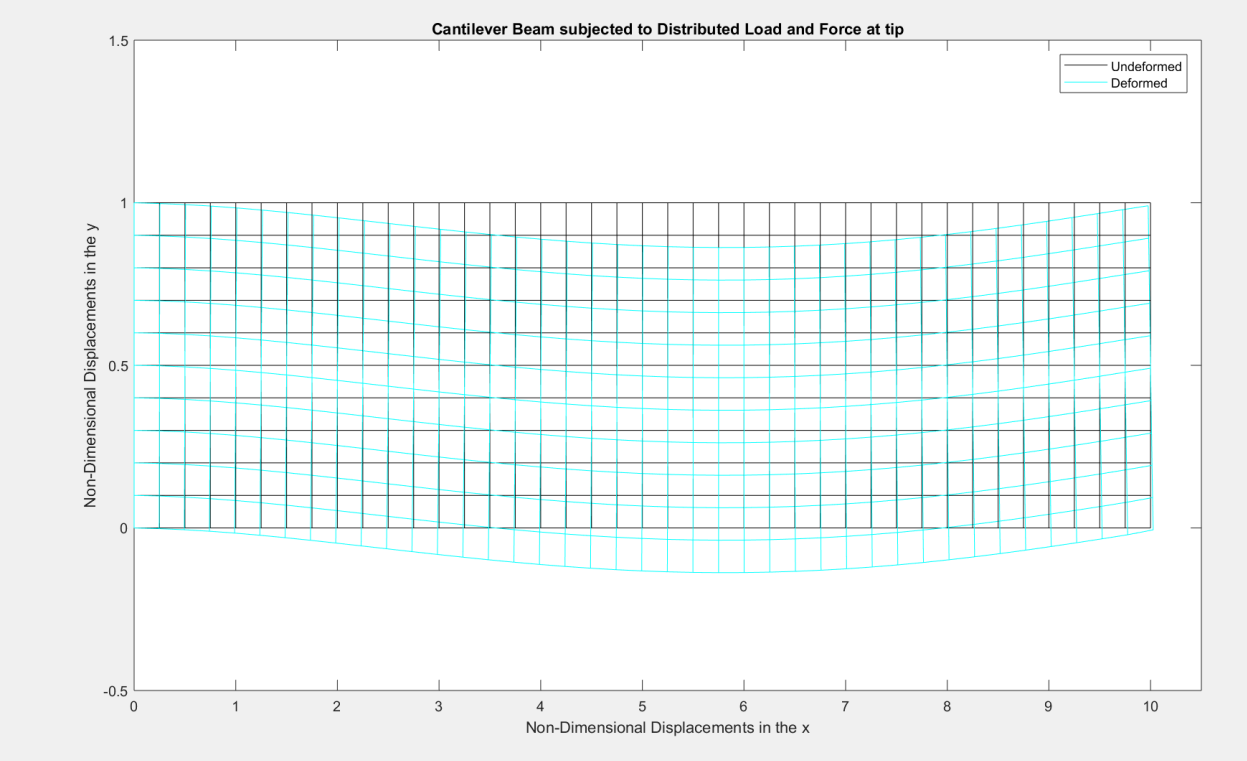
Mesh Generation

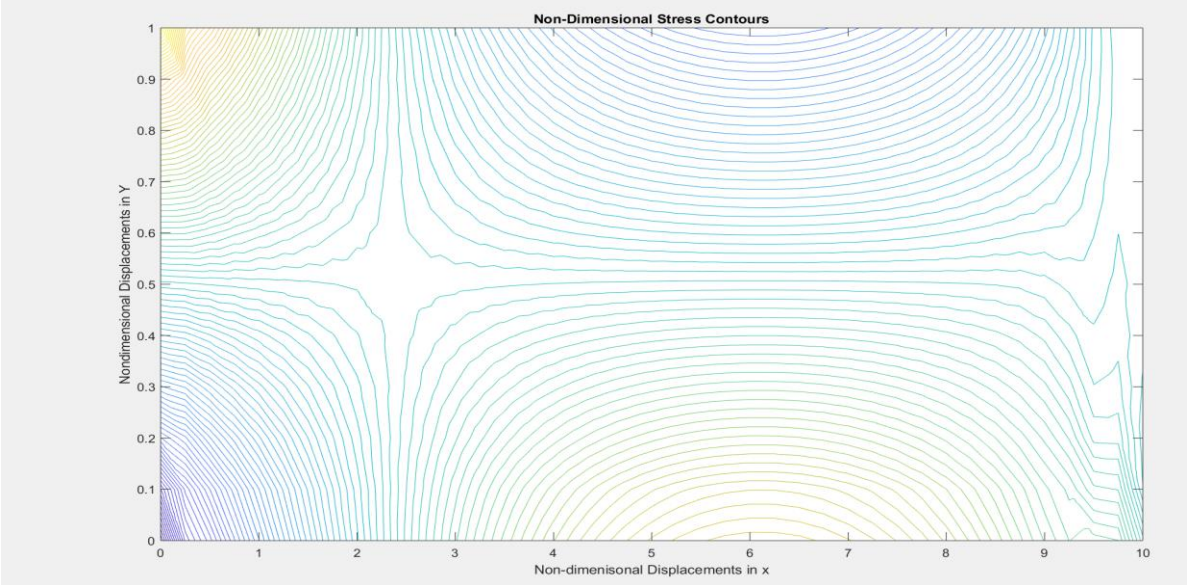
The formulation done in part a allowed one to create a Mesh of N by M elements. The following Mesh was obtained:



From our mesh we can see that the mesh deforms at the center and this is due to the vertical force exerted at the end of the beam.

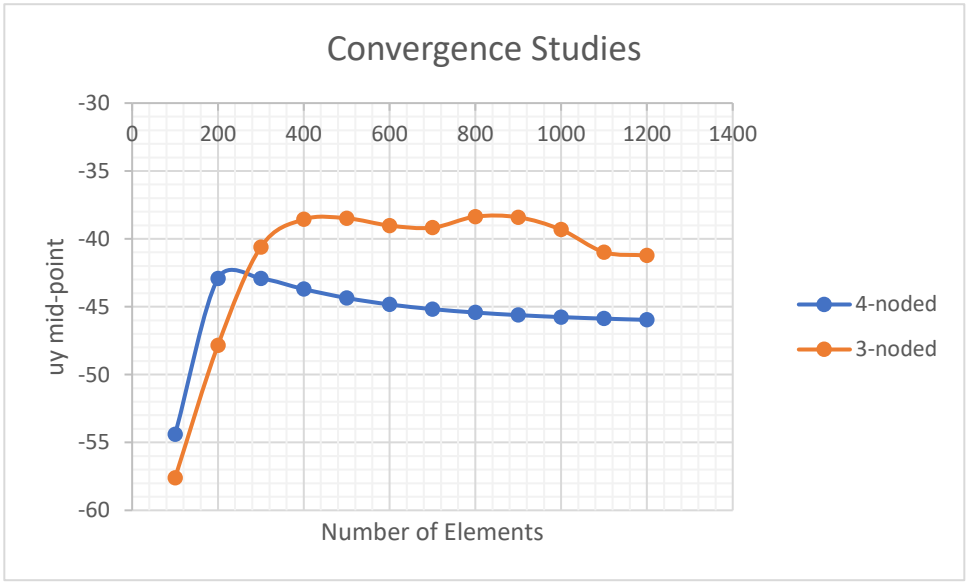
Non-Dimensional Stress Contours

After the mesh was created, the stress contours were then plotted which contain all the three stresses to see the distribution of stresses along the beam. From our graph on the next page we see that the top edge and the bottom edge of the beam are experiencing high stresses, we also observe that the stresses are symmetric and therefore only half of the beam could have been used to do the study. Each node had its own unique value of stress and varied from high low.



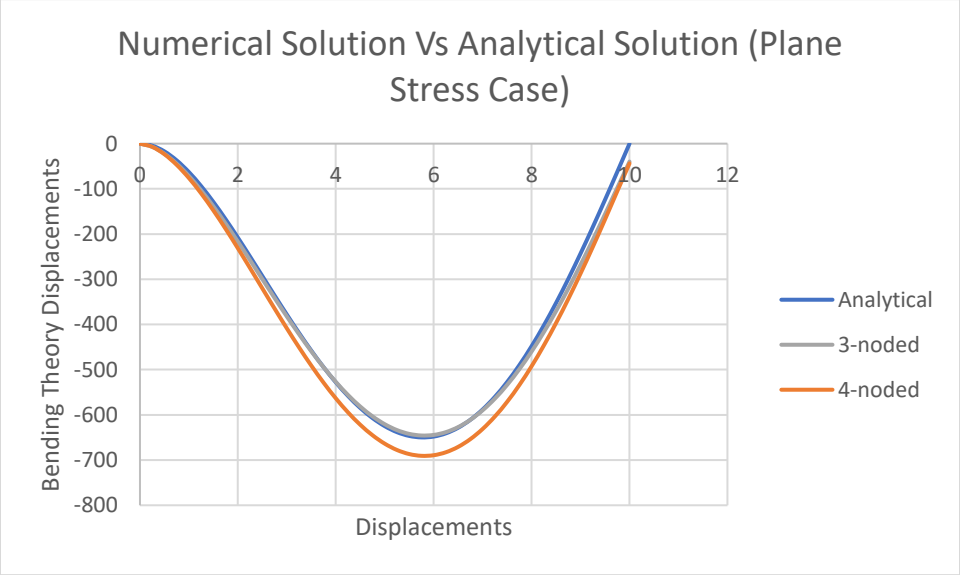
Convergence Studies

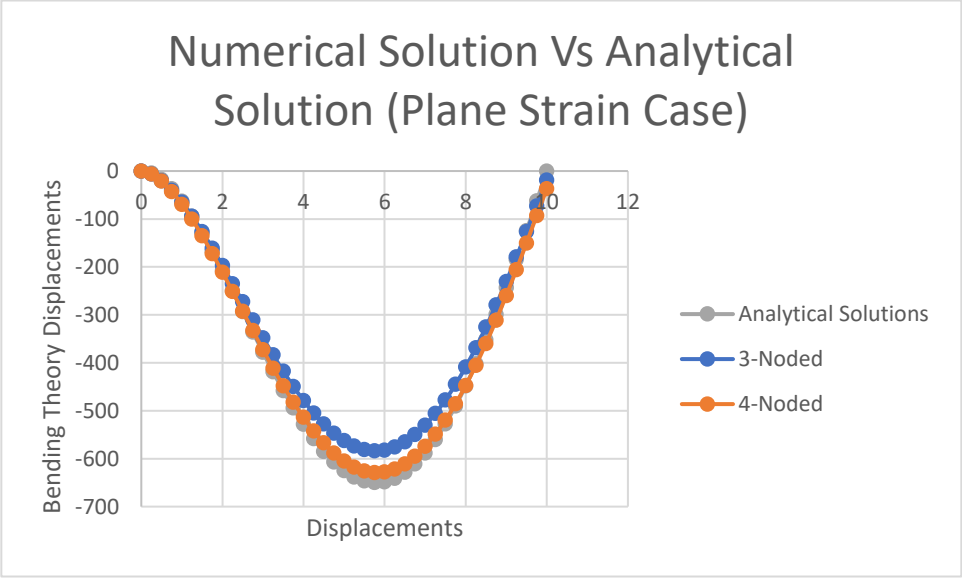
We then created plots for the convergence studies, where the values for n and m were varied to see the number of intervals it takes for the solution to converge. From the graphs we can see that the four noded converges to a value a lot quicker than the 3 noded. More intervals are needed before it converges to a solution.



Numerical Solution Vs Analytical Solution

The displacements plots were then made for the analytical solution, along with the 3-noded and the 4-noded to see the mid-point vertical displacements along the x-axis. One can clearly that the results are for the three cases are quite close. For the Plane Stress case, we notice that the mid-point vertical displacements obtained for the three noded are very close to our analytical solution, however for the plane strain case we notice that the analytical solution is slightly off when compared with the numerical solution.





Results

Please note that this section only contains values for the 50 elements and nodes along with the last 50 fifty element nodes.

4 Noded

Plain Stress case

Elastic matrix D

1.10	0.33	0.00
0.33	1.10	0.00
0.00	0.00	0.38

Displacement Boundary Conditions

node 1 2 3 4 5 6 7 8 9 10 11

ux=0.0 ux=0.0 ux=0.0 ux=0.0 ux=0.0 ux=0.0 ux=0.0 ux=0.0 ux=0.0 ux=0.0 ux=0.0

uy=0.0 uy=0.0 uy=0.0 uy=0.0 uy=0.0 uy=0.0 uy=0.0 uy=0.0 uy=0.0 uy=0.0 uy=0.0

Traction BC

node 11 22 33 44 55 66 77 88 99 110

fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0

fy=-0.13 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25

node 121 132 143 154 165 176 187 198 209 220

fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0

fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25

node 231 242 253 264 275 286 297 308 319 330

fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0

fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25

node 341 352 363 374 385 396 407 418 429 440

fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0

fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25

node 468

fx=0.0

fy= 0.25

d i s p l a c e m e n t s

.....

****

* * * * * * *

* node * ux * uy ** node * ux * uy *

* * * * * * *

****

*	1 *	0.0000 *	0.0000 **	2 *	0.0000 *	0.0000 *	
*	3 *	0.0000 *	0.0000 **	4 *	0.0000 *	0.0000 *	
*	5 *	0.0000 *	0.0000 **	6 *	0.0000 *	0.0000 *	
*	7 *	0.0000 *	0.0000 **	8 *	0.0000 *	0.0000 *	
*	9 *	0.0000 *	0.0000 **	10 *	0.0000 *	0.0000 *	
*	11 *	0.0000 *	0.0000 **	12 *	-18.1088 *	-11.7137 *	
*	13 *	-12.9548 *	-9.6403 **	14 *	-9.1536 *	-8.3530 *	
*	15 *	-5.8927 *	-7.5897 **	16 *	-2.8918 *	-7.1951 *	
*	17 *	-0.0007 *	-7.0922 **	18 *	2.8924 *	-7.2587 *	
*	19 *	5.8998 *	-7.7205 **	20 *	9.1751 *	-8.5589 *	
*	21 *	13.0085 *	-9.9357 **	22 *	18.2507 *	-12.1221 *	
*	23 *	-32.6494 *	-27.6873 **	24 *	-25.2122 *	-26.2597 *	
*	25 *	-18.2519 *	-25.0346 **	26 *	-11.8547 *	-24.1632 *	
*	27 *	-5.8235 *	-23.6659 **	28 *	0.0343 *	-23.5341 *	
*	29 *	5.8953 *	-23.7632 **	30 *	11.9370 *	-24.3595 *	
*	31 *	18.3528 *	-25.3322 **	32 *	25.3406 *	-26.6591 *	
*	33 *	32.7954 *	-28.1794 **	34 *	-45.4483 *	-50.4999 *	
*	35 *	-35.4501 *	-49.2102 **	36 *	-26.0579 *	-48.2283 *	
*	37 *	-17.0885 *	-47.5213 **	38 *	-8.4255 *	-47.1091 *	
*	39 *	0.0798 *	-47.0051 **	40 *	8.5874 *	-47.2134 *	
*	41 *	17.2568 *	-47.7295 **	42 *	26.2352 *	-48.5386 *	
*	43 *	35.6345 *	-49.6200 **	44 *	45.6369 *	-51.0087 *	
*	45 *	-56.2682 *	-79.0728 **	46 *	-44.1430 *	-78.0119 *	
*	47 *	-32.5809 *	-77.1844 **	48 *	-21.4470 *	-76.6064 *	
*	49 *	-10.5923 *	-76.2779 **	50 *	0.1228 *	-76.2021 *	

*	399 *	67.5907 *	-286.7345 **	400 *	45.3676 *	-287.0159 *	
*	401 *	23.2865 *	-287.1819 **	402 *	1.2865 *	-287.2386 *	
*	403 *	-20.6982 *	-287.1963 **	404 *	-42.7436 *	-287.0636 *	

* 405 *	-64.9346 *	-286.8439 **	406 *	-87.3606 *	-286.5334 *
* 407 *	-110.1104 *	-286.1198 **	408 *	117.1268 *	-227.5266 *
* 409 *	93.4970 *	-227.9197 **	410 *	70.1562 *	-228.2470 *
* 411 *	46.9869 *	-228.4342 **	412 *	23.9968 *	-228.4992 *
* 413 *	1.1519 *	-228.4770 **	414 *	-21.6227 *	-228.3939 *
* 415 *	-44.4257 *	-228.2634 **	416 *	-67.3692 *	-228.0897 *
* 417 *	-90.5757 *	-227.8690 **	418 *	-114.1737 *	-227.5871 *
* 419 *	120.7876 *	-167.0985 **	420 *	96.4385 *	-167.6035 *
* 421 *	71.9864 *	-167.7340 **	422 *	47.9015 *	-167.7066 *
* 423 *	24.2121 *	-167.6457 **	424 *	0.8089 *	-167.5936 *
* 425 *	-22.4493 *	-167.5517 **	426 *	-45.7039 *	-167.5064 *
* 427 *	-69.0897 *	-167.4422 **	428 *	-92.7422 *	-167.3498 *
* 429 *	-116.8231 *	-167.2313 **	430 *	124.8858 *	-105.0429 *
* 431 *	98.1056 *	-104.5309 **	432 *	72.5919 *	-104.3750 *
* 433 *	48.0425 *	-104.5333 **	434 *	24.1088 *	-104.8576 *
* 435 *	0.5234 *	-105.2302 **	436 *	-22.9157 *	-105.5780 *
* 437 *	-46.3665 *	-105.8611 **	438 *	-69.9549 *	-106.0622 *
* 439 *	-93.7795 *	-106.1807 **	440 *	-117.9128 *	-106.2376 *
* 441 *	129.0066 *	-29.5763 **	442 *	99.8884 *	-34.1306 *
* 443 *	73.6755 *	-37.6685 **	444 *	48.8214 *	-40.3377 *
* 445 *	24.6794 *	-42.3039 **	446 *	0.9065 *	-43.7046 *
* 447 *	-22.7134 *	-44.6501 **	448 *	-46.3334 *	-45.2359 *
* 449 *	-70.0639 *	-45.5554 **	450 *	-93.9699 *	-45.7088 *
* 451 *	-118.0355 *	-45.8052 **			

****

s t r e s s e s

.....

element	sx	sy	sxy
1	-56.3409	-0.5499	0.1284
1	-60.2874	-13.7046	-11.3163
1	-73.3670	-17.6285	-13.1580
1	-69.4206	-4.4737	-1.7133
2	-40.4195	-1.9729	-1.7388
2	-42.8698	-10.1405	-10.1797
2	-52.5166	-13.0345	-11.3231
2	-50.0663	-4.8669	-2.8822
3	-26.9466	-2.0642	-2.0331
3	-28.3994	-6.9068	-9.2742
3	-36.6749	-9.3895	-9.9522
3	-35.2221	-4.5468	-2.7111
4	-14.4729	-1.2298	-2.0947
4	-15.2239	-3.7333	-8.7585
4	-22.8397	-6.0180	-9.1090
4	-22.0886	-3.5145	-2.4452
5	-2.4210	0.0854	-2.1747
5	-2.6169	-0.5676	-8.5946
5	-9.9540	-2.7687	-8.6860
5	-9.7581	-2.1157	-2.2661
6	9.5933	1.5644	-2.3375
6	9.9103	2.6211	-8.7617
6	2.5683	0.4185	-8.6137
6	2.2513	-0.6382	-2.1895

*	7	*	21.9391 *	2.9399 *	-2.6047 *	
*	7	*	22.8180 *	5.8696 *	-9.2831 *	
*	7	*	15.1856 *	3.5799 *	-8.8729 *	
*	7	*	14.3067 *	0.6502 *	-2.1946 *	
*	8	*	35.1081 *	3.9203 *	-2.9597 *	
*	8	*	36.7038 *	9.2394 *	-10.2328 *	
*	8	*	28.3917 *	6.7458 *	-9.4881 *	
*	8	*	26.7960 *	1.4266 *	-2.2151 *	
*	9	*	50.0397 *	4.1535 *	-3.2100 *	
*	9	*	52.6603 *	12.8886 *	-11.7223 *	
*	9	*	42.9319 *	9.9701 *	-10.4994 *	
*	9	*	40.3114 *	1.2350 *	-1.9871 *	
*	10	*	69.6688 *	3.6572 *	-2.0369 *	
*	10	*	73.8302 *	17.5287 *	-13.6777 *	
*	10	*	60.5265 *	13.5376 *	-11.7357 *	
*	10	*	56.3651 *	-0.3340 *	-0.0949 *	
*	11	*	-50.8432 *	0.3877 *	1.3905 *	
*	11	*	-49.6140 *	4.4851 *	-3.6797 *	
*	11	*	-55.4086 *	2.7467 *	-3.1061 *	
*	11	*	-56.6378 *	-1.3506 *	1.9641 *	
*	12	*	-38.8447 *	0.7288 *	-1.4409 *	
*	12	*	-38.7262 *	1.1240 *	-8.4561 *	
*	12	*	-46.7434 *	-1.2812 *	-8.4007 *	
*	12	*	-46.8620 *	-1.6763 *	-1.3856 *	
*	13	*	-26.3223 *	0.5890 *	-3.4776 *	
*	13	*	-26.5281 *	-0.0971 *	-10.4419 *	
*	13	*	-34.4873 *	-2.4848 *	-10.5380 *	
*	13	*	-34.2815 *	-1.7988 *	-3.5736 *	
*	14	*	-14.1333 *	0.5163 *	-4.6390 *	

*	14	*	-14.3289 *	-0.1356 *	-11.3680 *
*	14	*	-22.0192 *	-2.4427 *	-11.4593 *
*	14	*	-21.8236 *	-1.7908 *	-4.7303 *
*	15	*	-2.1878 *	0.6003 *	-5.1861 *
*	15	*	-2.2427 *	0.4174 *	-11.7737 *
*	15	*	-9.7714 *	-1.8413 *	-11.7994 *
*	15	*	-9.7165 *	-1.6584 *	-5.2117 *
*	16	*	9.7311 *	0.7604 *	-5.2411 *
*	16	*	9.8502 *	1.1573 *	-11.8318 *
*	16	*	2.3180 *	-1.1024 *	-11.7762 *
*	16	*	2.1989 *	-1.4993 *	-5.1855 *
*	17	*	21.8464 *	0.8756 *	-4.7838 *
*	17	*	22.1024 *	1.7289 *	-11.5214 *
*	17	*	14.4023 *	-0.5811 *	-11.4019 *
*	17	*	14.1463 *	-1.4344 *	-4.6644 *
*	18	*	34.3112 *	0.8505 *	-3.6375 *
*	18	*	34.5668 *	1.7024 *	-10.6115 *
*	18	*	26.5966 *	-0.6887 *	-10.4922 *
*	18	*	26.3410 *	-1.5406 *	-3.5183 *
*	19	*	46.8679 *	0.6858 *	-1.4323 *
*	19	*	46.7729 *	0.3694 *	-8.4369 *
*	19	*	38.7676 *	-2.0322 *	-8.4812 *
*	19	*	38.8625 *	-1.7158 *	-1.4766 *
*	20	*	56.4015 *	0.3098 *	1.9538 *
*	20	*	55.1337 *	-3.9161 *	-2.9594 *
*	20	*	49.5186 *	-5.6007 *	-3.5510 *
*	20	*	50.7864 *	-1.3747 *	1.3622 *
*	21	*	-43.0327 *	0.2785 *	1.1097 *
*	21	*	-42.7702 *	1.1534 *	-4.5774 *

*	21	*	-49.2697 *	-0.7964 *	-4.4549 *
*	21	*	-49.5322 *	-1.6714 *	1.2322 *
*	22	*	-33.1646 *	0.3837 *	-1.4564 *
*	22	*	-32.7018 *	1.9263 *	-6.8564 *
*	22	*	-38.8733 *	0.0749 *	-6.6404 *
*	22	*	-39.3360 *	-1.4677 *	-1.2404 *
*	23	*	-22.9496 *	0.5322 *	-3.4750 *
*	23	*	-22.6366 *	1.5756 *	-9.1870 *
*	23	*	-29.1646 *	-0.3828 *	-9.0409 *
*	23	*	-29.4776 *	-1.4261 *	-3.3290 *
*	24	*	-12.4635 *	0.5635 *	-4.8586 *
*	24	*	-12.3016 *	1.1032 *	-10.7027 *
*	24	*	-18.9805 *	-0.9005 *	-10.6271 *
*	24	*	-19.1424 *	-1.4401 *	-4.7831 *
*	25	*	-1.8970 *	0.5293 *	-5.5394 *
*	25	*	-1.8441 *	0.7056 *	-11.4185 *
*	25	*	-8.5630 *	-1.3101 *	-11.3938 *
*	25	*	-8.6159 *	-1.4864 *	-5.5148 *
*	26	*	8.6736 *	0.4748 *	-5.5137 *
*	26	*	8.6340 *	0.3429 *	-11.3904 *
*	26	*	1.9177 *	-1.6720 *	-11.4089 *
*	26	*	1.9573 *	-1.5401 *	-5.5321 *
*	27	*	19.1859 *	0.4258 *	-4.7718 *
*	27	*	19.0333 *	-0.0830 *	-10.6070 *
*	27	*	12.3644 *	-2.0837 *	-10.6783 *
*	27	*	12.5170 *	-1.5749 *	-4.8430 *
*	28	*	29.4859 *	0.4087 *	-3.3060 *
*	28	*	29.1747 *	-0.6287 *	-8.9962 *
*	28	*	22.6716 *	-2.5796 *	-9.1414 *

*	28	*	22.9828 *	-1.5422 *	-3.4513 *
*	29	*	39.2715 *	0.4486 *	-1.2132 *
*	29	*	38.8042 *	-1.1090 *	-6.5682 *
*	29	*	32.6843 *	-2.9450 *	-6.7863 *
*	29	*	33.1516 *	-1.3873 *	-1.4313 *
*	30	*	49.4097 *	0.6578 *	1.2353 *
*	30	*	49.1592 *	-0.1772 *	-4.4219 *
*	30	*	42.6938 *	-2.1168 *	-4.5388 *
*	30	*	42.9443 *	-1.2818 *	1.1184 *
*	31	*	-36.5297 *	0.1338 *	0.6707 *
*	31	*	-36.0943 *	1.5852 *	-4.0522 *
*	31	*	-41.4918 *	-0.0341 *	-3.8490 *
*	31	*	-41.9272 *	-1.4854 *	0.8739 *
*	32	*	-27.8530 *	0.2447 *	-1.7917 *
*	32	*	-27.5589 *	1.2251 *	-6.6102 *
*	32	*	-33.0658 *	-0.4270 *	-6.4730 *
*	32	*	-33.3599 *	-1.4073 *	-1.6544 *
*	33	*	-19.1736 *	0.3010 *	-3.6409 *
*	33	*	-18.9283 *	1.1187 *	-8.4472 *
*	33	*	-24.4213 *	-0.5291 *	-8.3327 *
*	33	*	-24.6666 *	-1.3469 *	-3.5264 *
*	34	*	-10.4190 *	0.3363 *	-4.8802 *
*	34	*	-10.2596 *	0.8679 *	-9.7472 *
*	34	*	-15.8218 *	-0.8008 *	-9.6727 *
*	34	*	-15.9813 *	-1.3323 *	-4.8058 *
*	35	*	-1.5943 *	0.3386 *	-5.4936 *
*	35	*	-1.5406 *	0.5178 *	-10.4006 *
*	35	*	-7.1487 *	-1.1646 *	-10.3756 *
*	35	*	-7.2024 *	-1.3438 *	-5.4685 *

*	36	*	7.2354 *	0.3226 *	-5.4623 *
*	36	*	7.1786 *	0.1333 *	-10.3661 *
*	36	*	1.5743 *	-1.5480 *	-10.3926 *
*	36	*	1.6311 *	-1.3587 *	-5.4888 *
*	37	*	15.9994 *	0.3127 *	-4.7901 *
*	37	*	15.8369 *	-0.2290 *	-9.6472 *
*	37	*	10.2860 *	-1.8943 *	-9.7231 *
*	37	*	10.4485 *	-1.3526 *	-4.8660 *
*	38	*	24.6609 *	0.3311 *	-3.5063 *
*	38	*	24.4137 *	-0.4929 *	-8.2978 *
*	38	*	18.9378 *	-2.1357 *	-8.4131 *
*	38	*	19.1850 *	-1.3116 *	-3.6217 *
*	39	*	33.3331 *	0.3972 *	-1.6382 *
*	39	*	33.0407 *	-0.5772 *	-6.4446 *
*	39	*	27.5477 *	-2.2251 *	-6.5811 *
*	39	*	27.8400 *	-1.2507 *	-1.7746 *
*	40	*	41.8826 *	0.4787 *	0.8806 *
*	40	*	41.4479 *	-0.9701 *	-3.8323 *
*	40	*	36.0618 *	-2.5859 *	-4.0351 *
*	40	*	36.4964 *	-1.1371 *	0.6777 *
*	41	*	-30.0686 *	0.0163 *	0.3230 *
*	41	*	-29.6891 *	1.2812 *	-3.5595 *
*	41	*	-34.1262 *	-0.0499 *	-3.3824 *
*	41	*	-34.5057 *	-1.3149 *	0.5001 *
*	42	*	-22.9716 *	0.1127 *	-2.0312 *
*	42	*	-22.6651 *	1.1343 *	-5.9660 *
*	42	*	-27.1621 *	-0.2148 *	-5.8230 *
*	42	*	-27.4686 *	-1.2364 *	-1.8882 *
*	43	*	-15.7876 *	0.1670 *	-3.7889 *

*	43	*	-15.5759 *	0.8728 *	-7.7799 *
*	43	*	-20.1370 *	-0.4955 *	-7.6810 *
*	43	*	-20.3487 *	-1.2014 *	-3.6901 *
*	44	*	-8.5629 *	0.1862 *	-4.9514 *
*	44	*	-8.4350 *	0.6125 *	-8.9634 *
*	44	*	-13.0202 *	-0.7631 *	-8.9037 *
*	44	*	-13.1481 *	-1.1894 *	-4.8917 *
*	45	*	-1.3177 *	0.1876 *	-5.5203 *
*	45	*	-1.2757 *	0.3278 *	-9.5447 *
*	45	*	-5.8751 *	-1.0520 *	-9.5251 *
*	45	*	-5.9171 *	-1.1923 *	-5.5006 *
*	46	*	5.9278 *	0.1816 *	-5.4975 *
*	46	*	5.8824 *	0.0302 *	-9.5205 *
*	46	*	1.2846 *	-1.3492 *	-9.5417 *
*	46	*	1.3301 *	-1.1977 *	-5.5187 *
*	47	*	13.1531 *	0.1799 *	-4.8848 *
*	47	*	13.0227 *	-0.2551 *	-8.8931 *
*	47	*	8.4418 *	-1.6293 *	-8.9540 *
*	47	*	8.5723 *	-1.1944 *	-4.9457 *
*	48	*	20.3470 *	0.1943 *	-3.6818 *
*	48	*	20.1339 *	-0.5159 *	-7.6685 *
*	48	*	15.5777 *	-1.8828 *	-7.7679 *
*	48	*	15.7908 *	-1.1725 *	-3.7813 *
*	49	*	27.4601 *	0.2321 *	-1.8810 *
*	49	*	27.1531 *	-0.7913 *	-5.8118 *
*	49	*	22.6608 *	-2.1390 *	-5.9551 *
*	49	*	22.9679 *	-1.1156 *	-2.0243 *
*	50	*	34.4952 *	0.3130 *	0.5031 *
*	50	*	34.1169 *	-0.9483 *	-3.3786 *

*	50	*	29.6807 *	-2.2792 *	-3.5552 *
*	50	*	30.0591 *	-1.0179 *	0.3265 *
*	350	*	-23.9838 *	-1.7791 *	-0.7579 *
*	350	*	-23.8051 *	-1.1834 *	2.0876 *
*	350	*	-20.5532 *	-0.2078 *	2.1710 *
*	350	*	-20.7319 *	-0.8035 *	-0.6745 *
*	351	*	17.7162 *	-0.1412 *	-0.4010 *
*	351	*	17.5403 *	-0.7276 *	1.9891 *
*	351	*	20.2719 *	0.0919 *	1.9070 *
*	351	*	20.4478 *	0.6783 *	-0.4831 *
*	352	*	13.4108 *	-0.1128 *	0.7493 *
*	352	*	13.2397 *	-0.6832 *	3.1454 *
*	352	*	15.9781 *	0.1383 *	3.0656 *
*	352	*	16.1492 *	0.7087 *	0.6695 *
*	353	*	9.1293 *	-0.2213 *	1.6756 *
*	353	*	8.9975 *	-0.6606 *	4.0245 *
*	353	*	11.6819 *	0.1448 *	3.9630 *
*	353	*	11.8136 *	0.5840 *	1.6141 *
*	354	*	4.9015 *	-0.3108 *	2.3060 *
*	354	*	4.7930 *	-0.6725 *	4.6336 *
*	354	*	7.4531 *	0.1255 *	4.5830 *
*	354	*	7.5616 *	0.4872 *	2.2554 *
*	355	*	0.7165 *	-0.4419 *	2.6756 *
*	355	*	0.6355 *	-0.7117 *	4.9701 *
*	355	*	3.2578 *	0.0749 *	4.9323 *
*	355	*	3.3387 *	0.3447 *	2.6379 *
*	356	*	-3.4289 *	-0.6484 *	2.7604 *
*	356	*	-3.4678 *	-0.7781 *	5.0199 *
*	356	*	-0.8855 *	-0.0034 *	5.0017 *

*	356	*	-0.8466 *	0.1263 *	2.7422 *
*	357	*	-7.5580 *	-0.9217 *	2.5169 *
*	357	*	-7.5412 *	-0.8657 *	4.7602 *
*	357	*	-4.9774 *	-0.0966 *	4.7680 *
*	357	*	-4.9942 *	-0.1526 *	2.5248 *
*	358	*	-11.7175 *	-1.2288 *	1.9093 *
*	358	*	-11.6364 *	-0.9586 *	4.1712 *
*	358	*	-9.0514 *	-0.1831 *	4.2090 *
*	358	*	-9.1324 *	-0.4533 *	1.9471 *
*	359	*	-15.9681 *	-1.5220 *	0.9208 *
*	359	*	-15.8214 *	-1.0329 *	3.2448 *
*	359	*	-13.1654 *	-0.2361 *	3.3133 *
*	359	*	-13.3121 *	-0.7252 *	0.9892 *
*	360	*	-20.3747 *	-1.7479 *	-0.4417 *
*	360	*	-20.1688 *	-1.0616 *	1.9897 *
*	360	*	-17.3902 *	-0.2280 *	2.0858 *
*	360	*	-17.5961 *	-0.9143 *	-0.3456 *
*	361	*	14.7143 *	0.2019 *	-0.2953 *
*	361	*	14.4613 *	-0.6415 *	1.7663 *
*	361	*	16.8174 *	0.0654 *	1.6482 *
*	361	*	17.0704 *	0.9088 *	-0.4134 *
*	362	*	10.9889 *	-0.1181 *	0.9197 *
*	362	*	10.8607 *	-0.5455 *	2.9220 *
*	362	*	13.1490 *	0.1410 *	2.8622 *
*	362	*	13.2772 *	0.5684 *	0.8599 *
*	363	*	7.3141 *	0.1231 *	1.7514 *
*	363	*	7.1349 *	-0.4743 *	3.8524 *
*	363	*	9.5361 *	0.2461 *	3.7688 *
*	363	*	9.7154 *	0.8435 *	1.6677 *

*	364	*	3.6817 *		0.2408 *		2.5636 *	
*	364	*	3.4893 *		-0.4004 *		4.5823 *	
*	364	*	5.7964 *		0.2917 *		4.4925 *	
*	364	*	5.9888 *		0.9330 *		2.4738 *	
*	365	*	0.2110 *		0.1182 *		3.1985 *	
*	365	*	0.0609 *		-0.3821 *		5.0746 *	
*	365	*	2.2051 *		0.2612 *		5.0046 *	
*	365	*	2.3552 *		0.7615 *		3.1284 *	
*	366	*	-3.0844 *		-0.1799 *		3.4995 *	
*	366	*	-3.1621 *		-0.4389 *		5.2534 *	
*	366	*	-1.1576 *		0.1625 *		5.2172 *	
*	366	*	-1.0799 *		0.4214 *		3.4632 *	
*	367	*	-6.2584 *		-0.5682 *		3.3746 *	
*	367	*	-6.2541 *		-0.5540 *		5.0568 *	
*	367	*	-4.3316 *		0.0228 *		5.0588 *	
*	367	*	-4.3359 *		0.0086 *		3.3766 *	
*	368	*	-9.3979 *		-0.9849 *		2.7734 *	
*	368	*	-9.3105 *		-0.6937 *		4.4445 *	
*	368	*	-7.4007 *		-0.1207 *		4.4853 *	
*	368	*	-7.4880 *		-0.4119 *		2.8142 *	
*	369	*	-12.6173 *		-1.3885 *		1.6609 *	
*	369	*	-12.4462 *		-0.8183 *		3.3940 *	
*	369	*	-10.4655 *		-0.2240 *		3.4738 *	
*	369	*	-10.6366 *		-0.7943 *		1.7407 *	
*	370	*	-16.0521 *		-1.7188 *		0.0209 *	
*	370	*	-15.8013 *		-0.8829 *		1.9045 *	
*	370	*	-13.6486 *		-0.2371 *		2.0215 *	
*	370	*	-13.8994 *		-1.0730 *		0.1379 *	
*	371	*	12.0110 *		-1.2105 *		-0.2351 *	

*	371	*	12.2240	*	-0.5007	*	1.3621	*						
*	371	*	14.0493	*	0.0469	*	1.4615	*						
*	371	*	13.8364	*	-0.6629	*	-0.1357	*						
*	372	*	8.5098	*	0.8326	*	-0.1105	*						
*	372	*	8.1352	*	-0.4159	*	2.3576	*						
*	372	*	10.9558	*	0.4302	*	2.1828	*						
*	372	*	11.3303	*	1.6788	*	-0.2853	*						
*	373	*	4.8113	*	1.2634	*	1.4674	*						
*	373	*	4.4029	*	-0.0978	*	3.5004	*						
*	373	*	6.7263	*	0.5992	*	3.3098	*						
*	373	*	7.1347	*	1.9605	*	1.2768	*						
*	374	*	1.7091	*	0.8558	*	3.0350	*						
*	374	*	1.4694	*	0.0568	*	4.5878	*						
*	374	*	3.2440	*	0.5892	*	4.4760	*						
*	374	*	3.4837	*	1.3882	*	2.9231	*						
*	375	*	-0.8378	*	0.2067	*	4.0988	*						
*	375	*	-0.8948	*	0.0164	*	5.3383	*						
*	375	*	0.5218	*	0.4414	*	5.3117	*						
*	375	*	0.5789	*	0.6317	*	4.0722	*						
*	376	*	-3.0173	*	-0.3994	*	4.5550	*						
*	376	*	-2.9387	*	-0.1373	*	5.6291	*						
*	376	*	-1.7112	*	0.2310	*	5.6658	*						
*	376	*	-1.7898	*	-0.0311	*	4.5917	*						
*	377	*	-4.9905	*	-0.8642	*	4.4263	*						
*	377	*	-4.8284	*	-0.3239	*	5.4292	*						
*	377	*	-3.6822	*	0.0200	*	5.5049	*						
*	377	*	-3.8443	*	-0.5204	*	4.5019	*						
*	378	*	-6.8639	*	-1.1857	*	3.7535	*						
*	378	*	-6.6553	*	-0.4906	*	4.7356	*						

*	378	*	-5.5330 *	-0.1539 *	4.8329 *
*	378	*	-5.7415 *	-0.8490 *	3.8508 *
*	379	*	-8.7150 *	-1.4197 *	2.5394 *
*	379	*	-8.4708 *	-0.6056 *	3.5299 *
*	379	*	-7.3388 *	-0.2660 *	3.6438 *
*	379	*	-7.5830 *	-1.0801 *	2.6533 *
*	380	*	-10.6924 *	-1.6771 *	0.6820 *
*	380	*	-10.3816 *	-0.6411 *	1.7541 *
*	380	*	-9.1563 *	-0.2735 *	1.8992 *
*	380	*	-9.4671 *	-1.3095 *	0.8271 *
*	381	*	10.5655 *	6.1405 *	-4.3207 *
*	381	*	8.6297 *	-0.3121 *	1.0778 *
*	381	*	14.7995 *	1.5389 *	0.1745 *
*	381	*	16.7353 *	7.9915 *	-5.2241 *
*	382	*	3.9625 *	2.1429 *	0.1156 *
*	382	*	3.4174 *	0.3260 *	2.4727 *
*	382	*	6.1113 *	1.1342 *	2.2183 *
*	382	*	6.6564 *	2.9511 *	-0.1388 *
*	383	*	0.6586 *	-0.9936 *	3.2067 *
*	383	*	1.0121 *	0.1846 *	4.2384 *
*	383	*	2.1911 *	0.5383 *	4.4033 *
*	383	*	1.8377 *	-0.6399 *	3.3717 *
*	384	*	-1.0274 *	-2.7365 *	4.8683 *
*	384	*	-0.2944 *	-0.2930 *	5.4107 *
*	384	*	0.3255 *	-0.1070 *	5.7527 *
*	384	*	-0.4076 *	-2.5505 *	5.2104 *
*	385	*	-2.0183 *	-3.4345 *	5.5167 *
*	385	*	-1.2097 *	-0.7393 *	5.9215 *
*	385	*	-0.7471 *	-0.6005 *	6.2988 *

*	385	*	-1.5557	*	-3.2957	*	5.8941	*
*	386	*	-2.7570	*	-3.4814	*	5.4673	*
*	386	*	-2.0155	*	-1.0095	*	5.8690	*
*	386	*	-1.5564	*	-0.8718	*	6.2150	*
*	386	*	-2.2980	*	-3.3437	*	5.8134	*
*	387	*	-3.4345	*	-3.1672	*	4.9099	*
*	387	*	-2.8095	*	-1.0838	*	5.3455	*
*	387	*	-2.3117	*	-0.9345	*	5.6372	*
*	387	*	-2.9367	*	-3.0179	*	5.2016	*
*	388	*	-4.0932	*	-2.6788	*	3.9569	*
*	388	*	-3.5881	*	-0.9953	*	4.4069	*
*	388	*	-3.0738	*	-0.8410	*	4.6426	*
*	388	*	-3.5788	*	-2.5245	*	4.1926	*
*	389	*	-4.6432	*	-2.1320	*	2.6818	*
*	389	*	-4.2419	*	-0.7944	*	3.0639	*
*	389	*	-3.8053	*	-0.6634	*	3.2511	*
*	389	*	-4.2066	*	-2.0010	*	2.8691	*
*	390	*	-4.8066	*	-1.6403	*	1.1158	*
*	390	*	-4.4726	*	-0.5270	*	1.2322	*
*	390	*	-4.3395	*	-0.4871	*	1.3881	*
*	390	*	-4.6735	*	-1.6004	*	1.2716	*
*	391	*	-1.4763	*	-35.2797	*	-0.1373	*
*	391	*	8.1667	*	-3.1363	*	5.0542	*
*	391	*	14.0999	*	-1.3563	*	9.5543	*
*	391	*	4.4569	*	-33.4997	*	4.3628	*
*	392	*	-3.6774	*	-28.6759	*	3.5758	*
*	392	*	3.3532	*	-5.2406	*	5.1286	*
*	392	*	5.1278	*	-4.7082	*	8.4095	*
*	392	*	-1.9028	*	-28.1436	*	6.8568	*

*	393	*	-3.3436	*	-22.3891	*	4.2338 *
*	393	*	1.4354	*	-6.4590	*	4.9104 *
*	393	*	2.2086	*	-6.2271	*	7.1406 *
*	393	*	-2.5704	*	-22.1571	*	6.4640 *
*	394	*	-2.6367	*	-16.9832	*	4.0858 *
*	394	*	0.4886	*	-6.5655	*	4.5484 *
*	394	*	1.0174	*	-6.4069	*	6.0069 *
*	394	*	-2.1080	*	-16.8246	*	5.5442 *
*	395	*	-2.0435	*	-12.4477	*	3.7073 *
*	395	*	-0.0868	*	-5.9253	*	4.1236 *
*	395	*	0.3889	*	-5.7826	*	5.0367 *
*	395	*	-1.5678	*	-12.3050	*	4.6205 *
*	396	*	-1.6436	*	-8.6848	*	3.2306 *
*	396	*	-0.5059	*	-4.8925	*	3.6321 *
*	396	*	-0.0470	*	-4.7549	*	4.1630 *
*	396	*	-1.1847	*	-8.5472	*	3.7615 *
*	397	*	-1.4178	*	-5.6437	*	2.6595 *
*	397	*	-0.8416	*	-3.7231	*	3.0352 *
*	397	*	-0.4123	*	-3.5942	*	3.3041 *
*	397	*	-0.9885	*	-5.5149	*	2.9284 *
*	398	*	-1.3180	*	-3.3402	*	1.9701 *
*	398	*	-1.0928	*	-2.5895	*	2.2856 *
*	398	*	-0.7322	*	-2.4813	*	2.3907 *
*	398	*	-0.9574	*	-3.2320	*	2.0752 *
*	399	*	-1.2427	*	-1.8333	*	1.1653 *
*	399	*	-1.1762	*	-1.6113	*	1.3460 *
*	399	*	-0.9697	*	-1.5494	*	1.3771 *
*	399	*	-1.0362	*	-1.7713	*	1.1964 *
*	400	*	-0.8926	*	-1.1484	*	0.3707 *

* 400 * -0.8174 * -0.8978 * 0.2204 *

* 400 * -0.9891 * -0.9493 * 0.2555 *

* 400 * -1.0643 * -1.1999 * 0.4058 *

****

3-noded

Plain Strain case

Elastic matrix D

1.35	0.58	0.00
0.58	1.35	0.00
0.00	0.00	0.38

Displacement Boundary Conditions

node	1	2	3	4	5	6	7	8	9	10	11
	ux=0.0	ux=0.0	ux=0.0	ux=0.0	ux=0.0	ux=0.0	ux=0.0	ux=0.0	ux=0.0	ux=0.0	ux=0.0
	uy=0.0	uy=0.0	uy=0.0	uy=0.0	uy=0.0	uy=0.0	uy=0.0	uy=0.0	uy=0.0	uy=0.0	uy=0.0

Traction BC

node	11	22	33	44	55	66	77	88	99	110
	fx=0.0	fx=0.0	fx=0.0	fx=0.0	fx=0.0	fx=0.0	fx=0.0	fx=0.0	fx=0.0	fx=0.0
	fy=-0.13	fy=-0.25	fy=-0.25	fy=-0.25	fy=-0.25	fy=-0.25	fy=-0.25	fy=-0.25	fy=-0.25	fy=-0.25

node 121 132 143 154 165 176 187 198 209 220

fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0

fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25

node 231 242 253 264 275 286 297 308 319 330

fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0

fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25

node 341 352 363 374 385 396 407 418 429 440

fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0

fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25

node 468

fx=0.0

fy= 0.25

d i s p l a c e m e n t s

.....

****

node	ux	uy	node	ux	uy
1	0.0000	0.0000	2	0.0000	0.0000
3	0.0000	0.0000	4	0.0000	0.0000
5	0.0000	0.0000	6	0.0000	0.0000
7	0.0000	0.0000	8	0.0000	0.0000
9	0.0000	0.0000	10	0.0000	0.0000
11	0.0000	0.0000	12	-16.0080	-12.2395
13	-11.1853	-9.5594	14	-7.8332	-7.8403
15	-5.0160	-6.7916	16	-2.4515	-6.2338
17	0.0068	-6.0765	18	2.4667	-6.2934
19	5.0366	-6.9139	20	7.8656	-8.0322
21	11.2440	-9.8334	22	16.1436	-12.6156
23	-29.1703	-26.6113	24	-22.4511	-24.7569
25	-16.1581	-23.1645	26	-10.4478	-22.0225
27	-5.1105	-21.3601	28	0.0508	-21.1707
29	5.2151	-21.4496	30	10.5617	-22.2027
31	16.2888	-23.4370	32	22.6072	-25.1216
33	29.3432	-27.0587	34	-40.7924	-47.2844
35	-31.7562	-45.6074	36	-23.3008	-44.3254
37	-15.2451	-43.3968	38	-7.4910	-42.8475
39	0.1066	-42.6946	40	7.7062	-42.9429
41	15.4665	-43.5869	42	23.5308	-44.6086
43	31.9930	-45.9807	44	41.0330	-47.7472

```
 *  45 *    -50.6502 *    -73.1815 **  46 *    -39.6736 *    -71.8000 *
 *  47 *    -29.2387 *    -70.7211 **  48 *    -19.2162 *    -69.9624 *
 *  49 *     -9.4606 *    -69.5230 **  50 *      0.1606 *    -69.4071 *
* 399 *     62.0872 *   -258.9129 ** 400 *     41.8328 *   -259.2802 *
 * 401 *     21.7151 *   -259.4958 ** 402 *      1.6769 *   -259.5669 *
 * 403 *    -18.3449 *   -259.5042 ** 404 *    -38.4237 *   -259.3165 *
 * 405 *    -58.6418 *   -259.0068 ** 406 *    -79.0852 *   -258.5707 *
 * 407 *    -99.8394 *   -257.9954 ** 408 *    107.2653 *   -204.5035 *
 * 409 *     85.7090 *   -205.0331 ** 410 *     64.4188 *   -205.4554 *
 * 411 *     43.2973 *   -205.7084 ** 412 *     22.3542 *   -205.8130 *
 * 413 *      1.5535 *   -205.8039 ** 414 *    -19.1795 *   -205.7068 *
 * 415 *    -39.9406 *   -205.5354 ** 416 *    -60.8373 *   -205.2938 *
 * 417 *    -81.9865 *   -204.9786 ** 418 *   -103.5100 *   -204.5751 *
 * 419 *    110.6301 *   -149.3383 ** 420 *     88.3679 *   -149.9224 *
 * 421 *     66.0560 *   -150.1382 ** 422 *     44.1139 *   -150.1746 *
 * 423 *     22.5489 *   -150.1471 ** 424 *      1.2513 *   -150.0998 *
 * 425 *    -19.9146 *   -150.0393 ** 426 *    -41.0826 *   -149.9569 *
 * 427 *    -62.3807 *   -149.8410 ** 428 *    -83.9368 *   -149.6847 *
 * 429 *   -105.9040 *   -149.4907 ** 430 *    114.2083 *    -92.6331 *
 * 431 *     89.8260 *    -92.3783 ** 432 *     66.6245 *    -92.3046 *
 * 433 *     44.2960 *    -92.4529 ** 434 *     22.5217 *    -92.7268 *
 * 435 *      1.0588 *    -93.0365 ** 436 *    -20.2778 *    -93.3213 *
 * 437 *    -41.6345 *    -93.5463 ** 438 *    -63.1297 *    -93.6956 *
 * 439 *    -84.8575 *    -93.7691 ** 440 *   -106.8882 *    -93.7895 *
 * 441 *    118.3903 *    -24.0532 ** 442 *     91.9588 *    -28.0862 *
 * 443 *     68.0723 *    -31.2270 ** 444 *     45.3880 *    -33.6188 *
 * 445 *     23.3432 *    -35.3959 ** 446 *      1.6342 *    -36.6701 *
 * 447 *    -19.9340 *    -37.5343 ** 448 *    -41.5011 *    -38.0714 *
 * 449 *    -63.1697 *    -38.3641 ** 450 *    -85.0014 *    -38.5024 *
```

* 451 * -106.9852 * -38.5868 **

****

s t r e s s e s

....

element	sx	sy	sxy
1	-53.5218	0.2900	-0.9492
1	-62.4488	-20.5399	-11.6583
1	-77.4416	-26.9653	-14.0389
1	-68.5145	-6.1355	-3.3298
2	-38.1712	-1.4600	-2.4527
2	-43.8973	-14.8209	-9.8963
2	-54.3184	-19.2871	-11.4233
2	-48.5923	-5.9261	-3.9797
3	-25.4436	-1.8159	-2.2440
3	-28.9365	-9.9661	-8.4998
3	-37.6947	-13.7196	-9.4312
3	-34.2017	-5.5694	-3.1754
4	-13.5803	-0.9855	-1.9926
4	-15.4384	-5.3210	-7.6874
4	-23.4110	-8.7379	-8.1829
4	-21.5530	-4.4024	-2.4881
5	-2.0452	0.4863	-1.9428
5	-2.5690	-0.7358	-7.4016
5	-10.2113	-4.0111	-7.5413

*	5	*	-9.6876 *	-2.7890 *	-2.0824 *
*	6	*	9.4965 *	2.1908 *	-2.1499 *
*	6	*	10.2187 *	3.8759 *	-7.6123 *
*	6	*	2.5714 *	0.5985 *	-7.4197 *
*	6	*	1.8492 *	-1.0866 *	-1.9573 *
*	7	*	21.3725 *	3.7817 *	-2.6395 *
*	7	*	23.4393 *	8.6044 *	-8.3462 *
*	7	*	15.4499 *	5.1804 *	-7.7950 *
*	7	*	13.3830 *	0.3577 *	-2.0883 *
*	8	*	34.0459 *	4.8992 *	-3.4122 *
*	8	*	37.7707 *	13.5905 *	-9.6942 *
*	8	*	28.9759 *	9.8213 *	-8.7009 *
*	8	*	25.2511 *	1.1300 *	-2.4189 *
*	9	*	48.5049 *	5.1772 *	-4.2947 *
*	9	*	54.5045 *	19.1762 *	-11.7967 *
*	9	*	44.0016 *	14.6750 *	-10.1968 *
*	9	*	38.0020 *	0.6759 *	-2.6948 *
*	10	*	68.6925 *	5.3266 *	-3.6419 *
*	10	*	77.9598 *	26.9503 *	-14.5218 *
*	10	*	62.7280 *	20.4224 *	-12.0505 *
*	10	*	53.4607 *	-1.2013 *	-1.1707 *
*	11	*	-51.1144 *	0.3899 *	1.1892 *
*	11	*	-48.3643 *	6.8068 *	-3.0222 *
*	11	*	-54.2602 *	4.2800 *	-2.2888 *
*	11	*	-57.0103 *	-2.1369 *	1.9225 *
*	12	*	-38.8313 *	1.1510 *	-1.7210 *
*	12	*	-38.4092 *	2.1359 *	-8.2515 *
*	12	*	-47.5519 *	-1.7824 *	-8.1389 *
*	12	*	-47.9740 *	-2.7673 *	-1.6084 *

*	13	*	-26.0657 *	1.1613 *	-3.8514 *
*	13	*	-26.3764 *	0.4362 *	-10.2757 *
*	13	*	-35.3705 *	-3.4184 *	-10.3585 *
*	13	*	-35.0597 *	-2.6933 *	-3.9342 *
*	14	*	-13.7789 *	1.1312 *	-5.0309 *
*	14	*	-14.1272 *	0.3184 *	-11.1881 *
*	14	*	-22.7472 *	-3.3759 *	-11.2809 *
*	14	*	-22.3988 *	-2.5631 *	-5.1238 *
*	15	*	-1.7850 *	1.2422 *	-5.5779 *
*	15	*	-1.8923 *	0.9918 *	-11.5802 *
*	15	*	-10.2956 *	-2.6096 *	-11.6089 *
*	15	*	-10.1883 *	-2.3592 *	-5.6065 *
*	16	*	10.1883 *	1.4452 *	-5.6326 *
*	16	*	10.3953 *	1.9282 *	-11.6380 *
*	16	*	1.9878 *	-1.6750 *	-11.5828 *
*	16	*	1.7808 *	-2.1581 *	-5.5774 *
*	17	*	22.4078 *	1.6356 *	-5.1713 *
*	17	*	22.8494 *	2.6658 *	-11.3370 *
*	17	*	14.2173 *	-1.0336 *	-11.2193 *
*	17	*	13.7758 *	-2.0639 *	-5.0535 *
*	18	*	35.0786 *	1.7394 *	-3.9902 *
*	18	*	35.4650 *	2.6411 *	-10.4257 *
*	18	*	26.4552 *	-1.2203 *	-10.3227 *
*	18	*	26.0688 *	-2.1219 *	-3.8871 *
*	19	*	47.9800 *	1.7801 *	-1.6463 *
*	19	*	47.5916 *	0.8737 *	-8.1747 *
*	19	*	38.4518 *	-3.0433 *	-8.2783 *
*	19	*	38.8403 *	-2.1369 *	-1.7499 *
*	20	*	56.7788 *	1.0841 *	1.9200 *

*	20	*	53.9639 *	-5.4841 *	-2.1579 *
*	20	*	48.2549 *	-7.9308 *	-2.9085 *
*	20	*	51.0699 *	-1.3626 *	1.1693 *
*	21	*	-42.8494 *	0.4769 *	0.8514 *
*	21	*	-42.2583 *	1.8559 *	-4.2938 *
*	21	*	-49.4616 *	-1.2312 *	-4.1362 *
*	21	*	-50.0526 *	-2.6102 *	1.0090 *
*	22	*	-33.1469 *	0.6028 *	-1.6913 *
*	22	*	-32.1131 *	3.0152 *	-6.4929 *
*	22	*	-38.8353 *	0.1342 *	-6.2172 *
*	22	*	-39.8692 *	-2.2781 *	-1.4156 *
*	23	*	-22.8834 *	0.8923 *	-3.7370 *
*	23	*	-22.1726 *	2.5508 *	-8.9452 *
*	23	*	-29.4641 *	-0.5741 *	-8.7556 *
*	23	*	-30.1749 *	-2.2326 *	-3.5474 *
*	24	*	-12.2610 *	1.0442 *	-5.1615 *
*	24	*	-11.8843 *	1.9233 *	-10.5283 *
*	24	*	-19.3979 *	-1.2968 *	-10.4279 *
*	24	*	-19.7747 *	-2.1759 *	-5.0610 *
*	25	*	-1.5450 *	1.1035 *	-5.8607 *
*	25	*	-1.4234 *	1.3872 *	-11.2704 *
*	25	*	-8.9971 *	-1.8586 *	-11.2380 *
*	25	*	-9.1187 *	-2.1423 *	-5.8283 *
*	26	*	9.1724 *	1.1309 *	-5.8265 *
*	26	*	9.0704 *	0.8929 *	-11.2344 *
*	26	*	1.4993 *	-2.3519 *	-11.2616 *
*	26	*	1.6013 *	-2.1139 *	-5.8537 *
*	27	*	19.8154 *	1.1619 *	-5.0490 *
*	27	*	19.4521 *	0.3143 *	-10.4087 *

*	27	*	11.9485 *	-2.9016 *	-10.5056 *
*	27	*	12.3118 *	-2.0539 *	-5.1458 *
*	28	*	30.1822 *	1.2147 *	-3.5239 *
*	28	*	29.4739 *	-0.4381 *	-8.7137 *
*	28	*	22.2082 *	-3.5520 *	-8.9026 *
*	28	*	22.9165 *	-1.8992 *	-3.7128 *
*	29	*	39.8018 *	1.2535 *	-1.3881 *
*	29	*	38.7613 *	-1.1745 *	-6.1486 *
*	29	*	32.0966 *	-4.0308 *	-6.4261 *
*	29	*	33.1371 *	-1.6028 *	-1.6656 *
*	30	*	49.9244 *	1.5886 *	1.0127 *
*	30	*	49.3558 *	0.2618 *	-4.1037 *
*	30	*	42.1928 *	-2.8080 *	-4.2553 *
*	30	*	42.7615 *	-1.4812 *	0.8610 *
*	31	*	-36.5099 *	0.2207 *	0.4402 *
*	31	*	-35.5257 *	2.5173 *	-3.8685 *
*	31	*	-41.5579 *	-0.0679 *	-3.6060 *
*	31	*	-42.5422 *	-2.3646 *	0.7027 *
*	32	*	-27.7535 *	0.4338 *	-2.0174 *
*	32	*	-27.0772 *	2.0119 *	-6.4132 *
*	32	*	-33.2314 *	-0.6257 *	-6.2329 *
*	32	*	-33.9077 *	-2.2038 *	-1.8370 *
*	33	*	-19.0367 *	0.5732 *	-3.8655 *
*	33	*	-18.4710 *	1.8933 *	-8.2328 *
*	33	*	-24.5852 *	-0.7271 *	-8.0819 *
*	33	*	-25.1510 *	-2.0472 *	-3.7146 *
*	34	*	-10.2146 *	0.7057 *	-5.1086 *
*	34	*	-9.8484 *	1.5602 *	-9.5530 *
*	34	*	-16.0705 *	-1.1064 *	-9.4553 *

*	34	*	-16.4367 *	-1.9609 *	-5.0110 *
*	35	*	-1.2981 *	0.8031 *	-5.7242 *
*	35	*	-1.1745 *	1.0914 *	-10.2181 *
*	35	*	-7.4659 *	-1.6049 *	-10.1851 *
*	35	*	-7.5895 *	-1.8932 *	-5.6913 *
*	36	*	7.6232 *	0.8741 *	-5.6851 *
*	36	*	7.4940 *	0.5728 *	-10.1760 *
*	36	*	1.2068 *	-2.1217 *	-10.2104 *
*	36	*	1.3360 *	-1.8204 *	-5.7196 *
*	37	*	16.4551 *	0.9421 *	-4.9954 *
*	37	*	16.0835 *	0.0749 *	-9.4307 *
*	37	*	9.8741 *	-2.5863 *	-9.5298 *
*	37	*	10.2457 *	-1.7191 *	-5.0945 *
*	38	*	25.1445 *	1.0305 *	-3.6946 *
*	38	*	24.5753 *	-0.2976 *	-8.0478 *
*	38	*	18.4808 *	-2.9095 *	-8.1996 *
*	38	*	19.0500 *	-1.5814 *	-3.8464 *
*	39	*	33.8794 *	1.1917 *	-1.8208 *
*	39	*	33.2065 *	-0.3785 *	-6.2052 *
*	39	*	27.0683 *	-3.0091 *	-6.3846 *
*	39	*	27.7412 *	-1.4389 *	-2.0002 *
*	40	*	42.4964 *	1.3554 *	0.7093 *
*	40	*	41.5150 *	-0.9347 *	-3.5906 *
*	40	*	35.4952 *	-3.5146 *	-3.8523 *
*	40	*	36.4766 *	-1.2246 *	0.4476 *
*	41	*	-30.0834 *	0.0362 *	0.1269 *
*	41	*	-29.2177 *	2.0561 *	-3.4113 *
*	41	*	-34.1711 *	-0.0668 *	-3.1804 *
*	41	*	-35.0368 *	-2.0867 *	0.3577 *

*	42	*	-22.9204 *	0.2223 *	-2.2249 *
*	42	*	-22.2243 *	1.8466 *	-5.8140 *
*	42	*	-27.2490 *	-0.3069 *	-5.6283 *
*	42	*	-27.9451 *	-1.9311 *	-2.0393 *
*	43	*	-15.6664 *	0.3676 *	-3.9787 *
*	43	*	-15.1838 *	1.4936 *	-7.6237 *
*	43	*	-20.2868 *	-0.6934 *	-7.4950 *
*	43	*	-20.7693 *	-1.8194 *	-3.8500 *
*	44	*	-8.3800 *	0.4725 *	-5.1362 *
*	44	*	-8.0864 *	1.1578 *	-8.7979 *
*	44	*	-13.2128 *	-1.0393 *	-8.7196 *
*	44	*	-13.5065 *	-1.7245 *	-5.0579 *
*	45	*	-1.0743 *	0.5590 *	-5.6994 *
*	45	*	-0.9768 *	0.7866 *	-9.3741 *
*	45	*	-6.1214 *	-1.4182 *	-9.3481 *
*	45	*	-6.2189 *	-1.6458 *	-5.6734 *
*	46	*	6.2306 *	0.6362 *	-5.6702 *
*	46	*	6.1273 *	0.3951 *	-9.3435 *
*	46	*	0.9847 *	-1.8089 *	-9.3711 *
*	46	*	1.0880 *	-1.5678 *	-5.6978 *
*	47	*	13.5119 *	0.7154 *	-5.0511 *
*	47	*	13.2138 *	0.0199 *	-8.7091 *
*	47	*	8.0926 *	-2.1749 *	-8.7885 *
*	47	*	8.3907 *	-1.4794 *	-5.1306 *
*	48	*	20.7674 *	0.8122 *	-3.8418 *
*	48	*	20.2829 *	-0.3184 *	-7.4826 *
*	48	*	15.1858 *	-2.5028 *	-7.6118 *
*	48	*	15.6704 *	-1.3722 *	-3.9710 *
*	49	*	27.9362 *	0.9261 *	-2.0323 *

*	49	*	27.2397 *	-0.6991 *	-5.6174 *
*	49	*	22.2205 *	-2.8502 *	-5.8031 *
*	49	*	22.9171 *	-1.2249 *	-2.2180 *
*	50	*	35.0256 *	1.0840 *	0.3605 *
*	50	*	34.1623 *	-0.9303 *	-3.1766 *
*	50	*	29.2104 *	-3.0525 *	-3.4068 *
*	50	*	30.0736 *	-1.0382 *	0.1303 *
*	350	*	-24.3037 *	-2.2528 *	-0.6420 *
*	350	*	-23.8954 *	-1.3002 *	1.9458 *
*	350	*	-20.2724 *	0.2525 *	2.0547 *
*	350	*	-20.6807 *	-0.7001 *	-0.5332 *
*	351	*	17.6923 *	-0.1939 *	-0.2877 *
*	351	*	17.2803 *	-1.1553 *	1.8948 *
*	351	*	20.3358 *	0.1542 *	1.7849 *
*	351	*	20.7478 *	1.1156 *	-0.3976 *
*	352	*	13.3695 *	-0.2112 *	0.8686 *
*	352	*	13.0007 *	-1.0716 *	3.0480 *
*	352	*	16.0519 *	0.2361 *	2.9497 *
*	352	*	16.4207 *	1.0965 *	0.7703 *
*	353	*	9.0610 *	-0.3500 *	1.7866 *
*	353	*	8.7787 *	-1.0087 *	3.9243 *
*	353	*	11.7715 *	0.2740 *	3.8490 *
*	353	*	12.0538 *	0.9327 *	1.7113 *
*	354	*	4.8124 *	-0.4625 *	2.4135 *
*	354	*	4.5900 *	-0.9814 *	4.5312 *
*	354	*	7.5548 *	0.2892 *	4.4719 *
*	354	*	7.7772 *	0.8082 *	2.3542 *
*	355	*	0.6049 *	-0.6267 *	2.7833 *
*	355	*	0.4542 *	-0.9784 *	4.8658 *

*	355	*	3.3698 *	0.2711 *	4.8256 *	
*	355	*	3.5205 *	0.6228 *	2.7431 *	
*	356	*	-3.5682 *	-0.8770 *	2.8670 *	
*	356	*	-3.6204 *	-0.9989 *	4.9133 *	
*	356	*	-0.7556 *	0.2289 *	4.8993 *	
*	356	*	-0.7033 *	0.3508 *	2.8530 *	
*	357	*	-7.7316 *	-1.2019 *	2.6207 *	
*	357	*	-7.6617 *	-1.0388 *	4.6501 *	
*	357	*	-4.8206 *	0.1788 *	4.6687 *	
*	357	*	-4.8905 *	0.0157 *	2.6393 *	
*	358	*	-11.9319 *	-1.5665 *	2.0097 *	
*	358	*	-11.7254 *	-1.0848 *	4.0563 *	
*	358	*	-8.8602 *	0.1431 *	4.1114 *	
*	358	*	-9.0666 *	-0.3386 *	2.0648 *	
*	359	*	-16.2271 *	-1.9213 *	1.0187 *	
*	359	*	-15.8823 *	-1.1166 *	3.1243 *	
*	359	*	-12.9345 *	0.1467 *	3.2162 *	
*	359	*	-13.2794 *	-0.6580 *	1.1107 *	
*	360	*	-20.6780 *	-2.2113 *	-0.3451 *	
*	360	*	-20.2062 *	-1.1103 *	1.8622 *	
*	360	*	-17.1161 *	0.2140 *	1.9880 *	
*	360	*	-17.5879 *	-0.8870 *	-0.2192 *	
*	361	*	14.7344 *	0.1422 *	-0.1795 *	
*	361	*	14.2284 *	-1.0384 *	1.6853 *	
*	361	*	16.8391 *	0.0804 *	1.5504 *	
*	361	*	17.3451 *	1.2611 *	-0.3144 *	
*	362	*	10.9459 *	-0.1705 *	1.0002 *	
*	362	*	10.6298 *	-0.9079 *	2.8414 *	
*	362	*	13.2075 *	0.1968 *	2.7571 *	

*	362	*	13.5235 *	0.9342 *	0.9159 *
*	363	*	7.2734 *	0.0717 *	1.8490 *
*	363	*	6.8926 *	-0.8169 *	3.7745 *
*	363	*	9.5883 *	0.3384 *	3.6730 *
*	363	*	9.9692 *	1.2271 *	1.7474 *
*	364	*	3.6419 *	0.1537 *	2.6735 *
*	364	*	3.2719 *	-0.7096 *	4.5062 *
*	364	*	5.8377 *	0.3900 *	4.4075 *
*	364	*	6.2077 *	1.2533 *	2.5748 *
*	365	*	0.1578 *	-0.0187 *	3.3032 *
*	365	*	-0.1091 *	-0.6416 *	4.9965 *
*	365	*	2.2616 *	0.3744 *	4.9253 *
*	365	*	2.5285 *	0.9973 *	3.2320 *
*	366	*	-3.1668 *	-0.3702 *	3.5899 *
*	366	*	-3.2815 *	-0.6377 *	5.1693 *
*	366	*	-1.0704 *	0.3099 *	5.1387 *
*	366	*	-0.9557 *	0.5774 *	3.5593 *
*	367	*	-6.3825 *	-0.8138 *	3.4497 *
*	367	*	-6.3283 *	-0.6871 *	4.9646 *
*	367	*	-4.2074 *	0.2218 *	4.9791 *
*	367	*	-4.2617 *	0.0951 *	3.4642 *
*	368	*	-9.5733 *	-1.2902 *	2.8370 *
*	368	*	-9.3464 *	-0.7606 *	4.3442 *
*	368	*	-7.2364 *	0.1437 *	4.4047 *
*	368	*	-7.4634 *	-0.3860 *	2.8976 *
*	369	*	-12.8536 *	-1.7639 *	1.7193 *
*	369	*	-12.4508 *	-0.8241 *	3.2866 *
*	369	*	-10.2566 *	0.1162 *	3.3940 *
*	369	*	-10.6594 *	-0.8235 *	1.8267 *

*	370	*	-16.3520 *	-2.1749 *	0.0836 *
*	370	*	-15.7799 *	-0.8399 *	1.7918 *
*	370	*	-13.3885 *	0.1849 *	1.9444 *
*	370	*	-13.9607 *	-1.1500 *	0.2362 *
*	371	*	11.8169 *	-1.2284 *	-0.2468 *
*	371	*	11.9987 *	-0.8042 *	1.3207 *
*	371	*	14.1932 *	0.1363 *	1.3691 *
*	371	*	14.0114 *	-0.2879 *	-0.1983 *
*	372	*	8.4816 *	0.7841 *	0.0518 *
*	372	*	7.7935 *	-0.8214 *	2.3207 *
*	372	*	10.9700 *	0.5399 *	2.1372 *
*	372	*	11.6580 *	2.1454 *	-0.1317 *
*	373	*	4.8571 *	1.1790 *	1.6404 *
*	373	*	4.1356 *	-0.5045 *	3.4625 *
*	373	*	6.6866 *	0.5888 *	3.2701 *
*	373	*	7.4081 *	2.2723 *	1.4480 *
*	374	*	1.7536 *	0.7470 *	3.1600 *
*	374	*	1.3138 *	-0.2794 *	4.5411 *
*	374	*	3.2473 *	0.5492 *	4.4238 *
*	374	*	3.6872 *	1.5756 *	3.0427 *
*	375	*	-0.8355 *	0.0729 *	4.1761 *
*	375	*	-0.9629 *	-0.2242 *	5.2795 *
*	375	*	0.5819 *	0.4379 *	5.2455 *
*	375	*	0.7092 *	0.7349 *	4.1421 *
*	376	*	-3.0717 *	-0.5666 *	4.5990 *
*	376	*	-2.9498 *	-0.2822 *	5.5602 *
*	376	*	-1.6042 *	0.2944 *	5.5927 *
*	376	*	-1.7261 *	0.0101 *	4.6315 *
*	377	*	-5.1025 *	-1.0747 *	4.4494 *

*	377	*	-4.8059 *	-0.3827 *	5.3532 *
*	377	*	-3.5406 *	0.1596 *	5.4323 *
*	377	*	-3.8371 *	-0.5324 *	4.5285 *
*	378	*	-7.0319 *	-1.4483 *	3.7634 *
*	378	*	-6.6134 *	-0.4717 *	4.6546 *
*	378	*	-5.3657 *	0.0630 *	4.7662 *
*	378	*	-5.7843 *	-0.9136 *	3.8750 *
*	379	*	-8.9426 *	-1.7453 *	2.5420 *
*	379	*	-8.4134 *	-0.5106 *	3.4455 *
*	379	*	-7.1484 *	0.0315 *	3.5866 *
*	379	*	-7.6776 *	-1.2032 *	2.6831 *
*	380	*	-11.0113 *	-2.1008 *	0.6848 *
*	380	*	-10.3134 *	-0.4724 *	1.6702 *
*	380	*	-8.9338 *	0.1189 *	1.8563 *
*	380	*	-9.6317 *	-1.5096 *	0.8709 *
*	381	*	10.7104 *	5.4412 *	-3.7983 *
*	381	*	7.9162 *	-1.0788 *	0.9096 *
*	381	*	14.5072 *	1.7459 *	0.1645 *
*	381	*	17.3015 *	8.2659 *	-4.5434 *
*	382	*	4.1458 *	1.9147 *	0.3673 *
*	382	*	3.1815 *	-0.3354 *	2.3427 *
*	382	*	5.9470 *	0.8499 *	2.0855 *
*	382	*	6.9113 *	3.0999 *	0.1102 *
*	383	*	0.7010 *	-1.0691 *	3.2740 *
*	383	*	1.0737 *	-0.1993 *	4.1322 *
*	383	*	2.2752 *	0.3156 *	4.2316 *
*	383	*	1.9024 *	-0.5542 *	3.3734 *
*	384	*	-1.1211 *	-2.7904 *	4.8599 *
*	384	*	-0.1172 *	-0.4480 *	5.3245 *

*	384	*	0.5333	*	-0.1692	*	5.5922	*	
*	384	*	-0.4706	*	-2.5117	*	5.1276	*	
*	385	*	-2.2000	*	-3.5173	*	5.4904	*	
*	385	*	-1.0108	*	-0.7425	*	5.8574	*	
*	385	*	-0.4969	*	-0.5223	*	6.1745	*	
*	385	*	-1.6861	*	-3.2971	*	5.8075	*	
*	386	*	-2.9840	*	-3.6069	*	5.4457	*	
*	386	*	-1.8337	*	-0.9230	*	5.8249	*	
*	386	*	-1.3028	*	-0.6954	*	6.1316	*	
*	386	*	-2.4531	*	-3.3793	*	5.7524	*	
*	387	*	-3.6801	*	-3.3359	*	4.8978	*	
*	387	*	-2.6563	*	-0.9469	*	5.3165	*	
*	387	*	-2.0701	*	-0.6957	*	5.5895	*	
*	387	*	-3.0940	*	-3.0846	*	5.1709	*	
*	388	*	-4.3466	*	-2.8877	*	3.9502	*	
*	388	*	-3.4632	*	-0.8265	*	4.3879	*	
*	388	*	-2.8504	*	-0.5639	*	4.6234	*	
*	388	*	-3.7338	*	-2.6251	*	4.1857	*	
*	389	*	-4.9060	*	-2.3763	*	2.6696	*	
*	389	*	-4.1404	*	-0.5899	*	3.0510	*	
*	389	*	-3.6064	*	-0.3611	*	3.2552	*	
*	389	*	-4.3720	*	-2.1475	*	2.8737	*	
*	390	*	-5.0834	*	-1.9049	*	1.0820	*	
*	390	*	-4.3693	*	-0.2387	*	1.2229	*	
*	390	*	-4.1721	*	-0.1542	*	1.4133	*	
*	390	*	-4.8862	*	-1.8204	*	1.2725	*	
*	391	*	-4.2234	*	-36.1715	*	0.3109	*	
*	391	*	10.0584	*	-2.8473	*	4.8612	*	
*	391	*	16.4289	*	-0.1171	*	8.6697	*	

*	391	*	2.1471	*	-33.4412	*	4.1193	*	
*	392	*	-5.6256	*	-29.4603	*	3.6961	*	
*	392	*	5.0816	*	-4.4769	*	5.2174	*	
*	392	*	7.2113	*	-3.5642	*	8.0726	*	
*	392	*	-3.4959	*	-28.5476	*	6.5514	*	
*	393	*	-4.7784	*	-23.1213	*	4.2852	*	
*	393	*	2.6945	*	-5.6846	*	5.0751	*	
*	393	*	3.8003	*	-5.2107	*	7.0679	*	
*	393	*	-3.6726	*	-22.6474	*	6.2780	*	
*	394	*	-3.6883	*	-17.6179	*	4.1221	*	
*	394	*	1.3184	*	-5.9355	*	4.7230	*	
*	394	*	2.1597	*	-5.5749	*	6.0582	*	
*	394	*	-2.8470	*	-17.2573	*	5.4572	*	
*	395	*	-2.7971	*	-12.9614	*	3.7350	*	
*	395	*	0.4155	*	-5.4653	*	4.2814	*	
*	395	*	1.1805	*	-5.1375	*	5.1381	*	
*	395	*	-2.0322	*	-12.6336	*	4.5917	*	
*	396	*	-2.1645	*	-9.0790	*	3.2484	*	
*	396	*	-0.2346	*	-4.5760	*	3.7625	*	
*	396	*	0.4852	*	-4.2675	*	4.2772	*	
*	396	*	-1.4447	*	-8.7705	*	3.7630	*	
*	397	*	-1.7606	*	-5.9322	*	2.6681	*	
*	397	*	-0.7209	*	-3.5063	*	3.1355	*	
*	397	*	-0.0666	*	-3.2258	*	3.4128	*	
*	397	*	-1.1062	*	-5.6517	*	2.9454	*	
*	398	*	-1.5320	*	-3.5401	*	1.9722	*	
*	398	*	-1.0544	*	-2.4257	*	2.3573	*	
*	398	*	-0.5154	*	-2.1947	*	2.4846	*	
*	398	*	-0.9930	*	-3.3091	*	2.0996	*	

*	399 *	-1.3755 *	-1.9586 *	1.1629 *
*	399 *	-1.1597 *	-1.4550 *	1.3936 *
*	399 *	-0.8367 *	-1.3166 *	1.4512 *
*	399 *	-1.0525 *	-1.8201 *	1.2205 *
*	400 *	-0.9845 *	-1.2006 *	0.3571 *
*	400 *	-0.7714 *	-0.7034 *	0.2529 *
*	400 *	-0.9173 *	-0.7659 *	0.3097 *
*	400 *	-1.1304 *	-1.2632 *	0.4139 *

****

Connectivity Matrix

1	1	12	13	2
2	2	13	14	3
3	3	14	15	4
4	4	15	16	5
5	5	16	17	6
6	6	17	18	7
7	7	18	19	8
8	8	19	20	9
9	9	20	21	10
10	10	21	22	11
11	12	23	24	13
12	13	24	25	14
13	14	25	26	15
14	15	26	27	16
15	16	27	28	17
16	17	28	29	18
17	18	29	30	19
18	19	30	31	20
19	20	31	32	21

20	21	32	33	22
21	23	34	35	24
22	24	35	36	25
23	25	36	37	26
24	26	37	38	27
25	27	38	39	28
26	28	39	40	29
27	29	40	41	30
28	30	41	42	31
29	31	42	43	32
30	32	43	44	33
31	34	45	46	35
32	35	46	47	36
33	36	47	48	37
34	37	48	49	38
35	38	49	50	39
36	39	50	51	40
37	40	51	52	41
38	41	52	53	42
39	42	53	54	43
40	43	54	55	44
41	45	56	57	46
42	46	57	58	47
43	47	58	59	48
44	48	59	60	49
45	49	60	61	50
46	50	61	62	51
47	51	62	63	52
48	52	63	64	53

49	53	64	65	54
50	54	65	66	55
51	56	67	68	57
52	57	68	69	58
53	58	69	70	59
54	59	70	71	60
55	60	71	72	61
56	61	72	73	62
57	62	73	74	63
58	63	74	75	64
59	64	75	76	65
60	65	76	77	66
61	67	78	79	68
62	68	79	80	69
63	69	80	81	70
64	70	81	82	71
65	71	82	83	72
66	72	83	84	73
67	73	84	85	74
68	74	85	86	75
69	75	86	87	76
70	76	87	88	77
71	78	89	90	79
72	79	90	91	80
73	80	91	92	81
74	81	92	93	82
75	82	93	94	83
76	83	94	95	84
77	84	95	96	85

78	85	96	97	86
79	86	97	98	87
80	87	98	99	88
81	89	100	101	90
82	90	101	102	91
83	91	102	103	92
84	92	103	104	93
85	93	104	105	94
86	94	105	106	95
87	95	106	107	96
88	96	107	108	97
89	97	108	109	98
90	98	109	110	99
91	100	111	112	101
92	101	112	113	102
93	102	113	114	103
94	103	114	115	104
95	104	115	116	105
96	105	116	117	106
97	106	117	118	107
98	107	118	119	108
99	108	119	120	109
100	109	120	121	110
101	111	122	123	112
102	112	123	124	113
103	113	124	125	114
104	114	125	126	115
105	115	126	127	116
106	116	127	128	117

107	117	128	129	118
108	118	129	130	119
109	119	130	131	120
110	120	131	132	121
111	122	133	134	123
112	123	134	135	124
113	124	135	136	125
114	125	136	137	126
115	126	137	138	127
116	127	138	139	128
117	128	139	140	129
118	129	140	141	130
119	130	141	142	131
120	131	142	143	132
121	133	144	145	134
122	134	145	146	135
123	135	146	147	136
124	136	147	148	137
125	137	148	149	138
126	138	149	150	139
127	139	150	151	140
128	140	151	152	141
129	141	152	153	142
130	142	153	154	143
131	144	155	156	145
132	145	156	157	146
133	146	157	158	147
134	147	158	159	148
135	148	159	160	149

136	149	160	161	150
137	150	161	162	151
138	151	162	163	152
139	152	163	164	153
140	153	164	165	154
141	155	166	167	156
142	156	167	168	157
143	157	168	169	158
144	158	169	170	159
145	159	170	171	160
146	160	171	172	161
147	161	172	173	162
148	162	173	174	163
149	163	174	175	164
150	164	175	176	165
151	166	177	178	167
152	167	178	179	168
153	168	179	180	169
154	169	180	181	170
155	170	181	182	171
156	171	182	183	172
157	172	183	184	173
158	173	184	185	174
159	174	185	186	175
160	175	186	187	176
161	177	188	189	178
162	178	189	190	179
163	179	190	191	180
164	180	191	192	181

165	181	192	193	182
166	182	193	194	183
167	183	194	195	184
168	184	195	196	185
169	185	196	197	186
170	186	197	198	187
171	188	199	200	189
172	189	200	201	190
173	190	201	202	191
174	191	202	203	192
175	192	203	204	193
176	193	204	205	194
177	194	205	206	195
178	195	206	207	196
179	196	207	208	197
180	197	208	209	198
181	199	210	211	200
182	200	211	212	201
183	201	212	213	202
184	202	213	214	203
185	203	214	215	204
186	204	215	216	205
187	205	216	217	206
188	206	217	218	207
189	207	218	219	208
190	208	219	220	209
191	210	221	222	211
192	211	222	223	212
193	212	223	224	213

194	213	224	225	214
195	214	225	226	215
196	215	226	227	216
197	216	227	228	217
198	217	228	229	218
199	218	229	230	219
200	219	230	231	220
201	221	232	233	222
202	222	233	234	223
203	223	234	235	224
204	224	235	236	225
205	225	236	237	226
206	226	237	238	227
207	227	238	239	228
208	228	239	240	229
209	229	240	241	230
210	230	241	242	231
211	232	243	244	233
212	233	244	245	234
213	234	245	246	235
214	235	246	247	236
215	236	247	248	237
216	237	248	249	238
217	238	249	250	239
218	239	250	251	240
219	240	251	252	241
220	241	252	253	242
221	243	254	255	244
222	244	255	256	245

223	245	256	257	246
224	246	257	258	247
225	247	258	259	248
226	248	259	260	249
227	249	260	261	250
228	250	261	262	251
229	251	262	263	252
230	252	263	264	253
231	254	265	266	255
232	255	266	267	256
233	256	267	268	257
234	257	268	269	258
235	258	269	270	259
236	259	270	271	260
237	260	271	272	261
238	261	272	273	262
239	262	273	274	263
240	263	274	275	264
241	265	276	277	266
242	266	277	278	267
243	267	278	279	268
244	268	279	280	269
245	269	280	281	270
246	270	281	282	271
247	271	282	283	272
248	272	283	284	273
249	273	284	285	274
250	274	285	286	275
251	276	287	288	277

252	277	288	289	278
253	278	289	290	279
254	279	290	291	280
255	280	291	292	281
256	281	292	293	282
257	282	293	294	283
258	283	294	295	284
259	284	295	296	285
260	285	296	297	286
261	287	298	299	288
262	288	299	300	289
263	289	300	301	290
264	290	301	302	291
265	291	302	303	292
266	292	303	304	293
267	293	304	305	294
268	294	305	306	295
269	295	306	307	296
270	296	307	308	297
271	298	309	310	299
272	299	310	311	300
273	300	311	312	301
274	301	312	313	302
275	302	313	314	303
276	303	314	315	304
277	304	315	316	305
278	305	316	317	306
279	306	317	318	307
280	307	318	319	308

281	309	320	321	310
282	310	321	322	311
283	311	322	323	312
284	312	323	324	313
285	313	324	325	314
286	314	325	326	315
287	315	326	327	316
288	316	327	328	317
289	317	328	329	318
290	318	329	330	319
291	320	331	332	321
292	321	332	333	322
293	322	333	334	323
294	323	334	335	324
295	324	335	336	325
296	325	336	337	326
297	326	337	338	327
298	327	338	339	328
299	328	339	340	329
300	329	340	341	330
301	331	342	343	332
302	332	343	344	333
303	333	344	345	334
304	334	345	346	335
305	335	346	347	336
306	336	347	348	337
307	337	348	349	338
308	338	349	350	339
309	339	350	351	340

310	340	351	352	341
311	342	353	354	343
312	343	354	355	344
313	344	355	356	345
314	345	356	357	346
315	346	357	358	347
316	347	358	359	348
317	348	359	360	349
318	349	360	361	350
319	350	361	362	351
320	351	362	363	352
321	353	364	365	354
322	354	365	366	355
323	355	366	367	356
324	356	367	368	357
325	357	368	369	358
326	358	369	370	359
327	359	370	371	360
328	360	371	372	361
329	361	372	373	362
330	362	373	374	363
331	364	375	376	365
332	365	376	377	366
333	366	377	378	367
334	367	378	379	368
335	368	379	380	369
336	369	380	381	370
337	370	381	382	371
338	371	382	383	372

339	372	383	384	373
340	373	384	385	374
341	375	386	387	376
342	376	387	388	377
343	377	388	389	378
344	378	389	390	379
345	379	390	391	380
346	380	391	392	381
347	381	392	393	382
348	382	393	394	383
349	383	394	395	384
350	384	395	396	385
351	386	397	398	387
352	387	398	399	388
353	388	399	400	389
354	389	400	401	390
355	390	401	402	391
356	391	402	403	392
357	392	403	404	393
358	393	404	405	394
359	394	405	406	395
360	395	406	407	396
361	397	408	409	398
362	398	409	410	399
363	399	410	411	400
364	400	411	412	401
365	401	412	413	402
366	402	413	414	403
367	403	414	415	404

368	404	415	416	405
369	405	416	417	406
370	406	417	418	407
371	408	419	420	409
372	409	420	421	410
373	410	421	422	411
374	411	422	423	412
375	412	423	424	413
376	413	424	425	414
377	414	425	426	415
378	415	426	427	416
379	416	427	428	417
380	417	428	429	418
381	419	430	431	420
382	420	431	432	421
383	421	432	433	422
384	422	433	434	423
385	423	434	435	424
386	424	435	436	425
387	425	436	437	426
388	426	437	438	427
389	427	438	439	428
390	428	439	440	429
391	430	441	442	431
392	431	442	443	432
393	432	443	444	433
394	433	444	445	434
395	434	445	446	435
396	435	446	447	436

397	436	447	448	437
398	437	448	449	438
399	438	449	450	439
400	439	450	451	440

Coordinate Matrix

1	0.00000	0.00000
2	0.00000	0.10000
3	0.00000	0.20000
4	0.00000	0.30000
5	0.00000	0.40000
6	0.00000	0.50000
7	0.00000	0.60000
8	0.00000	0.70000
9	0.00000	0.80000
10	0.00000	0.90000
11	0.00000	1.00000
12	0.25000	0.00000
13	0.25000	0.10000
14	0.25000	0.20000
15	0.25000	0.30000
16	0.25000	0.40000
17	0.25000	0.50000
18	0.25000	0.60000
19	0.25000	0.70000
20	0.25000	0.80000
21	0.25000	0.90000
22	0.25000	1.00000
23	0.50000	0.00000
24	0.50000	0.10000

25	0.50000	0.20000
26	0.50000	0.30000
27	0.50000	0.40000
28	0.50000	0.50000
29	0.50000	0.60000
30	0.50000	0.70000
31	0.50000	0.80000
32	0.50000	0.90000
33	0.50000	1.00000
34	0.75000	0.00000
35	0.75000	0.10000
36	0.75000	0.20000
37	0.75000	0.30000
38	0.75000	0.40000
39	0.75000	0.50000
40	0.75000	0.60000
41	0.75000	0.70000
42	0.75000	0.80000
43	0.75000	0.90000
44	0.75000	1.00000
45	1.00000	0.00000
46	1.00000	0.10000
47	1.00000	0.20000
48	1.00000	0.30000
49	1.00000	0.40000
50	1.00000	0.50000
51	1.00000	0.60000
52	1.00000	0.70000
53	1.00000	0.80000

54	1.00000	0.90000
55	1.00000	1.00000
56	1.25000	0.00000
57	1.25000	0.10000
58	1.25000	0.20000
59	1.25000	0.30000
60	1.25000	0.40000
61	1.25000	0.50000
62	1.25000	0.60000
63	1.25000	0.70000
64	1.25000	0.80000
65	1.25000	0.90000
66	1.25000	1.00000
67	1.50000	0.00000
68	1.50000	0.10000
69	1.50000	0.20000
70	1.50000	0.30000
71	1.50000	0.40000
72	1.50000	0.50000
73	1.50000	0.60000
74	1.50000	0.70000
75	1.50000	0.80000
76	1.50000	0.90000
77	1.50000	1.00000
78	1.75000	0.00000
79	1.75000	0.10000
80	1.75000	0.20000
81	1.75000	0.30000
82	1.75000	0.40000

83	1.75000	0.50000
84	1.75000	0.60000
85	1.75000	0.70000
86	1.75000	0.80000
87	1.75000	0.90000
88	1.75000	1.00000
89	2.00000	0.00000
90	2.00000	0.10000
91	2.00000	0.20000
92	2.00000	0.30000
93	2.00000	0.40000
94	2.00000	0.50000
95	2.00000	0.60000
96	2.00000	0.70000
97	2.00000	0.80000
98	2.00000	0.90000
99	2.00000	1.00000
100	2.25000	0.00000
101	2.25000	0.10000
102	2.25000	0.20000
103	2.25000	0.30000
104	2.25000	0.40000
105	2.25000	0.50000
106	2.25000	0.60000
107	2.25000	0.70000
108	2.25000	0.80000
109	2.25000	0.90000
110	2.25000	1.00000
111	2.50000	0.00000

112	2.50000	0.10000
113	2.50000	0.20000
114	2.50000	0.30000
115	2.50000	0.40000
116	2.50000	0.50000
117	2.50000	0.60000
118	2.50000	0.70000
119	2.50000	0.80000
120	2.50000	0.90000
121	2.50000	1.00000
122	2.75000	0.00000
123	2.75000	0.10000
124	2.75000	0.20000
125	2.75000	0.30000
126	2.75000	0.40000
127	2.75000	0.50000
128	2.75000	0.60000
129	2.75000	0.70000
130	2.75000	0.80000
131	2.75000	0.90000
132	2.75000	1.00000
133	3.00000	0.00000
134	3.00000	0.10000
135	3.00000	0.20000
136	3.00000	0.30000
137	3.00000	0.40000
138	3.00000	0.50000
139	3.00000	0.60000
140	3.00000	0.70000

141	3.00000	0.80000
142	3.00000	0.90000
143	3.00000	1.00000
144	3.25000	0.00000
145	3.25000	0.10000
146	3.25000	0.20000
147	3.25000	0.30000
148	3.25000	0.40000
149	3.25000	0.50000
150	3.25000	0.60000
151	3.25000	0.70000
152	3.25000	0.80000
153	3.25000	0.90000
154	3.25000	1.00000
155	3.50000	0.00000
156	3.50000	0.10000
157	3.50000	0.20000
158	3.50000	0.30000
159	3.50000	0.40000
160	3.50000	0.50000
161	3.50000	0.60000
162	3.50000	0.70000
163	3.50000	0.80000
164	3.50000	0.90000
165	3.50000	1.00000
166	3.75000	0.00000
167	3.75000	0.10000
168	3.75000	0.20000
169	3.75000	0.30000

170	3.75000	0.40000
171	3.75000	0.50000
172	3.75000	0.60000
173	3.75000	0.70000
174	3.75000	0.80000
175	3.75000	0.90000
176	3.75000	1.00000
177	4.00000	0.00000
178	4.00000	0.10000
179	4.00000	0.20000
180	4.00000	0.30000
181	4.00000	0.40000
182	4.00000	0.50000
183	4.00000	0.60000
184	4.00000	0.70000
185	4.00000	0.80000
186	4.00000	0.90000
187	4.00000	1.00000
188	4.25000	0.00000
189	4.25000	0.10000
190	4.25000	0.20000
191	4.25000	0.30000
192	4.25000	0.40000
193	4.25000	0.50000
194	4.25000	0.60000
195	4.25000	0.70000
196	4.25000	0.80000
197	4.25000	0.90000
198	4.25000	1.00000

199	4.50000	0.00000
200	4.50000	0.10000
201	4.50000	0.20000
202	4.50000	0.30000
203	4.50000	0.40000
204	4.50000	0.50000
205	4.50000	0.60000
206	4.50000	0.70000
207	4.50000	0.80000
208	4.50000	0.90000
209	4.50000	1.00000
210	4.75000	0.00000
211	4.75000	0.10000
212	4.75000	0.20000
213	4.75000	0.30000
214	4.75000	0.40000
215	4.75000	0.50000
216	4.75000	0.60000
217	4.75000	0.70000
218	4.75000	0.80000
219	4.75000	0.90000
220	4.75000	1.00000
221	5.00000	0.00000
222	5.00000	0.10000
223	5.00000	0.20000
224	5.00000	0.30000
225	5.00000	0.40000
226	5.00000	0.50000
227	5.00000	0.60000

228	5.00000	0.70000
229	5.00000	0.80000
230	5.00000	0.90000
231	5.00000	1.00000
232	5.25000	0.00000
233	5.25000	0.10000
234	5.25000	0.20000
235	5.25000	0.30000
236	5.25000	0.40000
237	5.25000	0.50000
238	5.25000	0.60000
239	5.25000	0.70000
240	5.25000	0.80000
241	5.25000	0.90000
242	5.25000	1.00000
243	5.50000	0.00000
244	5.50000	0.10000
245	5.50000	0.20000
246	5.50000	0.30000
247	5.50000	0.40000
248	5.50000	0.50000
249	5.50000	0.60000
250	5.50000	0.70000
251	5.50000	0.80000
252	5.50000	0.90000
253	5.50000	1.00000
254	5.75000	0.00000
255	5.75000	0.10000
256	5.75000	0.20000

257	5.75000	0.30000
258	5.75000	0.40000
259	5.75000	0.50000
260	5.75000	0.60000
261	5.75000	0.70000
262	5.75000	0.80000
263	5.75000	0.90000
264	5.75000	1.00000
265	6.00000	0.00000
266	6.00000	0.10000
267	6.00000	0.20000
268	6.00000	0.30000
269	6.00000	0.40000
270	6.00000	0.50000
271	6.00000	0.60000
272	6.00000	0.70000
273	6.00000	0.80000
274	6.00000	0.90000
275	6.00000	1.00000
276	6.25000	0.00000
277	6.25000	0.10000
278	6.25000	0.20000
279	6.25000	0.30000
280	6.25000	0.40000
281	6.25000	0.50000
282	6.25000	0.60000
283	6.25000	0.70000
284	6.25000	0.80000
285	6.25000	0.90000

286	6.25000	1.00000
287	6.50000	0.00000
288	6.50000	0.10000
289	6.50000	0.20000
290	6.50000	0.30000
291	6.50000	0.40000
292	6.50000	0.50000
293	6.50000	0.60000
294	6.50000	0.70000
295	6.50000	0.80000
296	6.50000	0.90000
297	6.50000	1.00000
298	6.75000	0.00000
299	6.75000	0.10000
300	6.75000	0.20000
301	6.75000	0.30000
302	6.75000	0.40000
303	6.75000	0.50000
304	6.75000	0.60000
305	6.75000	0.70000
306	6.75000	0.80000
307	6.75000	0.90000
308	6.75000	1.00000
309	7.00000	0.00000
310	7.00000	0.10000
311	7.00000	0.20000
312	7.00000	0.30000
313	7.00000	0.40000
314	7.00000	0.50000

315	7.00000	0.60000
316	7.00000	0.70000
317	7.00000	0.80000
318	7.00000	0.90000
319	7.00000	1.00000
320	7.25000	0.00000
321	7.25000	0.10000
322	7.25000	0.20000
323	7.25000	0.30000
324	7.25000	0.40000
325	7.25000	0.50000
326	7.25000	0.60000
327	7.25000	0.70000
328	7.25000	0.80000
329	7.25000	0.90000
330	7.25000	1.00000
331	7.50000	0.00000
332	7.50000	0.10000
333	7.50000	0.20000
334	7.50000	0.30000
335	7.50000	0.40000
336	7.50000	0.50000
337	7.50000	0.60000
338	7.50000	0.70000
339	7.50000	0.80000
340	7.50000	0.90000
341	7.50000	1.00000
342	7.75000	0.00000
343	7.75000	0.10000

344	7.75000	0.20000
345	7.75000	0.30000
346	7.75000	0.40000
347	7.75000	0.50000
348	7.75000	0.60000
349	7.75000	0.70000
350	7.75000	0.80000
351	7.75000	0.90000
352	7.75000	1.00000
353	8.00000	0.00000
354	8.00000	0.10000
355	8.00000	0.20000
356	8.00000	0.30000
357	8.00000	0.40000
358	8.00000	0.50000
359	8.00000	0.60000
360	8.00000	0.70000
361	8.00000	0.80000
362	8.00000	0.90000
363	8.00000	1.00000
364	8.25000	0.00000
365	8.25000	0.10000
366	8.25000	0.20000
367	8.25000	0.30000
368	8.25000	0.40000
369	8.25000	0.50000
370	8.25000	0.60000
371	8.25000	0.70000
372	8.25000	0.80000

373	8.25000	0.90000
374	8.25000	1.00000
375	8.50000	0.00000
376	8.50000	0.10000
377	8.50000	0.20000
378	8.50000	0.30000
379	8.50000	0.40000
380	8.50000	0.50000
381	8.50000	0.60000
382	8.50000	0.70000
383	8.50000	0.80000
384	8.50000	0.90000
385	8.50000	1.00000
386	8.75000	0.00000
387	8.75000	0.10000
388	8.75000	0.20000
389	8.75000	0.30000
390	8.75000	0.40000
391	8.75000	0.50000
392	8.75000	0.60000
393	8.75000	0.70000
394	8.75000	0.80000
395	8.75000	0.90000
396	8.75000	1.00000
397	9.00000	0.00000
398	9.00000	0.10000
399	9.00000	0.20000
400	9.00000	0.30000
401	9.00000	0.40000

402	9.00000	0.50000
403	9.00000	0.60000
404	9.00000	0.70000
405	9.00000	0.80000
406	9.00000	0.90000
407	9.00000	1.00000
408	9.25000	0.00000
409	9.25000	0.10000
410	9.25000	0.20000
411	9.25000	0.30000
412	9.25000	0.40000
413	9.25000	0.50000
414	9.25000	0.60000
415	9.25000	0.70000
416	9.25000	0.80000
417	9.25000	0.90000
418	9.25000	1.00000
419	9.50000	0.00000
420	9.50000	0.10000
421	9.50000	0.20000
422	9.50000	0.30000
423	9.50000	0.40000
424	9.50000	0.50000
425	9.50000	0.60000
426	9.50000	0.70000
427	9.50000	0.80000
428	9.50000	0.90000
429	9.50000	1.00000
430	9.75000	0.00000

431	9.75000	0.10000
432	9.75000	0.20000
433	9.75000	0.30000
434	9.75000	0.40000
435	9.75000	0.50000
436	9.75000	0.60000
437	9.75000	0.70000
438	9.75000	0.80000
439	9.75000	0.90000
440	9.75000	1.00000
441	10.00000	0.00000
442	10.00000	0.10000
443	10.00000	0.20000
444	10.00000	0.30000
445	10.00000	0.40000
446	10.00000	0.50000
447	10.00000	0.60000
448	10.00000	0.70000
449	10.00000	0.80000
450	10.00000	0.90000
451	10.00000	1.00000

Three Noded

Plain Strain case

Elastic matrix D

1.20	0.40	0.00
0.40	1.20	0.00
0.00	0.00	0.40

d i s p l a c e m e n t s

.....

node	ux	uy	node	ux	uy
1	0.0000	0.0000	2	0.0000	0.0000
3	0.0000	0.0000	4	0.0000	0.0000
5	0.0000	0.0000	6	0.0000	0.0000
7	0.0000	0.0000	8	0.0000	0.0000
9	0.0000	0.0000	10	0.0000	0.0000
11	0.0000	0.0000	12	-15.0175	-10.0776
13	-11.1368	-8.5996	14	-7.9584	-7.5770
15	-5.1601	-6.9181	16	-2.5671	-6.5563
17	-0.0643	-6.4535	18	2.4439	-6.5997
19	5.0611	-7.0145	20	7.9347	-7.7542
21	11.3281	-8.9282	22	15.8035	-10.7355
23	-28.1395	-24.6340	24	-21.6175	-23.2105
25	-15.7164	-22.1367	26	-10.2556	-21.4039
27	-5.0686	-20.9982	28	-0.0055	-20.9104
29	5.0769	-21.1416	30	10.3270	-21.7033
31	15.9067	-22.6132	32	21.9824	-23.8763
33	28.6385	-25.4278	34	-39.4064	-44.8044
35	-30.6812	-43.5803	36	-22.5075	-42.6401

* 37 * -14.7474 * -41.9910 ** 38 * -7.2571 * -41.6360 *

* 39 * 0.1070 * -41.5761 ** 40 * 7.4867 * -41.8115 *

* 41 * 15.0211 * -42.3410 ** 42 * 22.8427 * -43.1584 *

* 43 * 31.0709 * -44.2494 ** 44 * 39.8201 * -45.6019 *

* 45 * -48.9010 * -69.9879 ** 46 * -38.3118 * -68.9716 *

* 47 * -28.2348 * -68.1899 ** 48 * -18.5431 * -67.6509 *

* 49 * -9.1066 * -67.3594 ** 50 * 0.2059 * -67.3178 *

* 399 * 59.9356 * -249.7968 ** 400 * 40.3523 * -250.0532 *

* 401 * 20.9123 * -250.1970 ** 402 * 1.5584 * -250.2386 *

* 403 * -17.7738 * -250.1869 ** 404 * -37.1560 * -250.0468 *

* 405 * -56.6646 * -249.8194 ** 406 * -76.3799 * -249.5012 *

* 407 * -96.3845 * -249.0841 ** 408 * 103.6734 * -197.3887 *

* 409 * 82.7785 * -197.8097 ** 410 * 62.1385 * -198.0976 *

* 411 * 41.7314 * -198.2680 ** 412 * 21.5211 * -198.3470 *

* 413 * 1.4459 * -198.3542 ** 414 * -18.5730 * -198.3009 *

* 415 * -38.6238 * -198.1915 ** 416 * -58.8001 * -198.0256 *

* 417 * -79.2001 * -197.7991 ** 418 * -99.9272 * -197.5046 *

* 419 * 107.0045 * -143.8761 ** 420 * 85.1873 * -144.2369 *

* 421 * 63.6991 * -144.4229 ** 422 * 42.5981 * -144.5347 *

* 423 * 21.8156 * -144.6086 ** 424 * 1.2466 * -144.6543 *

* 425 * -19.2222 * -144.6707 ** 426 * -39.7028 * -144.6523 *

* 427 * -60.3042 * -144.5934 ** 428 * -81.1341 * -144.4899 *

* 429 * -102.3047 * -144.3402 ** 430 * 109.9856 * -88.0036 *

* 431 * 86.7491 * -88.2723 ** 432 * 64.5548 * -88.6391 *

* 433 * 43.0363 * -89.0383 ** 434 * 21.9586 * -89.4253 *

* 435 * 1.1444 * -89.7687 ** 436 * -19.5518 * -90.0481 *

* 437 * -40.2550 * -90.2527 ** 438 * -61.0741 * -90.3805 *

* 439 * -82.1026 * -90.4387 ** 440 * -103.4175 * -90.4446 *

* 441 * 111.4401 * -26.0459 ** 442 * 88.1315 * -28.5240 *

*	443 *	65.7227 *	-30.6674 **	444 *	43.9728 *	-32.4533 *
*	445 *	22.6700 *	-33.8839 **	446 *	1.6405 *	-34.9768 *
*	447 *	-19.2597 *	-35.7616 **	448 *	-40.1505 *	-36.2779 *
*	449 *	-61.1303 *	-36.5767 **	450 *	-82.2719 *	-36.7226 *
*	451 *	-103.6094 *	-36.7966 **			

****

s t r e s s e s

.....

*	element	*	sx	*	sy	*	sxy	*
*	1	*	-72.0840	*	-24.0280	*	-16.1241	*
*	2	*	-47.5449	*	-0.0832	*	1.7633	*
*	3	*	-53.4568	*	-17.8189	*	-13.7593	*
*	4	*	-34.1097	*	-0.4620	*	0.5907	*
*	5	*	-38.2001	*	-12.7334	*	-12.1232	*
*	6	*	-22.1328	*	-0.3493	*	0.1242	*
*	7	*	-24.7684	*	-8.2561	*	-11.0689	*
*	8	*	-10.8750	*	0.2340	*	-0.1181	*
*	9	*	-12.3221	*	-4.1074	*	-10.4901	*

*	10	*	0.1024 *	1.1304 *	-0.3145 *
*	11	*	-0.3087 *	-0.1029 *	-10.3256 *
*	12	*	11.1458 *	2.1561 *	-0.5268 *
*	13	*	11.7305 *	3.9102 *	-10.5595 *
*	14	*	22.6342 *	3.1203 *	-0.7541 *
*	15	*	24.2934 *	8.0978 *	-11.2232 *
*	16	*	35.1280 *	3.8194 *	-0.9122 *
*	17	*	38.0867 *	12.6956 *	-12.4067 *
*	18	*	49.6789 *	4.0367 *	-0.7115 *
*	19	*	54.3750 *	18.1250 *	-14.2851 *
*	20	*	68.6277 *	3.5978 *	0.7248 *
*	21	*	-57.0735 *	-3.2594 *	-7.7676 *
*	22	*	-44.6134 *	0.3125 *	2.7104 *
*	23	*	-46.2168 *	-4.4977 *	-10.6636 *
*	24	*	-32.9433 *	0.4730 *	0.3089 *
*	25	*	-34.6029 *	-4.5060 *	-12.1025 *
*	26	*	-21.5271 *	0.6407 *	-1.3340 *
*	27	*	-23.0111 *	-3.8114 *	-12.8054 *
*	28	*	-10.3841 *	0.8666 *	-2.3591 *
*	29	*	-11.5960 *	-2.7691 *	-13.0958 *
*	30	*	0.6332 *	1.1468 *	-2.8788 *
*	31	*	-0.3023 *	-1.6600 *	-13.0984 *
*	32	*	11.7140 *	1.4387 *	-2.9374 *
*	33	*	10.9796 *	-0.7646 *	-12.7980 *
*	34	*	23.0294 *	1.6849 *	-2.5019 *
*	35	*	22.3175 *	-0.4508 *	-12.0077 *
*	36	*	34.6263 *	1.8373 *	-1.4555 *
*	37	*	33.5694 *	-1.3331 *	-10.2008 *
*	38	*	46.0880 *	1.8895 *	0.3858 *

*	39	*	43.9112 *	-4.6410 *	-6.0153 *
*	40	*	55.4016 *	1.9175 *	3.1167 *
*	41	*	-48.3875 *	-0.9456 *	-6.1848 *
*	42	*	-38.6098 *	0.1863 *	2.3090 *
*	43	*	-39.2106 *	-1.6161 *	-8.9872 *
*	44	*	-28.8360 *	0.4176 *	-0.1103 *
*	45	*	-29.6660 *	-2.0723 *	-10.9620 *
*	46	*	-18.9644 *	0.6022 *	-1.8989 *
*	47	*	-19.9377 *	-2.3179 *	-12.1914 *
*	48	*	-9.0849 *	0.7582 *	-3.0594 *
*	49	*	-10.1539 *	-2.4490 *	-12.7682 *
*	50	*	0.7798 *	0.8991 *	-3.6085 *
*	750	*	-1.1396 *	-0.8679 *	3.6436 *
*	751	*	-0.7434 *	0.3208 *	5.8444 *
*	752	*	-3.1814 *	-1.2346 *	3.9333 *
*	753	*	-2.6785 *	0.2742 *	5.6054 *
*	754	*	-5.1057 *	-1.5058 *	3.7405 *
*	755	*	-4.5155 *	0.2649 *	4.9574 *
*	756	*	-6.9840 *	-1.7002 *	3.0859 *
*	757	*	-6.3134 *	0.3117 *	3.8916 *
*	758	*	-8.8689 *	-1.8516 *	1.9751 *
*	759	*	-8.1053 *	0.4392 *	2.3864 *
*	760	*	-10.8139 *	-2.0086 *	0.3802 *
*	761	*	12.8663 *	0.4405 *	2.1274 *
*	762	*	6.4218 *	-0.7260 *	-3.4026 *
*	763	*	6.7528 *	0.2668 *	3.5904 *
*	764	*	2.6405 *	-3.0321 *	0.4769 *
*	765	*	3.6604 *	0.0274 *	4.8501 *
*	766	*	0.5068 *	-4.0891 *	2.7202 *

*	767	*	1.8079 *	-0.1856 *	5.6645 *
*	768	*	-0.8619 *	-4.4154 *	3.9824 *
*	769	*	0.5031 *	-0.3203 *	6.0171 *
*	770	*	-1.8639 *	-4.2845 *	4.5605 *
*	771	*	-0.5555 *	-0.3593 *	5.9420 *
*	772	*	-2.6994 *	-3.8799 *	4.6114 *
*	773	*	-1.5083 *	-0.3067 *	5.4739 *
*	774	*	-3.4690 *	-3.3384 *	4.2264 *
*	775	*	-2.4153 *	-0.1773 *	4.6338 *
*	776	*	-4.2071 *	-2.7664 *	3.4643 *
*	777	*	-3.2814 *	0.0109 *	3.4210 *
*	778	*	-4.8817 *	-2.2474 *	2.3677 *
*	779	*	-4.0507 *	0.2458 *	1.7992 *
*	780	*	-5.3654 *	-1.8522 *	0.9733 *
*	781	*	5.9069 *	-0.8976 *	6.1863 *
*	782	*	-3.2769 *	-27.5255 *	2.3628 *
*	783	*	5.1685 *	-2.1895 *	6.8201 *
*	784	*	-2.9677 *	-23.8523 *	3.1194 *
*	785	*	4.0092 *	-2.9216 *	6.6805 *
*	786	*	-2.6486 *	-19.9324 *	3.5360 *
*	787	*	2.9469 *	-3.1458 *	6.2249 *
*	788	*	-2.3072 *	-16.0280 *	3.6553 *
*	789	*	2.0411 *	-2.9829 *	5.6097 *
*	790	*	-1.9910 *	-12.3220 *	3.5488 *
*	791	*	1.2634 *	-2.5590 *	4.8822 *
*	792	*	-1.7371 *	-8.9496 *	3.2578 *
*	793	*	0.5836 *	-1.9876 *	4.0454 *
*	794	*	-1.5637 *	-6.0283 *	2.7963 *
*	795	*	-0.0101 *	-1.3674 *	3.0833 *

*	796	*	-1.4655 *	-3.6764 *	2.1668 *
*	797	*	-0.5026 *	-0.7877 *	1.9718 *
*	798	*	-1.3958 *	-2.0210 *	1.3796 *
*	799	*	-0.8363 *	-0.3425 *	0.6861 *
*	800	*	-1.2169 *	-1.1947 *	0.4868 *

****

Plain Stress case

Elastic matrix D

1.07	0.27	0.00
0.27	1.07	0.00
0.00	0.00	0.40

Displacement Boundary Conditions

node	1	2	3	4	5	6	7	8	9	10	11
	ux=0.0	ux=0.0	ux=0.0	ux=0.0	ux=0.0	ux=0.0	ux=0.0	ux=0.0	ux=0.0	ux=0.0	ux=0.0
	uy=0.0	uy=0.0	uy=0.0	uy=0.0	uy=0.0	uy=0.0	uy=0.0	uy=0.0	uy=0.0	uy=0.0	uy=0.0

Traction BC

node 11 22 33 44 55 66 77 88 99 110

fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0

fy=-0.13 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25

node 121 132 143 154 165 176 187 198 209 220

fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0

fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25

node 231 242 253 264 275 286 297 308 319 330

fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0

fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25

node 341 352 363 374 385 396 407 418 429 440

fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0 fx=0.0

fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25 fy=-0.25

node 242

fx=0.0

fy=-0.25

d i s p l a c e m e n t s

.....

node	ux	uy	node	ux	uy
1	0.0000	0.0000	2	0.0000	0.0000
3	0.0000	0.0000	4	0.0000	0.0000
5	0.0000	0.0000	6	0.0000	0.0000
7	0.0000	0.0000	8	0.0000	0.0000
9	0.0000	0.0000	10	0.0000	0.0000
11	0.0000	0.0000	12	-16.3864	-9.8324
13	-12.2275	-8.6642	14	-8.7647	-7.8718
15	-5.6879	-7.3717	16	-2.8220	-7.1046
17	-0.0495	-7.0371	18	2.7275	-7.1605
19	5.6129	-7.4929	20	8.7490	-8.0842
21	12.3766	-9.0309	22	16.9781	-10.5053
23	-30.3906	-25.2829	24	-23.4122	-24.1533
25	-17.0571	-23.3075	26	-11.1452	-22.7368
27	-5.5107	-22.4280	28	-0.0037	-22.3723

* 29 * 5.5183 * -22.5701 ** 30 * 11.2014 * -23.0317 *

* 31 * 17.2032 * -23.7733 ** 32 * 23.6875 * -24.8030 *

* 33 * 30.7655 * -26.0778 ** 34 * -42.3829 * -46.8065 *

* 35 * -33.0540 * -45.8351 ** 36 * -24.2854 * -45.0921 *

* 37 * -15.9364 * -44.5837 ** 38 * -7.8620 * -44.3122 *

* 39 * 0.0818 * -44.2780 ** 40 * 8.0372 * -44.4816 *

* 41 * 16.1442 * -44.9225 ** 42 * 24.5387 * -45.5969 *

* 43 * 33.3478 * -46.4948 ** 44 * 42.6934 * -47.6059 *

* 45 * -52.4866 * -73.6780 ** 46 * -41.1735 * -72.8716 *

* 47 * -30.3837 * -72.2540 ** 48 * -19.9872 * -71.8321 *

* 49 * -9.8516 * -71.6104 ** 50 * 0.1561 * -71.5910 *

* 399 * 63.2888 * -267.5474 ** 400 * 42.4687 * -267.7483 *

* 401 * 21.7952 * -267.8605 ** 402 * 1.2098 * -267.8939 *

* 403 * -19.3539 * -267.8568 ** 404 * -39.9691 * -267.7541 *

* 405 * -60.7142 * -267.5866 ** 406 * -81.6714 * -267.3511 *

* 407 * -102.9252 * -267.0399 ** 408 * 109.7641 * -212.1333 *

* 409 * 87.5716 * -212.4676 ** 410 * 65.6372 * -212.6902 *

* 411 * 43.9417 * -212.8170 ** 412 * 22.4471 * -212.8722 *

* 413 * 1.0907 * -212.8741 ** 414 * -20.2079 * -212.8332 *

* 415 * -41.5388 * -212.7533 ** 416 * -62.9981 * -212.6336 *

* 417 * -84.6865 * -212.4702 ** 418 * -106.7106 * -212.2558 *

* 419 * 113.3091 * -155.3072 ** 420 * 90.1359 * -155.5966 *

* 421 * 67.3035 * -155.7369 ** 422 * 44.8661 * -155.8215 *

* 423 * 22.7565 * -155.8839 ** 424 * 0.8689 * -155.9319 *

* 425 * -20.9123 * -155.9626 ** 426 * -42.7024 * -155.9692 *

* 427 * -64.6140 * -155.9450 ** 428 * -86.7585 * -155.8849 *

* 429 * -109.2521 * -155.7873 ** 430 * 116.4837 * -95.9473 *

* 431 * 91.7946 * -96.1821 ** 432 * 68.1820 * -96.5411 *

* 433 * 45.2829 * -96.9520 ** 434 * 22.8535 * -97.3623 *

*	435 *	0.7072 *	-97.7350 **	436 *	-21.3093 *	-98.0461 *
*	437 *	-43.3272 *	-98.2826 **	438 *	-65.4602 *	-98.4418 *
*	439 *	-87.8050 *	-98.5302 **	440 *	-110.4406 *	-98.5648 *
*	441 *	117.8681 *	-30.0433 **	442 *	93.0141 *	-32.7269 *
*	443 *	69.1722 *	-35.0429 **	444 *	46.0539 *	-36.9683 *
*	445 *	23.4224 *	-38.5077 **	446 *	1.0871 *	-39.6823 *
*	447 *	-21.1074 *	-40.5255 **	448 *	-43.2896 *	-41.0817 *
*	449 *	-65.5641 *	-41.4069 **	450 *	-88.0073 *	-41.5709 *
*	451 *	-110.6540 *	-41.6606 **			

****

s t r e s s e s

.....

*	element	*	sx	*	sy	*	sxy	*
*	1	*	-69.9154	*	-17.4789	*	-15.7319	*
*	2	*	-49.0553	*	-0.5818	*	2.7731	*
*	3	*	-52.1705	*	-13.0426	*	-13.8628	*
*	4	*	-35.2828	*	-0.8967	*	1.2563	*
*	5	*	-37.3959	*	-9.3490	*	-12.5949	*
*	6	*	-22.9347	*	-0.7327	*	0.5124	*
*	7	*	-24.2683	*	-6.0671	*	-11.7947	*
*	8	*	-11.3284	*	-0.1612	*	0.0960	*
*	9	*	-12.0406	*	-3.0102	*	-11.3674	*
*	10	*	-0.0310	*	0.6679	*	-0.1692	*

*	11	*	-0.2112 *	-0.0528 *	-11.2593 *
*	12	*	11.3079 *	1.5922 *	-0.3490 *
*	13	*	11.6372 *	2.9093 *	-11.4569 *
*	14	*	23.0623 *	2.4421 *	-0.4467 *
*	15	*	23.9485 *	5.9871 *	-11.9886 *
*	16	*	35.7523 *	3.0250 *	-0.3904 *
*	17	*	37.3292 *	9.3323 *	-12.9347 *
*	18	*	50.2823 *	3.1039 *	0.0609 *
*	19	*	52.8068 *	13.2017 *	-14.4494 *
*	20	*	68.5078 *	2.3823 *	1.5973 *
*	21	*	-56.6361 *	-2.4770 *	-8.0849 *
*	22	*	-44.7092 *	0.1185 *	3.1314 *
*	23	*	-45.6083 *	-3.4780 *	-10.9314 *
*	24	*	-33.1254 *	0.1767 *	0.7233 *
*	25	*	-34.0473 *	-3.5108 *	-12.3899 *
*	26	*	-21.7628 *	0.2659 *	-0.9367 *
*	27	*	-22.5723 *	-2.9722 *	-13.1207 *
*	28	*	-10.6480 *	0.4264 *	-1.9793 *
*	29	*	-11.2914 *	-2.1472 *	-13.4273 *
*	30	*	0.3440 *	0.6430 *	-2.5084 *
*	31	*	-0.1338 *	-1.2682 *	-13.4285 *
*	32	*	11.3802 *	0.8670 *	-2.5672 *
*	33	*	11.0214 *	-0.5682 *	-13.1134 *
*	34	*	22.6133 *	1.0377 *	-2.1297 *
*	35	*	22.2673 *	-0.3463 *	-12.3177 *
*	36	*	34.0933 *	1.1070 *	-1.0955 *
*	37	*	33.5466 *	-1.0801 *	-10.5923 *
*	38	*	45.5137 *	1.0809 *	0.7017 *
*	39	*	44.3278 *	-3.6627 *	-6.8296 *

*	40	*	55.4269 *	1.1093 *	3.3961 *
*	41	*	-48.1547 *	-0.7429 *	-6.5239 *
*	42	*	-38.5481 *	0.0773 *	2.6246 *
*	43	*	-38.8831 *	-1.2627 *	-9.2704 *
*	44	*	-28.8596 *	0.2152 *	0.2191 *
*	45	*	-29.3192 *	-1.6232 *	-11.2078 *
*	46	*	-19.0869 *	0.3119 *	-1.5589 *
*	47	*	-19.6189 *	-1.8163 *	-12.4169 *
*	48	*	-9.3084 *	0.3883 *	-2.7172 *
*	49	*	-9.8839 *	-1.9140 *	-12.9867 *
*	50	*	0.4557 *	0.4552 *	-3.2739 *
*	750	*	-1.0745 *	-0.7490 *	3.5569 *
*	751	*	-0.8371 *	0.2003 *	5.9132 *
*	752	*	-3.0874 *	-1.0786 *	3.8681 *
*	753	*	-2.7925 *	0.1011 *	5.6694 *
*	754	*	-4.9826 *	-1.3118 *	3.6940 *
*	755	*	-4.6460 *	0.0346 *	5.0172 *
*	756	*	-6.8301 *	-1.4650 *	3.0555 *
*	757	*	-6.4591 *	0.0193 *	3.9481 *
*	758	*	-8.6804 *	-1.5696 *	1.9586 *
*	759	*	-8.2687 *	0.0772 *	2.4401 *
*	760	*	-10.5836 *	-1.6697 *	0.3751 *
*	761	*	12.7732 *	0.2991 *	2.2833 *
*	762	*	6.4509 *	-0.7351 *	-3.6931 *
*	763	*	6.7028 *	0.2729 *	3.7337 *
*	764	*	2.7908 *	-2.8932 *	0.2631 *
*	765	*	3.5228 *	0.0349 *	4.9637 *
*	766	*	0.6826 *	-3.9378 *	2.5947 *
*	767	*	1.6118 *	-0.2212 *	5.7527 *

*	768	*	-0.6805 *	-4.2734 *	3.9168 *
*	769	*	0.2855 *	-0.4090 *	6.0840 *
*	770	*	-1.6835 *	-4.1481 *	4.5301 *
*	771	*	-0.7714 *	-0.4996 *	5.9902 *
*	772	*	-2.5230 *	-3.7410 *	4.6004 *
*	773	*	-1.7112 *	-0.4939 *	5.5060 *
*	774	*	-3.2966 *	-3.1896 *	4.2267 *
*	775	*	-2.6011 *	-0.4078 *	4.6522 *
*	776	*	-4.0347 *	-2.6007 *	3.4734 *
*	777	*	-3.4500 *	-0.2620 *	3.4272 *
*	778	*	-4.7005 *	-2.0588 *	2.3884 *
*	779	*	-4.2045 *	-0.0750 *	1.7931 *
*	780	*	-5.1632 *	-1.6367 *	1.0134 *
*	781	*	5.2809 *	-1.0276 *	6.6900 *
*	782	*	-1.9528 *	-27.3240 *	2.1123 *
*	783	*	4.2459 *	-2.5294 *	7.0781 *
*	784	*	-1.9512 *	-23.6478 *	3.0296 *
*	785	*	3.1292 *	-3.3262 *	6.8007 *
*	786	*	-1.8448 *	-19.7157 *	3.5008 *
*	787	*	2.1955 *	-3.5544 *	6.2561 *
*	788	*	-1.6777 *	-15.8130 *	3.6410 *
*	789	*	1.4333 *	-3.3688 *	5.5825 *
*	790	*	-1.5116 *	-12.1236 *	3.5433 *
*	791	*	0.7912 *	-2.9124 *	4.8184 *
*	792	*	-1.3871 *	-8.7789 *	3.2551 *
*	793	*	0.2307 *	-2.3077 *	3.9610 *
*	794	*	-1.3225 *	-5.8924 *	2.7926 *
*	795	*	-0.2639 *	-1.6580 *	2.9898 *
*	796	*	-1.3105 *	-3.5795 *	2.1580 *

```
*    797    *    -0.6790 *    -1.0535 *     1.8767 *
*    798    *    -1.3006 *    -1.9656 *     1.3621 *
*    799    *    -0.9554 *    -0.5847 *     0.5921 *
*    800    *    -1.1494 *    -1.1839 *     0.4598 *
************************************************************
C234567
C A FINITE ELEMENT CODE FOR SOLVING PLANE ELASTICITY
C PISS=1 SOLVES PLANE STRAIN PROBLEM
C PISS NON EQUAL TO 1 SOLVES THE PLANE STRESS PROBLEM

C234567
      PROGRAM PROJECT2
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION IELMN(800,3),SLM(6,6),GSM(902,24),KK(6),CORD(451,2),
     *B(3,6),B1(3,6),BT(6,3),D(3,3),IGG(22),F(902),STRAIN(3,1),
     *UEL(6),STRESS(3,1)



C     ELASTIC CONSTANTS EY=YOUNG MODULUS, V=POISSON'S RATIO, D=ELASIC MATRIX

      PISS=11.0
      IF(PISS.EQ.1)GO TO 6
      WRITE(6,99)
      WIDTH=1.0
      GO TO 7
 6    WRITE(6,98)
      WIDTH=1.0
 7    EY=210000
```

```
      V=0.25
      CALL ELASTI (EY,V,PISS,D)
      WRITE(6,100) ((D(I,J),J=1,3),I=1,3)

C     DADA----GEOMETRY OF THE BODY
C     XL------LENGTH OF THE BODY
C     YL------HEIGHT OF THE BODY
C     NGP-----# OF GRID POINTS:::NEL=# OF ELEMENTS
C     IBAND-----BANDWIDTH
C     H-----DISPLACEMENT BOUNDARY LENGTH
C     QP-----APPLIED TRACTION IN THE Y-DIRECTION
C
C
C      PI=DATAN(1.0D0)*4.0D0
C
C
C234567
      XL=10.0
      YL=1.0
      H=YL
      QP=-10.0
      N=40
      M=10

      DX=XL/N
      DY=YL/M
      XII=0.25
      N1=N+1
      M1=M+1
```

```
      M22=M1*2
      NGP=N1*M1
      NGP2=NGP*2
      IBAND=(M1+1)*2
      NEL=M*N*2

C MATRIX DESCRIPTION
C CORD(NGP,2)----COORDINATE'S MATRIX
C IELMN(NEL,3)----ASSOCIATES EACH ELEMENT WITH ITS NODES
C B1=D*B(3,6)
C BT(6,3)----TRANSPOSE OF B1
C SLM(6,6)----ELEMENT STIFFNESS MATRIX IN GLOBAL SENSE
C GSM(NGP2,IBAND)----GLOBAL STIFFNESS MATRIX
C F(NGP2)-----GLOBAL FORCE VECTOR

C234567
      DO 8 I=1,NGP2
      DO 8 J=1,IBAND
 8    GSM(I,J)=0.0
      CALL CRDIXY(M1,N1,NGP,DX,DY,CORD)
      CALL INODES(M,N,M1,NEL,IELMN)

      DO 40 L=1,NEL
      CALL BETA (L,CORD,IELMN,NGP,NEL,DA,B)
      CALL MULT (D,B,B1,3,3,6)
      DO 41 I=1,3
      DO 41 J=1,6
 41   BT(J,I)=B(I,J)
      CALL MULT(BT,B1,SLM,3,6,6)
```

```
      DO 42 IS=1,6
      DO 42 JS=1,6
   42 SLM(IS,JS)=SLM(IS,JS)*DA*WIDTH/2.0
      CALL ASSEMB(IELMN,SLM,NEL,NGP2,IBAND,L,GSM)
   40 CONTINUE

C APPLY GEOMETRIC BCS

      DO 50 I=1,NGP2
   50 F(I)=0.0
      NGP3=NGP-M1
      DO 60 I=M1,NGP3,M1
      F(2*I-1)=F(2*I-1)+0.0
      F(2*I)=F(2*I)+QP*XII/2.0
      F(2*(I+M1)-1)=F(2*(I+M1)-1)+0.0
   60 F(2*(I+M1))=F(2*(I+M1))+QP*XII/2.0
      DO 70 I=1,M1
      SOF=(I-1)*DY+H
      IF(SOF.LT.YL) GO TO 70
      IGG(2*I-1)=2*I-1
      IGG(2*I)=2*I
   70 CONTINUE
      WRITE(6,101)
      WRITE(6,102)(I,I=1,M1)
      WRITE(6,103)
      WRITE(6,104)
      DO 80 KX=1,4
      WRITE(6,105)(((I-1)*M1+11),I=((KX-1)*10+1),KX*10)
   80 WRITE(6,106)(F(((I-1)*M1+11)*2),I=((KX-1)*10+1),KX*10)
```

```
      I=451
      WRITE(6,107) I,F(I*2)

      CALL BOUNDA (M22,NGP2,IBAND,IGG,GSM,F)

C SOLVE FOR NODAL DISPLACEMENTS

      CALL HALLEY (1,GSM,F,NGP2,IBAND)
      CALL HALLEY (2,GSM,F,NGP2,IBAND)
      WRITE(6,111)
      WRITE(6,112)(I,F(2*I-1),F(2*I),I=1,NGP)

C CALCULATION OF THE STRESS FIELD STRESS=STRAIN *D

      WRITE(6,113)
      DO 200 I=1,NEL
      DO 199 J=1,3
      JPN=IELMN(I,J)
      UEL(2*J-1)=F(2*JPN-1)
  199 UEL(2*J)=F(2*JPN)

      CALL BETA (I,CORD,IELMN,NGP,NEL,DA,B)
      CALL MULT (B,UEL,STRAIN,6,3,1)
      CALL MULT (D,STRAIN,STRESS,3,3,1)

      WRITE(6,114)I,(STRESS(IT,1),IT=1,3)
  200 CONTINUE
      WRITE(6,115)
```

```
98   FORMAT(//,10X,'PLAIN STRAIN CASE')

99   FORMAT(//,10X,'PLAIN STRESS CASE')

100  FORMAT(//,10X,'ELASTIC MATRIX D',/(35X,3(F10.2,2X)))

101  FORMAT(5(/),10X,'DISPLACEMENT BOUNDARY CONDITIONS')

102  FORMAT(//,4X,'NODE',11(4X,I2,4X))

103  FORMAT(/,10X,11(2X,'UX=0.0',2X)/10X,11(2X,'UY=0.0',2X))

104  FORMAT(5(/),10X,'TRACTION BC')

105  FORMAT(//,4X,'NODE',10(4X,I3,4X))

106  FORMAT(/,10X,10(2X,'FX=0.0',3X)/10X,10(2X,'FY=',F5.2,1X))

107  FORMAT(//,4X,'NODE',4X,I3,//,12X,'FX=0.0'/12X,'FY=',F5.2)

111  FORMAT(///,47X,'D',2X,'I',2X,'S',2X,'P',2X,'L',2X,'A',2X,'C',
    *2X,'E',2X,'M',2X,'E',2X,'N',2X,'T',2X,'S',/,40X,50('.'),5(/),
    :28X,76('*'),/,28X,2('*',6X,'*',2(14X,'*')),/,28X,2('*',1X,'NODE',
    *1X,'*',6X,'UX',6X,'*',6X,'UY',
    *6X,'*'),/,28X,2('*',6X,'*',2(14X,'*')),/,28X,76('*'))
```

```
 112  FORMAT((28X,2('*',1X,I4,1X,'*',2(1X,F12.8,1X,'*'))))

 113  FORMAT(28X,76('*'),6(/),55X,'S',2X,'T',2X,'R',2X,'E',2X,'S',2X,
     *'S',2X,'E',2X,'S',/,50X,32('.'),5(/),35X,61('*'),/,35X,'*',4(14X,
     *'*'),/,35X,'*',
     *4X,'ELEMENT',3X,'*',6X,'SX',6X,'*',6X,'SY',6X,'*',6X,'SXY',5X,'*',
     */,35X,'*',4(14X,'*'),/,35X,61('*'))

 114  FORMAT((35X,'*',5X,I4,5X,'*',3(1X,F12.4,1X,'*')))
 115  FORMAT(35X,61('*'))

      STOP
      END
C
C*****************************************************
C
      SUBROUTINE CRDIXY(M1,N1,NGP,DX,DY,CORD)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION CORD(NGP,2)
C
      DO 1 I=1,M1
      DO 1 J=1,N1

        JP=(J-1)*M1+I
        CORD(JP,1)=(J-1)*DX
        CORD(JP,2)=(I-1)*DY
 1      CONTINUE
C
```

```
C
      WRITE(6,100)(I,(CORD(I,J),J=1,2),I=1,NGP)
 100  FORMAT("COORDINATE MATRIX",/,((I5,2X,F12.5,F12.5)))
C
C
      RETURN
      END
C
C*****************************************************
C
      SUBROUTINE INODES(M,N,M1,NEL,IELMN)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION IELMN(NEL,3)
C
C
      DO 10 J=1,N
      LP=(J-1)*M1
      KP=LP+1
      DO 10 I=1,M
      KK1=(J-1)*M*2+2*I-1
      KK2=KK1+1
      LP=LP+1
      KP=KP+1
C
C
      IELMN(KK1,1)=LP
      IELMN(KK1,2)=LP+M+1
      IELMN(KK1,3)=LP+1
```

```
      IELMN(KK2,1)=KP
      IELMN(KK2,2)=KP+M
      IELMN(KK2,3)=KP+M+1

   10 CONTINUE

      WRITE(6,200) (I,(IELMN(I,J),J=1,3),I=1,NEL)
  200 FORMAT("CONNECTIVITY MATRIX",/,((I5,2X,I5,2X,I5,2X,I5)))
      RETURN
      END

C
C*************************************************************
C ASSEMBLES THE GLOBAL STIFFNESS MATRIX

      SUBROUTINE ASSEMB(IELMN,SLM,NEL,NGP2,IBAND,L,GSM)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION IELMN(NEL,3),SLM(6,6),GSM(NGP2,IBAND),KK(6)

      DO 10 INODE=1,3
      II=2*INODE
      KK(II)=2*IELMN(L,INODE)
      KK(II-1)=KK(II)-1
   10 CONTINUE

      DO 30 I=1,6
      K=KK(I)
      DO 30 J=1,6
      IF(KK(J).LT.K) GO TO 30
```

```
      LM=KK(J)-K+1
      GSM(K,LM)=GSM(K,LM)+SLM(I,J)
   30 CONTINUE

      RETURN
      END
C
C*************************************************************
C COMPUTES THE B ELEMENT MATRIX

      SUBROUTINE BETA(L,CORD,IELMN,NGP,NEL,DA,B)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION CORD(NGP,2),IELMN(NEL,3),B(3,6),X(3),Y(3)

      DO 3 I=1,3

      MS=IELMN(L,I)

      X(I)=CORD(MS,1)

    3 Y(I)=CORD(MS,2)

      DA=X(1)*Y(2)-X(1)*Y(3)+X(2)*Y(3)-X(2)*Y(1)+X(3)*Y(1)-X(3)*Y(2)

      DO 4 I=1,3

      DO 4 J=1,6

    4 B(I,J)=0.0
```

```
      B(1,1)=(Y(2)-Y(3))/DA

      B(1,3)=(Y(3)-Y(1))/DA


      B(1,5)=(Y(1)-Y(2))/DA

      B(2,2)=(X(3)-X(2))/DA

      B(2,4)=(X(1)-X(3))/DA

      B(2,6)=(X(2)-X(1))/DA

      DO 5 I=1,5,2

 5    B(3,I)=B(2,I+1)

      DO 6 I=2,6,2

 6    B(3,I)=B(1,I-1)

      RETURN
      END
C
C*************************************************************
C MULTIPLICATION OF TWO MATRICES OF THE FORM S(M4,L4)*Q(L4,N4)

      SUBROUTINE MULT(S,Q,C,L4,M4,N4)
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION S(M4,L4),C(M4,N4),Q(L4,N4)

      DO 10 I=1,M4
      DO 10 J=1,N4
      C(I,J)=0.0
      DO 20 KY=I,L4
   20 C(I,J)=C(I,J)+S(I,KY)*Q(KY,J)
   10 CONTINUE

   90 RETURN
      END
C
C*************************************************************
C COMPUTATION OF ELASTIC MATRIX...D...
      SUBROUTINE ELASTI (EY,V,PISS,D)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION D(3,3)

      IF(PISS.EQ.1) GO TO 1

      DD=EY/(1.0-V**2)

      DOF=V*DD

      D3=EY/(2*(1.0+V))

      GO TO 2
```

```
1    DD=(EY*(1.0-V))/((1.0+V)*(1.0-2.0*V))

     DOF=V*DD/(1.0-V)

     D3=EY/(2.0*(1.0+V))

2    DO 3 I=1,2

     DO 3 J=1,2

     IF(I.EQ.J) GO TO 4

     D(I,J)=DOF

     GO TO 3

4    D(I,J)=DD

3    CONTINUE

     DO 5 I=1,2

     D(I,3)=0.0

5    D(3,I)=0.0

     D(3,3)=D3
     RETURN
     END
```

```
C
C************************************************************
C IMPOSING BC---UNSCRAMBLING THE SYSTEM OF EQNS
C234567
      SUBROUTINE BOUNDA (M22,NGP2,IBAND,IGG,GSM,F)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION IGG(M22),GSM(NGP2,IBAND),F(NGP2)
C
      DO 20 I=1,M22
      KM=IGG(I)
      F(KM)=0.0
      GSM(KM,1)=1.0
C
      DO 20 J=2,IBAND
      KMJ=KM-J+1
      IF(KMJ.LE.0) GO TO 21
      F(KMJ)=F(KMJ)-GSM(KMJ,J)*F(KM)
      GSM(KMJ,J)=0.0
C
 21   KMJ=KM+J-1
      IF(KMJ.GT.NGP2) GO TO 20
      F(KMJ)=F(KMJ)-GSM(KM,J)*F(KM)
      GSM(KM,J)=0.0
 20   CONTINUE
C
      RETURN
      END

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
C234567
  320  Q(N)=Q(N)-AK(N,K)*Q(L)
  400  RETURN
      END
call station (im2,z,w)call station (im2,z,w)call station (im2,z,w)c234567
c A finite element code for solving plane elasticity
c Piss=1 Solves Plane Strain Problem
c Piss Non Equal to 1 solves the plane stress problem

c234567
      program project2
      implicit real*8(a-h,o-z)
      dimension ielmn(400,4),slm(8,8),gsm(902,26),kk(8),cord(451,2)
```

```
      dimension b(3,8),b1(3,8),bt(8,3),d(3,3),igg(22),f(902),strain(3,1)
      dimension uel(8,1),stress(3,1),xq(4),yq(4),dj(2,2)
c     dimension slmsum(8,8)



c     Elastic Constants EY=Young Modulus, v=Poisson's Ratio, D=Elasic Matrix

      piss=11.0
      if(piss.eq.1)go to 6
      write(6,99)
      width=1.0
      go to 7
 6    write(6,98)
      width=1.0
 7    ey=1.0
      v=0.30
      call elasti (ey,v,piss,d)
      write(6,100) ((d(i,j),j=1,3),i=1,3)

c     dada----geometry of the body
c     xl------length of the body
c     yl------height of the body
c     ngp-----# of grid points:::nel=# of elements
c     iband-----bandwidth
c     h-----displacement boundary length
c     qp-----applied traction in the y-direction
c
c
```

```
c      pi=datan(1.0d0)*4.0d0
c
c
c234567
      xl=10.0
      yl=1.0
      h=yl
      qp=-1.0
      n=40
      m=10

      dx=xl/n
      dy=yl/m
      xii=0.25
      n1=n+1
      m1=m+1
      m22=m1*2
      ngp=n1*m1
      ngp2=ngp*2
      iband=(m1+2)*2
      nel=m*n

c Matrix Description
c cord(ngp,2)----coordinate's matrix
c ielmn(nel,3)----associates each element with its nodes
c b1=d*b(3,6)
c bt(6,3)----transpose of b1
c slm(6,6)----element stiffness matrix in global sense
c gsm(ngp2,iband)----global stiffness matrix
```

```
c f(ngp2)-----global force vector


c234567
      do 8 i=1,ngp2
      do 8 j=1,iband
 8    gsm(i,j)=0.0
      call inodes(m,n,m1,nel,ielmn)
      call crdixy(m1,n1,ngp,dx,dy,cord)
      do 40 l=1,nel
c     slmsum(is,js)=0.0d0
      do 40  im=1,4
c     call localuxy(l,ielmn,f,nel,ngp2,ulocal)
      call station (im,z,w)
      call beta(l,cord,ielmn,z,w,ngp,nel,b,da)
      call elasti (ey,v,piss,d)
      call mult(d,b,b1,3,3,8)
      do 41 i=1,3
      do 41 j=1,8
 41   bt(j,i)=b(i,j)
      call mult(bt,b1,slm,3,8,8)
      do 43 is=1,8
      do 43 js=1,8
c     slmsum(is,js)=0.0d0
c     do 42 ls=1,3
c     do 42 ms=1,8
c      do iss=1,8
c      do jss=1,8

 43   slm(is,js)=slm(is,js)*da*width
```

```
c      slmsum(is,js)=0.0d0
c      slmsum(is,js)=slmsum(is,js)+slm(is,js)*da*width
c      call assemb(ielmn,slm,nel,ngp2,iband,l,gsm)
c      enddo
c      enddo
       call assemb(ielmn,slm,nel,ngp2,iband,l,gsm)
 40    continue
c      call assemb(ielmn,slm,nel,ngp2,iband,l,gsm)

c      call  assemb(ielmn,slm,nel,ngp2,iband,l,gsm)
c      write(6,120)z,w,determ
c 120  format(3(f8.3,2x))
c      write(6,121)((B(i,j),j=1,8),i=1,3)
c 121  format(/,8(f8.4,1x))



c



c Apply Geometric BCs

       do 50 i=1,ngp2
 50    f(i)=0.0
       ngp3=ngp-m1
       do 60 i=m1,ngp3,m1
       f(2*i-1)=f(2*i-1)+0.0

       f(2*i)=f(2*i)+qp*xii/2.0
       f(2*(i+m1)-1)=f(2*(i+m1)-1)+0.0
```

```
      f((2*ngp)-(2*m))=-(3*qp*xl)/8
   60 f(2*(i+m1))=f(2*(i+m1))+qp*xii/2.0
      do 70 i=1,m1
      sof=(i-1)*dy+h
      if(sof.lt.yl) go to 70
      igg(2*i-1)=2*i-1
      igg(2*i)=2*i
   70 continue
      write(6,101)
      write(6,102)(i,i=1,m1)
      write(6,103)
      write(6,104)
      do 80 kx=1,4
      write(6,105)(((i-1)*m1+11),i=((kx-1)*10+1),kx*10)
   80 write(6,106)(f(((i-1)*m1+11)*2),i=((kx-1)*10+1),kx*10)
      i=451
      write(6,107) i,f(i*2)

      call bounda (m22,ngp2,iband,igg,gsm,f)

c Solve for nodal displacements

      call halley (1,gsm,f,ngp2,iband)
      call halley (2,gsm,f,ngp2,iband)
      write(6,111)
      write(6,112)(i,f(2*i-1),f(2*i),i=1,ngp)
```

```
c calculation of the stress field stress=strain *d

      write(6,113)
      do 200 i=1,nel
      do 200 im2=1,4
      do 199 j=1,4
      jpn=ielmn(i,j)
      uel(2*j-1,1)=f(2*jpn-1)
 199  uel(2*j,1)=f(2*jpn)
      call station (im2,z,w)
      call beta (i,cord,ielmn,z,w,ngp,nel,b,da)
      call elasti (ey,v,piss,d)
      call mult (b,uel,strain,8,3,1)
      call mult (d,strain,stress,3,3,1)
      write(6,114)i,(stress(it,1),it=1,3)
 200  continue

      write(6,115)
c234567
 98   format(//,10x,'Plain Strain case')

 99   format(//,10x,'Plain Stress case')

 100  format(//,10x,'Elastic matrix D',/(35x,3(f10.2,2x)))

 101  format(5(/),10x,'Displacement Boundary Conditions')

 102  format(//,4x,'node',11(4x,i2,4x))
```

```
103  format(/,10x,11(2x,'ux=0.0',2x)/10x,11(2x,'uy=0.0',2x))

104  format(5(/),10x,'Traction BC')

105  format(//,4x,'node',10(4x,i3,4x))

106  format(/,10x,10(2x,'fx=0.0',3x)/10x,10(2x,'fy=',f5.2,1x))

107  format(//,4x,'node',4x,i3,//,12x,'fx=0.0'/12x,'fy=',f5.2)

111  format(///,47x,'d',2x,'i',2x,'s',2x,'p',2x,'l',2x,'a',2x,'c',
    *2x,'e',2x,'m',2x,'e',2x,'n',2x,'t',2x,'s',/,40x,50('.'),5(/),
    :28x,76('*'),/,28x,2('*',6x,'*',2(14x,'*')),/,28x,2('*',1x,'node',
    *1x,'*',6x,'ux',6x,'*',6x,'uy',
    *6x,'*'),/,28x,2('*',6x,'*',2(14x,'*')),/,28x,76('*'))

112  format((28x,2('*',1x,i4,1x,'*',2(1x,f12.4,1x,'*'))))

113  format(28x,76('*'),6(/),55x,'s',2x,'t',2x,'r',2x,'e',2x,'s',2x,
    *'s',2x,'e',2x,'s',/,50x,32('.'),5(/),35x,61('*'),/,35x,'*',4(14x,
    *'*'),/,35x,'*',
    *4x,'element',3x,'*',6x,'sx',6x,'*',6x,'sy',6x,'*',6x,'sxy',5x,'*',
    */,35x,'*',4(14x,'*'),/,35x,61('*'))
```

```
 114  format((35x,'*',5x,i4,5x,'*',3(1x,f12.4,1x,'*')))
 115  format(35x,61('*'))

      stop
      end
c
c*************************************************
c
      subroutine crdixy(m1,n1,ngp,dx,dy,cord)
      implicit real*8(a-h,o-z)
      dimension cord(ngp,2)
c
      do 1 i=1,m1
      do 1 j=1,n1

      jp=(j-1)*m1+i
      cord(jp,1)=(j-1)*dx
      cord(jp,2)=(i-1)*dy
 1    continue

c     do k=1,n
c     cord(2,1)=cord(2,1)-dx
c     cord(2,2)=-cord(2,1)
c     enddo
c
c
      write(6,100)(i,(cord(i,j),j=1,2),i=1,ngp)
 100  format("Coordinate Matrix",/,((i5,2x,f12.5,f12.5)))
c
```

```
c
      return
      end
c
c*************************************************
c
      subroutine inodes(m,n,m1,nel,ielmn)
      implicit real*8(a-h,o-z)
      dimension ielmn(nel,4)



      do 10  i=1,n
      do 10  j=1,m
      kl=m*(i-1)+j
      kn1=(i-1)*m1+j
      kn2=kn1+m1
      kn3=kn2+1
      kn4=kn1+1

      ielmn(kl,1)=kn1
      ielmn(kl,2)=kn2
      ielmn(kl,3)=kn3
      ielmn(kl,4)=kn4

 10   continue
      write(6,200) (i,(ielmn(i,j),j=1,4),i=1,nel)
 200  format("Connectivity Matrix",/,((i5,2x,i5,2x,i5,2x,i5,2x,i5)))
      return
      end
```

```
c
c************************************************************
c Assembles the global stiffness matrix

      subroutine assemb(ielmn,slm,nel,ngp2,iband,l,gsm)
      implicit real*8(a-h,o-z)
      dimension ielmn(nel,4),slm(8,8),gsm(ngp2,iband),kk(8)

      do 10 inode=1,4
        ii=2*inode
        kk(ii)=2*ielmn(l,inode)
        kk(ii-1)=kk(ii)-1
 10     continue

        do 30 i=1,8
        k=kk(i)
        do 30 j=1,8
        if(kk(j).lt.k) go to 30
        lm=kk(j)-k+1
        gsm(k,lm)=gsm(k,lm)+slm(i,j)
 30     continue

        return
        end
c
c************************************************************
c computes the b element matrix
c234567
```

```
      subroutine beta(l,cord,ielmn,z,w,ngp,nel,b,da)
      implicit real*8(a-h,o-z)
      dimension cord(ngp,2),ielmn(nel,4),b(3,8),xq(4),yq(4),dj(2,2)

      do 1 i=1,4

      ms=ielmn(l,i)

      xq(i)=cord(ms,1)

1     yq(i)=cord(ms,2)

      d11=((w-1.0d0)*(xq(1)-xq(2))+(w+1.0d0)*(xq(3)-xq(4)))/4.0

      d12=((w-1.0d0)*(yq(1)-yq(2))+(w+1.0d0)*(yq(3)-yq(4)))/4.0

      d21=((z-1.0d0)*(xq(1)-xq(4))+(z+1.0d0)*(xq(3)-xq(2)))/4.0

      d22=((z-1.0d0)*(yq(1)-yq(4))+(z+1.0d0)*(yq(3)-yq(2)))/4.0

      da=d11*d22-d12*d21

      dj(1,1)=d22/da
      dj(1,2)=-d12/da
      dj(2,1)=-d21/da
      dj(2,2)=d11/da

      do 2 ks=1,2
```

```
      do 2 kl=1,8
    2 b(ks,kl)=0.0
      b(1,1)=(dj(1,1)*(w-1.0d0)+dj(1,2)*(z-1.0d0))/4.0
      b(2,2)=(dj(2,1)*(w-1.0d0)+dj(2,2)*(z-1.0d0))/4.0
      b(1,3)=-(dj(1,1)*(w-1.0d0)+dj(1,2)*(z+1.0d0))/4.0
      b(2,4)=-(dj(2,1)*(w-1.0d0)+dj(2,2)*(z+1.0d0))/4.0
      b(1,5)=(dj(1,1)*(w+1.0d0)+dj(1,2)*(z+1.0d0))/4.0
      b(2,6)=(dj(2,1)*(w+1.0d0)+dj(2,2)*(z+1.0d0))/4.0
      b(1,7)=-(dj(1,1)*(w+1.0d0)+dj(1,2)*(z-1.0d0))/4.0
      b(2,8)=-(dj(2,1)*(w+1.0d0)+dj(2,2)*(z-1.0d0))/4.0
      do 3 ls=1,7,2
      b(3,ls)=b(2,ls+1)
    3 b(3,ls+1)=b(1,ls)
c     write(6,200)(xq(i),yq(i),i=1,4)
c 200  format(2(f10.3,2x))
c     write(6,201)z,w,determ
c 201  format(3(f8.3,2x))
```

```
      return
      end
c
c
c*************************************************************
c subroutine station chooses the values of z and w to be used
c In Gauss Integration for each integration station
c234567
      subroutine station (ms,z,w)
      implicit real*8(a-h,o-z)
c
      cs1=1.0/sqrt(3.0)
      if (ms-2)1,2,3
 1    z=cs1
      w=z
      go to 5
 2    z=-cs1
      w=-z
      go to 5
 3    if(ms.gt.3) go to 4
      z=-cs1
      w=z
      go to 5
 4    z=cs1
      w=-z
c
 5    return
      end
c*************************************************************
```

```
c Multiplication of two matrices of the form s(m4,l4)*q(l4,n4)

      subroutine mult(s,q,c,l4,m4,n4)
      implicit real*8(a-h,o-z)
      dimension s(m4,l4),c(m4,n4),q(l4,n4)

      do 10 i=1,m4
      do 10 j=1,n4
      c(i,j)=0.0
      do 20 ky=1,l4
   20 c(i,j)=c(i,j)+s(i,ky)*q(ky,j)
   10 continue

   90 return
      end
c
c*************************************************************
c computation of elastic matrix...d...
      subroutine elasti (ey,v,piss,d)
      implicit real*8(a-h,o-z)
      dimension d(3,3)

      if(piss.eq.1) go to 1

      dd=ey/(1.0-v**2)

      dof=v*dd

      d3=ey/(2*(1.0+v))
```

```
      go to 2
1     dd=(ey*(1.0-v))/((1.0+v)*(1.0-2.0*v))
      dof=v*dd/(1.0-v)
      d3=ey/(2.0*(1.0+v))
2     do 3 i=1,2
      do 3 j=1,2
      if(i.eq.j) go to 4
      d(i,j)=dof
      go to 3
4     d(i,j)=dd
3     continue
      do 5 i=1,2
      d(i,3)=0.0
5     d(3,i)=0.0
```

```
      d(3,3)=d3
      return
      end
c
c*************************************************************
c Imposing BC---Unscrambling the system of eqns
c234567
      subroutine bounda (m22,ngp2,iband,igg,gsm,f)
      implicit real*8(a-h,o-z)
      dimension igg(m22),gsm(ngp2,iband),f(ngp2)
c
      do 20 i=1,m22
      km=igg(i)
      f(km)=0.0
      gsm(km,1)=1.0
c
      do 20 j=2,iband
      kmj=km-j+1
      if(kmj.le.0) go to 21
      f(kmj)=f(kmj)-gsm(kmj,j)*f(km)
      gsm(kmj,j)=0.0
c
 21   kmj=km+j-1
      if(kmj.gt.ngp2) go to 20
      f(kmj)=f(kmj)-gsm(km,j)*f(km)
      gsm(km,j)=0.0
 20   continue
c
      return
```

```
      end


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
c234567
 320  q(n)=q(n)-ak(n,k)*q(l)
 400  return
      end

c*************************************************************
```

MATLAB Code

```
cord =  xlsread('cord');
ielmn = xlsread('ielmn');
disp = xlsread('displacements');
defcord = zeros(451,2);

for j = 2:3
    for k = 1:451
       defcord(k,(j-1)) = (cord(k,(j-1))) + ((disp(k,j))/5000);
    end
end


for i = 1:length(ielmn)
    xu = [cord(ielmn(i,1),1),cord(ielmn(i,2),1)];
    yu = [cord(ielmn(i,1),2),cord(ielmn(i,2),2)];
    plot(xu,yu,'k'); hold on;
    xlim([0 10.5]);
    ylim([-.5 1.5]);

    xd = [defcord(ielmn(i,1),1),defcord(ielmn(i,2),1)];
    yd = [defcord(ielmn(i,1),2),defcord(ielmn(i,2),2)];
    plot(xd,yd,'c'); hold on;
end


for i = 1:length(ielmn)
    xu = [cord(ielmn(i,2),1),cord(ielmn(i,3),1)];
    yd = [cord(ielmn(i,2),2),cord(ielmn(i,3),2)];
    plot(xu,yd,'k'); hold on;

    xd = [defcord(ielmn(i,2),1),defcord(ielmn(i,3),1)];
    yd = [defcord(ielmn(i,2),2),defcord(ielmn(i,3),2)];
    plot(xd,yd,'c'); hold on;
end


for i = 1:length(ielmn)
    xu = [cord(ielmn(i,3),1),cord(ielmn(i,4),1)];
    yu = [cord(ielmn(i,3),2),cord(ielmn(i,4),2)];
    plot(xu,yu,'k'); hold on;

    xd = [defcord(ielmn(i,3),1),defcord(ielmn(i,4),1)];
    yd = [defcord(ielmn(i,3),2),defcord(ielmn(i,4),2)];
    plot(xd,yd,'c'); hold on;
end


for i = 1:length(ielmn)
    xu = [cord(ielmn(i,4),1),cord(ielmn(i,1),1)];
    yu = [cord(ielmn(i,4),2),cord(ielmn(i,1),2)];
    plot(xu,yu,'k'); hold on;

    xd = [defcord(ielmn(i,4),1),defcord(ielmn(i,1),1)];
    yd = [defcord(ielmn(i,4),2),defcord(ielmn(i,1),2)];
    plot(xd,yd,'c'); hold on;
end

title('Cantilever Beam subjected to Distributed Load and Force at tip')
xlabel('Non-Dimensional Displacements in the x')
ylabel('Non-Dimensional Displacements in the y')
legend('Undeformed','Deformed')
```

```
for i = 1:1600
    sigma(i,1) = i;
end
sigma2 = zeros(451,2);
cord = zeros(451,2);

cord2 = xlsread('stresses');
sigma(:,2) = cord2(:,3);
dx = 10/40;
dy = 1/10;

p = 1;
for i = 0:40
    for j = 0:10
    cord(p,1) = 0+dx*i;
    cord(p,2) = 0+dy*j;
    p=p+1;
    end
end

j = [1;4;8;12;16;20;24;28;32;36;40];
k = [1562;1566;1570;1574;1578;1582;1586;1590;1594;1598;1599];
l = [441;442;443;444;445;446;447;448;449;450;451];

q = 1;
for i = 1:40;
    for p = 1:11
    sigma2(q,2) = sigma(j(p)+40*(i-1),2);
    q = q+1;
    end
end

    for p = 1:11
    sigma2(l(p),2) = sigma(k(p),2);
    end

for i = 1:451
    sigma2(i,1) = i;
end


x=min(cord(:,1)):(max(cord(:,1))-min(cord(:,1)))/200:max(cord(:,1));
y=min(cord(:,2)):(max(cord(:,2))-min(cord(:,2)))/200:max(cord(:,2));

[X, Y] = meshgrid(x,y);
Z =griddata(cord(:,1),cord(:,2),sigma2(:,2),X,Y);
contour(X,Y,Z,100)
title('Non-Dimensional Stress Contours')
xlabel('Non-dimenisonal Displacements in x')
ylabel('Nondimensional Displacements in Y')
```