ENME 815

The Theory of Finite Elements

Project 1

Trusses

Syed Ali

Contents

PROJECT STATEMENT

The overall objective of this project was to develop a finite element algorithm that allows a user to solve a truss structure with n number of elements. The algorithm allows the user to achieve the following:

i. The algorithm provides a mesh generator that conforms to a specific truss as shown in the figure below:

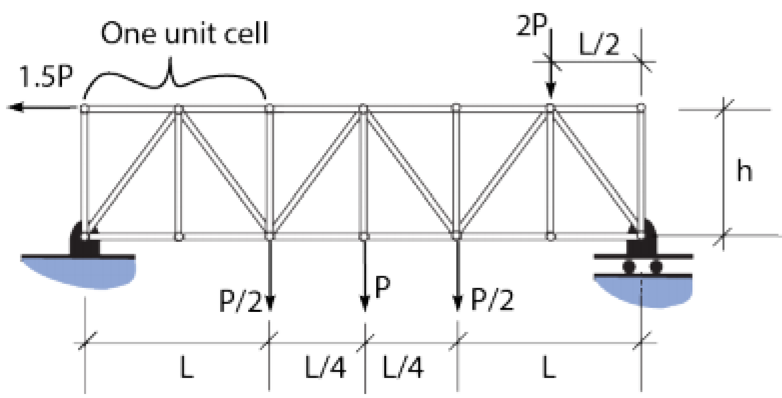


Figure 1 The truss structure used in this study

ii. The mesh generated in (i) allows the user to take the mesh information and solves for a truss structure composed of 10-unit cells which are supported and loaded as shown in the above diagram.

Assumptions:

The following assumptions were made when solving the truss problem above:

1. The physical parameters (Area, length, etc.) were kept a constant value of 1, and so were the material properties.
2. Neglected any forces exerted from gravity
3. Considered the problem as a 2D problem and not a 3D problem.

Problem Formulation:

3.1 Truss Structure

The following truss structure was used in this study:

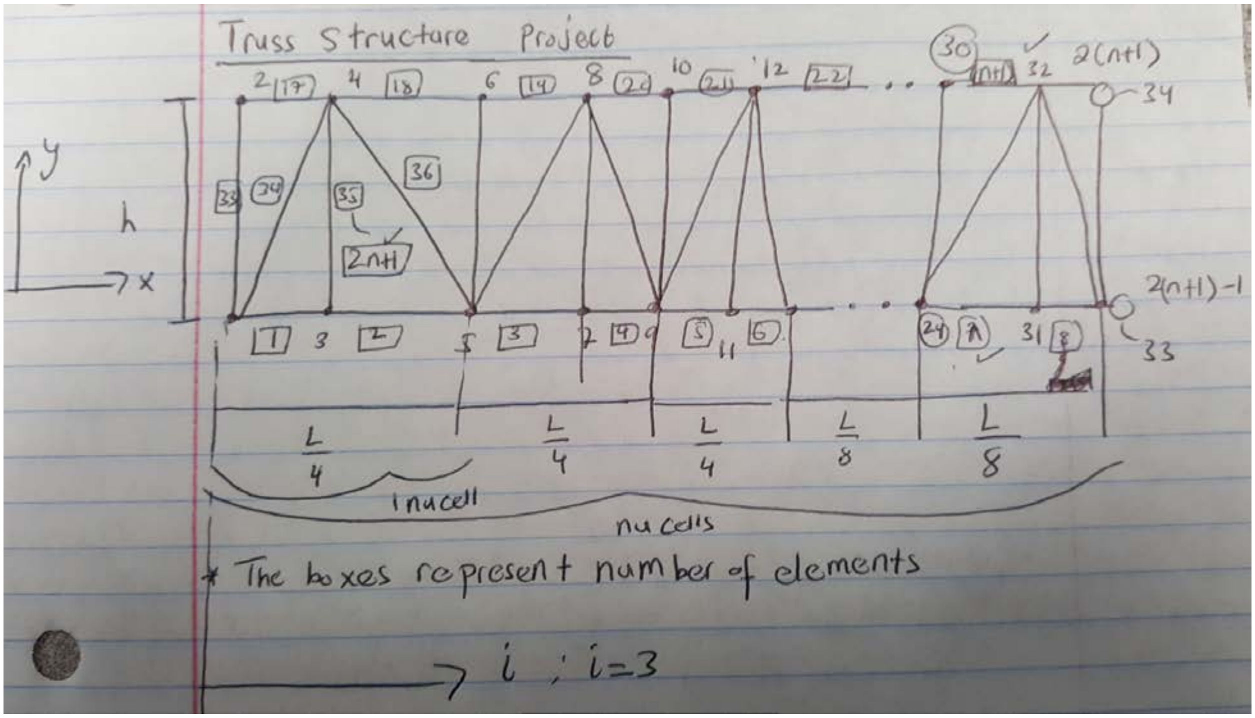


Figure 2 Truss structure used in this study

The above truss structure was divided into n, n+1, and 2n+1 elements, along with 2(n+1)-1 and 2(n+1) nodes respectively. The division of elements and nodes as prescribed above allowed one to create a mesh for n number of elements and nodes.

3.2 Mesh Generation

If we look at the element connectivity matrix for a single unit cell, we observe that for every n element, they are connected to 2(n+1) nodes and for every (n+1) element they are connected to 2(n+1)-1 nodes. Likewise, for every 2n+1 element, they are connected to n number of nodes, and for every 2n+2 element, they are connected to 3(n+1)-2 nodes. It is assumed that n starts from 0. A table below summarizes the statement given above.

Number of Unit cells	Elements	Nodes
1	1	1,3
	2	3,5
	33	1,2
	34	1,4
	35	3,4
	36	5,4
	17	2,4
	18	4,6
	37	5,6
2	3	5,7
	4	7,9
	37	5,6
	38	5,8
	39	7,8
	40	8,9
	41	9,10
n	n	2(n+1)-1
	n+1	2(n+1)
	2n+2	3(n+1)-2
	2n+1	n

Table 1: A table showing the number of elements and their connectivity

With the above information, one can now easily create a mesh for a truss which has n number of elements. A mesh for an 82-node truss is shown on the next page:

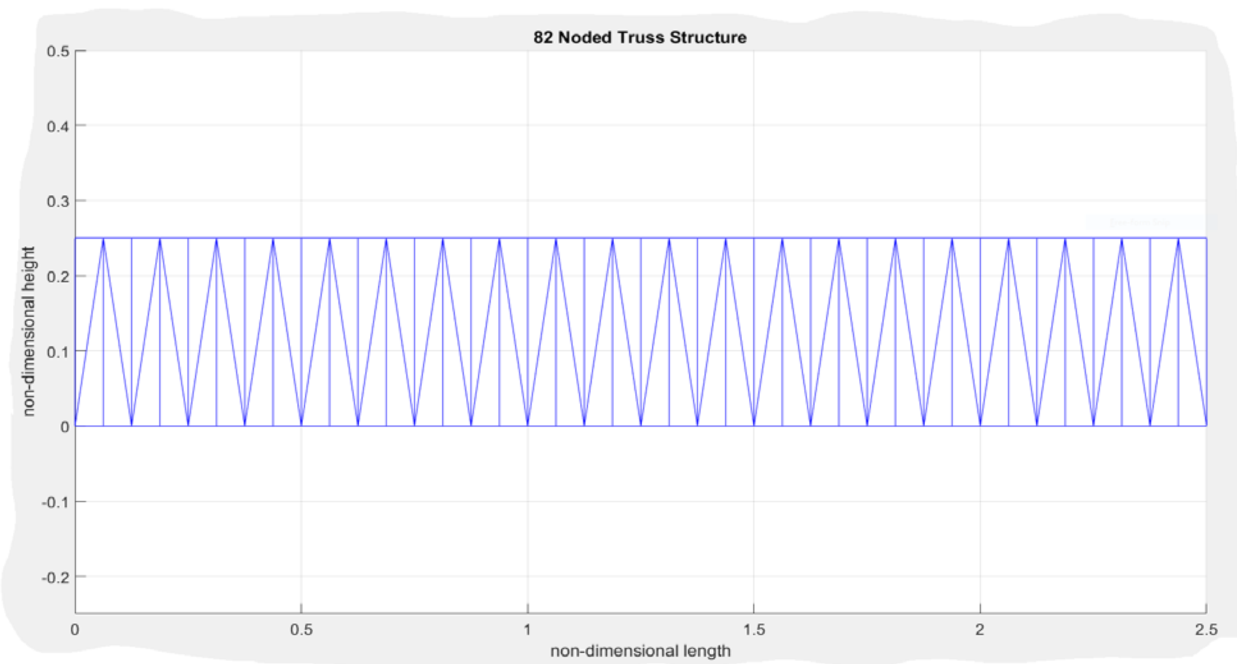
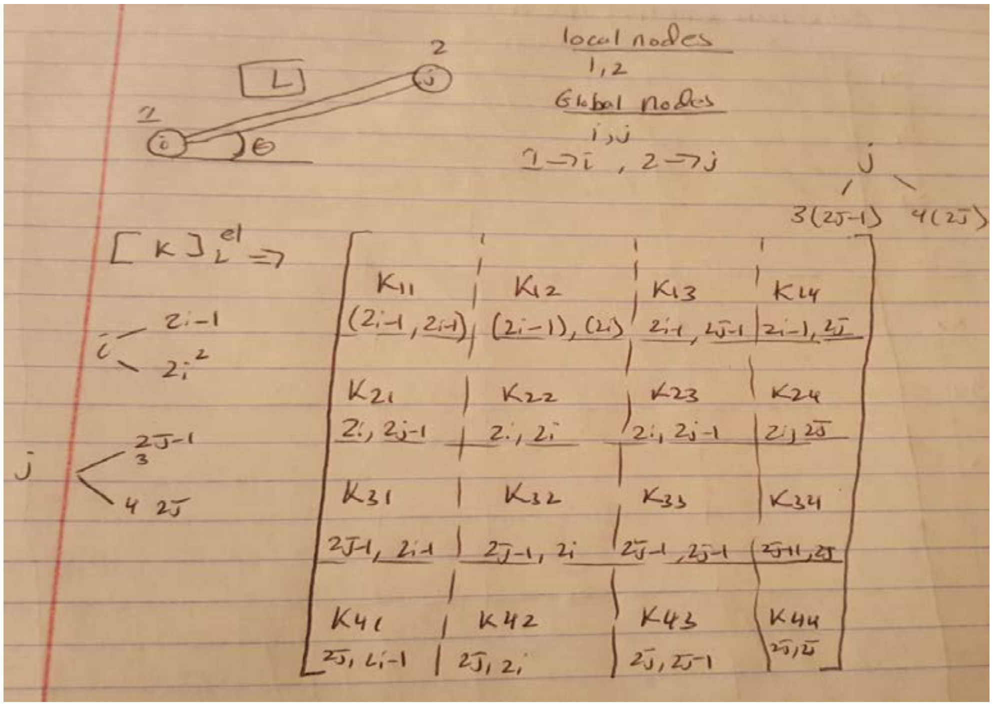


Figure 3 Mesh Generation for an 82-node truss structure created in MATLAB

3.3 Stiffness Matrix Assembly

With the mesh created, it was now time to determine the local stiffness matrix assembly. The stiffness matrix was formulated the following way:

The local stiffness matrix was formulated the following way:



where K11, K21, K31, K41, K12, K22, K32, K42, K13, K23, K33, K43, K14, K24, K34, K44 equal to the following:

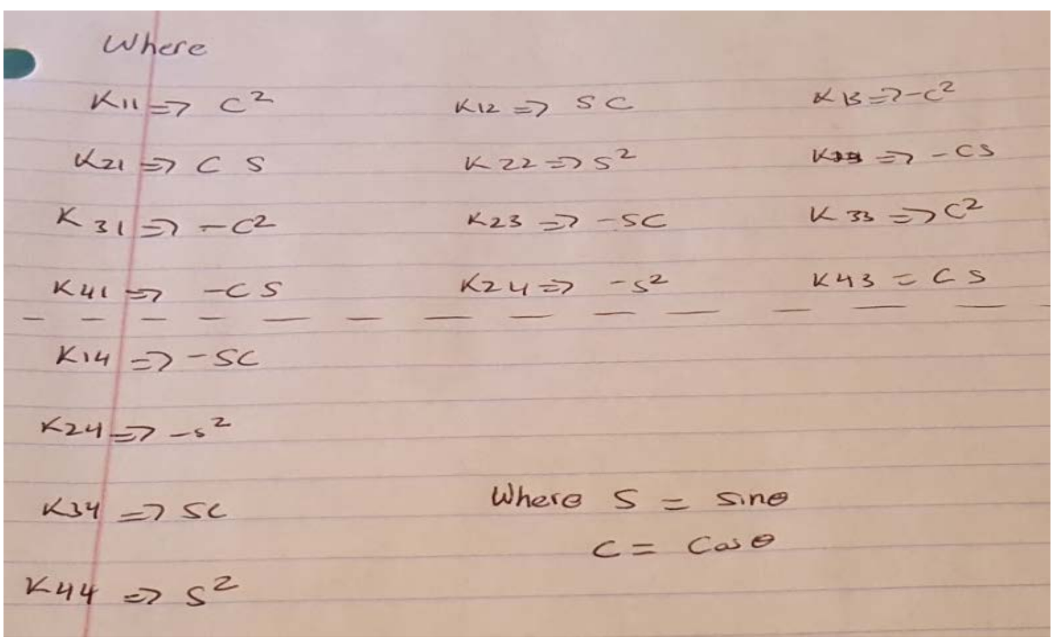


Figure 4 The formulation of local stiffness matrix

The formulations for the local stiffness matrix and element connectivity matrix allowed one to create a banded stiffness matrix. From our element connectivity matrix, we can see that iband is 8 since the maximum the difference between the two nodes is 3, which would mean that our banded stiffness matrix would be something like this:

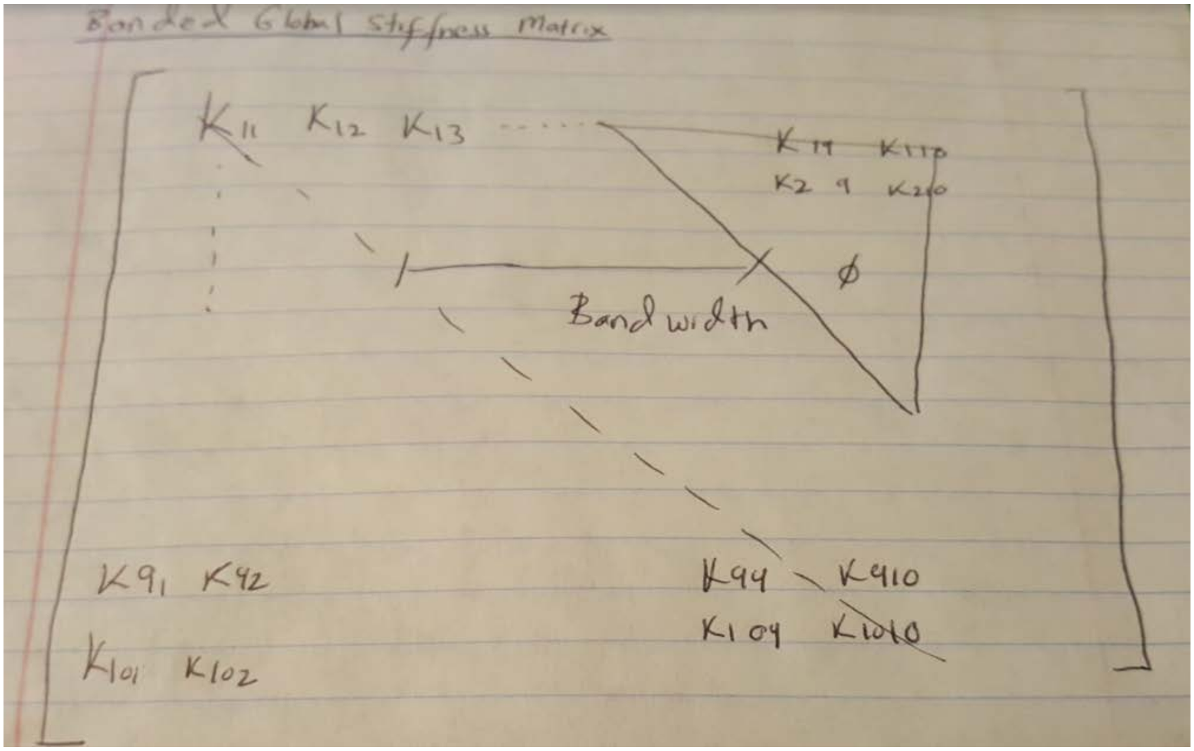


Figure 5 The location of the banded stiffness matrices

So, in summary the FE equations can be re-written as follows:

FE equations in summary

$$[K]\{u_N\} = \{F_N\}$$

$[K]$ - Global Stiffness Matrix

$\{u_N\} \Rightarrow$ Global Displacement Vector

$\{F_N\} \Rightarrow$ Global force vector

$$\begin{Bmatrix} F_x^1 \\ F_y^1 \\ F_x^2 \\ F_y^2 \\ \vdots \\ F_x^5 \\ F_y^5 \end{Bmatrix} \Rightarrow \begin{bmatrix} K_{11} & K_{12} & K_{13} & \cdots & K_{19} & K_{110} \\ K_{21} & K_{22} & K_{23} & & K_{29} & K_{210} \\ \vdots & & & & & \\ K_{91} & K_{92} & & & K_{99} & K_{910} \\ K_{101} & K_{102} & & & K_{109} & K_{1010} \end{bmatrix} \begin{Bmatrix} u_x^1 \\ u_y^1 \\ u_x^2 \\ u_y^2 \\ \vdots \\ u_x^i \\ u_y^i \\ \vdots \end{Bmatrix}$$

Include Knowns and Unknowns

In Clude Knowns and Unknown

Figure 6 Summary of FE equations

3.4 Imposing Boundary Conditions

With the stiffness matrix created, it was now time to impose boundary conditions so that the equations on the previous page can be further reduced. A free body diagram is created for the 82-node truss to see where the forces are acting on the truss. The positions of the load applied were determined by dividing up the truss into 5-unit cells, where each unit cells were composed of 9 nodes. A brief formulation is shown below. The formulation describes on how the positions of the loads were determined on the 82-node truss. A free body diagram is then created to reduce the FE equations given on the previous page.

nucells 4 = 5

nucells 2 => 2 x nucells 4 => 10

nucells => 2 x nucells 2 => 20

n => 2 x nucells = 40

$dx = 2.5 / 40 \Rightarrow \underline{0.0625}$

load-nodal position 1 $\Rightarrow \left(\frac{1}{0.0625} \times 2\right) + 1 = \underline{33}$

where +1 is the location of where the forces are applied. Since the bottom truss is divided into <u>odd nodes</u>, and top part of the truss is divided into <u>even nodes</u>.

Figure 7 A brief formulation identifying the position of the loads applied on the 82-node structure

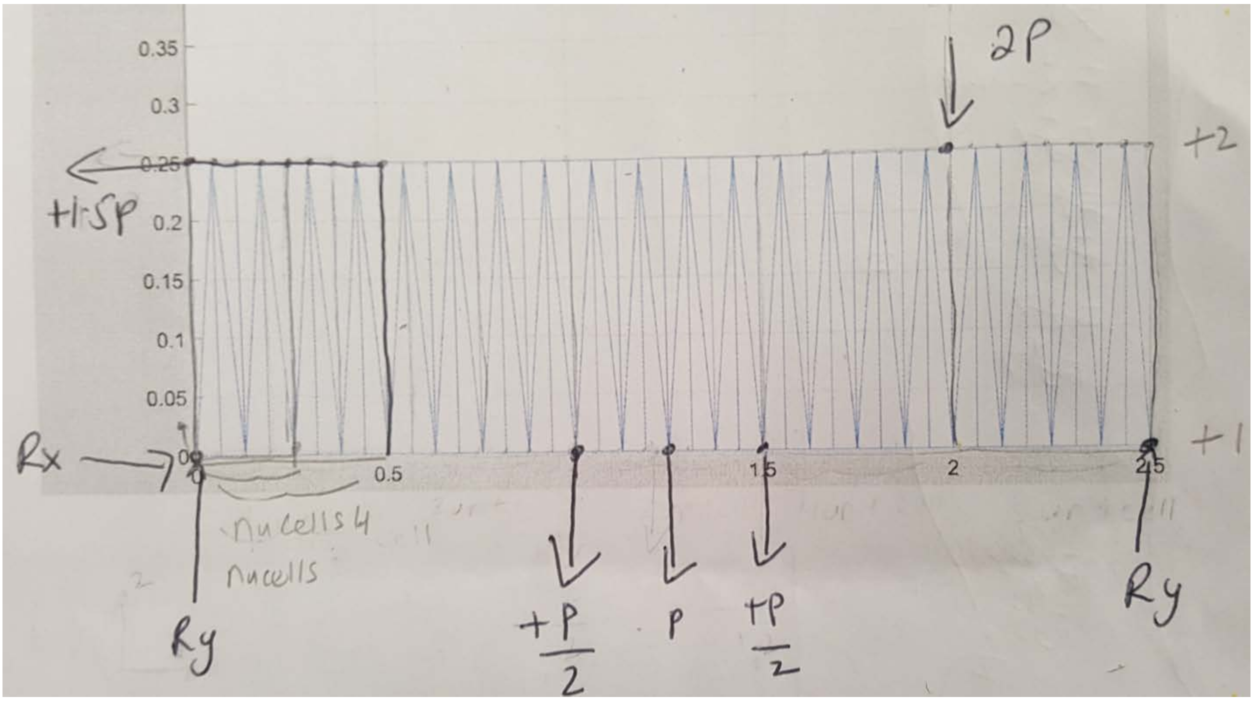


Figure 8 A free body diagram of the 82-node truss, where +1 and +2 are positions of the nodes.

Finally, the FE equations given on page 6 are reduced to the following:

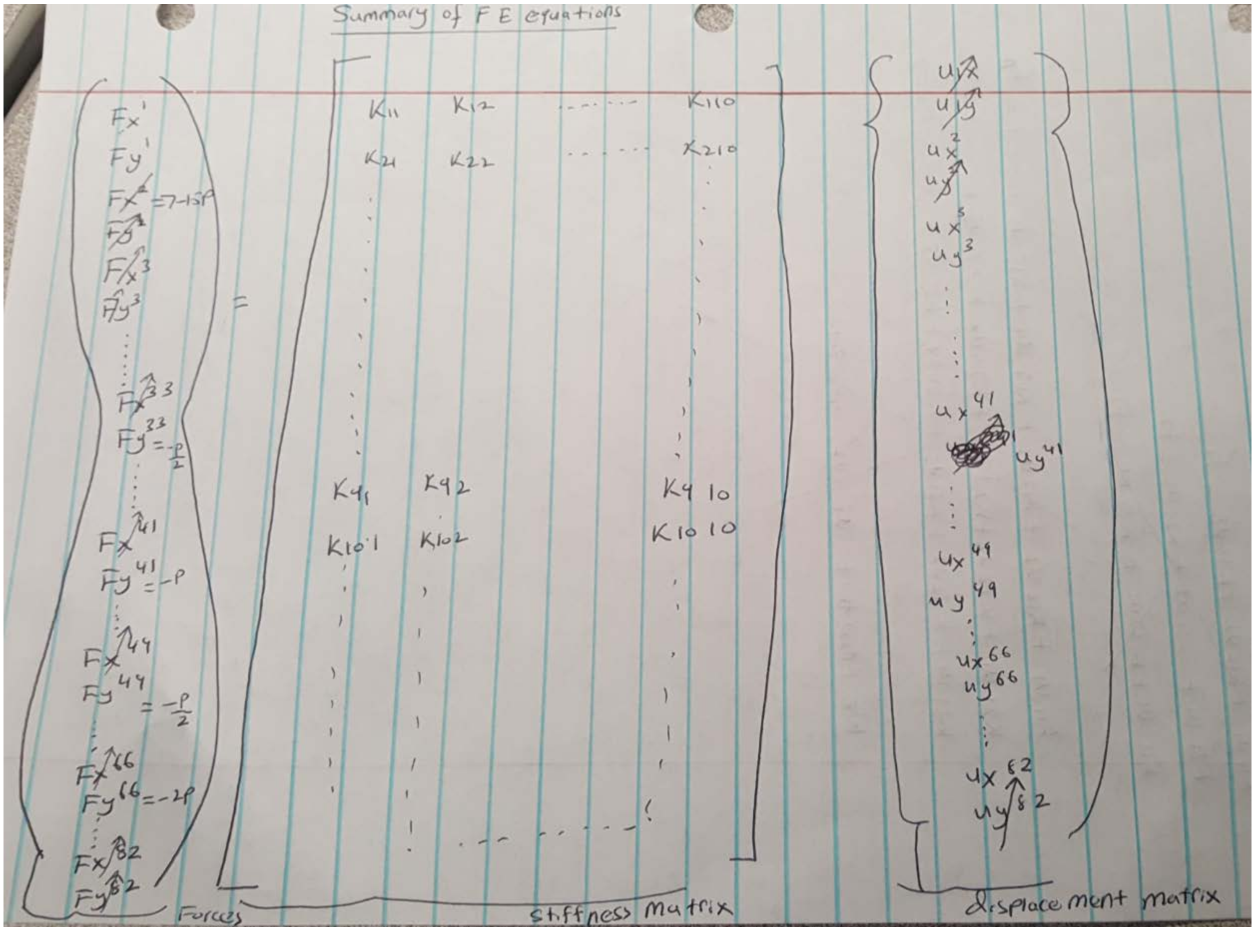


Figure 9 Summary of reduced FE equations

3.5 Non-Dimensionalization of FE Equations

The FE equations given on the figure above were non-dimensionalized to obtain accurate results. Non-dimensionalization allows one to create equations that simplify and parameterize problems where measured units are involved. The process of non-dimensionalization is done from figures 10-12

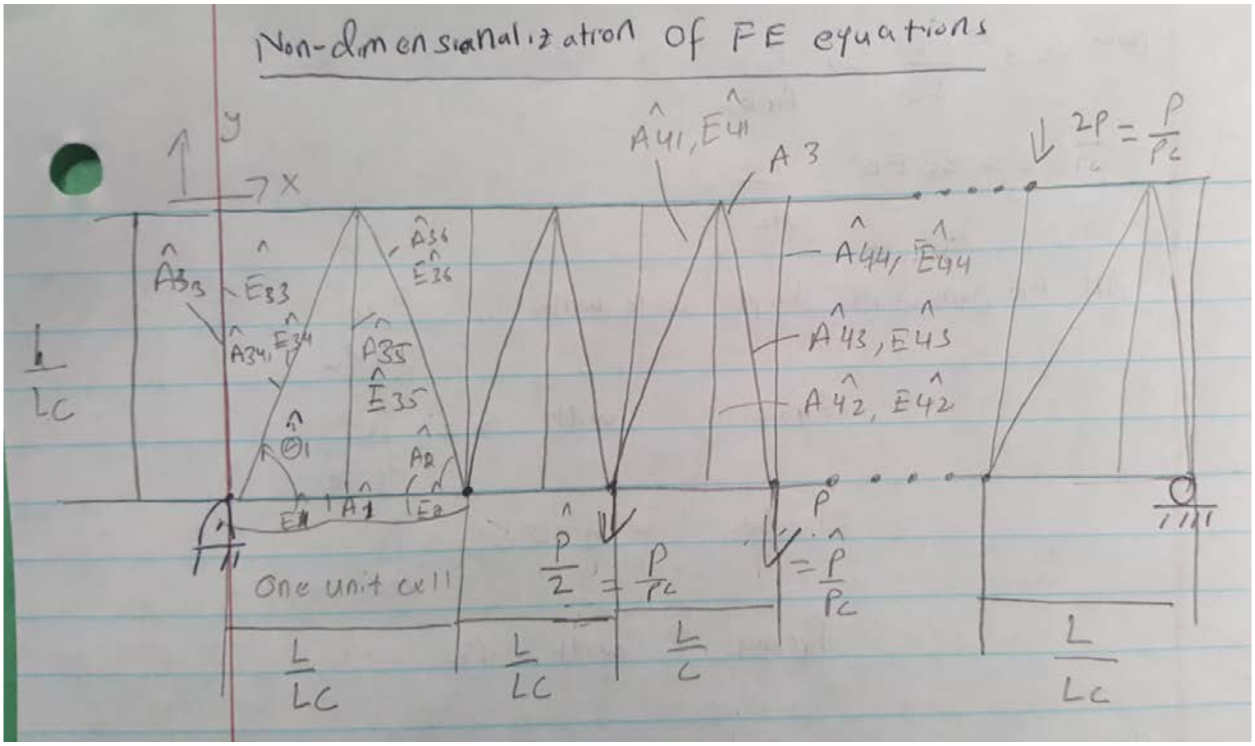


Figure 10 Creation of a n-node Truss

Recall $[K]_{el} = \frac{AE}{L} \overset{4\times4}{\begin{bmatrix} \cos \\ \sin \\ \vdots \end{bmatrix}}$

Let $[K] = \frac{A_c E_c}{L_c} [\hat{K}]$ ← non-dimensional stiffness matrix

FE equations $[K]\{U_N\} = \{F_N\}$

$\Rightarrow \frac{A_c E_c}{L_c} [\hat{K}]\, U_c \{\hat{U}_N\} = P_c \{\hat{F}_N\}$

$\Rightarrow [\hat{K}]\{\hat{u}_N\} \Rightarrow \frac{P_c L_c}{U_c A_c E_c} \{F_N\}$

Now let $\frac{P_c L_c}{U_c A_c E_c} = 1 \Rightarrow U_c = \frac{P_c L_c}{A_c E_c}$

and $\boxed{[\hat{K}]\{\hat{U}_N\} = \{\hat{F}_N\}}$ Non-dimensionalization

Figure 11 Non-dimensionalization of FE of Equations now $\varepsilon_c = \dfrac{U_c}{L_c} \Rightarrow \dfrac{P_c}{A_c E_c}$

$\sigma_c = \varepsilon_c E_c = \dfrac{P_c}{A_c}$

All FE non-dim. displ- scale with $U_c = \dfrac{P_c L_c}{A_c E_c}$

Strains with $\varepsilon_c = \dfrac{P_c}{A_c E_c}$

Stresses with $\sigma_c = \dfrac{P_c}{A_c}$

forces with P_c

Also Note:- In non-dimensional FE model $\hat{A}_i = \dfrac{A_i}{A_c}$; $\hat{E}_i = \dfrac{E_i}{E_c}$

Figure 12 Non-Dimensionalization of stress, strains, and forces

3.6 Unscrambling the FE Equations

Once the non-dimensionalization of FE equations was completed, it was now time to unscramble the FE equations. Figures 13 and 14 describe the process of unscrambling the FE equations

In the x-direction :-

$$\hat K_{11}\hat u_1 + \hat K_{12}\hat u_2 + \hat K_{13}\hat u_3 + \hat K_{14}\hat u_4 = \hat F_{1x}$$
$$\hat K_{21}\hat u_1 + \hat K_{22}\hat u_2 + \hat K_{23}\hat u_3 + \hat K_{24}\hat u_4 = \hat F_{2x}$$
$$\hat K_{31}\hat u_1 + \hat K_{32}\hat u_2 + \hat K_{33}\hat u_3 + \hat K_{34}\hat u_4 = \hat F_{3x}$$

We know that $\hat u_1 \Rightarrow 0$, $F_2 \Rightarrow$ u·k (un known)

$$0\,\hat u_1 + \hat K_{12}\hat u_2 + \hat K_{13}\hat u_3 + \hat K_{14}\hat u_4 = \hat F_{1x} - \hat K_{11}\hat u_1$$
$$1\,\hat u_1 + 0\,\hat u_2 + 0\,\hat u_3 + 0\,\hat u_4 = \hat u_1$$
$$0\,\hat u_1 + \hat K_{32}\hat u_2 + \hat K_{33}\hat u_3 + \hat K_{34}\hat u_{34} = \hat F_{3x} - \hat K_{31}\hat u_1$$

Figure 13 The process of unscrambling FE equations. Here the displacements, forces, and stiffness are all non-dimensional

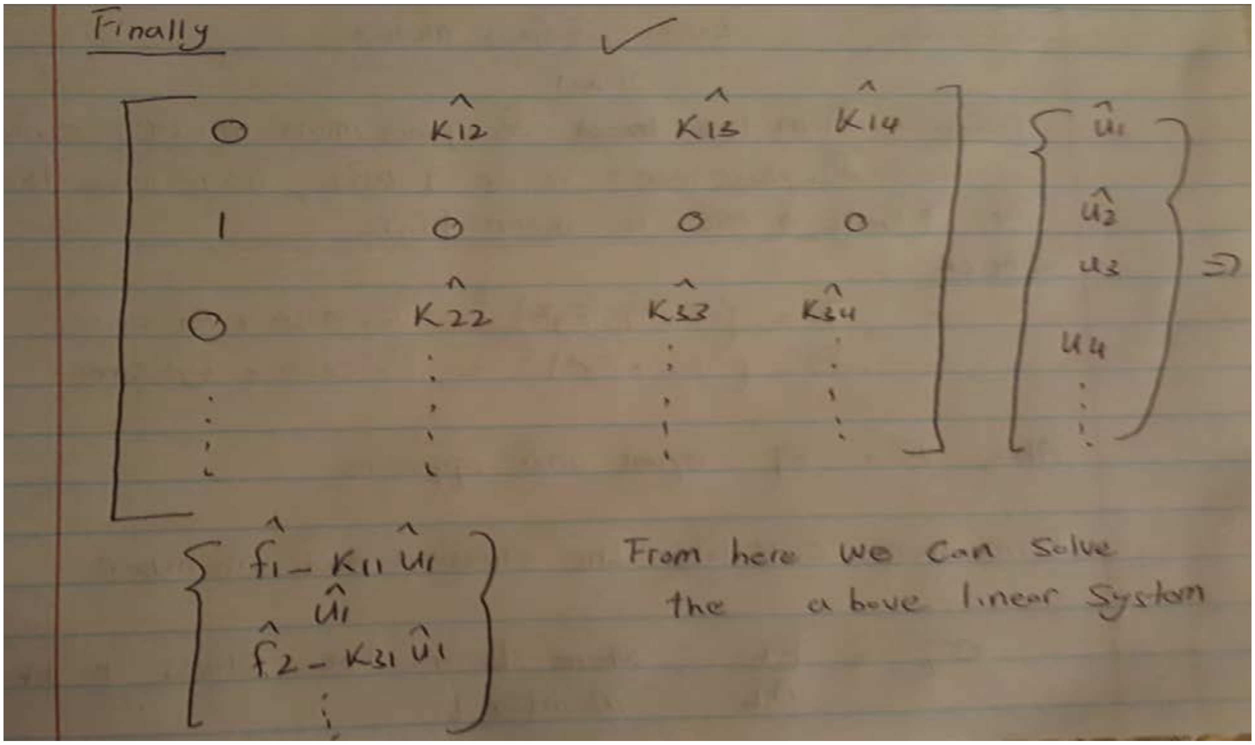


Figure 14 The result obtained. From here we can solve for a linear system of equations

3.7 Solving for The Member Forces, Stresses and Displacements

Once we have unscrambled the FE equations, we can then go ahead and do the post processing. This is the place where we will be solving for the member forces, displacements, etc. Figures 15-16 on the next page illustrate the process of solving for the member forces and displacements.

Solving for the Member forces

Recall

$$\begin{Bmatrix} Fx^1 \\ Fy^1 \\ Fx^2 \\ Fy^2 \\ Fx^3 \\ Fy^3 \\ \vdots \end{Bmatrix} = \frac{AE}{L} \begin{bmatrix} c^2 & sc & -c^2 & sc \\ sc & s^2 & -sc & -s^2 \\ -c^2 & -sc & c^2 & sc \\ -sc & -s^2 & sc & s^2 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{Bmatrix} ux^1 \\ uy^1 \\ ux^2 \\ uy^2 \\ \vdots \end{Bmatrix}$$

Local Forces Local stiffness matrix Local displacement

Global

Process :-- From the ~~[illegible]~~ displacement vector, construct a local displacement vector $\{u_N\}_L$. Then take the product of $\{k_N\}_L \{u_N\}_L$ to obtain $\{F_n\}_L$

Check

$$F_i = (Fx^2 + Fy^2)^{\frac{1}{2}} = Fx^1 \cos\theta + Fy^1 \sin\theta$$

$$F_j = (Fx^2 + Fy^2)^{\frac{1}{2}} = Fx^j \cos\theta + Fy^j \sin\theta$$

Also, $F_i = -F_j$ equal and opposite

From there calculate the stress in each member

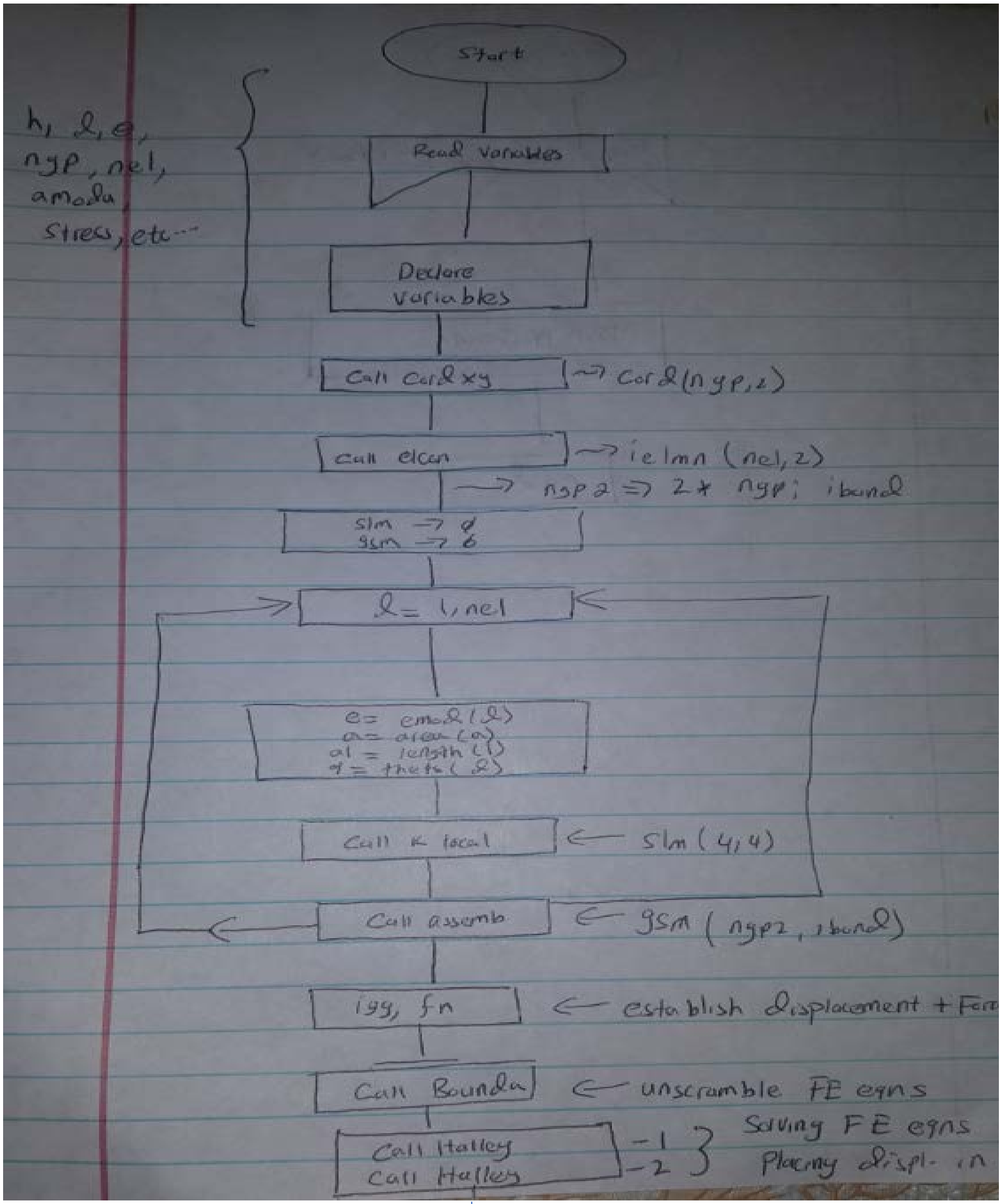
$$\sigma_L = \frac{F_L}{A_L}$$, where L is the stress in the Member L.

Figures 15-16 Illustrating the process of post processing

Results

4.1 Flowchart

Once the formulation was completed, it was now time to create a FORTRAN algorithm that allows the user to solve for an 82-node truss structure. The FORTRAN algorithm followed a series of instructions to print out the results. A programming flowchart on the next page is created to illustrate the sequence of instructions followed by FORTRAN.



Post processing

End

Figure 17 A programming Flowchart showing a series of instructions followed by FORTRAN

The following series of results were obtained, once the program finished running

4.2 Coordinate Matrix:

1	0.00000	0.00000
2	0.00000	0.25000
3	0.06250	0.00000
4	0.06250	0.25000
5	0.12500	0.00000
6	0.12500	0.25000
7	0.18750	0.00000
8	0.18750	0.25000
9	0.25000	0.00000
10	0.25000	0.25000
11	0.31250	0.00000
12	0.31250	0.25000
13	0.37500	0.00000
14	0.37500	0.25000
15	0.43750	0.00000
16	0.43750	0.25000
17	0.50000	0.00000
18	0.50000	0.25000
19	0.56250	0.00000
20	0.56250	0.25000
21	0.62500	0.00000
22	0.62500	0.25000
23	0.68750	0.00000
24	0.68750	0.25000
25	0.75000	0.00000

26	0.75000	0.25000
27	0.81250	0.00000
28	0.81250	0.25000
29	0.87500	0.00000
30	0.87500	0.25000
31	0.93750	0.00000
32	0.93750	0.25000
33	1.00000	0.00000
34	1.00000	0.25000
35	1.06250	0.00000
36	1.06250	0.25000
37	1.12500	0.00000
38	1.12500	0.25000
39	1.18750	0.00000
40	1.18750	0.25000
41	1.25000	0.00000
42	1.25000	0.25000
43	1.31250	0.00000
44	1.31250	0.25000
45	1.37500	0.00000
46	1.37500	0.25000
47	1.43750	0.00000
48	1.43750	0.25000
49	1.50000	0.00000
50	1.50000	0.25000
51	1.56250	0.00000
52	1.56250	0.25000
53	1.62500	0.00000
54	1.62500	0.25000

55	1.68750	0.00000
56	1.68750	0.25000
57	1.75000	0.00000
58	1.75000	0.25000
59	1.81250	0.00000
60	1.81250	0.25000
61	1.87500	0.00000
62	1.87500	0.25000
63	1.93750	0.00000
64	1.93750	0.25000
65	2.00000	0.00000
66	2.00000	0.25000
67	2.06250	0.00000
68	2.06250	0.25000
69	2.12500	0.00000
70	2.12500	0.25000
71	2.18750	0.00000
72	2.18750	0.25000
73	2.25000	0.00000
74	2.25000	0.25000
75	2.31250	0.00000
76	2.31250	0.25000
77	2.37500	0.00000
78	2.37500	0.25000
79	2.43750	0.00000
80	2.43750	0.25000
81	2.50000	0.00000
82	2.50000	0.25000

4.3 Element Connectivity

1	1	3
2	3	5
3	5	7
4	7	9
5	9	11
6	11	13
7	13	15
8	15	17
9	17	19
10	19	21
11	21	23
12	23	25
13	25	27
14	27	29
15	29	31
16	31	33
17	33	35
18	35	37
19	37	39
20	39	41
21	41	43
22	43	45
23	45	47
24	47	49
25	49	51
26	51	53
27	53	55

28	55	57
29	57	59
30	59	61
31	61	63
32	63	65
33	65	67
34	67	69
35	69	71
36	71	73
37	73	75
38	75	77
39	77	79
40	79	81
41	2	4
42	4	6
43	6	8
44	8	10
45	10	12
46	12	14
47	14	16
48	16	18
49	18	20
50	20	22
51	22	24
52	24	26
53	26	28
54	28	30
55	30	32
56	32	34

57	34	36
58	36	38
59	38	40
60	40	42
61	42	44
62	44	46
63	46	48
64	48	50
65	50	52
66	52	54
67	54	56
68	56	58
69	58	60
70	60	62
71	62	64
72	64	66
73	66	68
74	68	70
75	70	72
76	72	74
77	74	76
78	76	78
79	78	80
80	80	82
81	1	2
82	1	4
83	3	4
84	4	5
85	5	6

86	5	8
87	7	8
88	8	9
89	9	10
90	9	12
91	11	12
92	12	13
93	13	14
94	13	16
95	15	16
96	16	17
97	17	18
98	17	20
99	19	20
100	20	21
101	21	22
102	21	24
103	23	24
104	24	25
105	25	26
106	25	28
107	27	28
108	28	29
109	29	30
110	29	32
111	31	32
112	32	33
113	33	34
114	33	36

115	35	36
116	36	37
117	37	38
118	37	40
119	39	40
120	40	41
121	41	42
122	41	44
123	43	44
124	44	45
125	45	46
126	45	48
127	47	48
128	48	49
129	49	50
130	49	52
131	51	52
132	52	53
133	53	54
134	53	56
135	55	56
136	56	57
137	57	58
138	57	60
139	59	60
140	60	61
141	61	62
142	61	64
143	63	64

144	64	65
145	65	66
146	65	68
147	67	68
148	68	69
149	69	70
150	69	72
151	71	72
152	72	73
153	73	74
154	73	76
155	75	76
156	76	77
157	77	78
158	77	80
159	79	80
160	80	81
161	81	82

The element connectivity matrix along with the coordinate matrix allowed one to create a mesh.

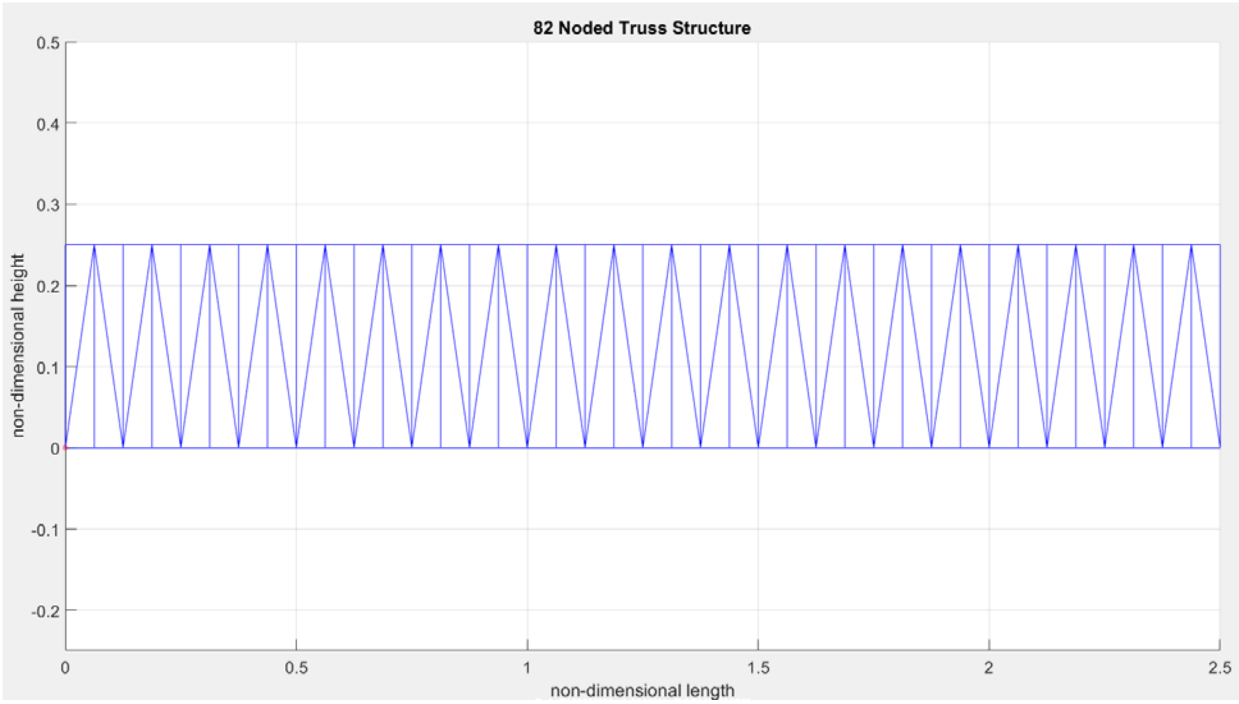


Figure 18 Mesh created through element connectivity and coordinate matrix

4.4 Displacement Matrix

Node 1 u_x= 0.000000 u_y= 0.000000

Node 2 u_x= 6.659939 u_y= -0.000000

Node 3 u_x= -0.069531 u_y= -2.112812

Node 4 u_x= 6.753689 u_y= -2.112812

Node 5 u_x= -0.139063 u_y= -4.260390

Node 6 u_x= 6.799001 u_y= -4.260390

Node 7 u_x= -0.160156 u_y= -6.430624

Node 8 u_x= 6.844314 u_y= -6.430624

Node 9 u_x= -0.181250 u_y= -8.611405

Node 10 u_x= 6.841189 u_y= -8.611405

Node 11 u_x= -0.153906 u_y=-10.790623

Node 12 u_x= 6.838064 u_y=-10.790623

Node 13 u_x= -0.126563 u_y=-12.956170

Node 14 u_x= 6.786501 u_y=-12.956170

Node 15 u_x= -0.050781 u_y=-15.095935

Node 16 u_x= 6.734939 u_y=-15.095935

Node 17 u_x= 0.025000 u_y=-17.197810

Node 18 u_x= 6.634939 u_y=-17.197810

Node 19 u_x= 0.149219 u_y=-19.249684

Node 20 u_x= 6.534939 u_y=-19.249684

Node 21 u_x= 0.273437 u_y=-21.239450

Node 22 u_x= 6.386501 u_y=-21.239450

Node 23 u_x= 0.446094 u_y=-23.154996

Node 24 u_x= 6.238064 u_y=-23.154996

Node 25 u_x= 0.618750 u_y=-24.984215

Node 26 u_x= 6.041189 u_y=-24.984215

Node 27 u_x= 0.839844 u_y=-26.714996

Node 28 u_x= 5.844314 u_y=-26.714996

Node 29 u_x= 1.060937 u_y=-28.335230

Node 30 u_x= 5.599001 u_y=-28.335230

Node 31 u_x= 1.330469 u_y=-29.832807

Node 32 u_x= 5.353689 u_y=-29.832807

Node 33 u_x= 1.600000 u_y=-31.195620

Node 34 u_x= 5.059939 u_y=-31.195620

Node 35 u_x= 1.910156 u_y=-32.274657

Node 36 u_x= 4.766189 u_y=-32.274657

Node 37 u_x= 2.220312 u_y=-33.198616

Node 38 u_x= 4.439626 u_y=-33.198616

Node 39 u_x= 2.563281 u_y=-33.959294

Node 40 u_x= 4.113064 u_y=-33.959294

Node 41 u_x= 2.906250 u_y=-34.548487

Node 42 u_x= 3.753689 u_y=-34.548487

Node 43 u_x= 3.266406 u_y=-34.684193

Node 44 u_x= 3.394314 u_y=-34.684193

Node 45 u_x= 3.626562 u_y=-34.639821

Node 46 u_x= 3.033376 u_y=-34.639821

Node 47 u_x= 3.988281 u_y=-34.414980

Node 48 u_x= 2.672439 u_y=-34.414980

Node 49 u_x= 4.350000 u_y=-34.009280

Node 50 u_x= 2.309939 u_y=-34.009280

Node 51 u_x= 4.705469 u_y=-33.285429

Node 52 u_x= 1.947439 u_y=-33.285429

Node 53 u_x= 5.060937 u_y=-32.383845

Node 54 u_x= 1.599001 u_y=-32.383845

Node 55 u_x= 5.402344 u_y=-31.308041

Node 56 u_x= 1.250564 u_y=-31.308041

Node 57 u_x= 5.743750 u_y=-30.061535

Node 58 u_x= 0.916189 u_y=-30.061535

Node 59 u_x= 6.071094 u_y=-28.647841

Node 60 u_x= 0.581814 u_y=-28.647841

Node 61 u_x= 6.398437 u_y=-27.070475

Node 62 u_x= 0.261501 u_y=-27.070475

Node 63 u_x= 6.711719 u_y=-25.332952

Node 64 u_x= -0.058811 u_y=-25.332952

Node 65 u_x= 7.025000 u_y=-23.438790

Node 66 u_x= -0.365061 u_y=-23.938790

Node 67 u_x= 7.292969 u_y=-20.843902

Node 68 u_x= -0.671311 u_y=-20.843902

Node 69 u_x= 7.560937 u_y=-18.115030

Node 70 u_x= -0.900999 u_y=-18.115030

Node 71 u_x= 7.752344 u_y=-15.271314

Node 72 u_x= -1.130686 u_y=-15.271314

Node 73 u_x= 7.943750 u_y=-12.331895

Node 74 u_x= -1.283811 u_y=-12.331895

Node 75 u_x= 8.058594 u_y= -9.315913

Node 76 u_x= -1.436936 u_y= -9.315913

Node 77 u_x= 8.173437 u_y= -6.242510

Node 78 u_x= -1.513499 u_y= -6.242510

Node 79 u_x= 8.211719 u_y= -3.130825

Node 80 u_x= -1.590061 u_y= -3.130825

Node 81 u_x= 8.250000 u_y= 0.000000

Node 82 u_x= -1.590061 u_y= 0.000000

Based on displacement matrix, we could now create a deformed mesh. A figure below shows the original mesh and the deformed mesh together in a single figure:

4.5 Deformed Mesh and Undeformed Mesh

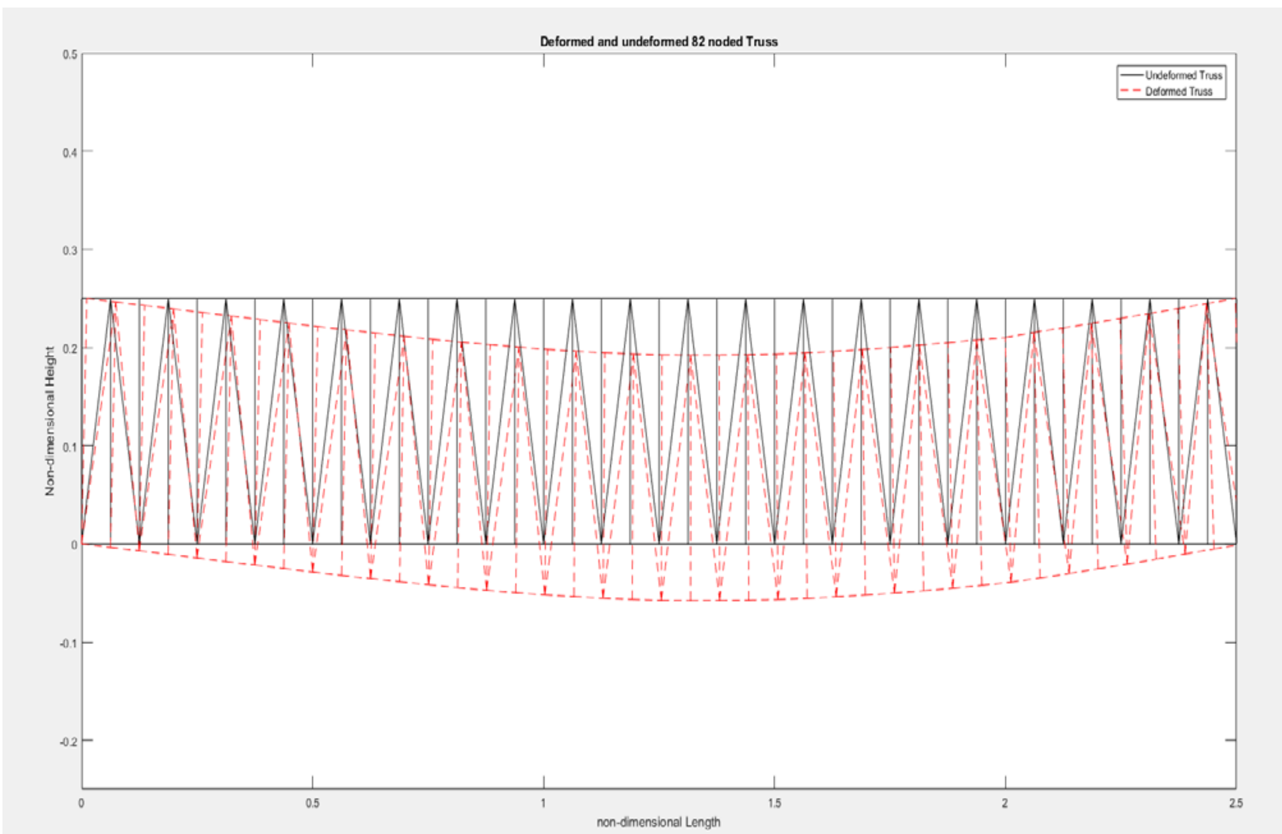


Figure 19 Mesh created through element connectivity and coordinate matrix, and a deformed Mesh created through element connectivity, coordinate matrix, and displacement matrix

4.6 Force Applied Matrix

Constrained Degrees of Freedom 1 2 162

Node 1 F_x= 0.000000 F_y= 0.000000

Node 2 F_x= -1.500000 F_y= 0.000000

Node 3 F_x= 0.000000 F_y= 0.000000

Node 4 F_x= 0.000000 F_y= 0.000000

Node 5 F_x= 0.000000 F_y= 0.000000

Node 6 F_x= 0.000000 F_y= 0.000000

Node 7 F_x= 0.000000 F_y= 0.000000

Node 8 F_x= 0.000000 F_y= 0.000000

Node 9 F_x= 0.000000 F_y= 0.000000

Node 10 F_x= 0.000000 F_y= 0.000000

Node 11 F_x= 0.000000 F_y= 0.000000

Node 12 F_x= 0.000000 F_y= 0.000000

Node 13 F_x= 0.000000 F_y= 0.000000

Node 14 F_x= 0.000000 F_y= 0.000000

Node 15 F_x= 0.000000 F_y= 0.000000

Node 16 F_x= 0.000000 F_y= 0.000000

Node 17 F_x= 0.000000 F_y= 0.000000

Node 18 F_x= 0.000000 F_y= 0.000000

Node 19 F_x= 0.000000 F_y= 0.000000

Node 20 F_x= 0.000000 F_y= 0.000000

Node 21 F_x= 0.000000 F_y= 0.000000

Node 22 F_x= 0.000000 F_y= 0.000000

Node 23 F_x= 0.000000 F_y= 0.000000

Node 24 F_x= 0.000000 F_y= 0.000000

Node 25 F_x= 0.000000 F_y= 0.000000

Node 26 F_x= 0.000000 F_y= 0.000000

Node 27 F_x= 0.000000 F_y= 0.000000

Node 28 F_x= 0.000000 F_y= 0.000000

Node 29 F_x= 0.000000 F_y= 0.000000

Node 30 F_x= 0.000000 F_y= 0.000000

Node 31 F_x= 0.000000 F_y= 0.000000

Node 32 F_x= 0.000000 F_y= 0.000000

Node 33 F_x= 0.000000 F_y= -0.500000

Node 34 F_x= 0.000000 F_y= 0.000000

Node 35 F_x= 0.000000 F_y= 0.000000

Node 36 F_x= 0.000000 F_y= 0.000000

Node 37 F_x= 0.000000 F_y= 0.000000

Node 38 F_x= 0.000000 F_y= 0.000000

Node 39 F_x= 0.000000 F_y= 0.000000

Node 40 F_x= 0.000000 F_y= 0.000000

Node 41 F_x= 0.000000 F_y= -1.000000

Node 42 F_x= 0.000000 F_y= 0.000000

Node 43 F_x= 0.000000 F_y= 0.000000

Node 44 F_x= 0.000000 F_y= 0.000000

Node 45 F_x= 0.000000 F_y= 0.000000

Node 46 F_x= 0.000000 F_y= 0.000000

Node 47 F_x= 0.000000 F_y= 0.000000

Node 48 F_x= 0.000000 F_y= 0.000000

Node 49 F_x= 0.000000 F_y= -0.500000

Node 50 F_x= 0.000000 F_y= 0.000000

Node 51 F_x= 0.000000 F_y= 0.000000

Node 52 F_x= 0.000000 F_y= 0.000000

Node 53 F_x= 0.000000 F_y= 0.000000

Node 54 F_x= 0.000000 F_y= 0.000000

Node 55 F_x= 0.000000 F_y= 0.000000

Node 56 F_x= 0.000000 F_y= 0.000000

Node 57 F_x= 0.000000 F_y= 0.000000

Node 58 F_x= 0.000000 F_y= 0.000000

Node 59 F_x= 0.000000 F_y= 0.000000

Node 60 F_x= 0.000000 F_y= 0.000000

Node 61 F_x= 0.000000 F_y= 0.000000

Node 62 F_x= 0.000000 F_y= 0.000000

Node 63 F_x= 0.000000 F_y= 0.000000

Node 64 F_x= 0.000000 F_y= 0.000000

Node 65 F_x= 0.000000 F_y= 0.000000

Node 66 F_x= 0.000000 F_y= -2.000000

Node 67 F_x= 0.000000 F_y= 0.000000

Node 68 F_x= 0.000000 F_y= 0.000000

Node 69 F_x= 0.000000 F_y= 0.000000

Node 70 F_x= 0.000000 F_y= 0.000000

Node 71 F_x= 0.000000 F_y= 0.000000

Node 72 F_x= 0.000000 F_y= 0.000000

Node 73 F_x= 0.000000 F_y= 0.000000

Node 74 F_x= 0.000000 F_y= 0.000000

Node 75 F_x= 0.000000 F_y= 0.000000

Node 76 F_x= 0.000000 F_y= 0.000000

Node 77 F_x= 0.000000 F_y= 0.000000

Node 78 F_x= 0.000000 F_y= 0.000000

Node 79 F_x= 0.000000 F_y= 0.000000

Node 80 F_x= 0.000000 F_y= 0.000000

Node 81 F_x= 0.000000 F_y= 0.000000

Node 82 F_x= 0.000000 F_y= 0.000000

4.7 El # El length El angle El Axial Force El Axial Stress

1	0.062500	0.000000	1.112500	1.112500
2	0.062500	0.000000	1.112500	1.112500
3	0.062500	0.000000	0.337500	0.337500
4	0.062500	0.000000	0.337500	0.337500
5	0.062500	0.000000	0.437500	0.437500
6	0.062500	0.000000	0.437500	0.437500
7	0.062500	0.000000	1.212500	1.212500
8	0.062500	0.000000	1.212500	1.212500
9	0.062500	0.000000	1.987500	1.987500
10	0.062500	0.000000	1.987500	1.987500
11	0.062500	0.000000	2.762500	2.762500
12	0.062500	0.000000	2.762500	2.762500
13	0.062500	0.000000	3.537500	3.537500
14	0.062500	0.000000	3.537500	3.537500
15	0.062500	0.000000	4.312500	4.312500
16	0.062500	0.000000	4.312500	4.312500
17	0.062500	0.000000	4.962500	4.962500
18	0.062500	0.000000	4.962500	4.962500
19	0.062500	0.000000	5.487500	5.487500
20	0.062500	0.000000	5.487500	5.487500
21	0.062500	0.000000	5.762500	5.762500
22	0.062500	0.000000	5.762500	5.762500
23	0.062500	0.000000	5.787500	5.787500
24	0.062500	0.000000	5.787500	5.787500

25	0.062500	0.000000	5.687500	5.687500
26	0.062500	0.000000	5.687500	5.687500
27	0.062500	0.000000	5.462500	5.462500
28	0.062500	0.000000	5.462500	5.462500
29	0.062500	0.000000	5.237500	5.237500
30	0.062500	0.000000	5.237500	5.237500
31	0.062500	0.000000	5.012500	5.012500
32	0.062500	0.000000	5.012500	5.012500
33	0.062500	0.000000	4.287500	4.287500
34	0.062500	0.000000	4.287500	4.287500
35	0.062500	0.000000	3.062500	3.062500
36	0.062500	0.000000	3.062500	3.062500
37	0.062500	0.000000	1.837500	1.837500
38	0.062500	0.000000	1.837500	1.837500
39	0.062500	0.000000	0.612500	0.612500
40	0.062500	0.000000	0.612500	0.612500
41	0.062500	0.000000	1.500000	1.500000
42	0.062500	0.000000	0.725000	0.725000
43	0.062500	0.000000	0.725000	0.725000
44	0.062500	0.000000	0.050000	0.050000
45	0.062500	0.000000	0.050000	0.050000
46	0.062500	0.000000	0.825000	0.825000
47	0.062500	0.000000	0.825000	0.825000
48	0.062500	0.000000	1.600000	1.600000
49	0.062500	0.000000	1.600000	1.600000
50	0.062500	0.000000	2.375000	2.375000
51	0.062500	0.000000	2.375000	2.375000
52	0.062500	0.000000	3.150000	3.150000
53	0.062500	0.000000	3.150000	3.150000

54	0.062500	0.000000	3.925000	3.925000
55	0.062500	0.000000	3.925000	3.925000
56	0.062500	0.000000	4.700000	4.700000
57	0.062500	0.000000	4.700000	4.700000
58	0.062500	0.000000	5.225000	5.225000
59	0.062500	0.000000	5.225000	5.225000
60	0.062500	0.000000	5.750000	5.750000
61	0.062500	0.000000	5.750000	5.750000
62	0.062500	0.000000	5.775000	5.775000
63	0.062500	0.000000	5.775000	5.775000
64	0.062500	0.000000	5.800000	5.800000
65	0.062500	0.000000	5.800000	5.800000
66	0.062500	0.000000	5.575000	5.575000
67	0.062500	0.000000	5.575000	5.575000
68	0.062500	0.000000	5.350000	5.350000
69	0.062500	0.000000	5.350000	5.350000
70	0.062500	0.000000	5.125000	5.125000
71	0.062500	0.000000	5.125000	5.125000
72	0.062500	0.000000	4.900000	4.900000
73	0.062500	0.000000	4.900000	4.900000
74	0.062500	0.000000	3.675000	3.675000
75	0.062500	0.000000	3.675000	3.675000
76	0.062500	0.000000	2.450000	2.450000
77	0.062500	0.000000	2.450000	2.450000
78	0.062500	0.000000	1.225000	1.225000
79	0.062500	0.000000	1.225000	1.225000
80	0.062500	0.000000	0.000000	0.000000
81	0.250000	90.000000	0.000000	0.000000
82	0.257694	75.963757	1.597703	1.597703

83	0.250000	90.000000	0.000000	0.000000
84	0.257694	-75.963757	1.597703	1.597703
85	0.250000	90.000000	0.000000	0.000000
86	0.257694	75.963757	1.597703	1.597703
87	0.250000	90.000000	0.000000	0.000000
88	0.257694	-75.963757	1.597703	1.597703
89	0.250000	90.000000	0.000000	0.000000
90	0.257694	75.963757	1.597703	1.597703
91	0.250000	90.000000	0.000000	0.000000
92	0.257694	-75.963757	1.597703	1.597703
93	0.250000	90.000000	0.000000	0.000000
94	0.257694	75.963757	1.597703	1.597703
95	0.250000	90.000000	0.000000	0.000000
96	0.257694	-75.963757	1.597703	1.597703
97	0.250000	90.000000	0.000000	0.000000
98	0.257694	75.963757	1.597703	1.597703
99	0.250000	90.000000	0.000000	0.000000
100	0.257694	-75.963757	1.597703	1.597703
101	0.250000	90.000000	0.000000	0.000000
102	0.257694	75.963757	1.597703	1.597703
103	0.250000	90.000000	0.000000	0.000000
104	0.257694	-75.963757	1.597703	1.597703
105	0.250000	90.000000	0.000000	0.000000
106	0.257694	75.963757	1.597703	1.597703
107	0.250000	90.000000	0.000000	0.000000
108	0.257694	-75.963757	1.597703	1.597703
109	0.250000	90.000000	0.000000	0.000000
110	0.257694	75.963757	1.597703	1.597703
111	0.250000	90.000000	0.000000	0.000000

112	0.257694	-75.963757	1.597703	1.597703
113	0.250000	90.000000	0.000000	0.000000
114	0.257694	75.963757	1.082315	1.082315
115	0.250000	90.000000	0.000000	0.000000
116	0.257694	-75.963757	1.082315	1.082315
117	0.250000	90.000000	0.000000	0.000000
118	0.257694	75.963757	1.082315	1.082315
119	0.250000	90.000000	0.000000	0.000000
120	0.257694	-75.963757	1.082315	1.082315
121	0.250000	90.000000	0.000000	0.000000
122	0.257694	75.963757	0.051539	0.051539
123	0.250000	90.000000	0.000000	0.000000
124	0.257694	-75.963757	0.051539	0.051539
125	0.250000	90.000000	0.000000	0.000000
126	0.257694	75.963757	0.051539	0.051539
127	0.250000	90.000000	0.000000	0.000000
128	0.257694	-75.963757	0.051539	0.051539
129	0.250000	90.000000	0.000000	0.000000
130	0.257694	75.963757	0.463849	0.463849
131	0.250000	90.000000	0.000000	0.000000
132	0.257694	-75.963757	0.463849	0.463849
133	0.250000	90.000000	0.000000	0.000000
134	0.257694	75.963757	0.463849	0.463849
135	0.250000	90.000000	0.000000	0.000000
136	0.257694	-75.963757	0.463849	0.463849
137	0.250000	90.000000	0.000000	0.000000
138	0.257694	75.963757	0.463849	0.463849
139	0.250000	90.000000	0.000000	0.000000
140	0.257694	-75.963757	0.463849	0.463849

141	0.250000	90.000000	0.000000	0.000000
142	0.257694	75.963757	0.463849	0.463849
143	0.250000	90.000000	0.000000	0.000000
144	0.257694	-75.963757	0.463849	0.463849
145	0.250000	90.000000	2.000000	2.000000
146	0.257694	75.963757	2.525402	2.525402
147	0.250000	90.000000	0.000000	0.000000
148	0.257694	-75.963757	2.525402	2.525402
149	0.250000	90.000000	0.000000	0.000000
150	0.257694	75.963757	2.525402	2.525402
151	0.250000	90.000000	0.000000	0.000000
152	0.257694	-75.963757	2.525402	2.525402
153	0.250000	90.000000	0.000000	0.000000
154	0.257694	75.963757	2.525402	2.525402
155	0.250000	90.000000	0.000000	0.000000
156	0.257694	-75.963757	2.525402	2.525402
157	0.250000	90.000000	0.000000	0.000000
158	0.257694	75.963757	2.525402	2.525402
159	0.250000	90.000000	0.000000	0.000000
160	0.257694	-75.963757	2.525402	2.525402
161	0.250000	90.000000	0.000000	0.000000

4.8 Typical Local Stiffness Matrix

Local Stiffness Matrix

0.22827	0.91308	-0.22827	-0.91308
0.91308	3.65230	-0.91308	-3.65230
-0.22827	-0.91308	0.22827	0.91308
-0.91308	-3.65230	0.91308	3.65230

4.9 Typical Banded Stiffness Matrix

Banded Stiffness Matrix

16.228	0.913	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
7.652	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
16.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913

11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000

32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000

4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000

32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	-0.000	-0.000	-16.000	0.000	0.000	0.000
4.000	-0.000	-4.000	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.228	0.913	-16.000	0.000	0.000	0.000
11.305	0.913	-3.652	0.000	0.000	0.000	0.000	0.000
32.457	0.000	-0.000	-0.000	-16.000	0.000	-0.228	-0.913
11.305	-0.000	-4.000	0.000	0.000	-0.913	-3.652	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000

0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.228	0.913	0.000	0.000	-16.000	0.000	0.000	0.000
3.652	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	0.000	0.000	-16.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
16.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
16.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

As we can see from the banded stiffness matrix, the iband is 8 as was discussed in page 6 of this report.

Appendix A Fortran Code

```
c234567
      program project1
      implicit real*8(a-h,o-z)
      dimension gsm(164,164),slm(4,4),cord(82,2),ielmn(161,2)
      dimension elength(161),thetal(161),thetald(161)
      dimension gsmb(164,8),fn(164),igg(3)
      dimension ulocal(4),fnlocal(4),an(161),stress(161)
c
      area=1.0d0
      amodu=1.0d0
c
c      node1=0
c      node2=0

      pi=datan(1.0d0)*4.0d0

c
      ell=2.50d0
      h=0.25d0

      x0=0.0d0
      y0=0.0d0
c
c     read(5,601) nucells4
c     601 format(i5)
c
      nucells4=5
      nucells2=2*nucells4
```

```
      nucells=2*nucells2
      n=2*nucells
      n1=n+1
      ngp=2*n1
      nel=4*n+1
      ngp2=2*ngp
      iband=8

c     call cordxy(cord,ngp,el1,el2,el3,h1,h2,h3,th1,th2)
c     call econ(ielmn,nel)
      dx=ell/dfloat(n)
      np1=(1.0d0/dx)*2+1
      np2=(1.25d0/dx)*2+1
      np3=(1.50d0/dx)*2+1
      np4=(2.00d0/dx)*2+2
      np5=(0.0d0/dx)*2+2
c
      write(6,*)np1,np2,np3,np4,np5

      call cordxyp(cord,ngp,x0,y0,ell,h,nucells)
      call econp(ielmn,nel,nucells)
c      call mesh2(ielmn,cord,nel,ngp)
c
       do i=1,ngp2
       do j=1,ngp2
       gsm(i,j)=0.0d0
       enddo
       fn(i)=0.0d0
       enddo
```

```
c
      do i=1,ngp2
      do j=1,iband
      gsmb(i,j)=0.0d0
      enddo
      enddo

      do i=1,4
      do j=1,4
      slm(i,j)=0.0d0
      enddo
      enddo
c
c     l=2*n+2

     call geometry(ielmn,cord,ngp,nel,elength,thetal,thetald)
c

     do l=1,nel
     alength=elength(l)
     theta=thetal(l)
c
c
     call klocal(area,alength,amodu,theta,slm)
     call assemb2(l,ngp2,ielmn,nel,slm,gsm)
     call assemb(ielmn,slm,nel,ngp2,iband,l,gsmb)
     enddo
c
     igg(1)=1
```

```
      igg(2)=2
c     igg(3)=2*(2*n+1)-1
c     igg(3)=0
      igg(3)=2*(2*n+1)
      m12=3
      write(7,400)(igg(i),i=1,m12)
 400  format("Constrained Degrees of Freedom",2x,4(i4,1x))
c



      fn(2*np1)=-0.5d0
      fn(2*np2)=-1.0d0
      fn(2*np3)=-0.5d0
      fn(2*np4)=-2.0d0
      fn(2*np5-1)=-1.5d0
cxc
c
      write(7,300) (i,fn(2*i-1),fn(2*i),i=1,ngp)
 300    format("Node",1x,i3,1x,"F_x=",f10.6,2x,"F_y=",f10.6)
c
      call bounda(m12,ngp2,iband,igg,gsmb,fn)
c
      call halley(1,gsmb,fn,ngp2,iband)
      call halley(2,gsmb,fn,ngp2,iband)
c
        write(7,301)(i,fn(2*i-1),fn(2*i),i=1,ngp)
 301    format("Node",1x,i3,1x,"u_x=",f10.6,2x,"u_y=",f10.6)
c 301     format(1x,i3,1x,f10.6,2x,f10.6)
c
```

```
c     post processing
c
      do l=1,nel
      call localuxy(l,ielmn,fn,nel,ulocal,node1,node2)
      alength=elength(l)
      theta=thetal(l)
      call klocal(area,alength,amodu,theta,slm)
      call mult(fnlocal,slm,ulocal)
      an(l)=dsqrt(fnlocal(1)**2+fnlocal(2)**2)
      stress(l)=an(l)/area
      enddo

      write(7,501)
 501  format(/,"El #",2x,"El length",2x,"El angle",2x,"El Axial Force"
     +,2x,"El Axial Stress",/)
      write(7,500)(i,elength(i),thetald(i),an(i),stress(i),i=1,nel)
 500  format((i3,4(f10.6,2x)))

      stop
      end
c
c*******************************************************
c
      subroutine cordxyp(cord,ngp,x0,y0,ell,h,nucells)
      implicit real*8(a-h,o-z)
      dimension cord(ngp,2)

      n=2*nucells
      n1=n+1
```

```
      dx=ell/dfloat(n)
c

c234567
      do i=1,n1
        cord(2*i-1,1)=(dfloat(i-1)*dx)+x0
        cord(2*i-1,2)=y0

        cord(2*i,1)=cord(2*i-1,1)
        cord(2*i,2)=y0+h
      enddo
c234567
      write(6,100)(i,(cord(i,j),j=1,2),i=1,ngp)
 100    format("Coordinate Matrix",/,((i5,2x,f12.5,2x,f12.5)))
      return
      end
c
c***************************
c
      subroutine econp(ielmn,nel,nucells)
      implicit real*8(a-h,o-z)
      dimension ielmn(nel,2)
c
      n=2*nucells
c234567

      do i=1,n
        ielmn(i,1)=2*i-1
        ielmn(i,2)=2*i+1
```

```
        ielmn(n+i,1)=2*i
        ielmn(n+i,2)=2*i+2
      enddo
c234567
      do j=1,nucells

        jp=4*(j-1)+1
        jp1=2*n+4*j
        ielmn(2*n+4*j-3,1)=jp
        ielmn(2*n+4*j-3,2)=jp+1
        ielmn(2*n+4*j-2,1)=jp
        ielmn(2*n+4*j-2,2)=jp+3
        ielmn(2*n+4*j-1,1)=jp+2
        ielmn(2*n+4*j-1,2)=jp+3
        ielmn(2*n+4*j,1)=jp+3
        ielmn(2*n+4*j,2)=jp+4
      enddo

      ielmn(nel,1)=2*(n+1)-1
      ielmn(nel,2)=2*(n+1)
c234567
      write(6,100)(l,(ielmn(l,j),j=1,2),l=1,nel)
 100    format("Connectivity Matrix",/,((i5,2x,i5,2x,i5)))

       return
       end
c
c*****************************************
c
```

```
      subroutine klocal(area,alength,amodu,theta,slm)
      implicit real*8(a-h,o-z)
      dimension slm(4,4)
c
      pi=datan(1.0d0)*4.0d0
      q=theta
c      q=theta*pi/180.0d0

      factor=amodu*area/alength
      c=dcos(q)
      s=dsin(q)
c
      slm(1,1)=factor*c**2
      slm(1,2)=factor*c*s
      slm(1,3)=-factor*c**2
      slm(1,4)=-factor*c*s
c
      slm(2,2)=factor*s**2
      slm(2,3)=-factor*c*s
      slm(2,4)=-factor*s**2
c
      slm(3,3)=factor*c**2
      slm(3,4)=factor*c*s
c
      slm(4,4)=factor*s**2

      do i=2,4
      do j=1,i-1
      slm(i,j)=slm(j,i)
```

```
      enddo
      enddo
c
      write(6,101)
 101  format(/,"Local Stiffness Matrix",/)
      write(6,100)((slm(i,j),i=1,4),j=1,4)
 100  format(/,4(f12.5,2x))
c
      return
      end
c
c************************************
c
      subroutine assemb2(l,ngp2,ielmn,nel,slm,gsm)
      implicit real*8(a-h,o-z)
      dimension slm(4,4),ielmn(nel,2),gsm(ngp2,ngp2),kk(4)
      do inode=1,2
      kk(2*inode)=ielmn(l,inode)*2
      kk(2*inode-1)=kk(2*inode)-1
      enddo
c
c
      do i=1,4
      do j=1,4
      k1=kk(i)
      k2=kk(j)
      gsm(k1,k2)=gsm(k1,k2)+slm(i,j)
      enddo
      enddo
```

```
c
c
c     write(6,101)
 101    format(/,"Square Stiffness Matrix",/)
c     write(6,100)((gsm(i,j),i=1,ngp2),j=1,ngp2)
 100    format(10(f7.4,1x))
c
c
      return
      end
c
c*****************************************
c
      subroutine geometry(ielmn,cord,ngp,nel,elength,thetal,thetald)
      implicit real*8(a-h,o-z)
      dimension  ielmn(nel,2),cord(ngp,2),elength(nel),thetal(nel)
      dimension thetald(nel)
c
      pi=4.0d0*datan(1.0d0)
c
      do l=1,nel
      jp1=ielmn(l,1)
      jp2=ielmn(l,2)
c
      dx=cord(jp2,1)-cord(jp1,1)
      dy=cord(jp2,2)-cord(jp1,2)
c
      elength(l)=dsqrt(dx**2+dy**2)
      thetal(l)=datan(dy/dx)
```

```
      thetald(l)=thetal(l)*180.0d0/pi
      enddo
      write(6,101)
 101   format(/,"Element Connectivity, Element Length and Element Angle"
     +,/)
      write(6,100)(l,ielmn(l,1),ielmn(l,2),elength(l),thetal(l)
     +,thetald(l),l=1,nel)
 100   format((3(i5,1x),3(f12.5,2x)))
c
      return
      end
c
c
c
c
*********************************************************************
c
c  assembles the banded global stiffness matrix
c
      subroutine assemb(ielmn,slm,nel,ngp2,iband,l,gsmb)
      implicit real*8(a-h,o-z)
      dimension ielmn(nel,2),slm(4,4),gsmb(ngp2,iband),kk(4)
c
      do 10 inode=1,2
      ii=2*inode
      kk(ii)=2*ielmn(l,inode)
      kk(ii-1)=kk(ii)-1
 10    continue
c
```

```
      do 30 i=1,4
      k=kk(i)
      do 30 j=1,4
      if(kk(j).lt.k) go to 30
      lm=kk(j)-k+1
      gsmb(k,lm)=gsmb(k,lm)+slm(i,j)
 30   continue
c
c     write(6,201)
 201  format(/,"Local Stiffness Matrix",/)
c     write(6,200)((slm(i,j),i=1,4),j=1,4)
 200  format(/,4(f12.5,2x))
c
c
c     write(6,101)
 101  format(/,"Banded Stiffness Matrix",/)
c     write(6,100)((gsmb(i,j),j=1,iband),i=1,ngp2)
 100  format(8(f9.3,1x))
      return
      end
c
c
c
c*****************************************************************
c
c  imposing boundary contitions--unscrambling the system of eqns
c
      subroutine bounda (m12,ngp2,iband,igg,gsmb,fn)
      implicit real*8(a-h,o-z)
```

```
      dimension igg(m12),gsmb(ngp2,iband),fn(ngp2)
c
      do 20 i=1,m12
      km=igg(i)
      fn(km)=0.0d0
      gsmb(km,1)=1.0d0
c
      do 20 j=2,iband
      kmj=km-j+1
      if(kmj.le.0) go to 21
      fn(kmj)=fn(kmj)-gsmb(kmj,j)*fn(km)
      gsmb(kmj,j)=0.0d0
c
 21   kmj=km+j-1
      if(kmj.gt.ngp2) go to 20
      fn(kmj)=fn(kmj)-gsmb(km,j)*fn(km)
      gsmb(km,j)=0.0d0
 20   continue
c
      write(6,301)
 301  format(/,"fn",/)
      write(6,300)((fn))
 300  format(/,f12.5,/)

      return
      end
c
c
c***********************************************************
```

```
c234567
      subroutine halley(kkk,ak,q,mdim,ndim)
      implicit real*8(a-h,o-z)
c  symmetric banded matrix equation solver
c
c  kkk=1 triangularizes the banded symmetric stiffness matrix ak(mdim,ndim)
c  kkk=2 solves for right hand side q(mdim), solution returns in q(mdim)
c
      dimension ak(mdim,ndim),q(mdim)
      ner=mdim
      iband=ndim
      nrs=ner-1
      nr=ner
      if (kkk.eq.2) go to 200
      do 120 n=1,nrs
      m=n-1
      mr=min0(iband,nr-m)
      pivot=ak(n,1)
      do 120 l=2,mr
      cp=ak(n,l)/pivot
      i=m+l
      j=0
      do 110 k=l,mr
      j=j+1
 110   ak(i,j)=ak(i,j)-cp*ak(n,k)
 120    ak(n,l)=cp
      go to 400
 200   do 220 n=1,nrs
      m=n-1
```

```
      mr=min0(iband,nr-m)
      cp=q(n)
      q(n)=cp/ak(n,1)
      do 220 l=2,mr
      i=m+l
 220  q(i)=q(i)-ak(n,l)*cp
      q(nr)=q(nr)/ak(nr,1)
      do 320 i=1,nrs
      n=nr-i
      m=n-1
      mr=min0(iband,nr-m)
      do 320 k=2,mr
      l=m+k
c
c  store computed displacements in load vector q
c
 320  q(n)=q(n)-ak(n,k)*q(l)
 400  return
      end
c
c**********************************************************
c234567

      subroutine localuxy(l,ielmn,fn,nel,ulocal,node1,node2)
      implicit real*8(a-h,o-z)
      dimension ulocal(4),ielmn(nel,2),fn(4)
c     do inode=1,2
c        fn(2*inode)=ielmn(l,inode)*2
c        fn(2*inode-1)=fn(2*inode)-1
```

```
c      enddo



       node1=ielmn(l,1)

       node2=ielmn(l,2)



       ulocal(1)=fn(((2*node1)-1))

       ulocal(2)=fn(((2*node1)))

       ulocal(3)=fn(((2*node2-1)))

       ulocal(4)=fn(((2*node2)))



      write(6,301)
 301  format(/,"L",/)
      write(6,300)((l))
 300  format(/,i5,/)

      write(6,201)
 201  format(/,"Node 1",/)
      write(6,200)((node1))
 200  format(/,i5,/)

      write(6,401)
 401   format(/,"Node 2",/)
      write(6,400)((node2))
 400   format(/,i5,/)
```

```
      write(6,101)

 101  format(/,"u Local",/)
      write(6,100)((ulocal(i)),i=1,4)
 100  format(/,4(f12.5,2x),/)
c
      return
      end
c
c*************************************************************
c234567
      subroutine mult(fnlocal,slm,ulocal)
      implicit real*8(a-h,o-z)
      dimension fnlocal(4),ulocal(4),slm(4,4)
c
      do i=1,4
       fnlocal(i)=0.0*d0
      enddo

       do i=1,4
        do m=1,4
          fnlocal(i)=fnlocal(i)+(slm(i,m)*ulocal(m))
        enddo
       enddo

c     write(6,101)
c 101  format(/,"local force vectors",/)
      return
      end
```

Appendix B MATLAB Code for plotting Meshes

Undeformed Mesh

```
%Syed Ali
%ENME 815 HW2 Problem 1 Part b

clear
clc

cord = importdata('hw2cordxy.txt');
ielmn = importdata('hw2ielmnxy.txt');

figure; grid on; hold on;
set(gca,'Fontsize',16)
for i = 1:length(ielmn)
    xx = [cord(ielmn(i,2),2),cord(ielmn(i,3),2)];
    yy = [cord(ielmn(i,2),3),cord(ielmn(i,3),3)];
    plot(xx,yy,'b-');
    xlim([0 2.5]);
    ylim([-0.25 0.5]);
end
title('82 Noded Truss Structure (Undeformed)')
xlabel('non-dimensional length')
ylabel('non-dimensional height')
```

Deformed Mesh

```
%%

cord = importdata('cordxy.txt');
ielmn = importdata('ielmnxy.txt');
disp = importdata('displace.txt');
deformed_cord = zeros(82,2);

for l = 2:3 %Calling Rows 2 and 3 in the Element
Connectivity Matrix
    for m = 1:82 %From nodes 1-82
       deformed_cord(m,(l-1)) = (cord(m,l)) +
((disp(m,l))/600);
    end
```

```
end

figure; grid on;
set(gca,'Fontsize',20);
for i = 1:length(ielmn)
    x = [cord(ielmn(i,2),2),cord(ielmn(i,3),2)];
    y = [cord(ielmn(i,2),3),cord(ielmn(i,3),3)];
    plot(x,y,'k'); hold on;
    xlim([0 2.5]);
    ylim([-.25 .5]);

    xd =
[deformed_cord(ielmn(i,2),1),deformed_cord(ielmn(i,
3),1)];
    yd =
[deformed_cord(ielmn(i,2),2),deformed_cord(ielmn(i,
3),2)];
    plot(xd,yd,'r--'); hold on;
    title('Deformed and undeformed 82 noded Truss')
    xlabel('non-dimensional Length')
    ylabel('Non-dimensional Height')
    legend('Undeformed Truss', 'Deformed Truss')
end
```