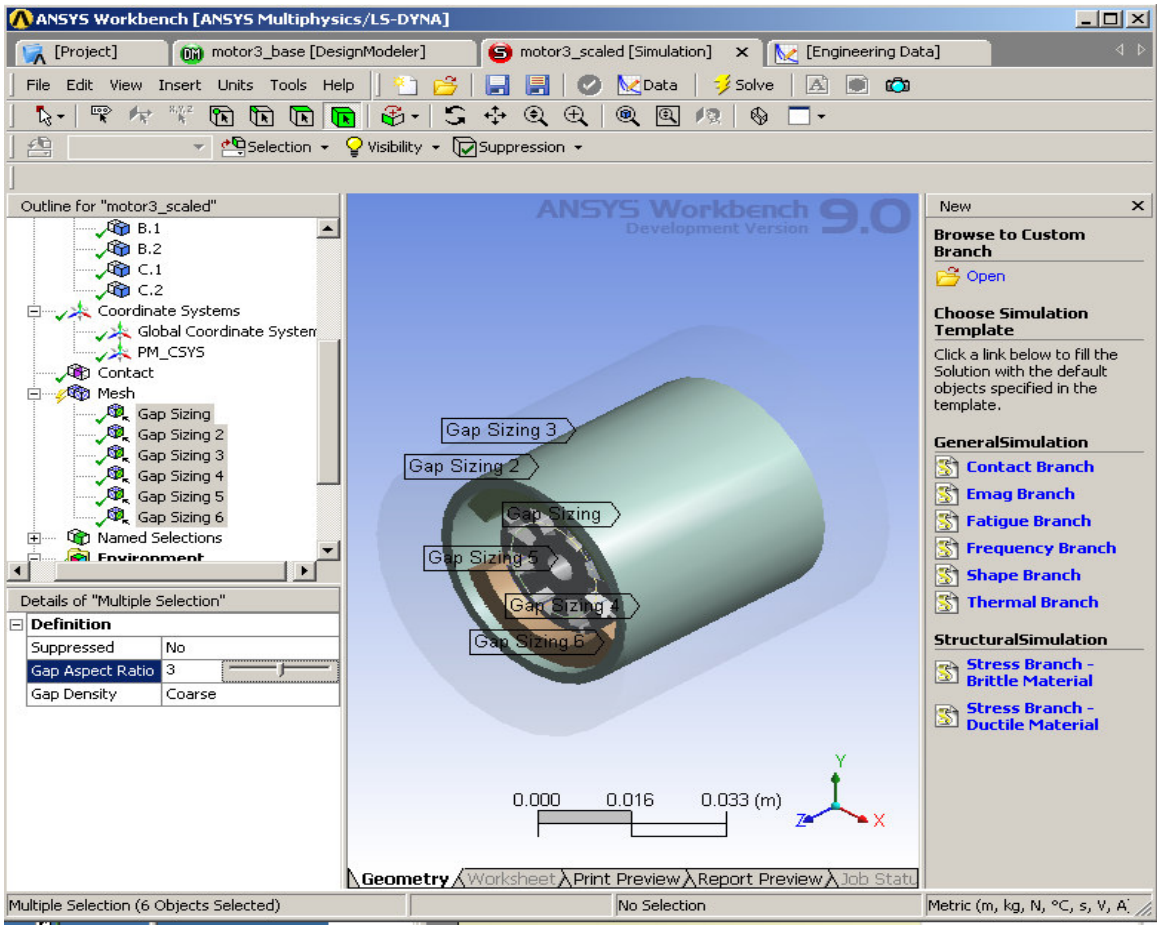
Click a link below to fill the Solution with the default objects specified in the template.

GeneralSimulation

- Contact Branch
- Emag Branch
- Fatigue Branch
- Frequency Branch
- Shape Branch
- Thermal Branch

StructuralSimulation

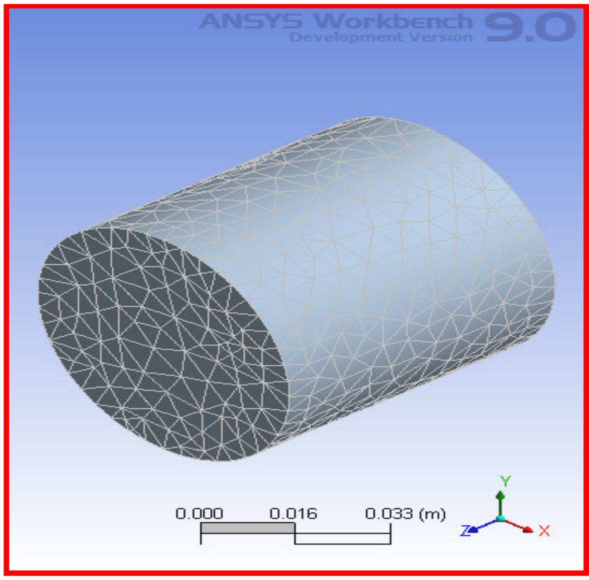
- Stress Branch - Brittle Material
- Stress Branch - Ductile Material

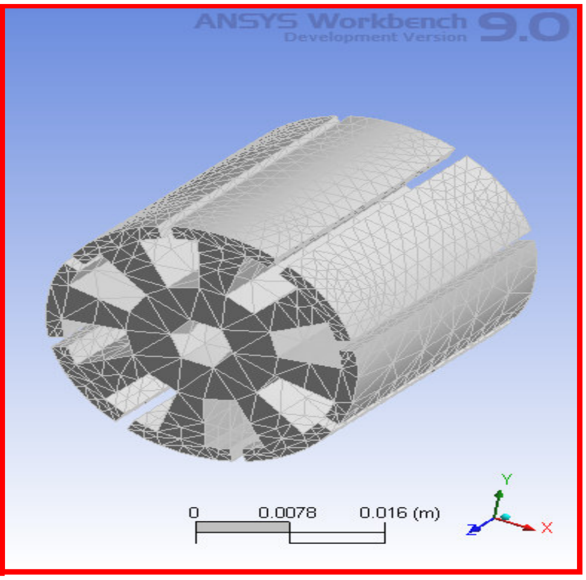


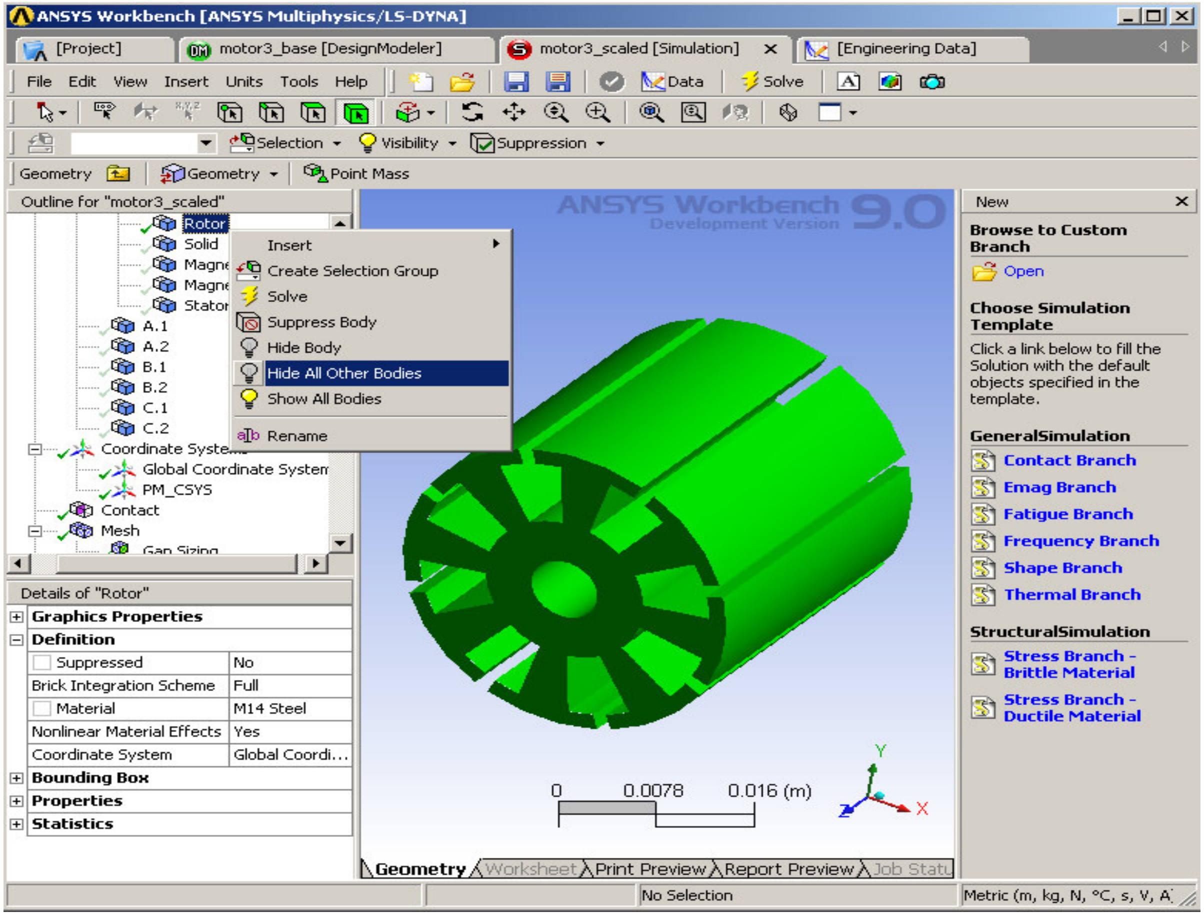
Details of "Multiple Selection"

Definition	
Suppressed	No
Gap Aspect Ratio	3
Gap Density	Coarse

Click a link below to fill the Solution with the default objects specified in the template.







Details of "Rotor"

⊞ Graphics Properties

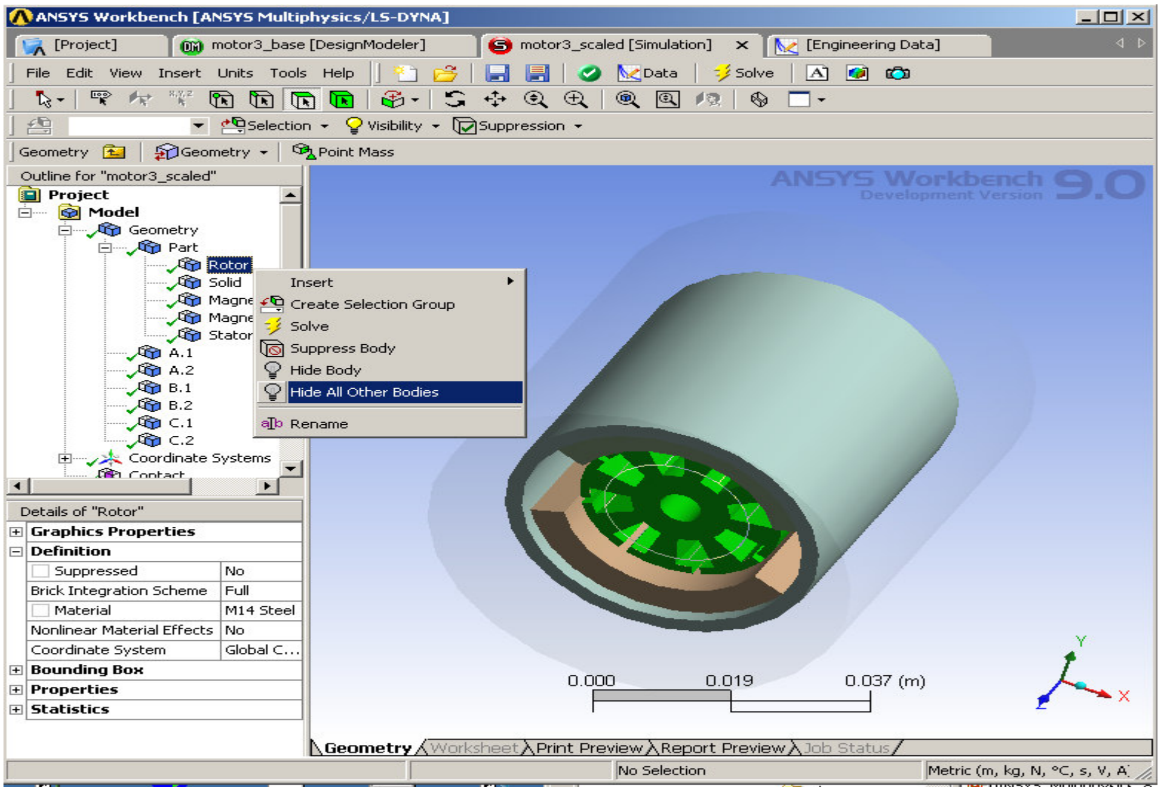
⊟ Definition

□ Suppressed	No
Brick Integration Scheme	Full
□ Material	M14 Steel
Nonlinear Material Effects	Yes
Coordinate System	Global Coordi...

⊞ Bounding Box

⊞ Properties

⊞ Statistics



Details of "Rotor"

⊞ Graphics Properties

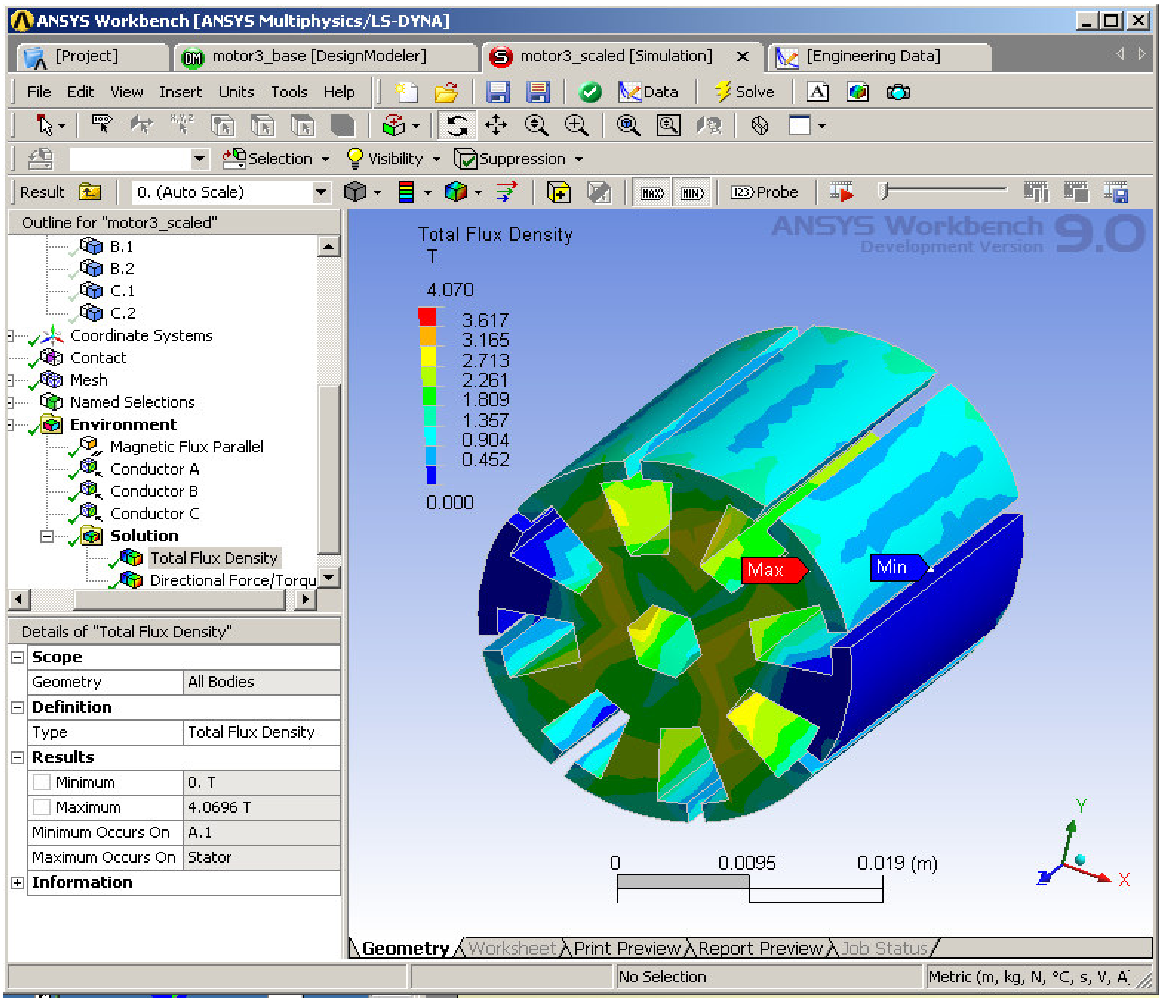
⊟ Definition

□ Suppressed	No
Brick Integration Scheme	Full
□ Material	M14 Steel
Nonlinear Material Effects	No
Coordinate System	Global C...

⊞ Bounding Box

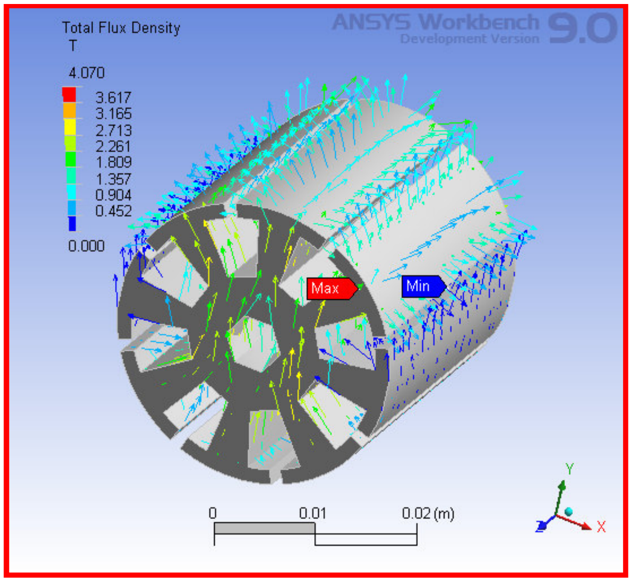
⊞ Properties

⊞ Statistics

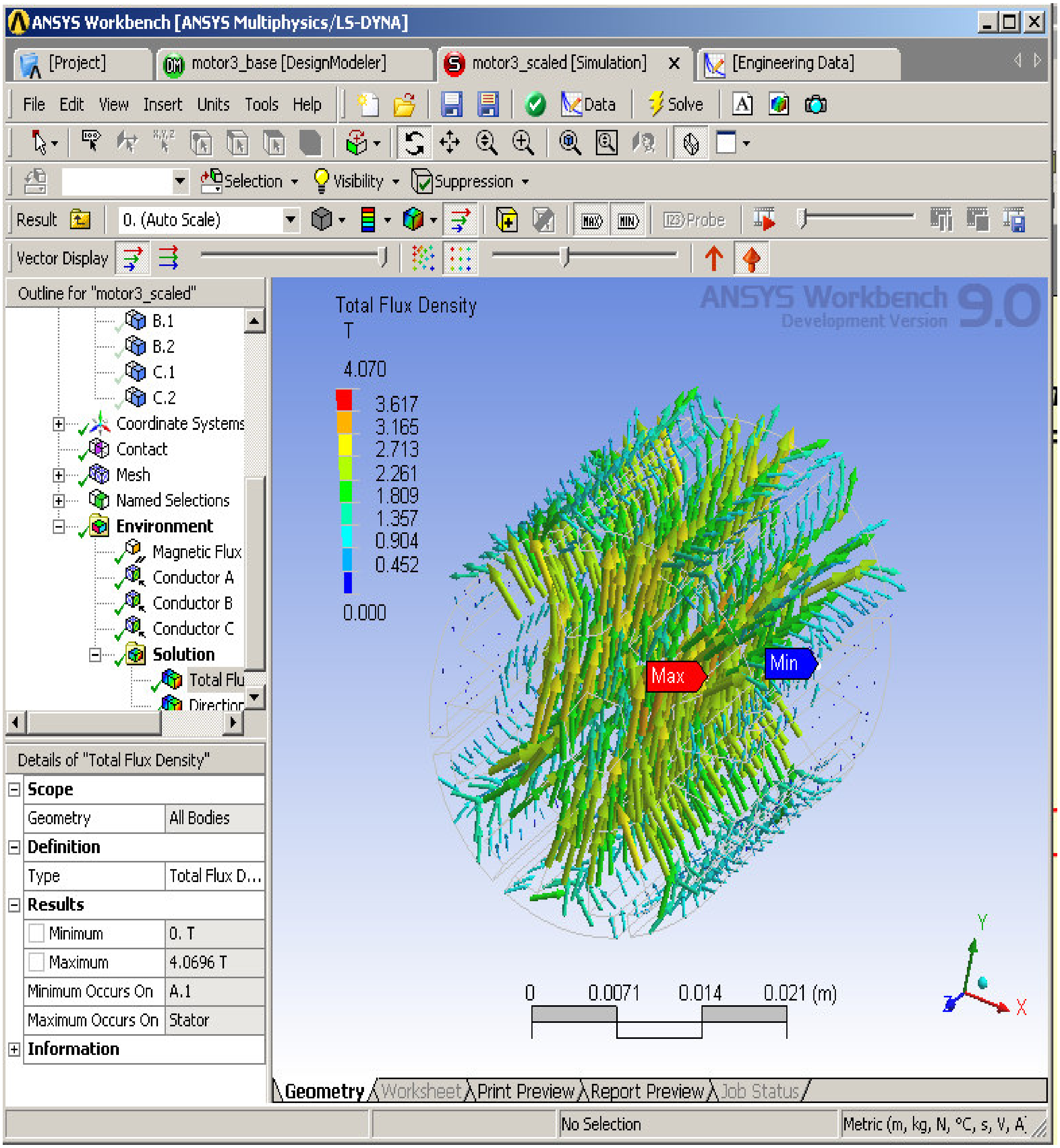


Details of "Total Flux Density"

Scope	
Geometry	All Bodies
Definition	
Type	Total Flux Density
Results	
□ Minimum	0. T
□ Maximum	4.0696 T
Minimum Occurs On	A.1
Maximum Occurs On	Stator
Information	

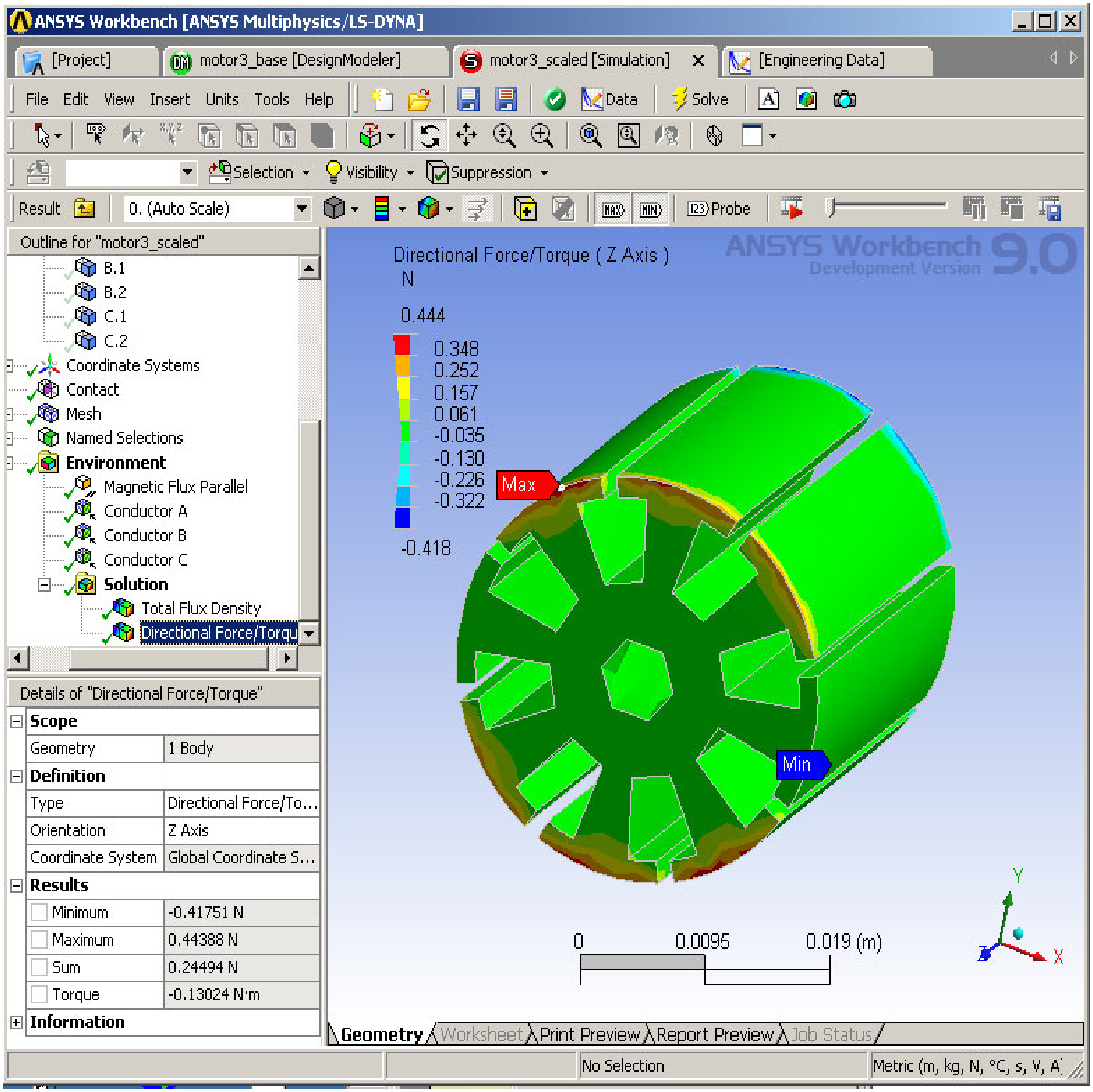


This figure is showing the electric flux lines in a motor.



Details of "Total Flux Density"

Scope	
Geometry	All Bodies
Definition	
Type	Total Flux D...
Results	
□ Minimum	0. T
□ Maximum	4.0696 T
Minimum Occurs On	A.1
Maximum Occurs On	Stator
Information	



Details of "Directional Force/Torque"

Scope	
Geometry	1 Body
Definition	
Type	Directional Force/To...
Orientation	Z Axis
Coordinate System	Global Coordinate S...
Results	
□ Minimum	-0.41751 N
□ Maximum	0.44388 N
□ Sum	0.24494 N
□ Torque	-0.13024 N·m
Information	

Conclusion and future work:

Two numerical methods are discussed; e.g. finite difference method and finite element method. Both methods are used to find two dimensional electric field distributions with given boundary conditions using MATLAB. Electric field distributions in more practical three dimensional cases with non-uniformly distributed dielectric of a capacitor in a DC busbar has found using C-programming. Also, electromagnetic field calculations of electric motor have been done in ANSYS.FEM and ANSYS can be used in order find the field distributions of high voltage insulators and cables with or without faults, which I left as a task for future batches to take over and complete the project.

Reference:-

[1] Sadiku Mathew n. o, Elements of electromagnetic, India, oxford, 4th edition, 2007.

[2] Hayt W H, Engineering electromagnetic, India, TMH, 7th edition, 2007.

[3] Zaengl W.S, Kuffel E, High voltage engineering, USA, Springer, 1st edition, 2002.

[4] Madenci Erdogan, Guven Ibrahim, The finite element method and applications in engineering using ANSYS,USA, Springer,1st edition, 2007.

[5] Sadiku Mathew n.o., Numerical techniques in electromagnetic, USA, CRC press, 1st edition, 2007.

[6] Ida Nathan, electromagnetic theory, USA, 2nd edition, Springer, 2008.

[7] Sawhney A.K, Electrical machine design, India, dhanpat rai,6th edition, 2006.

[8] Tadasu takuma, “recent development in electric field calculation”, IEEE transaction on magnetic, vol-33, no-2, march-1997.

[9] ANSYS-10 released helps files.

[10] ANSYS-9 workbench software package.

[11] Matlab-7 released helps files.

APPENDIX A: MATLAB

A.1 MATLAB program for 2D problem using finite difference method.

```
v1=0.0;

v2=100.0;

v3=0.0;

ni=200;

nx=9;

ny=9;

v=zeros(nx,ny);

for i=2:8

   v(i,1)=v1;

end

for j=2:8

   v((10-j),j)=v2;

end

for j=2:8

   v(1,j)=v3;

end

v(1,1)=0.5*(v1+v3);

v(9,1)=0.5*(v1+v2);

v(1,9)=0.5*(v3+v2);

for k=1:ni

   for i=2:8

      for j=2:8

         if (i+j)<10
         v(i,j)=0.25*(v(i+1,j)+v(i-1,j)+v(i,j+1)+v(i,j-1));
         end
      end
   end
end
v
```

A.2 MATLAB program for 2D problem using finite element method.

```
NE=25;
ND=21;
NP=15;
NL=[1 2 7
   2 8 7
  2 3 8
  3 9 8
  3 4 9
  4 10 9
  4 5 10
  5 11 10
  5 6 11
  7 8 12
  8 13 12
  8 9 13
  9 14 13
  9 10 14
  10 15 14
  10 11 15
  12 13 16
  13 17 16
  13 14 17
  14 18 17
  14 15 18
  16 17 19
  17 20 19
  17 18 20
  19 20 21];
X=[0.0 0.2 0.4 0.6 0.8 1.0 0.0 0.2 0.4 0.6 0.8 0.0 0.2 0.4 0.6 0.0 0.2 0.4 0.0 0.2 0.0];
Y=[0.0 0.0 0.0 0.0 0.0 0.0 0.2 0.2 0.2 0.2 0.2 0.4 0.4 0.4 0.4 0.6 0.6 0.6 0.8 0.8 1.0];
NDP=[1 2 3 4 5 6 11 15 18 20 21 19 16 12 7];
VAL=[0.0 0.0 0.0 0.0 0.0 50.0 100.0 100.0 100.0 100.0 50.0 0.0 0.0 0.0 0.0];
B=zeros(ND,1);
C=zeros(ND,ND);
for I=1:NE
   K=NL(I,[1:3]);
   XL=X(K);
   YL=Y(K);
   P=zeros(3,1);
   Q=zeros(3,1);
   P(1)=YL(2)-YL(3);
   P(2)=YL(3)-YL(1);
   P(3)=YL(1)-YL(2);
```

```
    Q(1)=XL(3)-XL(2);

    Q(2)=XL(1)-XL(3);

    Q(3)=XL(2)-XL(1);

    AREA =0.5*abs(P(2)*Q(3)-Q(2)*P(3));

    CE=(P*P'+Q*Q')/(4.0*(AREA+0.000000000000000000000001));

    for J=1:3

    IR=NL(I,J);

    IFLAG1=0;

    for K=1:NP

    if(IR==NDP(K))

       C(IR,IR)=1.0;

       B(IR)=VAL(K);

       IFLAG1=1;

    end

end

       if(IFLAG1==0)

          for L=1:3

          IC=NL(I,L);

          IFLAG2=0;

         for K=1:NP

            if (IC==NDP(K)),

               B(IR)=B(IR)-CE(J,L)*VAL(K);

          IFLAG2=1;

            end

         end
```

```
           if(IFLAG2==0)
               C(IR,IC)=C(IR,IC)+CE(J,L);
           end
            end
        end
    end
end
V=inv(C)*B;
V=V';
[ND, NE, NP]
[ [1:ND]' X' Y' V
```

APPENDIX B: C programming

B.1 C programming for Three-dimensional Laplace’s equation.

```
/* fdm3d.c
*/
#include "fdm2.dat"
```

```
/* Table of constant values */

static integer c__9 = 9;

static integer c__1 = 1;

static integer c__3 = 3;

static integer c__4 = 4;

/*     ITERATIVE FINITE DIFFERENCE SOLUTION */

/*     THE FOLLOWING DATA ARE USED: */

/*     VK = MATRIX OF ALL POTENTIALS */

/*     V  = VECTOR OF POTENTIALS AT INTERNAL NODES */

/*     N1X= NUMBER OF DIVISIONS IN THE X DIRECTION */

/*     N1Y= NUMBER OF DIVISIONS IN THE Y DIRECTION */

/*     VX1,VX2, BOUNDARY POTENTIALS ON THE LOWER AND UPPER BOUNDARIES */

/*     VY1,VY2, BOUNDARY POTENTIALS ON THE LEFT AND RIGHT BOUNDARIES */

/*     EPS, TOLERANCE ERROR PER NODE. */

/* Main program */ MAIN__()

{

   /* Format strings */

   static char fmt_11[] = "(1x,\002ITERATION NO:\002,i5)";

   static char fmt_7011[] = "(i5)";

   static char fmt_7010[] = "(3i5,4e10.4)";

   static char fmt_1[] = "(1x,\002NUMBER OF ITERATIONS = \002,i5,/)";

   static char fmt_2[] = "(3(\002   NODE:\002,i3,\002,  V=\002,f9.4))";

   /* System generated locals */

   integer i__1, i__2, i__3;

   real r__1;

   olist o__1;

   /* Builtin functions */

   integer f_open(), s_wsle(), do_lio(), e_wsle(), s_rsle(), e_rsle(),
           s_wsfe(), do_fio(), e_wsfe();
   /* Subroutine */ int s_stop();
   /* Local variables */
   static integer i, j, k, n;
   static real v[10000], x, y, z;
   static integer n1, n2, n3, kk;
   static real dx;
   static integer nn;
   static real vk[27000]      /* was [30][30][30] */, vv;
   static integer n1x, n1y, n1z;
   static real vv1, vx1, vx2, vy1, vy2, vz1, vz2, eps;
   /* Fortran I/O blocks */
   static cilist io___1 = { 0, 6, 0, 0, 0 };
   static cilist io___2 = { 0, 5, 0, 0, 0 };
   static cilist io___4 = { 0, 6, 0, 0, 0 };
   static cilist io___5 = { 0, 5, 0, 0, 0 };
   static cilist io___7 = { 0, 6, 0, 0, 0 };
   static cilist io___8 = { 0, 5, 0, 0, 0 };
   static cilist io___10 = { 0, 6, 0, 0, 0 };
   static cilist io___11 = { 0, 5, 0, 0, 0 };
   static cilist io___13 = { 0, 6, 0, 0, 0 };
   static cilist io___14 = { 0, 5, 0, 0, 0 };
   static cilist io___16 = { 0, 6, 0, 0, 0 };
   static cilist io___17 = { 0, 5, 0, 0, 0 };
   static cilist io___19 = { 0, 6, 0, 0, 0 };
   static cilist io___20 = { 0, 5, 0, 0, 0 };
   static cilist io___22 = { 0, 6, 0, 0, 0 };
   static cilist io___23 = { 0, 5, 0, 0, 0 };
```

```
    static cilist io___25 = { 0, 6, 0, 0, 0 };
    static cilist io___26 = { 0, 5, 0, 0, 0 };
    static cilist io___28 = { 0, 6, 0, 0, 0 };
    static cilist io___29 = { 0, 5, 0, 0, 0 };
    static cilist io___42 = { 0, 6, 0, fmt_11, 0 };
    static cilist io___46 = { 0, 3, 0, fmt_7011, 0 };
    static cilist io___50 = { 0, 3, 0, fmt_7010, 0 };
    static cilist io___51 = { 0, 1, 0, fmt_1, 0 };
    static cilist io___52 = { 0, 1, 0, 0, 0 };
    static cilist io___53 = { 0, 1, 0, fmt_2, 0 };
/* ***  ENTER DATA */

    o__1.oerr = 0;
    o__1.ounit = 1;
    o__1.ofnmlen = 4;
    o__1.ofnm = "OUT1";
    o__1.orl = 0;
    o__1.osta = 0;
    o__1.oacc = 0;
    o__1.ofm = "FORMATTED";
    o__1.oblnk = 0;
    f_open(&o__1);
    o__1.oerr = 0;
    o__1.ounit = 3;
    o__1.ofnmlen = 4;
    o__1.ofnm = "out2";
    o__1.orl = 0;
    o__1.osta = 0;
```

```
o__1.oacc = 0;
o__1.ofm = "formatted";
o__1.oblnk = 0;
f_open(&o__1);
s_wsle(&io___1);
do_lio(&c__9, &c__1, "ENTER NUMBER OF DIVISIONS IN THE X DIRECTION", 44L);
e_wsle();
s_rsle(&io___2);
do_lio(&c__3, &c__1, (char *)&n1x, (ftnlen)sizeof(integer));
e_rsle();
s_wsle(&io___4);
do_lio(&c__9, &c__1, "ENTER NUMBER OF DIVISIONS IN THE Y DIRECTION", 44L);
e_wsle();
s_rsle(&io___5);
do_lio(&c__3, &c__1, (char *)&n1y, (ftnlen)sizeof(integer));
e_rsle();
s_wsle(&io___7);
do_lio(&c__9, &c__1, "ENTER NUMBER OF DIVISIONS IN THE Z DIRECTION", 44L);
e_wsle();
s_rsle(&io___8);
do_lio(&c__3, &c__1, (char *)&n1z, (ftnlen)sizeof(integer));
e_rsle();
s_wsle(&io___10);
do_lio(&c__9, &c__1, "ENTER BOUNDARY CONDITION ON BOTTOM BOUNDARY", 43L);
e_wsle();
s_rsle(&io___11);
do_lio(&c__4, &c__1, (char *)&vx1, (ftnlen)sizeof(real));
e_rsle();
```

```
s_wsle(&io___13);
do_lio(&c__9, &c__1, "ENTER BOUNDARY CONDITION ON TOP BOUNDARY", 40L);
e_wsle();
s_rsle(&io___14);
do_lio(&c__4, &c__1, (char *)&vx2, (ftnlen)sizeof(real));
e_rsle();
s_wsle(&io___16);
do_lio(&c__9, &c__1, "ENTER BOUNDARY CONDITION ON LEFT BOUNDARY", 41L);
e_wsle();
s_rsle(&io___17);
do_lio(&c__4, &c__1, (char *)&vy1, (ftnlen)sizeof(real));
e_rsle();
s_wsle(&io___19);
do_lio(&c__9, &c__1, "ENTER BOUNDARY CONDITION ON RIGHT BOUNDARY", 42L);
e_wsle();
s_rsle(&io___20);
do_lio(&c__4, &c__1, (char *)&vy2, (ftnlen)sizeof(real));
e_rsle();
s_wsle(&io___22);
do_lio(&c__9, &c__1, "ENTER BOUNDARY CONDITION ON FRONT BOUNDARY", 42L);
e_wsle();
s_rsle(&io___23);
do_lio(&c__4, &c__1, (char *)&vz1, (ftnlen)sizeof(real));
e_rsle();
s_wsle(&io___25);
do_lio(&c__9, &c__1, "ENTER BOUNDARY CONDITION ON BACK BOUNDARY", 41L);
e_wsle();
s_rsle(&io___26);
```

```
    do_lio(&c__4, &c__1, (char *)&vz2, (ftnlen)sizeof(real));

    e_rsle();

    s_wsle(&io___28);

    do_lio(&c__9, &c__1, "ENTER REQUIRED TOLLERANCE ERROR", 31L);

    e_wsle();

    s_rsle(&io___29);

    do_lio(&c__4, &c__1, (char *)&eps, (ftnlen)sizeof(real));

    e_rsle();


/* **** START CALCULATION */


    n1 = n1x + 1;

    n2 = n1y + 1;

    n3 = n1z + 1;

    i__1 = n1;

    for (i = 1; i <= i__1; ++i) {

            i__2 = n2;

            for (j = 1; j <= i__2; ++j) {

                i__3 = n3;

                for (k = 1; k <= i__3; ++k) {

                        vk[i + (j + k * 30) * 30 - 931] = (float)0.;

/* L100: */

                }

            }

    }


/* **** SET BOUNDARY CONDITIONS ON UPPER, LOWER, LEFT AND RIGHT SURFACES

*/
```

```
    i__3 = n1;
    for (i = 1; i <= i__3; ++i) {
            i__2 = n2;
            for (j = 1; j <= i__2; ++j) {
                vk[i + (j + 30) * 30 - 931] = vz1;
                vk[i + (j + n3 * 30) * 30 - 931] = vz2;
/* L120: */
            }
    }
    i__2 = n2;
    for (j = 1; j <= i__2; ++j) {
            i__3 = n3;
            for (k = 1; k <= i__3; ++k) {
                vk[(j + k * 30) * 30 - 930] = vx1;
                vk[n1 + (j + k * 30) * 30 - 931] = vx2;
/* L130: */
            }
    }
    i__3 = n1;
    for (i = 1; i <= i__3; ++i) {
            i__2 = n3;
            for (k = 1; k <= i__2; ++k) {
                vk[i + (k * 30 + 1) * 30 - 931] = vy1;
                vk[i + (n2 + k * 30) * 30 - 931] = vy2;
/* L140: */
            }
    }
```

```
/* **** START ITERATIONS */


    n = (n1 - 2) * (n2 - 2) * (n3 - 2);

    kk = 0;

    vv = (float)0.;

L1000:

    vv1 = vv;

    i__2 = n1 - 1;

    for (i = 2; i <= i__2; ++i) {

             i__3 = n2 - 1;

             for (j = 2; j <= i__3; ++j) {

                 i__1 = n3 - 1;

                 for (k = 2; k <= i__1; ++k) {

                          vk[i + (j + k * 30) * 30 - 931] = vk[i + (j - 1 + k * 30) *

                                       30 - 931] + vk[i + (j + 1 + k * 30) * 30 - 931] + vk[

                                       i - 1 + (j + k * 30) * 30 - 931] + vk[i + 1 + (j + k *

                                        30) * 30 - 931] + vk[i + (j + (k - 1) * 30) * 30 -

                                       931] + vk[i + (j + (k + 1) * 30) * 30 - 931];

                          vk[i + (j + k * 30) * 30 - 931] /= (float)6.;

                          vv += (r__1 = vk[i + (j + k * 30) * 30 - 931], dabs(r__1));

/* L160: */

                 }

             }

    }

    vv /= n;    ++kk;

/*      WRITE(*,*)'ENTER 1 IF YOU WANT DISPLAY OF CURRENT ITERATION' */

/*      READ(*,*)II */
/*     IF(II.NE.1) GO TO 111 */

/*    N=0 */

/*    DO 181 I=2,N1-1 */

/*    DO 181 J=2,N2-1 */

/*    DO 181 K=2,N3-1 */

/*    N=N+1 */

/*    V(N)=VK(I,J,K) */

/* 181 CONTINUE */

/*    WRITE(1,2)(K,V(K),K=1,N) */

/* 111 CONTINUE */

    s_wsfe(&io___42);

    do_fio(&c__1, (char *)&kk, (ftnlen)sizeof(integer));

    e_wsfe();

    if ((r__1 = vv - vv1, dabs(r__1)) > eps) {

            goto L1000;

    }


/*    THE ERROR CRITERION HAS BEEN MET. THE CORRECT SOLUTION IS IN VK */

/* **** REWRITE THE TWO-DIMENSIONAL VECTOR VK INTO A ONE DIMENSIONA */

/*    VECTOR V WHICH CONTAINS ONLY THE INTERIOR NODES. */

    n = 0;

    i__1 = n1 - 1;

    for (i = 2; i <= i__1; ++i) {

            i__3 = n2 - 1;

            for (j = 2; j <= i__3; ++j) {

                i__2 = n2 - 1;

                for (k = 2; k <= i__2; ++k) {

                        ++n;
                        v[n - 1] = vk[i + (j + k * 30) * 30 - 931];
/* L180: */
                }
            }
    }
/* **** WRITE THE SOLUTION VECTOR */
   dx = (float)1.;
    nn = n1 * n2 * n3;
    s_wsfe(&io___46);
    do_fio(&c__1, (char *)&nn, (ftnlen)sizeof(integer));
    e_wsfe();
    i__2 = n3;
    for (i = 1; i <= i__2; ++i) {
            i__3 = n2;
            for (j = 1; j <= i__3; ++j) {
                i__1 = n1;
                for (k = 1; k <= i__1; ++k) {
                        x = dx * (k - 1);
                        y = dx * (j - 1);
                        z = dx * (i - 1);
                        s_wsfe(&io___50);
                        do_fio(&c__1, (char *)&i, (ftnlen)sizeof(integer));
                        do_fio(&c__1, (char *)&j, (ftnlen)sizeof(integer));
                        do_fio(&c__1, (char *)&k, (ftnlen)sizeof(integer));
                        do_fio(&c__1, (char *)&x, (ftnlen)sizeof(real));
                        do_fio(&c__1, (char *)&y, (ftnlen)sizeof(real));
                        do_fio(&c__1, (char *)&z, (ftnlen)sizeof(real));
                        do_fio(&c__1, (char *)&vk[i + (j + k * 30) * 30 - 931], (
                        ftnlen)sizeof(real));
                e_wsfe();
/* L7001: */
            }
        }
    }
    s_wsfe(&io___51);
    do_fio(&c__1, (char *)&kk, (ftnlen)sizeof(integer));
    e_wsfe();
    s_wsle(&io___52);
    do_lio(&c__9, &c__1, "SOLUTION:", 9L);
    e_wsle();
    s_wsfe(&io___53);
    i__1 = n;
    for (k = 1; k <= i__1; ++k) {
        do_fio(&c__1, (char *)&k, (ftnlen)sizeof(integer));
        do_fio(&c__1, (char *)&v[k - 1], (ftnlen)sizeof(real));
    }
    e_wsfe();
    s_stop("", 0L);
} /* MAIN__ */
```

Input data:- 26 34 2 1

```
1 2 7 1
2 8 7 1
2 3 8 1
3 9 8 1
3 4 9 1
4 10 9 1
4 5 10 1
```

```
5 11 10 1
5 6 11 1
6 12 11 1
7 8 13 1
8 14 13 1
8 9 14 1
9 15 14 1
9 10 15 1
10 16 15 1
10 11 16 1
11 17 16 1
11 12 17 1
12 18 17 1
13 14 19 1
14 20 19 1
14 22 20 1
22 21 20 1
14 15 22 1
15 24 22 1
15 16 24 1
22 24 21 1
24 23 21 1
16 17 24 1
24 17 25 1
24 25 23 1
17 26 25 1
17 18 26 1
0.0 0.0
0.2 0.0
0.4 0.0
0.6 0.0
0.8 0.0
```

```
1.0 0.0

0.0 0.16

0.2 0.16

0.4 0.16

0.6 0.16

0.8 0.16

1.0 0.16

0.0 0.34

0.2 0.34

0.4 0.34

0.6 0.34

0.8 0.34

1.0 0.34

0.05 0.46

0.2 0.55

0.4 0.6

0.4 0.46

0.6 0.6

0.6 0.46

0.8 0.55

0.95 0.46
```

B.2 C programming for non-uniformly distributed dielectric of capacitor.

```
/* fem1.c

*/

#include "fdm3.dat"

/* Table of constant values */

static integer c__3 = 3;

static integer c__1 = 1;

static integer c__4 = 4;

/* ***  FINITE ELEMENT PROGRAM FOR ELECTROSTATIC PROBLEMS */
/* Main program */ MAIN__()
{
    /* Format strings */
    static char fmt_31[] = "(2i3)";
    static char fmt_171[] = "(i3,5x,e10.4)";
    static char fmt_170[] = "(\002 NODE-\002,i3,\002    POTENTIAL=\002,e10.4)"
            ;
    static char fmt_191[] = "(i3,3(2x,e10.4))";
    static char fmt_190[] = "(\002 ELEMENT-\002,i3,\002  EX=\002,e10.4,\002 \
 EY=\002,e10.4,\002  EM=\002,e10.4)";

    /* System generated locals */
    integer i__1, i__2, i__3;
    olist o__1;

    /* Builtin functions */
    integer f_open(), s_rsle(), do_lio(), e_rsle(), s_wsfe(), do_fio(),
              e_wsfe();
    double sqrt();
    /* Subroutine */ int s_stop();

    /* Local variables */
    static real fact, emod, emax;
    static integer ncon, nmat, naux[3];
    static real xeps;
    static integer i, j, k, l, m;
    static real q[150], coeff, s[9]       /* was [3][3] */, x[150], y[150];
    static integer nelem, nnode, n1, n2, n3;
    static real q1, q2, q3, r1, r2, r3, coeff1;
    static integer jj;
    static real qe[3];
    static integer kk, nm, nn;
    static real ex;
    static integer np[600]     /* was [200][3] */;
    static real vi[30], ey, ro[10], ss[22500]      /* was [150][150] */, vv[150];
    static integer nbound[300]           /* was [10][30] */;
    static real det;
    static integer mat[200];
    static real eps[10];
    static integer nox;
    static real sum, xro;
    static integer max1, max2, max3;

    /* Fortran I/O blocks */
    static cilist io___11 = { 0, 1, 0, 0, 0 };
    static cilist io___16 = { 0, 3, 0, fmt_31, 0 };
    static cilist io___17 = { 0, 1, 0, 0, 0 };
    static cilist io___20 = { 0, 1, 0, 0, 0 };
    static cilist io___23 = { 0, 1, 0, 0, 0 };
    static cilist io___25 = { 0, 1, 0, 0, 0 };
    static cilist io___26 = { 0, 1, 0, 0, 0 };
    static cilist io___55 = { 0, 3, 0, fmt_171, 0 };
    static cilist io___56 = { 0, 2, 0, fmt_170, 0 };
    static cilist io___61 = { 0, 3, 0, fmt_191, 0 };
    static cilist io___62 = { 0, 2, 0, fmt_190, 0 };
/* **** ZERO THE VARIOUS ARRAYS */
```

```
o__1.oerr = 0;
o__1.ounit = 1;
o__1.ofnmlen = 4;
o__1.ofnm = "DAT1";
o__1.orl = 0;
o__1.osta = 0;
o__1.oacc = 0;
o__1.ofm = 0;
o__1.oblnk = 0;
f_open(&o__1);
o__1.oerr = 0;
o__1.ounit = 2;
o__1.ofnmlen = 4;
o__1.ofnm = "OUT2";
o__1.orl = 0;
o__1.osta = 0;
o__1.oacc = 0;
o__1.ofm = "FORMATTED";
o__1.oblnk = 0;
f_open(&o__1);
o__1.oerr = 0;
o__1.ounit = 3;
o__1.ofnmlen = 4;
o__1.ofnm = "OUT1";
o__1.orl = 0;
o__1.osta = 0;
o__1.oacc = 0;
o__1.ofm = "formatted";
```

```
    o__1.oblnk = 0;
    f_open(&o__1);
    max1 = 150;
    max2 = 10;
    max3 = 30;
    i__1 = max1;
    for (i = 1; i <= i__1; ++i) {
            vv[i - 1] = (float)0.;
            q[i - 1] = (float)0.;
            ro[i - 1] = (float)0.;
            i__2 = max1;
            for (j = 1; j <= i__2; ++j) {
/* L10: */
                ss[i + j * 150 - 151] = (float)0.;
            }
    }
    i__2 = max2;
    for (i = 1; i <= i__2; ++i) {
            i__1 = max2;
            for (j = 1; j <= i__1; ++j) {
/* L20: */
                nbound[i + j * 10 - 11] = 0;
            }
    }

/* **** READ INPUT DATA */

    s_rsle(&io___11);
    do_lio(&c__3, &c__1, (char *)&nnode, (ftnlen)sizeof(integer));
    do_lio(&c__3, &c__1, (char *)&nelem, (ftnlen)sizeof(integer));
    do_lio(&c__3, &c__1, (char *)&ncon, (ftnlen)sizeof(integer));
    do_lio(&c__3, &c__1, (char *)&nmat, (ftnlen)sizeof(integer));
    e_rsle();
/* ---------------READ THE MESH STRUCTURE */
    s_wsfe(&io___16);
    do_fio(&c__1, (char *)&nnode, (ftnlen)sizeof(integer));
    do_fio(&c__1, (char *)&nelem, (ftnlen)sizeof(integer));
    e_wsfe();
    i__1 = nelem;
    for (i = 1; i <= i__1; ++i) {
/*          write(3,31)i,i */
/* L30: */
            s_rsle(&io___17);
            do_lio(&c__3, &c__1, (char *)&np[i - 1], (ftnlen)sizeof(integer));
            do_lio(&c__3, &c__1, (char *)&np[i + 199], (ftnlen)sizeof(integer));
            do_lio(&c__3, &c__1, (char *)&np[i + 399], (ftnlen)sizeof(integer));
            do_lio(&c__3, &c__1, (char *)&mat[i - 1], (ftnlen)sizeof(integer));
            e_rsle();
    }
/* ---------------READ NODE COORDINATES */
    i__1 = nnode;
    for (i = 1; i <= i__1; ++i) {
/*          write(3,31)i,i */
/* L40: */
            s_rsle(&io___20);
            do_lio(&c__4, &c__1, (char *)&x[i - 1], (ftnlen)sizeof(real));
```

```
          do_lio(&c__4, &c__1, (char *)&y[i - 1], (ftnlen)sizeof(real));

          e_rsle();

   }

/* -----READ BOUNDARY CONDITIONS */

   i__1 = ncon;

   for (i = 1; i <= i__1; ++i) {

          s_rsle(&io___23);

          do_lio(&c__4, &c__1, (char *)&vi[i - 1], (ftnlen)sizeof(real));

          e_rsle();

          s_rsle(&io___25);

          for (j = 1; j <= 20; ++j) {

             do_lio(&c__3, &c__1, (char *)&nbound[i + j * 10 - 11], (ftnlen)

                       sizeof(integer));

          }

          e_rsle();

/* L50: */

   }

/* -----READ PERMITTIVITIES OF MATERIALS */

   i__1 = nmat;

   for (i = 1; i <= i__1; ++i) {

/*   60 READ(1,*)EPS(I), RO(I) */

          ro[i - 1] = (float)0.;

/* L60: */

          s_rsle(&io___26);

          do_lio(&c__4, &c__1, (char *)&eps[i - 1], (ftnlen)sizeof(real));

          e_rsle();

   }
```

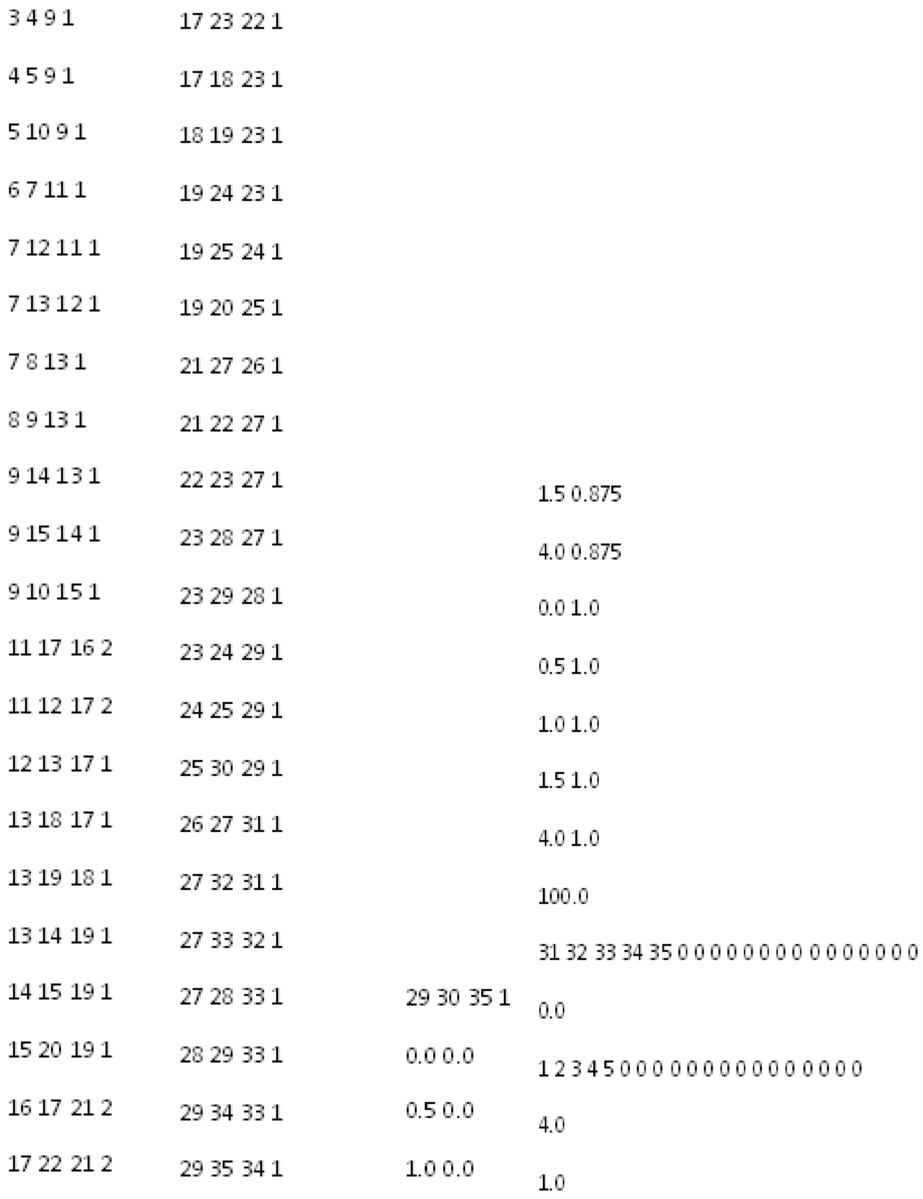
```
/* **** FORM THE GLOBAL MATRIX SS() */
/* ---------------DO FOR NELEM ELEMENTS */
    i__1 = nelem;
    for (i = 1; i <= i__1; ++i) {
/*          write(3,31)i,i */
            n1 = np[i - 1];
            n2 = np[i + 199];
            n3 = np[i + 399];
            nm = mat[i - 1];
/* ---------------CALCULATE Qi, Qj, Qk, Ri, Rj, Rk */
            q1 = y[n2 - 1] - y[n3 - 1];
            q2 = y[n3 - 1] - y[n1 - 1];
            q3 = y[n1 - 1] - y[n2 - 1];
            r1 = x[n3 - 1] - x[n2 - 1];
            r2 = x[n1 - 1] - x[n3 - 1];
            r3 = x[n2 - 1] - x[n1 - 1];
            xeps = eps[nm - 1] * (float)8.854e-12;
            xro = ro[nm - 1];
/* ---------------CALCULATE DETERMINANT, TWICE THE AREA OF TRIANGLE */
            det = x[n2 - 1] * y[n3 - 1] + x[n1 - 1] * y[n2 - 1] + x[n3 - 1] * y[
                        n1 - 1] - x[n1 - 1] * y[n3 - 1] - x[n3 - 1] * y[n2 - 1] - x[
                        n2 - 1] * y[n1 - 1];
            coeff = xeps / det / (float)2.;
            coeff1 = xro * det / (float)6.;
/* ---------------CALCULATE THE TERMS S(3,3) */
            s[0] = coeff * (q1 * q1 + r1 * r1);
            s[3] = coeff * (q1 * q2 + r1 * r2);
            s[6] = coeff * (q1 * q3 + r1 * r3);
            s[1] = s[3];
            s[4] = coeff * (q2 * q2 + r2 * r2);
            s[7] = coeff * (q2 * q3 + r2 * r3);
            s[2] = s[6];
            s[5] = s[7];
            s[8] = coeff * (q3 * q3 + r3 * r3);
            qe[0] = coeff1;
            qe[1] = coeff1;
            qe[2] = coeff1;
/* ---------------ASSEMBLE THE S(3,3) INTO THE MATRIX SS(NNODE,NNODE)
*/
            naux[0] = n1;
            naux[1] = n2;
            naux[2] = n3;
            for (k = 1; k <= 3; ++k) {
                kk = naux[k - 1];
                q[kk - 1] += qe[k - 1];
                for (j = 1; j <= 3; ++j) {
                         jj = naux[j - 1];
/* L80: */
                         ss[kk + jj * 150 - 151] += s[k + j * 3 - 4];
                }
            }
/* L70: */
    }

/* **** INSERT BOUNDARY CONDITIONS */
```

```
   i__1 = ncon;

   for (i = 1; i <= i__1; ++i) {

          for (j = 1; j <= 20; ++j) {

             nox = nbound[i + j * 10 - 11];

             if (nox == 0) {

                    goto L90;

             }

/* ---------------ZERO THE COEFFICIENTS IN LINE OF MATRIX SS */

             i__2 = nnode;

             for (l = 1; l <= i__2; ++l) {

/* L110: */

                    ss[nox + l * 150 - 151] = (float)0.;

             }

/* ---------------SET THE DIAGONAL TO 1. */

             ss[nox + nox * 150 - 151] = (float)1.;

/* ---------------PLACE THE IMPOSED POTENTIAL IN THE RIGHT HAND SI

DE */

             q[nox - 1] = vi[i - 1];

/* L100: */

          }

L90:

          ;

   }

/* **** SOLVE THE MATRIX SYSTEM */

/* ---------------GAUSSIAN ELIMINATION */


   nn = nnode - 1;

   i__1 = nn;
```

```
    for (i = 1; i <= i__1; ++i) {
        i__2 = nnode;
        for (m = i + 1; m <= i__2; ++m) {
            fact = ss[m + i * 150 - 151] / ss[i + i * 150 - 151];
            q[m - 1] -= q[i - 1] * fact;
            i__3 = nnode;
            for (j = i + 1; j <= i__3; ++j) {
/* L130: */
                ss[m + j * 150 - 151] -= ss[i + j * 150 - 151] * fact;
            }
/* L120: */
        }
    }
    vv[nnode - 1] = q[nnode - 1] / ss[nnode + nnode * 150 - 151];
    for (i = nn; i >= 1; --i) {
        sum = (float)0.;
        i__2 = nnode;
        for (j = i + 1; j <= i__2; ++j) {
/* L150: */
            sum += ss[i + j * 150 - 151] * vv[j - 1];
        }
        vv[i - 1] = (q[i - 1] - sum) / ss[i + i * 150 - 151];
/* L140: */
    }

/* **** PRINT THE RESULTS */

    i__2 = nnode;
    for (i = 1; i <= i__2; ++i) {

             s_wsfe(&io___55);

             do_fio(&c__1, (char *)&i, (ftnlen)sizeof(integer));

             do_fio(&c__1, (char *)&vv[i - 1], (ftnlen)sizeof(real));

             e_wsfe();

/* L160: */

             s_wsfe(&io___56);

             do_fio(&c__1, (char *)&i, (ftnlen)sizeof(integer));

             do_fio(&c__1, (char *)&vv[i - 1], (ftnlen)sizeof(real));

             e_wsfe();

    }

/* ---------------PRINT THE FIELDS IN THE ELEMENTS */



    emax = (float)0.;

    i__2 = nelem;

    for (i = 1; i <= i__2; ++i) {


/* **** CALCULATE THE FIELDS OR GRADIENTS */


             n1 = np[i - 1];

             n2 = np[i + 199];

             n3 = np[i + 399];

             q1 = y[n2 - 1] - y[n3 - 1];

             q2 = y[n3 - 1] - y[n1 - 1];

             q3 = y[n1 - 1] - y[n2 - 1];

             r1 = x[n3 - 1] - x[n2 - 1];

             r2 = x[n1 - 1] - x[n3 - 1];
```

```
        r3 = x[n2 - 1] - x[n1 - 1];
/* ---------------CALCULATE DETERMINANT, TWICE THE AREA OF TRIANGLE */
        det = x[n2 - 1] * y[n3 - 1] + x[n1 - 1] * y[n2 - 1] + x[n3 - 1] * y[
                n1 - 1] - x[n1 - 1] * y[n3 - 1] - x[n3 - 1] * y[n2 - 1] - x[
                n2 - 1] * y[n1 - 1];
        ex = -(doublereal)(q1 * vv[n1 - 1] + q2 * vv[n2 - 1] + q3 * vv[n3 - 1]
                ) / det;
        ey = -(doublereal)(r1 * vv[n1 - 1] + r2 * vv[n2 - 1] + r3 * vv[n3 - 1]
                ) / det;
        emod = sqrt(ex * ex + ey * ey);
/*  180 CONTINUE */
        s_wsfe(&io___61);
        do_fio(&c__1, (char *)&i, (ftnlen)sizeof(integer));
        do_fio(&c__1, (char *)&ex, (ftnlen)sizeof(real));
        do_fio(&c__1, (char *)&ey, (ftnlen)sizeof(real));
        do_fio(&c__1, (char *)&emod, (ftnlen)sizeof(real));
        e_wsfe();
/* L180: */
        s_wsfe(&io___62);
        do_fio(&c__1, (char *)&i, (ftnlen)sizeof(integer));
        do_fio(&c__1, (char *)&ex, (ftnlen)sizeof(real));
        do_fio(&c__1, (char *)&ey, (ftnlen)sizeof(real));
        do_fio(&c__1, (char *)&emod, (ftnlen)sizeof(real));
        e_wsfe();
  }
  s_stop("", 0L);
} /* MAIN__
```

Input data:



```
3 4 9 1
4 5 9 1
5 10 9 1
6 7 11 1
7 12 11 1
7 13 12 1
7 8 13 1
8 9 13 1
9 14 13 1
9 15 14 1
9 10 15 1
11 17 16 2
11 12 17 2
12 13 17 1
13 18 17 1
13 19 18 1
13 14 19 1
14 15 19 1
15 20 19 1
16 17 21 2
17 22 21 2
17 23 22 1
17 18 23 1
18 19 23 1
19 24 23 1
19 25 24 1
19 20 25 1
21 27 26 1
21 22 27 1
22 23 27 1
23 28 27 1
23 29 28 1
23 24 29 1
24 25 29 1
25 30 29 1
26 27 31 1
27 32 31 1
27 33 32 1
27 28 33 1
28 29 33 1
29 34 33 1
29 35 34 1
29 30 35 1
0.0 0.0
0.5 0.0
1.0 0.0
1.5 0.875
4.0 0.875
0.0 1.0
0.5 1.0
1.0 1.0
1.5 1.0
4.0 1.0
100.0
31 32 33 34 35 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.0
1 2 3 4 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4.0
1.0
```

B.3 input data for DC busbar.

39 52 2 1 1 2 6 12 7 6 1 2 3 7 1 3 8 7 1 3 4 8 1 4 9 8 1 4 5 9 1 5 10 9 1 6 7 11 1 7 12 11 1

7 8 12 1 8 13 12 1 8 9 13 1 9 14 13 1 9 10 14 1 10 15 14 1 12 13 16 1 13 17 16 1 13 14 17 1

14 18 17 1 14 15 18 1 15 19 18 1 16 17 21 1 17 22 21 1 17 18 22 1 18 23 22 1 18 19 23 1

```
19 24 23 1  20 21 25 1  21 26 25 1  21 22 26 1  22 27 26 1  22 23 27 1  23 28 27 1  23 24 28 1
24 29 28 1   25 26 30 1   26 31 30 1   26 27 31 1   27 32 31 1   27 28 32 1   28 33 32 1   28 29 33 1
29 34 33 1   30 31 35 1   31 36 35 1  31 32 36 1  32 37 36 1  32 33 37 1  33 38 37 1   33 34 38 1
34 39 38 1  0.0 0.0  0.01 0.0   0.025 0.0  0.05 0.0  0.1 0.0  0.0 0.0025  0.01 0.0025   0.025 0.0025
0.05 0.0025  0.1 0.0025  0.0 0.005  0.01 0.005  0.025 0.005  0.05 0.005  0.1 0.005   0.01 0.01
0.025 0.01   0.05 0.01  0.1 0.01  0.0 0.015  0.01 0.015   0.025 0.015  0.05 0.015    0.1 0.015
0.0 0.035  0.01 0.035   0.025 0.035  0.05 0.035  0.1 0.035  0.0 0.06  0.01 0.06    0.025 0.06
0.05 0.06   0.1 0.06  0.0 0.12   0.01 0.12   0.025 0.12  0.05 0.12  0.1 0.12  220.0
11 12 16 21 20 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0    0.0
1 2 3 4 5 10 15 19 24 29 34 39 38 37 36 35 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0   1.0
```