APPENDIX 1. Input Data for Prosthesis F076

F076 : designation of the prosthesis
5
14
18
23 : number of the basis-scan
11 : last 8 mm step
28 : total number of steps
73 : number of points per step
129 : neck angle in degrees
1
7
110

No.	X	Y	Z	No.	X	Y	Z
1	3.31	0.00	8.00	2	3.32	.29	8.00
3	3.34	.59	8.00	4	3.27	.88	8.00
5	3.30	1.20	8.00	6	3.36	1.57	8.00
7	3.35	1.93	8.00	8	3.27	2.29	8.00
9	3.13	2.63	8.00	10	2.97	2.97	8.00
11	2.76	3.29	8.00	12	2.52	3.60	8.00
13	2.30	3.98	8.00	14	2.04	4.37	8.00
15	1.71	4.69	8.00	16	1.37	5.13	8.00
17	.97	5.50	8.00	18	.49	5.56	8.00
19	0.00	5.72	8.00	20	-.49	5.64	8.00
21	-.97	5.50	8.00	22	-1.43	5.34	8.00
23	-1.88	5.17	8.00	24	-2.22	4.77	8.00
25	-2.51	4.34	8.00	26	-2.72	3.89	8.00
27	-2.91	3.47	8.00	28	-3.13	3.13	8.00
29	-3.35	2.81	8.00	30	-3.51	2.46	8.00
31	-3.58	2.07	8.00	32	-3.78	1.76	8.00
33	-3.81	1.39	8.00	34	-3.93	1.05	8.00
35	-4.08	.72	8.00	36	-4.07	.36	8.00
37	-4.16	0.00	8.00	38	-4.28	-.37	8.00
39	-4.36	-.77	8.00	40	-4.48	-1.20	8.00
41	-4.42	-1.61	8.00	42	-4.41	-2.06	8.00
43	-4.34	-2.51	8.00	44	-4.22	-2.95	8.00
45	-4.05	-3.39	8.00	46	-3.91	-3.91	8.00
47	-3.72	-4.44	8.00	48	-3.50	-5.01	8.00
49	-3.17	-5.48	8.00	50	-2.79	-5.99	8.00
51	-2.34	-6.44	8.00	52	-1.81	-6.75	8.00
53	-1.24	-7.04	8.00	54	-.61	-7.02	8.00
55	-0.00	-6.86	8.00	56	.58	-6.67	8.00
57	1.10	-6.23	8.00	58	1.56	-5.81	8.00
59	1.95	-5.35	8.00	60	2.31	-4.94	8.00
61	2.52	-4.36	8.00	62	2.72	-3.89	8.00
63	2.98	-3.55	8.00	64	3.20	-3.20	8.00
65	3.32	-2.78	8.00	66	3.48	-2.44	8.00
67	3.50	-2.02	8.00	68	3.56	-1.66	8.00
69	3.48	-1.27	8.00	70	3.50	-.94	8.00
71	3.45	-.61	8.00	72	3.38	-.30	8.00
73	3.31	-0.00	8.00				
1	3.84	0.00	16.00	2	3.80	.33	16.00
3	3.83	.67	16.00	4	3.77	1.01	16.00
5	3.69	1.34	16.00	6	3.59	1.68	16.00
7	3.55	2.05	16.00	8	3.56	2.50	16.00
9	3.48	2.92	16.00	10	3.27	3.27	16.00

11	3.12	3.72	16.00	12	2.98	4.26	16.00
13	2.73	4.74	16.00	14	2.35	5.05	16.00
15	1.94	5.33	16.00	16	1.51	5.62	16.00
17	1.04	5.88	16.00	18	.52	5.90	16.00
19	0.00	5.86	16.00	20	-.49	5.65	16.00
21	-.97	5.50	16.00	22	-1.43	5.34	16.00
23	-1.88	5.15	16.00	24	-2.31	4.96	16.00
25	-2.67	4.63	16.00	26	-2.96	4.22	16.00
27	-3.22	3.83	16.00	28	-3.46	3.46	16.00
29	-3.71	3.11	16.00	30	-3.78	2.64	16.00
31	-3.97	2.29	16.00	32	-4.11	1.92	16.00
33	-4.22	1.53	16.00	34	-4.37	1.17	16.00
35	-4.36	.77	16.00	36	-4.46	.39	16.00
37	-4.51	0.00	16.00	38	-4.60	-.40	16.00
39	-4.60	-.81	16.00	40	-4.51	-1.21	16.00
41	-4.42	-1.61	16.00	42	-4.41	-2.06	16.00
43	-4.34	-2.51	16.00	44	-4.22	-2.95	16.00
45	-4.05	-3.39	16.00	46	-3.91	-3.91	16.00
47	-3.72	-4.44	16.00	48	-3.51	-5.01	16.00
49	-3.17	-5.48	16.00	50	-2.79	-5.99	16.00
51	-2.34	-6.44	16.00	52	-1.81	-6.75	16.00
53	-1.24	-7.04	16.00	54	-.61	-7.02	16.00
55	-0.00	-6.86	16.00	56	.58	-6.67	16.00
57	1.10	-6.23	16.00	58	1.56	-5.81	16.00
59	1.95	-5.35	16.00	60	2.30	-4.94	16.00
61	2.52	-4.36	16.00	62	2.72	-3.89	16.00
63	2.98	-3.55	16.00	64	3.21	-3.21	16.00
65	3.41	-2.86	16.00	66	3.60	-2.52	16.00
67	3.71	-2.14	16.00	68	3.68	-1.72	16.00
69	3.75	-1.37	16.00	70	3.77	-1.01	16.00
71	3.83	-.68	16.00	72	3.90	-.34	16.00
73	3.84	-0.00	16.00				
1	4.42	0.00	24.00	2	4.50	.39	24.00
3	4.52	.80	24.00	4	4.48	1.20	24.00
5	4.40	1.60	24.00	6	4.40	2.05	24.00
7	4.33	2.50	24.00	8	4.18	2.93	24.00
9	4.11	3.45	24.00	10	3.82	3.82	24.00
11	3.54	4.22	24.00	12	3.24	4.62	24.00
13	2.94	5.09	24.00	14	2.53	5.42	24.00
15	2.02	5.55	24.00	16	1.54	5.73	24.00
17	1.04	5.92	24.00	18	.52	5.91	24.00
19	0.00	5.84	24.00	20	-.52	5.92	24.00
21	-1.05	5.97	24.00	22	-1.56	5.84	24.00
23	-1.97	5.42	24.00	24	-2.36	5.06	24.00
25	-2.74	4.74	24.00	26	-3.12	4.46	24.00
27	-3.43	4.08	24.00	28	-3.70	3.70	24.00
29	-3.81	3.20	24.00	30	-4.01	2.81	24.00
31	-4.10	2.37	24.00	32	-4.12	1.92	24.00
33	-4.21	1.53	24.00	34	-4.37	1.17	24.00
35	-4.36	.77	24.00	36	-4.46	.39	24.00
37	-4.51	0.00	24.00	38	-4.60	-.40	24.00
39	-4.60	-.81	24.00	40	-4.51	-1.21	24.00
41	-4.42	-1.61	24.00	42	-4.41	-2.06	24.00
43	-4.34	-2.51	24.00	44	-4.22	-2.95	24.00
45	-4.05	-3.39	24.00	46	-3.91	-3.91	24.00
47	-3.72	-4.44	24.00	48	-3.51	-5.01	24.00
49	-3.17	-5.48	24.00	50	-2.79	-5.99	24.00
51	-2.34	-6.44	24.00	52	-1.81	-6.75	24.00
53	-1.24	-7.04	24.00	54	-.61	-7.01	24.00
55	-0.00	-6.93	24.00	56	.58	-6.68	24.00
57	1.10	-6.23	24.00	58	1.56	-5.81	24.00

59	1.96	-5.38	24.00	60	2.31	-4.95	24.00
61	2.52	-4.36	24.00	62	2.72	-3.88	24.00
63	3.01	-3.59	24.00	64	3.23	-3.23	24.00
65	3.41	-2.86	24.00	66	3.63	-2.54	24.00
67	3.74	-2.16	24.00	68	3.86	-1.80	24.00
69	3.95	-1.44	24.00	70	4.07	-1.09	24.00
71	4.17	-.74	24.00	72	4.24	-.37	24.00
73	4.42	-0.00	24.00				
1	4.49	0.00	32.00	2	4.69	.41	32.00
3	4.87	.86	32.00	4	4.98	1.33	32.00
5	5.02	1.83	32.00	6	4.92	2.29	32.00
7	4.81	2.78	32.00	8	4.69	3.28	32.00
9	4.55	3.82	32.00	10	4.30	4.30	32.00
11	4.08	4.86	32.00	12	3.76	5.37	32.00
13	3.31	5.73	32.00	14	2.83	6.06	32.00
15	2.33	6.41	32.00	16	1.77	6.61	32.00
17	1.19	6.75	32.00	18	.60	6.84	32.00
19	0.00	6.77	32.00	20	-.58	6.68	32.00
21	-1.16	6.59	32.00	22	-1.73	6.44	32.00
23	-2.22	6.11	32.00	24	-2.66	5.70	32.00
25	-3.01	5.21	32.00	26	-3.32	4.74	32.00
27	-3.61	4.30	32.00	28	-3.88	3.88	32.00
29	-3.99	3.35	32.00	30	-4.05	2.84	32.00
31	-4.13	2.38	32.00	32	-4.18	1.95	32.00
33	-4.22	1.54	32.00	34	-4.43	1.19	32.00
35	-4.40	.78	32.00	36	-4.46	.39	32.00
37	-4.55	0.00	32.00	38	-4.66	-.41	32.00
39	-4.61	-.81	32.00	40	-4.51	-1.21	32.00
41	-4.42	-1.61	32.00	42	-4.41	-2.06	32.00
43	-4.34	-2.51	32.00	44	-4.22	-2.95	32.00
45	-4.05	-3.39	32.00	46	-3.91	-3.91	32.00
47	-3.72	-4.44	32.00	48	-3.51	-5.01	32.00
49	-3.17	-5.48	32.00	50	-2.79	-5.99	32.00
51	-2.34	-6.44	32.00	52	-1.81	-6.75	32.00
53	-1.24	-7.04	32.00	54	-.61	-7.01	32.00
55	-0.00	-6.94	32.00	56	.58	-6.66	32.00
57	1.14	-6.44	32.00	58	1.61	-6.00	32.00
59	1.99	-5.46	32.00	60	2.32	-4.97	32.00
61	2.61	-4.52	32.00	62	2.89	-4.12	32.00
63	3.13	-3.74	32.00	64	3.38	-3.38	32.00
65	3.46	-2.90	32.00	66	3.65	-2.55	32.00
67	3.75	-2.16	32.00	68	3.87	-1.80	32.00
69	4.07	-1.48	32.00	70	4.17	-1.12	32.00
71	4.39	-.77	32.00	72	4.40	-.39	32.00
73	4.49	-0.00	32.00				
1	5:13	0.00	40.00	2	5.17	.45	40.00
3	5.11	.90	40.00	4	5.11	1.37	40.00
5	5.05	1.84	40.00	6	5.14	2.40	40.00
7	5.11	2.95	40.00	8	4.99	3.49	40.00
9	4.86	4.07	40.00	10	4.58	4.58	40.00
11	4.24	5.05	40.00	12	3.79	5.41	40.00
13	3.31	5.73	40.00	14	2.83	6.06	40.00
15	2.33	6.41	40.00	16	1.77	6.61	40.00
17	1.19	6.75	40.00	18	.60	6.84	40.00
19	0.00	6.77	40.00	20	-.58	6.68	40.00
21	-1.16	6.59	40.00	22	-1.73	6.44	40.00
23	-2.22	6.11	40.00	24	-2.66	5.70	40.00
25	-3.01	5.21	40.00	26	-3.32	4.74	40.00
27	-3.61	4.30	40.00	28	-3.88	3.88	40.00
29	-3.99	3.35	40.00	30	-4.05	2.84	40.00

31	-4.12	2.38	40.00	32	-4.29	2.00	40.00
33	-4.41	1.61	40.00	34	-4.46	1.19	40.00
35	-4.39	.77	40.00	36	-4.46	.39	40.00
37	-4.55	0.00	40.00	38	-4.66	-.41	40.00
39	-4.61	-.81	40.00	40	-4.51	-1.21	40.00
41	-4.42	-1.61	40.00	42	-4.41	-2.06	40.00
43	-4.34	-2.51	40.00	44	-4.22	-2.95	40.00
45	-4.05	-3.39	40.00	46	-3.91	-3.91	40.00
47	-3.72	-4.44	40.00	48	-3.51	-5.01	40.00
49	-3.17	-5.49	40.00	50	-2.79	-5.98	40.00
51	-2.38	-6.53	40.00	52	-1.83	-6.84	40.00
53	-1.25	-7.09	40.00	54	-.62	-7.14	40.00
55	-0.00	-7.04	40.00	56	.58	-6.68	40.00
57	1.13	-6.43	40.00	58	1.64	-6.13	40.00
59	2.04	-5.61	40.00	60	2.50	-5.37	40.00
61	2.83	-4.90	40.00	62	3.17	-4.53	40.00
63	3.57	-4.25	40.00	64	3.71	-3.71	40.00
65	3.93	-3.30	40.00	66	4.19	-2.93	40.00
67	4.44	-2.57	40.00	68	4.52	-2.11	40.00
69	4.73	-1.72	40.00	70	4.90	-1.31	40.00
71	5.02	-.88	40.00	72	5.08	-.44	40.00
73	5.13	-0.00	40.00				
1	5.36	0.00	48.00	2	5.59	.49	48.00
3	5.60	.99	48.00	4	5.47	1.47	48.00
5	5.49	2.00	48.00	6	5.39	2.51	48.00
7	5.32	3.07	48.00	8	5.30	3.71	48.00
9	4.99	4.18	48.00	10	4.63	4.63	48.00
11	4.28	5.10	48.00	12	3.90	5.56	48.00
13	3.44	5.95	48.00	14	2.90	6.23	48.00
15	2.36	6.48	48.00	16	1.77	6.61	48.00
17	1.19	6.75	48.00	18	.60	6.84	48.00
19	0.00	6.77	48.00	20	-.58	6.68	48.00
21	-1.16	6.59	48.00	22	-1.73	6.44	48.00
23	-2.22	6.11	48.00	24	-2.66	5.70	48.00
25	-3.01	5.21	48.00	26	-3.32	4.74	48.00
27	-3.61	4.30	48.00	28	-3.88	3.88	48.00
29	-3.99	3.35	48.00	30	-4.05	2.84	48.00
31	-4.12	2.38	48.00	32	-4.29	2.00	48.00
33	-4.41	1.61	48.00	34	-4.46	1.19	48.00
35	-4.39	.77	48.00	36	-4.46	.39	48.00
37	-4.55	0.00	48.00	38	-4.66	-.41	48.00
39	-4.61	-.81	48.00	40	-4.51	-1.21	48.00
41	-4.42	-1.61	48.00	42	-4.41	-2.06	48.00
43	-4.34	-2.51	48.00	44	-4.22	-2.95	48.00
45	-4.05	-3.39	48.00	46	-3.91	-3.91	48.00
47	-3.72	-4.44	48.00	48	-3.51	-5.01	48.00
49	-3.17	-5.49	48.00	50	-2.79	-5.98	48.00
51	-2.38	-6.53	48.00	52	-1.83	-6.84	48.00
53	-1.25	-7.09	48.00	54	-.62	-7.14	48.00
55	-0.00	-7.04	48.00	56	.58	-6.67	48.00
57	1.14	-6.45	48.00	58	1.64	-6.13	48.00
59	2.06	-5.65	48.00	60	2.52	-5.41	48.00
61	2.87	-4.97	48.00	62	3.18	-4.54	48.00
63	3.57	-4.26	48.00	64	3.88	-3.88	48.00
65	4.14	-3.48	48.00	66	4.27	-2.99	48.00
67	4.45	-2.57	48.00	68	4.52	-2.11	48.00
69	4.72	-1.72	48.00	70	4.98	-1.33	48.00
71	5.13	-.90	48.00	72	5.33	-.47	48.00
73	5.36	-0.00	48.00				
1	5.73	0.00	56.00	2	5.79	.51	56.00

3	5.76	1.02	56.00	4	5.88	1.58	56.00
5	5.89	2.14	56.00	6	5.77	2.69	56.00
7	5.78	3.34	56.00	8	5.65	3.96	56.00
9	5.53	4.64	56.00	10	5.17	5.17	56.00
11	4.75	5.66	56.00	12	4.22	6.03	56.00
13	3.67	6.35	56.00	14	3.06	6.56	56.00
15	2.51	6.89	56.00	16	1.86	6.94	56.00
17	1.27	7.17	56.00	18	.63	7.23	56.00
19	0.00	7.20	56.00	20	-.63	7.17	56.00
21	-1.22	6.91	56.00	22	-1.75	6.54	56.00
23	-2.23	6.12	56.00	24	-2.66	5.70	56.00
25	-3.01	5.21	56.00	26	-3.32	4.74	56.00
27	-3.61	4.30	56.00	28	-3.88	3.88	56.00
29	-3.99	3.35	56.00	30	-4.05	2.84	56.00
31	-4.12	2.38	56.00	32	-4.29	2.00	56.00
33	-4.41	1.61	56.00	34	-4.46	1.19	56.00
35	-4.39	.77	56.00	36	-4.46	.39	56.00
37	-4.55	0.00	56.00	38	-4.66	-.41	56.00
39	-4.61	-.81	56.00	40	-4.51	-1.21	56.00
41	-4.42	-1.61	56.00	42	-4.41	-2.06	56.00
43	-4.34	-2.51	56.00	44	-4.22	-2.95	56.00
45	-4.05	-3.39	56.00	46	-3.91	-3.91	56.00
47	-3.72	-4.44	56.00	48	-3.51	-5.01	56.00
49	-3.17	-5.49	56.00	50	-2.79	-5.98	56.00
51	-2.38	-6.53	56.00	52	-1.83	-6.84	56.00
53	-1.25	-7.09	56.00	54	-.62	-7.14	56.00
55	-0.00	-7.03	56.00	56	.59	-6.69	56.00
57	1.15	-6.50	56.00	58	1.69	-6.30	56.00
59	2.13	-5.85	56.00	60	2.61	-5.59	56.00
61	3.02	-5.23	56.00	62	3.41	-4.87	56.00
63	3.74	-4.45	56.00	64	4.03	-4.03	56.00
65	4.31	-3.61	56.00	66	4.61	-3.23	56.00
67	4.67	-2.70	56.00	68	4.93	-2.30	56.00
69	5.09	-1.85	56.00	70	5.24	-1.40	56.00
71	5.39	-.95	56.00	72	5.59	-.49	56.00
73	5.73	-0.00	56.00				
1	6.60	0.00	64.00	2	6.75	.59	64.00
3	6.95	1.23	64.00	4	6.84	1.83	64.00
5	6.71	2.44	64.00	6	6.65	3.10	64.00
7	6.46	3.73	64.00	8	6.24	4.37	64.00
9	5.79	4.86	64.00	10	5.30	5.30	64.00
11	4.77	5.68	64.00	12	4.22	6.03	64.00
13	3.67	6.36	64.00	14	3.06	6.56	64.00
15	2.51	6.89	64.00	16	1.86	6.94	64.00
17	1.27	7.17	64.00	18	.63	7.23	64.00
19	0.00	7.20	64.00	20	-.63	7.17	64.00
21	-1.22	6.91	64.00	22	-1.75	6.54	64.00
23	-2.23	6.12	64.00	24	-2.66	5.70	64.00
25	-3.01	5.21	64.00	26	-3.32	4.74	64.00
27	-3.61	4.30	64.00	28	-3.88	3.88	64.00
29	-4.05	3.40	64.00	30	-4.06	2.84	64.00
31	-4.12	2.38	64.00	32	-4.28	2.00	64.00
33	-4.44	1.62	64.00	34	-4.58	1.23	64.00
35	-4.63	.82	64.00	36	-4.70	.41	64.00
37	-4.68	0.00	64.00	38	-4.67	-.41	64.00
39	-4.61	-.81	64.00	40	-4.51	-1.21	64.00
41	-4.42	-1.61	64.00	42	-4.41	-2.06	64.00
43	-4.34	-2.51	64.00	44	-4.22	-2.95	64.00
45	-4.05	-3.39	64.00	46	-3.91	-3.91	64.00
47	-3.72	-4.44	64.00	48	-3.51	-5.01	64.00
49	-3.17	-5.48	64.00	50	-2.79	-5.98	64.00

51	-2.37	-6.52	64.00	52	-1.89	-7.07	64.00
53	-1.29	-7.34	64.00	54	-.66	-7.49	64.00
55	-0.00	-7.30	64.00	56	.63	-7.15	64.00
57	1.24	-7.03	64.00	58	1.79	-6.69	64.00
59	2.30	-6.32	64.00	60	2.82	-6.04	64.00
61	3.28	-5.67	64.00	62	3.73	-5.33	64.00
63	4.22	-5.03	64.00	64	4.58	-4.58	64.00
65	4.89	-4.10	64.00	66	5.20	-3.64	64.00
67	5.51	-3.18	64.00	68	5.83	-2.72	64.00
69	6.21	-2.26	64.00	70	6.29	-1.68	64.00
71	6.38	-1.13	64.00	72	6.49	-.57	64.00
73	6.60	-0.00	64.00				
1	7.29	0.00	72.00	2	7.15	.63	72.00
3	7.20	1.27	72.00	4	7.30	1.96	72.00
5	7.21	2.62	72.00	6	7.07	3.30	72.00
7	6.78	3.91	72.00	8	6.28	4.40	72.00
9	5.85	4.91	72.00	10	5.31	5.31	72.00
11	4.77	5.68	72.00	12	4.22	6.03	72.00
13	3.67	6.36	72.00	14	3.06	6.56	72.00
15	2.51	6.89	72.00	16	1.86	6.94	72.00
17	1.27	7.17	72.00	18	.63	7.23	72.00
19	0.00	7.21	72.00	20	-.63	7.16	72.00
21	-1.25	7.08	72.00	22	-1.80	6.73	72.00
23	-2.32	6.38	72.00	24	-2.73	5.85	72.00
25	-3.19	5.52	72.00	26	-3.55	5.07	72.00
27	-3.71	4.42	72.00	28	-3.98	3.98	72.00
29	-4.15	3.48	72.00	30	-4.29	3.00	72.00
31	-4.45	2.57	72.00	32	-4.49	2.09	72.00
33	-4.58	1.67	72.00	34	-4.60	1.23	72.00
35	-4.62	.82	72.00	36	-4.70	.41	72.00
37	-4.68	0.00	72.00	38	-4.67	-.41	72.00
39	-4.61	-.81	72.00	40	-4.51	-1.21	72.00
41	-4.42	-1.61	72.00	42	-4.41	-2.06	72.00
43	-4.34	-2.51	72.00	44	-4.15	-2.95	72.00
45	-4.05	-3.39	72.00	46	-3.91	-3.91	72.00
47	-3.72	-4.44	72.00	48	-3.51	-5.01	72.00
49	-3.17	-5.48	72.00	50	-2.79	-5.98	72.00
51	-2.37	-6.52	72.00	52	-1.89	-7.07	72.00
53	-1.29	-7.34	72.00	54	-.66	-7.55	72.00
55	-0.00	-7.54	72.00	56	.64	-7.29	72.00
57	1.25	-7.09	72.00	58	1.80	-6.70	72.00
59	2.30	-6.32	72.00	60	2.82	-6.04	72.00
61	3.29	-5.69	72.00	62	3.83	-5.47	72.00
63	4.29	-5.12	72.00	64	4.64	-4.64	72.00
65	5.13	-4.30	72.00	66	5.55	-3.88	72.00
67	5.86	-3.39	72.00	68	6.16	-2.87	72.00
69	6.43	-2.34	72.00	70	6.70	-1.79	72.00
71	6.92	-1.22	72.00	72	7.23	-.63	72.00
73	7.29	-0.00	72.00				
1	8.23	0.00	80.00	2	8.21	.72	80.00
3	8.33	1.47	80.00	4	8.09	2.17	80.00
5	7.72	2.81	80.00	6	7.40	3.45	80.00
7	7.07	4.08	80.00	8	6.62	4.64	80.00
9	5.90	4.95	80.00	10	5.31	5.31	80.00
11	4.77	5.68	80.00	12	4.22	6.03	80.00
13	3.67	6.36	80.00	14	3.06	6.56	80.00
15	2.51	6.89	80.00	16	1.86	6.94	80.00
17	1.27	7.17	80.00	18	.63	7.23	80.00
19	0.00	7.20	80.00	20	-.63	7.18	80.00
21	-1.25	7.08	80.00	22	-1.80	6.73	80.00

23	-2.38	6.55	80.00	24	-2.84	6.09	80.00
25	-3.20	5.55	80.00	26	-3.56	5.09	80.00
27	-3.72	4.43	80.00	28	-3.98	3.98	80.00
29	-4.17	3.50	80.00	30	-4.38	3.07	80.00
31	-4.46	2.58	80.00	32	-4.56	2.13	80.00
33	-4.60	1.67	80.00	34	-4.60	1.23	80.00
35	-4.62	.82	80.00	36	-4.70	.41	80.00
37	-4.68	0.00	80.00	38	-4.67	-.41	80.00
39	-4.61	-.81	80.00	40	-4.51	-1.21	80.00
41	-4.42	-1.61	80.00	42	-4.41	-2.06	80.00
43	-4.34	-2.51	80.00	44	-4.22	-2.95	80.00
45	-4.05	-3.39	80.00	46	-3.91	-3.91	80.00
47	-3.72	-4.44	80.00	48	-3.51	-5.01	80.00
49	-3.17	-5.48	80.00	50	-2.79	-5.98	80.00
51	-2.37	-6.52	80.00	52	-1.89	-7.06	80.00
53	-1.32	-7.50	80.00	54	-.67	-7.69	80.00
55	-0.00	-7.93	80.00	56	.68	-7.78	80.00
57	1.31	-7.41	80.00	58	1.88	-7.03	80.00
59	2.44	-6.71	80.00	60	3.00	-6.44	80.00
61	3.57	-6.19	80.00	62	4.09	-5.84	80.00
63	4.63	-5.52	80.00	64	5.03	-5.03	80.00
65	5.44	-4.57	80.00	66	5.96	-4.18	80.00
67	6.47	-3.73	80.00	68	6.99	-3.26	80.00
69	7.45	-2.71	80.00	70	7.91	-2.12	80.00
71	8.14	-1.44	80.00	72	8.27	-.72	80.00
73	8.23	-0.00	80.00				
1	8.68	0.00	88.00	2	8.80	.77	88.00
3	8.66	1.53	88.00	4	8.39	2.25	88.00
5	8.02	2.92	88.00	6	7.67	3.58	88.00
7	7.21	4.16	88.00	8	6.64	4.65	88.00
9	5.90	4.95	88.00	10	5.31	5.31	88.00
11	4.77	5.68	88.00	12	4.22	6.03	88.00
13	3.67	6.36	88.00	14	3.06	6.56	88.00
15	2.51	6.89	88.00	16	1.86	6.94	88.00
17	1.27	7.18	88.00	18	.63	7.22	88.00
19	0.00	7.33	88.00	20	-.64	7.31	88.00
21	-1.26	7.17	88.00	22	-1.88	7.00	88.00
23	-2.44	6.71	88.00	24	-2.89	6.19	88.00
25	-3.35	5.80	88.00	26	-3.71	5.30	88.00
27	-4.01	4.78	88.00	28	-4.34	4.34	88.00
29	-4.44	3.72	88.00	30	-4.62	3.24	88.00
31	-4.66	2.69	88.00	32	-4.67	2.18	88.00
33	-4.71	1.72	88.00	34	-4.74	1.27	88.00
35	-4.80	.85	88.00	36	-4.78	.42	88.00
37	-4.68	0.00	88.00	38	-4.69	-.41	88.00
39	-4.62	-.82	88.00	40	-4.51	-1.21	88.00
41	-4.42	-1.61	88.00	42	-4.41	-2.06	88.00
43	-4.34	-2.51	88.00	44	-4.22	-2.95	88.00
45	-4.05	-3.39	88.00	46	-3.91	-3.91	88.00
47	-3.74	-4.45	88.00	48	-3.55	-5.08	88.00
49	-3.34	-5.78	88.00	50	-3.04	-6.51	88.00
51	-2.59	-7.11	88.00	52	-2.06	-7.67	88.00
53	-1.41	-8.02	88.00	54	-.71	-8.17	88.00
55	-0.00	-8.18	88.00	56	.68	-7.81	88.00
57	1.31	-7.41	88.00	58	1.88	-7.03	88.00
59	2.44	-6.71	88.00	60	3.00	-6.44	88.00
61	3.57	-6.19	88.00	62	4.09	-5.84	88.00
63	4.63	-5.52	88.00	64	5.03	-5.03	88.00
65	5.44	-4.56	88.00	66	5.96	-4.18	88.00
67	6.46	-3.73	88.00	68	7.13	-3.33	88.00
69	7.72	-2.81	88.00	70	8.19	-2.19	88.00

71	8.43	-1.49	88.00	72	8.58	-.75	88.00
73	8.68	-0.00	88.00				
1	9.21	0.00	92.00	2	9.35	.82	92.00
3	9.20	1.62	92.00	4	8.68	2.32	92.00
5	8.42	3.07	92.00	6	7.98	3.72	92.00
7	7.34	4.24	92.00	8	6.65	4.66	92.00
9	5.90	4.95	92.00	10	5.31	5.31	92.00
11	4.77	5.68	92.00	12	4.22	6.03	92.00
13	3.67	6.36	92.00	14	3.06	6.56	92.00
15	2.51	6.89	92.00	16	1.86	6.94	92.00
17	1.27	7.18	92.00	18	.63	7.22	92.00
19	0.00	7.33	92.00	20	-.64	7.31	92.00
21	-1.26	7.17	92.00	22	-1.88	7.00	92.00
23	-2.44	6.71	92.00	24	-2.89	6.19	92.00
25	-3.35	5.80	92.00	26	-3.71	5.30	92.00
27	-4.01	4.78	92.00	28	-4.34	4.34	92.00
29	-4.44	3.72	92.00	30	-4.62	3.24	92.00
31	-4.66	2.69	92.00	32	-4.67	2.18	92.00
33	-4.71	1.72	92.00	34	-4.74	1.27	92.00
35	-4.80	.85	92.00	36	-4.78	.42	92.00
37	-4.68	0.00	92.00	38	-4.69	-.41	92.00
39	-4.62	-.82	92.00	40	-4.51	-1.21	92.00
41	-4.42	-1.61	92.00	42	-4.41	-2.06	92.00
43	-4.34	-2.51	92.00	44	-4.22	-2.95	92.00
45	-4.05	-3.39	92.00	46	-3.91	-3.91	92.00
47	-3.74	-4.45	92.00	48	-3.55	-5.08	92.00
49	-3.34	-5.78	92.00	50	-3.04	-6.51	92.00
51	-2.59	-7.11	92.00	52	-2.06	-7.67	92.00
53	-1.41	-8.02	92.00	54	-.71	-8.17	92.00
55	-0.00	-8.18	92.00	56	.68	-7.81	92.00
57	1.31	-7.41	92.00	58	1.88	-7.03	92.00
59	2.44	-6.71	92.00	60	3.00	-6.44	92.00
61	3.57	-6.19	92.00	62	4.09	-5.84	92.00
63	4.63	-5.52	92.00	64	5.03	-5.03	92.00
65	5.44	-4.56	92.00	66	5.96	-4.18	92.00
67	6.46	-3.73	92.00	68	7.13	-3.33	92.00
69	7.72	-2.81	92.00	70	8.18	-2.19	92.00
71	8.62	-1.52	92.00	72	8.80	-.77	92.00
73	9.21	-0.00	92.00				
1	10.30	0.00	96.00	2	10.57	.92	96.00
3	10.55	1.86	96.00	4	9.90	2.65	96.00
5	9.41	3.42	96.00	6	8.86	4.13	96.00
7	7.98	4.61	96.00	8	7.26	5.08	96.00
9	6.48	5.44	96.00	10	5.68	5.68	96.00
11	5.02	5.98	96.00	12	4.35	6.22	96.00
13	3.79	6.57	96.00	14	3.13	6.72	96.00
15	2.51	6.91	96.00	16	1.86	6.94	96.00
17	1.27	7.18	96.00	18	.63	7.21	96.00
19	0.00	7.49	96.00	20	-.67	7.64	96.00
21	-1.34	7.62	96.00	22	-2.04	7.61	96.00
23	-2.66	7.31	96.00	24	-3.22	6.90	96.00
25	-3.54	6.13	96.00	26	-3.88	5.54	96.00
27	-4.15	4.95	96.00	28	-4.38	4.38	96.00
29	-4.46	3.75	96.00	30	-4.62	3.23	96.00
31	-4.78	2.76	96.00	32	-4.89	2.28	96.00
33	-4.79	1.74	96.00	34	-4.82	1.29	96.00
35	-4.86	.86	96.00	36	-4.97	.44	96.00
37	-4.97	0.00	96.00	38	-4.91	-.43	96.00
39	-4.80	-.8	96.00	40	-4.83	-1.29	96.00
41	-4.78	-1.74	96.00	42	-4.60	-2.14	96.00

43	-4.45	-2.57	96.00	44	-4.23	-2.96	96.00
45	-4.04	-3.39	96.00	46	-3.91	-3.91	96.00
47	-3.78	-4.51	96.00	48	-3.57	-5.09	96.00
49	-3.33	-5.77	96.00	50	-3.09	-6.63	96.00
51	-2.60	-7.14	96.00	52	-2.05	-7.67	96.00
53	-1.41	-8.02	96.00	54	-.71	-8.17	96.00
55	-0.00	-8.18	96.00	56	.68	-7.81	96.00
57	1.31	-7.41	96.00	58	1.88	-7.03	96.00
59	2.44	-6.71	96.00	60	3.00	-6.44	96.00
61	3.57	-6.19	96.00	62	4.09	-5.84	96.00
63	4.63	-5.52	96.00	64	5.03	-5.03	96.00
65	5.44	-4.56	96.00	66	5.96	-4.18	96.00
67	6.46	-3.73	96.00	68	7.13	-3.33	96.00
69	7.71	-2.81	96.00	70	8.28	-2.22	96.00
71	9.12	-1.61	96.00	72	9.85	-.86	96.00
73	10.30	-0.00	96.00				
1	10.40	0.00	100.00	2	10.76	.94	100.00
3	10.61	1.87	100.00	4	10.14	2.72	100.00
5	9.65	3.51	100.00	6	9.05	4.22	100.00
7	8.14	4.70	100.00	8	7.33	5.13	100.00
9	6.65	5.58	100.00	10	6.01	6.01	100.00
11	5.40	6.44	100.00	12	4.71	6.73	100.00
13	4.02	6.97	100.00	14	3.38	7.25	100.00
15	2.79	7.67	100.00	16	2.10	7.83	100.00
17	1.40	7.92	100.00	18	.69	7.93	100.00
19	0.00	7.87	100.00	20	-.69	7.83	100.00
21	-1.38	7.81	100.00	22	-2.10	7.83	100.00
23	-2.73	7.50	100.00	24	-3.38	7.25	100.00
25	-3.91	6.77	100.00	26	-4.33	6.18	100.00
27	-4.74	5.65	100.00	28	-4.99	4.99	100.00
29	-5.20	4.36	100.00	30	-5.31	3.71	100.00
31	-5.38	3.11	100.00	32	-5.24	2.44	100.00
33	-5.23	1.90	100.00	34	-5.33	1.43	100.00
35	-5.20	.92	100.00	36	-5.07	.44	100.00
37	-5.03	0.00	100.00	38	-5.04	-.44	100.00
39	-5.10	-.90	100.00	40	-5.03	-1.35	100.00
41	-4.93	-1.79	100.00	42	-4.90	-2.29	100.00
43	-4.82	-2.78	100.00	44	-4.69	-3.28	100.00
45	-4.44	-3.73	100.00	46	-4.29	-4.29	100.00
47	-4.14	-4.94	100.00	48	-4.00	-5.71	100.00
49	-3.82	-6.61	100.00	50	-3.36	-7.22	100.00
51	-2.82	-7.74	100.00	52	-2.20	-8.22	100.00
53	-1.49	-8.43	100.00	54	-.75	-8.62	100.00
55	-0.00	-8.26	100.00	56	.68	-7.79	100.00
57	1.32	-7.50	100.00	58	1.89	-7.05	100.00
59	2.44	-6.71	100.00	60	3.00	-6.44	100.00
61	3.57	-6.19	100.00	62	4.09	-5.84	100.00
63	4.63	-5.52	100.00	64	5.03	-5.03	100.00
65	5.44	-4.56	100.00	66	5.96	-4.18	100.00
67	6.46	-3.73	100.00	68	7.13	-3.33	100.00
69	7.71	-2.81	100.00	70	8.28	-2.22	100.00
71	9.12	-1.61	100.00	72	9.89	-.87	100.00
73	10.40	-0.00	100.00				
1	10.92	0.00	104.00	2	11.36	.99	104.00
3	11.20	1.98	104.00	4	10.97	2.94	104.00
5	10.31	3.75	104.00	6	9.56	4.46	104.00
7	8.74	5.05	104.00	8	7.96	5.58	104.00
9	7.26	6.09	104.00	10	6.36	6.36	104.00
11	5.58	6.65	104.00	12	4.95	7.07	104.00
13	4.27	7.39	104.00	14	3.51	7.54	104.00

15	2.81	7.71	104.00	16	2.10	7.82	104.00
17	1.41	7.98	104.00	18	.71	8.15	104.00
19	0.00	8.30	104.00	20	-.74	8.43	104.00
21	-1.51	8.54	104.00	22	-2.24	8.34	104.00
23	-3.00	8.24	104.00	24	-3.62	7.77	104.00
25	-4.20	7.27	104.00	26	-4.70	6.71	104.00
27	-5.00	5.96	104.00	28	-5.28	5.28	104.00
29	-5.31	4.45	104.00	30	-5.35	3.75	104.00
31	-5.39	3.11	104.00	32	-5.45	2.54	104.00
33	-5.49	2.00	104.00	34	-5.48	1.47	104.00
35	-5.69	1.00	104.00	36	-5.70	.50	104.00
37	-5.72	0.00	104.00	38	-5.71	-.50	104.00
39	-5.58	-.98	104.00	40	-5.48	-1.47	104.00
41	-5.54	-2.02	104.00	42	-5.53	-2.58	104.00
43	-5.54	-3.20	104.00	44	-5.43	-3.80	104.00
45	-5.29	-4.43	104.00	46	-5.09	-5.09	104.00
47	-4.90	-5.84	104.00	48	-4.64	-6.30	104.00
49	-4.22	-7.32	104.00	50	-3.79	-8.13	104.00
51	-3.20	-8.80	104.00	52	-2.43	-9.08	104.00
53	-1.60	-9.10	104.00	54	-.77	-8.77	104.00
55	-0.00	-8.25	104.00	56	.68	-7.79	104.00
57	1.32	-7.50	104.00	58	1.89	-7.05	104.00
59	2.44	-6.71	104.00	60	3.00	-6.44	104.00
61	3.57	-6.18	104.00	62	4.15	-5.92	104.00
63	4.80	-5.72	104.00	64	5.36	-5.36	104.00
65	5.88	-4.93	104.00	66	6.53	-4.57	104.00
67	7.07	-4.08	104.00	68	7.59	-3.54	104.00
69	8.27	-3.01	104.00	70	9.00	-2.41	104.00
71	9.85	-1.74	104.00	72	10.41	-.91	104.00
73	10.92	-0.00	104.00				
1	11.82	0.00	108.00	2	11.95	1.05	108.00
3	11.67	2.06	108.00	4	11.27	3.02	108.00
5	10.73	3.91	108.00	6	10.12	4.72	108.00
7	9.28	5.36	108.00	8	8.37	5.86	108.00
9	7.50	6.29	108.00	10	6.66	6.66	108.00
11	5.92	7.06	108.00	12	5.20	7.43	108.00
13	4.47	7.75	108.00	14	3.80	8.15	108.00
15	3.06	8.42	108.00	16	2.28	8.50	108.00
17	1.51	8.55	108.00	18	.77	8.77	108.00
19	0.00	9.03	108.00	20	-.80	9.10	108.00
21	-1.63	9.26	108.00	22	-2.48	9.27	108.00
23	-3.28	9.00	108.00	24	-3.93	8.43	108.00
25	-4.53	7.84	108.00	26	-5.01	7.16	108.00
27	-5.44	6.49	108.00	28	-5.77	5.77	108.00
29	-6.09	5.11	108.00	30	-6.18	4.33	108.00
31	-6.30	3.64	108.00	32	-6.27	2.92	108.00
33	-6.16	2.24	108.00	34	-6.10	1.64	108.00
35	-6.12	1.08	108.00	36	-6.09	.53	108.00
37	-5.96	0.00	108.00	38	-5.86	-.51	108.00
39	-5.83	-1.03	108.00	40	-5.95	-1.59	108.00
41	-5.90	-2.15	108.00	42	-5.79	-2.70	108.00
43	-5.72	-3.30	108.00	44	-5.62	-3.94	108.00
45	-5.63	-4.72	108.00	46	-5.54	-5.54	108.00
47	-5.39	-6.43	108.00	48	-5.01	-7.16	108.00
49	-4.52	-7.82	108.00	50	-3.98	-8.54	108.00
51	-3.26	-8.95	108.00	52	-2.44	-9.10	108.00
53	-1.60	-9.10	108.00	54	-.77	-8.77	108.00
55	-0.00	-8.25	108.00	56	.68	-7.79	108.00
57	1.32	-7.50	108.00	58	1.89	-7.05	108.00
59	2.44	-6.71	108.00	60	3.00	-6.44	108.00
61	3.57	-6.18	108.00	62	4.15	-5.92	108.00

63	4.80	-5.72	108.00	64	5.36	-5.36	108.00
65	5.88	-4.93	108.00	66	6.53	-4.57	108.00
67	7.07	-4.08	108.00	68	7.59	-3.54	108.00
69	8.26	-3.01	108.00	70	9.18	-2.46	108.00
71	10.30	-1.82	108.00	72	11.10	-.97	108.00
73	11.82	-0.00	108.00				
1	12.25	0.00	112.00	2	12.94	1.13	112.00
3	12.50	2.20	112.00	4	11.68	3.13	112.00
5	10.96	3.99	112.00	6	10.37	4.84	112.00
7	9.64	5.57	112.00	8	8.90	6.23	112.00
9	7.93	6.65	112.00	10	7.13	7.13	112.00
11	6.17	7.36	112.00	12	5.40	7.71	112.00
13	4.70	8.14	112.00	14	3.89	8.35	112.00
15	3.07	8.45	112.00	16	2.34	8.74	112.00
17	1.58	8.95	112.00	18	.80	9.14	112.00
19	0.00	9.34	112.00	20	-.84	9.58	112.00
21	-1.67	9.45	112.00	22	-2.52	9.40	112.00
23	-3.45	9.47	112.00	24	-4.38	9.38	112.00
25	-5.13	8.88	112.00	26	-5.91	8.45	112.00
27	-6.58	7.85	112.00	28	-7.14	7.14	112.00
29	-7.68	6.44	112.00	30	-7.97	5.58	112.00
31	-8.19	4.73	112.00	32	-8.31	3.87	112.00
33	-8.27	3.01	112.00	34	-8.28	2.22	112.00
35	-8.30	1.46	112.00	36	-8.19	.72	112.00
37	-8.19	0.00	112.00	38	-7.92	-.69	112.00
39	-7.81	-1.38	112.00	40	-7.57	-2.03	112.00
41	-7.45	-2.71	112.00	42	-7.39	-3.44	112.00
43	-7.09	-4.10	112.00	44	-7.01	-4.91	112.00
45	-6.66	-5.59	112.00	46	-6.31	-6.31	112.00
47	-5.86	-6.99	112.00	48	-5.37	-7.67	112.00
49	-4.76	-8.24	112.00	50	-4.06	-8.72	112.00
51	-3.29	-9.05	112.00	52	-2.45	-9.15	112.00
53	-1.61	-9.10	112.00	54	-.77	-8.77	112.00
55	-0.00	-8.25	112.00	56	.68	-7.79	112.00
57	1.32	-7.49	112.00	58	1.89	-7.05	112.00
59	2.44	-6.71	112.00	60	3.00	-6.44	112.00
61	3.57	-6.18	112.00	62	4.15	-5.92	112.00
63	4.80	-5.72	112.00	64	5.36	-5.36	112.00
65	5.88	-4.93	112.00	66	6.53	-4.57	112.00
67	7.08	-4.08	112.00	68	7.59	-3.54	112.00
69	8.26	-3.01	112.00	70	9.18	-2.46	112.00
71	10.29	-1.81	112.00	72	11.27	-.99	112.00
73	12.25	-0.00	112.00				
1	14.06	0.00	116.00	2	14.22	1.24	116.00
3	13.71	2.42	116.00	4	13.21	3.54	116.00
5	12.36	4.50	116.00	6	11.61	5.42	116.00
7	10.65	6.15	116.00	8	9.51	6.66	116.00
9	8.60	7.22	116.00	10	7.79	7.79	116.00
11	6.89	8.22	116.00	12	6.00	8.57	116.00
13	5.14	8.91	116.00	14	4.25	9.11	116.00
15	3.41	9.38	116.00	16	2.52	9.42	116.00
17	1.69	9.60	116.00	18	.86	9.86	116.00
19	0.00	10.08	116.00	20	-.89	10.21	116.00
21	-1.82	10.35	116.00	22	-2.82	10.54	116.00
23	-3.90	10.71	116.00	24	-5.07	10.87	116.00
25	-6.11	10.58	116.00	26	-7.03	10.03	116.00
27	-7.82	9.32	116.00	28	-8.22	8.22	116.00
29	-8.73	7.33	116.00	30	-9.16	6.41	116.00
31	-9.49	5.48	116.00	32	-9.65	4.50	116.00
33	-9.88	3.60	116.00	34	-9.88	2.65	116.00

35	-10.23	1.80	116.00	36	-10.29	.90	116.00
37	-10.14	0.00	116.00	38	-9.85	-.86	116.00
39	-9.53	-1.68	116.00	40	-9.23	-2.47	116.00
41	-8.93	-3.25	116.00	42	-8.63	-4.02	116.00
43	-8.33	-4.81	116.00	44	-8.02	-5.62	116.00
45	-7.70	-6.46	116.00	46	-7.36	-7.36	116.00
47	-6.95	-8.28	116.00	48	-6.43	-9.18	116.00
49	-5.70	-9.87	116.00	50	-4.79	-10.28	116.00
51	-3.77	-10.37	116.00	52	-2.70	-10.06	116.00
53	-1.65	-9.36	116.00	56	-.77	-8.77	116.00
55	-0.00	-8.33	116.00	56	.70	-7.95	116.00
57	1.33	-7.52	116.00	58	1.89	-7.05	116.00
59	2.44	-6.71	116.00	60	3.00	-6.44	116.00
61	3.57	-6.18	116.00	62	4.15	-5.92	116.00
63	4.79	-5.71	116.00	64	5.48	-5.48	116.00
65	6.04	-5.07	116.00	66	6.72	-4.71	116.00
67	7.61	-4.40	116.00	68	8.55	-3.99	116.00
69	9.68	-3.52	116.00	70	10.71	-2.87	116.00
71	11.80	-2.08	116.00	72	13.04	-1.14	116.00
73	14.06	-0.00	116.00				
1	16.57	0.00	120.00	2	16.06	1.41	120.00
3	15.52	2.74	120.00	4	14.27	3.82	120.00
5	13.08	4.76	120.00	6	11.93	5.56	120.00
7	10.89	6.29	120.00	8	9.87	6.91	120.00
9	8.85	7.43	120.00	10	7.95	7.95	120.00
11	6.91	8.24	120.00	12	6.00	8.56	120.00
13	5.21	9.02	120.00	14	4.39	9.42	120.00
15	3.46	9.49	120.00	16	2.58	9.63	120.00
17	1.77	10.07	120.00	18	.90	10.34	120.00
19	0.00	10.70	120.00	20	-.95	10.91	120.00
21	-2.00	11.34	120.00	22	-3.16	11.79	120.00
23	-4.35	11.96	120.00	24	-5.63	12.07	120.00
25	-6.95	12.04	120.00	26	-7.97	11.38	120.00
27	-8.78	10.47	120.00	28	-9.46	9.46	120.00
29	-10.26	8.61	120.00	30	-10.78	7.55	120.00
31	-11.41	6.59	120.00	32	-11.82	5.51	120.00
33	-11.87	4.32	120.00	34	-11.67	3.13	120.00
35	-11.26	1.99	120.00	36	-10.86	.95	120.00
37	-10.50	0.00	120.00	38	-10.16	-.89	120.00
39	-9.83	-1.73	120.00	40	-9.52	-2.55	120.00
41	-9.21	-3.35	120.00	42	-8.90	-4.15	120.00
43	-8.59	-4.96	120.00	44	-8.27	-5.79	120.00
45	-7.94	-6.66	120.00	46	-7.54	-7.54	120.00
47	-7.06	-8.41	120.00	48	-6.40	-9.14	120.00
49	-5.58	-9.67	120.00	50	-4.66	-10.00	120.00
51	-3.67	-10.08	120.00	52	-2.66	-9.94	120.00
53	-1.65	-9.34	120.00	54	-.78	-8.86	120.00
55	-0.00	-8.35	120.00	56	.70	-7.94	120.00
57	1.33	-7.52	120.00	58	1.89	-7.05	120.00
59	2.44	-6.71	120.00	60	3.00	-6.44	120.00
61	3.57	-6.18	120.00	62	4.15	-5.92	120.00
63	4.79	-5.71	120.00	64	5.48	-5.48	120.00
65	6.04	-5.07	120.00	66	6.72	-4.71	120.00
67	7.61	-4.39	120.00	68	8.69	-4.05	120.00
69	9.99	-3.64	120.00	70	11.59	-3.11	120.00
71	14.21	-2.51	120.00	72	15.94	-1.39	120.00
73	16.57	-0.00	120.00				
1	18.67	0.00	124.00	2	18.55	1.62	124.00
3	17.93	3.16	124.00	4	17.04	4.57	124.00
5	15.71	5.72	124.00	6	14.21	6.63	124.00

7	12.71	7.34	124.00	8	11.29	7.91	124.00
9	9.91	8.32	124.00	10	8.44	8.44	124.00
11	7.48	8.92	124.00	12	6.51	9.30	124.00
13	5.56	9.63	124.00	14	4.63	9.92	124.00
15	3.71	10.19	124.00	16	2.81	10.47	124.00
17	1.87	10.61	124.00	18	.96	10.95	124.00
19	0.00	11.23	124.00	20	-.98	11.18	124.00
21	-2.04	11.56	124.00	22	-3.16	11.81	124.00
23	-4.42	12.15	124.00	24	-5.84	12.53	124.00
25	-7.23	12.53	124.00	26	-8.61	12.29	124.00
27	-9.85	11.74	124.00	28	-10.88	10.88	124.00
29	-11.65	9.78	124.00	30	-12.10	8.47	124.00
31	-12.18	7.03	124.00	32	-11.96	5.58	124.00
33	-11.46	4.17	124.00	34	-10.98	2.94	124.00
35	-10.57	1.86	124.00	36	-10.20	.89	124.00
37	-9.85	0.00	124.00	38	-9.53	-.83	124.00
39	-9.23	-1.63	124.00	40	-8.94	-2.39	124.00
41	-8.65	-3.15	124.00	42	-8.36	-3.90	124.00
43	-8.07	-4.66	124.00	44	-7.77	-5.44	124.00
45	-7.42	-6.23	124.00	46	-7.00	-7.00	124.00
47	-6.46	-7.69	124.00	48	-5.80	-8.28	124.00
49	-5.05	-8.75	124.00	50	-4.23	-9.07	124.00
51	-3.37	-9.25	124.00	52	-2.48	-9.27	124.00
53	-1.61	-9.13	124.00	54	-.77	-8.85	124.00
55	-0.00	-8.44	124.00	56	.70	-7.96	124.00
57	1.33	-7.52	124.00	58	1.89	-7.05	124.00
59	2.44	-6.71	124.00	60	3.00	-6.44	124.00
61	3.57	-6.18	124.00	62	4.15	-5.92	124.00
63	4.79	-5.71	124.00	64	5.48	-5.48	124.00
65	6.04	-5.07	124.00	66	6.72	-4.71	124.00
67	7.61	-4.39	124.00	68	8.69	-4.05	124.00
69	9.99	-3.64	124.00	70	11.58	-3.10	124.00
71	14.63	-2.58	124.00	72	17.36	-1.52	124.00
73	18.67	-0.00	124.00				
1	20.78	0.00	128.00	2	21.21	1.86	128.00
3	20.42	3.60	128.00	4	19.68	5.27	128.00
5	18.25	6.64	128.00	6	16.46	7.67	128.00
7	14.37	8.29	128.00	8	12.64	8.85	128.00
9	11.04	9.26	128.00	10	9.71	9.71	128.00
11	8.28	9.86	128.00	12	7.11	10.15	128.00
13	6.05	10.49	128.00	14	5.08	10.89	128.00
15	4.03	11.07	128.00	16	3.00	11.19	128.00
17	2.04	11.59	128.00	18	1.02	11.63	128.00
19	0.00	11.75	128.00	20	-1.06	12.13	128.00
21	-2.18	12.35	128.00	22	-3.39	12.66	128.00
23	-4.69	12.89	128.00	24	-6.08	13.03	128.00
25	-7.45	12.91	128.00	26	-8.71	12.44	128.00
27	-9.78	11.66	128.00	28	-10.62	10.62	128.00
29	-11.17	9.37	128.00	30	-11.33	7.94	128.00
31	-11.19	6.46	128.00	32	-10.70	4.99	128.00
33	-10.21	3.72	128.00	34	-9.79	2.62	128.00
35	-9.42	1.66	128.00	36	-9.09	.80	128.00
37	-8.78	0.00	128.00	38	-8.50	-.74	128.00
39	-8.23	-1.45	128.00	40	-7.96	-2.13	128.00
41	-7.71	-2.80	128.00	42	-7.45	-3.47	128.00
43	-7.19	-4.15	128.00	44	-6.92	-4.85	128.00
45	-6.62	-5.55	128.00	46	-6.25	-6.25	128.00
47	-5.78	-6.88	128.00	48	-5.21	-7.44	128.00
49	-4.56	-7.90	128.00	50	-3.85	-8.25	128.00
51	-3.09	-8.48	128.00	52	-2.30	-8.59	128.00
53	-1.51	-8.58	128.00	54	-.74	-8.44	128.00

55	-0.00	-8.18	128.00	56	.68	-7.81	128.00
57	1.29	-7.34	128.00	58	1.88	-7.02	128.00
59	2.44	-6.72	128.00	60	3.00	-6.44	128.00
61	3.57	-6.18	128.00	62	4.15	-5.92	128.00
63	4.79	-5.71	128.00	64	5.48	-5.48	128.00
65	6.04	-5.07	128.00	66	6.72	-4.71	128.00
67	7.61	-4.39	128.00	68	8.69	-4.05	128.00
69	9.99	-3.64	128.00	70	11.59	-3.11	128.00
71	14.59	-2.57	128.00	72	18.62	-1.63	128.00
73	20.78	-0.00	128.00				
1	24.35	0.00	132.00	2	24.23	2.12	132.00
3	23.81	4.20	132.00	4	22.51	6.03	132.00
5	20.33	7.40	132.00	6	17.98	8.38	132.00
7	15.91	9.18	132.00	8	13.67	9.57	132.00
9	12.05	10.11	132.00	10	10.67	10.67	132.00
11	9.17	10.93	132.00	12	7.80	11.14	132.00
13	6.55	11.34	132.00	14	5.46	11.70	132.00
15	4.40	12.10	132.00	16	3.35	12.51	132.00
17	2.27	12.85	132.00	18	1.14	13.04	132.00
19	0.00	13.23	132.00	20	-1.17	13.39	132.00
21	-2.43	13.79	132.00	22	-3.79	14.16	132.00
23	-5.17	14.20	132.00	24	-6.50	13.94	132.00
25	-7.74	13.40	132.00	26	-8.82	12.59	132.00
27	-9.70	11.56	132.00	28	-10.33	10.33	132.00
29	-10.67	8.95	132.00	30	-10.66	7.46	132.00
31	-10.42	6.01	132.00	32	-9.92	4.63	132.00
33	-9.46	3.44	132.00	34	-9.08	2.43	132.00
35	-8.73	1.54	132.00	36	-8.42	.74	132.00
37	-8.14	0.00	132.00	38	-7.88	-.69	132.00
39	-7.62	-1.34	132.00	40	-7.38	-1.98	132.00
41	-7.14	-2.60	132.00	42	-6.90	-3.22	132.00
43	-6.66	-3.85	132.00	44	-6.42	-4.49	132.00
45	-6.13	-5.15	132.00	46	-5.81	-5.81	132.00
47	-5.42	-6.45	132.00	48	-4.93	-7.04	132.00
49	-4.37	-7.57	132.00	50	-3.74	-8.03	132.00
51	-3.06	-8.40	132.00	52	-2.33	-8.68	132.00
53	-1.56	-8.86	132.00	54	-.78	-8.94	132.00
55	-0.00	-8.92	132.00	56	.77	-8.80	132.00
57	1.52	-8.62	132.00	58	2.22	-8.29	132.00
59	2.91	-7.99	132.00	60	3.63	-7.78	132.00
61	4.36	-7.55	132.00	62	5.09	-7.28	132.00
63	5.88	-7.01	132.00	64	6.79	-6.79	132.00
65	7.82	-6.56	132.00	66	8.94	-6.26	132.00
67	10.09	-5.82	132.00	68	11.40	-5.32	132.00
69	13.23	-4.82	132.00	70	15.23	-4.08	132.00
71	19.01	-3.35	132.00	72	22.77	-1.99	132.00
73	24.35	-0.00	132.00				
1	27.92	0.00	136.00	2	27.87	2.44	136.00
3	26.96	4.75	136.00	4	25.28	6.77	136.00
5	22.79	8.30	136.00	6	20.27	9.45	136.00
7	17.89	10.33	136.00	8	15.97	11.18	136.00
9	14.06	11.80	136.00	10	12.10	12.10	136.00
11	10.60	12.64	136.00	12	9.04	12.91	136.00
13	7.59	13.14	136.00	14	6.21	13.33	136.00
15	4.94	13.57	136.00	16	3.74	13.98	136.00
17	2.53	14.33	136.00	18	1.26	14.39	136.00
19	0.00	14.60	136.00	20	-1.30	14.91	136.00
21	-2.64	14.96	136.00	22	-3.99	14.88	136.00
23	-5.30	14.56	136.00	24	-6.53	14.01	136.00
25	-7.63	13.21	136.00	26	-8.56	12.22	136.00

27	-9.29	11.07	136.00	28	-9.78	9.78	136.00
29	-10.01	8.40	136.00	30	-9.93	6.95	136.00
31	-9.72	5.61	136.00	32	-9.27	4.32	136.00
33	-8.84	3.22	136.00	34	-8.48	2.27	136.00
35	-8.16	1.44	136.00	36	-7.87	.69	136.00
37	-7.61	0.00	136.00	38	-7.36	-.64	136.00
39	-7.12	-1.26	136.00	40	-6.90	-1.85	136.00
41	-6.67	-2.43	136.00	42	-6.45	-3.01	136.00
43	-6.23	-3.59	136.00	44	-5.98	-4.19	136.00
45	-5.70	-4.78	136.00	46	-5.36	-5.36	136.00
47	-4.95	-5.90	136.00	48	-4.49	-6.41	136.00
49	-3.97	-6.88	136.00	50	-3.41	-7.31	136.00
51	-2.79	-7.68	136.00	52	-2.14	-7.98	136.00
53	-1.45	-8.22	136.00	54	-.73	-8.39	136.00
55	-0.00	-8.48	136.00	56	.74	-8.49	136.00
57	1.49	-8.42	136.00	58	2.22	-8.28	136.00
59	2.91	-8.00	136.00	60	3.63	-7.78	136.00
61	4.36	-7.55	136.00	62	5.09	-7.28	136.00
63	5.88	-7.01	136.00	64	6.79	-6.79	136.00
65	7.82	-6.56	136.00	66	8.94	-6.26	136.00
67	10.09	-5.82	136.00	68	11.40	-5.31	136.00
69	13.26	-4.83	136.00	70	16.01	-4.29	136.00
71	21.90	-3.86	136.00	72	26.22	-2.29	136.00
73	27.92	-0.00	136.00				

1	34.50	-0.00	140.00	2	35.12	3.07	140.00
3	34.21	6.04	140.00	4	31.37	8.40	140.00
5	26.63	9.69	140.00	6	22.50	10.50	140.00
7	19.32	11.16	140.00	8	16.71	11.70	140.00
9	14.49	12.16	140.00	10	12.57	12.56	140.00
11	10.84	12.92	140.00	12	9.28	13.25	140.00
13	7.82	13.55	140.00	14	6.45	13.84	140.00
15	5.14	14.11	140.00	16	3.85	14.38	140.00
17	2.58	14.64	140.00	18	1.31	14.91	140.00
19	0.00	15.18	140.00	20	-1.35	15.44	140.00
21	-2.73	15.49	140.00	22	-4.09	15.27	140.00
23	-5.39	14.79	140.00	24	-6.56	14.06	140.00
25	-7.57	13.11	140.00	26	-8.38	11.97	140.00
27	-8.96	10.67	140.00	28	-9.26	9.26	140.00
29	-9.26	7.77	140.00	30	-8.91	6.24	140.00
31	-8.40	4.85	140.00	32	-7.96	3.71	140.00
33	-7.60	2.76	140.00	34	-7.28	1.95	140.00
35	-7.01	1.24	140.00	36	-6.76	.59	140.00
37	-6.53	0.00	140.00	38	-6.32	-.55	140.00
39	-6.12	-1.08	140.00	40	-5.92	-1.59	140.00
41	-5.73	-2.09	140.00	42	-5.54	-2.59	140.00
43	-5.35	-3.09	140.00	44	-5.15	-3.61	140.00
45	-4.94	-4.15	140.00	46	-4.70	-4.70	140.00
47	-4.41	-5.26	140.00	48	-4.06	-5.79	140.00
49	-3.64	-6.31	140.00	50	-3.17	-6.80	140.00
51	-2.64	-7.25	140.00	52	-2.05	-7.65	140.00
53	-1.41	-8.00	140.00	54	-.72	-8.28	140.00
55	-0.00	-8.50	140.00	56	.76	-8.63	140.00
57	1.53	-8.68	140.00	58	2.32	-8.65	140.00
59	3.10	-8.53	140.00	60	3.91	-8.39	140.00
61	4.76	-8.24	140.00	62	5.66	-8.09	140.00
63	6.65	-7.92	140.00	64	7.73	-7.73	140.00
65	8.96	-7.52	140.00	66	10.39	-7.27	140.00
67	12.09	-6.98	140.00	68	14.19	-6.62	140.00
69	16.90	-6.15	140.00	70	20.57	-5.51	140.00
71	26.11	-4.60	140.00	72	31.97	-2.80	140.00
73	34.50	-0.00	140.00				

1	46.45	0.00	144.00	2	45.76	4.00	144.00
3	44.38	7.83	144.00	4	41.38	11.09	144.00
5	34.14	12.43	144.00	6	27.71	12.92	144.00
7	23.05	13.31	144.00	8	19.45	13.61	144.00
9	16.52	13.86	144.00	10	14.07	14.07	144.00
11	11.96	14.25	144.00	12	10.09	14.40	144.00
13	8.40	14.55	144.00	14	6.85	14.68	144.00
15	5.39	14.80	144.00	16	3.99	14.92	144.00
17	2.65	15.03	144.00	18	1.32	15.14	144.00
19	-0.00	15.26	144.00	20	-1.35	15.35	144.00
21	-2.69	15.25	144.00	22	-3.99	14.89	144.00
23	-5.20	14.29	144.00	24	-6.28	13.46	144.00
25	-7.18	12.43	144.00	26	-7.87	11.24	144.00
27	-8.32	9.91	144.00	28	-8.49	8.49	144.00
29	-8.36	7.02	144.00	30	-7.91	5.54	144.00
31	-7.43	4.29	144.00	32	-7.04	3.28	144.00
33	-6.72	2.45	144.00	34	-6.45	1.73	144.00
35	-6.20	1.09	144.00	36	-5.98	.52	144.00
37	-5.78	0.00	144.00	38	-5.60	-.49	144.00
39	-5.42	-.96	144.00	40	-5.24	-1.40	144.00
41	-5.08	-1.85	144.00	42	-4.91	-2.29	144.00
43	-4.74	-2.73	144.00	44	-4.56	-3.19	144.00
45	-4.37	-3.67	144.00	46	-4.18	-4.18	144.00
47	-3.96	-4.71	144.00	48	-3.68	-5.25	144.00
49	-3.34	-5.78	144.00	50	-2.94	-6.30	144.00
51	-2.47	-6.80	144.00	52	-1.95	-7.26	144.00
53	-1.35	-7.68	144.00	54	-.70	-8.04	144.00
55	-0.00	-8.34	144.00	56	.75	-8.57	144.00
57	1.53	-8.71	144.00	58	2.35	-8.76	144.00
59	3.18	-8.73	144.00	60	4.04	-8.66	144.00
61	4.96	-8.60	144.00	62	5.97	-8.52	144.00
63	7.08	-8.44	144.00	64	8.35	-8.35	144.00
65	9.82	-8.24	144.00	66	11.58	-8.11	144.00
67	13.78	-7.95	144.00	68	16.60	-7.75	144.00
69	20.50	-7.46	144.00	70	26.25	-7.03	144.00
71	36.61	-6.46	144.00	72	44.22	-3.86	144.00
73	46.45	0.00	144.00				

1	52.39	-.01	148.00	2	50.90	4.46	148.00
3	49.31	8.70	148.00	4	46.68	12.51	148.00
5	38.50	14.02	148.00	6	30.73	14.33	148.00
7	25.20	14.55	148.00	8	21.01	14.72	148.00
9	17.69	14.85	148.00	10	14.96	14.95	148.00
11	12.62	15.05	148.00	12	10.59	15.13	148.00
13	8.77	15.20	148.00	14	7.12	15.26	148.00
15	5.58	15.33	148.00	16	4.12	15.38	148.00
17	2.72	15.44	148.00	18	1.36	15.49	148.00
19	-0.00	15.55	148.00	20	-1.35	15.50	148.00
21	-2.68	15.21	148.00	22	-3.93	14.66	148.00
23	-5.05	13.88	148.00	24	-6.01	12.88	148.00
25	-6.75	11.70	148.00	26	-7.26	10.37	148.00
27	-7.49	8.93	148.00	28	-7.40	7.40	148.00
29	-6.95	5.83	148.00	30	-6.45	4.51	148.00
31	-6.06	3.50	148.00	32	-5.74	2.68	148.00
33	-5.48	1.99	148.00	34	-5.25	1.41	148.00
35	-5.06	.89	148.00	36	-4.88	.43	148.00
37	-4.71	0.00	148.00	38	-4.56	-.40	148.00
39	-4.41	-.78	148.00	40	-4.27	-1.14	148.00
41	-4.14	-1.51	148.00	42	-4.00	-1.86	148.00
43	-3.86	-2.23	148.00	44	-3.71	-2.60	148.00
45	-3.56	-2.99	148.00	46	-3.41	-3.41	148.00

47	-3.23	-3.85	148.00	48	-3.04	-4.35	148.00
49	-2.83	-4.90	148.00	50	-2.56	-5.49	148.00
51	-2.22	-6.09	148.00	52	-1.79	-6.68	148.00
53	-1.28	-7.25	148.00	54	-.68	-7.78	148.00
55	0.00	-8.26	148.00	56	.76	-8.66	148.00
57	1.58	-8.98	148.00	58	2.46	-9.19	148.00
59	3.38	-9.29	148.00	60	4.33	-9.28	148.00
61	5.34	-9.26	148.00	62	6.46	-9.23	148.00
63	7.72	-9.20	148.00	64	9.16	-9.16	148.00
65	10.87	-9.12	148.00	66	12.94	-9.06	148.00
67	15.58	-9.00	148.00	68	19.11	-8.91	148.00
69	24.13	-8.78	148.00	70	31.99	-8.57	148.00
71	46.19	-8.14	148.00	72	51.71	-4.53	148.00
73	52.39	-.01	148.00				
1	56.66	.01	152.00	2	55.03	4.82	152.00
3	53.38	9.41	152.00	4	49.63	13.30	152.00
5	39.63	14.42	152.00	6	31.32	14.61	152.00
7	25.64	14.81	152.00	8	21.35	14.96	152.00
9	17.96	15.08	152.00	10	15.18	15.17	152.00
11	12.80	15.26	152.00	12	10.74	15.33	152.00
13	8.89	15.40	152.00	14	7.21	15.45	152.00
15	5.65	15.51	152.00	16	4.17	15.56	152.00
17	2.75	15.61	152.00	18	1.37	15.66	152.00
19	-0.00	15.66	152.00	20	-1.35	15.43	152.00
21	-2.63	14.93	152.00	22	-3.80	14.19	152.00
23	-4.81	13.22	152.00	24	-5.62	12.04	152.00
25	-6.18	10.70	152.00	26	-6.45	9.22	152.00
27	-6.39	7.62	152.00	28	-5.90	5.90	152.00
29	-5.37	4.51	152.00	30	-4.98	3.49	152.00
31	-4.68	2.70	152.00	32	-4.44	2.07	152.00
33	-4.23	1.54	152.00	34	-4.06	1.09	152.00
35	-3.91	.69	152.00	36	-3.77	.33	152.00
37	-3.64	0.00	152.00	38	-3.52	-.31	152.00
39	-3.41	-.60	152.00	40	-3.30	-.88	152.00
41	-3.20	-1.16	152.00	42	-3.09	-1.44	152.00
43	-2.98	-1.72	152.00	44	-2.87	-2.01	152.00
45	-2.75	-2.31	152.00	46	-2.63	-2.63	152.00
47	-2.50	-2.98	152.00	48	-2.35	-3.36	152.00
49	-2.19	-3.79	152.00	50	-2.00	-4.28	152.00
51	-1.77	-4.87	152.00	52	-1.50	-5.59	152.00
53	-1.12	-6.37	152.00	54	-.62	-7.14	152.00
55	-0.00	-7.87	152.00	56	.75	-8.53	152.00
57	1.60	-9.10	152.00	58	2.55	-9.53	152.00
59	3.58	-9.83	152.00	60	4.65	-9.96	152.00
61	5.74	-9.95	152.00	62	6.93	-9.90	152.00
63	8.26	-9.84	152.00	64	9.78	-9.78	152.00
65	11.57	-9.70	152.00	66	13.73	-9.61	152.00
67	16.45	-9.50	152.00	68	20.05	-9.35	152.00
69	25.10	-9.14	152.00	70	32.79	-8.79	152.00
71	48.10	-8.49	152.00	72	55.50	-4.85	152.00
73	56.66	.01	152.00				
1	61.98	-0.00	156.00	2	60.08	5.26	156.00
3	58.41	10.30	156.00	4	52.58	14.08	156.00
5	41.10	14.95	156.00	6	32.25	15.04	156.00
7	26.30	15.19	156.00	8	21.84	15.30	156.00
9	18.33	15.38	156.00	10	15.46	15.46	156.00
11	13.02	15.52	156.00	12	10.90	15.57	156.00
13	9.01	15.61	156.00	14	7.30	15.66	156.00
15	5.71	15.70	156.00	16	4.21	15.73	156.00
17	2.78	15.77	156.00	18	1.38	15.80	156.00

19	0.00	15.67	156.00	20	-1.34	15.27	156.00
21	-2.57	14.61	156.00	22	-3.67	13.69	156.00
23	-4.57	12.55	156.00	24	-5.23	11.22	156.00
25	-5.60	9.70	156.00	26	-5.62	8.03	156.00
27	-5.12	6.10	156.00	28	-4.52	4.52	156.00
29	-4.11	3.45	156.00	30	-3.81	2.67	156.00
31	-3.58	2.07	156.00	32	-3.39	1.58	156.00
33	-3.24	1.18	156.00	34	-3.10	.83	156.00
35	-2.99	.53	156.00	36	-2.88	.25	156.00
37	-2.79	0.00	156.00	38	-2.69	-.24	156.00
39	-2.61	-.46	156.00	40	-2.53	-.68	156.00
41	-2.44	-.89	156.00	42	-2.36	-1.10	156.00
43	-2.28	-1.32	156.00	44	-2.19	-1.54	156.00
45	-2.11	-1.77	156.00	46	-2.01	-2.01	156.00
47	-1.91	-2.28	156.00	48	-1.80	-2.57	156.00
49	-1.67	-2.90	156.00	50	-1.53	-3.28	156.00
51	-1.36	-3.72	156.00	52	-1.14	-4.27	156.00
53	-.88	-4.97	156.00	54	-.51	-5.87	156.00
55	-0.00	-6.79	156.00	56	.67	-7.64	156.00
57	1.48	-8.40	156.00	58	2.42	-9.03	156.00
59	3.46	-9.50	156.00	60	4.57	-9.81	156.00
61	5.73	-9.91	156.00	62	6.93	-9.89	156.00
63	8.26	-9.84	156.00	64	9.80	-9.79	156.00
65	11.60	-9.73	156.00	66	13.80	-9.66	156.00
67	16.58	-9.57	156.00	68	20.27	-9.45	156.00
69	25.49	-9.28	156.00	70	33.56	-8.99	156.00
71	49.72	-8.77	156.00	72	60.28	-5.27	156.00
73	61.98	-0.00	156.00				

Coordinates of the centre of the head of the prosthesis (x,y,z)

40.70 3.98 161.00

APPENDIX 2. FEM-model. Programs PROEM and COSCF

Program PROEM

```
C MESH GENERATOR FOR FEM STRESS ANALYSIS WITH PAFEC OF EMBEDDED
C PROSTHESIS WITH SMOOTH SHAPE
C
      PROGRAM PROEM
      CHARACTER NAME*10
      INTEGER STEP,L1,L2,CCD,UEB,NPPS,NSCH
      INTEGER I,J,K,ISHOR,IDIV,NSTEP,NSEMB,NBLOCK
      INTEGER NRNOD,NSTEM,NTOP,NEMBE,LEV,LESS
      REAL FORCE
      DIMENSION POINTS(30,73,3),RKMP(3),CKMP(3),SKMP(73,3)
      DIMENSION NSCN(5),CMID(30,3),GEM(6,10,3)
C
C NAME: NEAME OF THE FILE CONTAINING THE GEOMETRY OF THE PROSTHESIS
C STEP, L1, L2, UEB: PARAMETERS
C CCD: NECK ANGLE
C NSCH: NUMBER OF STEPS (CONTOURS) OF THE PROSTHESIS
C NPPS: NUMBER OF POINTS PER STEP (CONTOUR)
C I,J,K: POINTERS
C ISHOR: NUMBER OF POINTS PER CONTOUR FOR FEM MODEL
C IDIV: NUMBER OF DIVISIONS OF THE SECTION ITO ELEMENTS
C       IDIV=2 => ONE LAYER OF BRICK + ONE LAYER OF WEDGE ELEMENTS
C       IDIV=3 => TWO LAYERS OF BRICK + ONE LAYER OF WEDGE ELEMENTS
C NSTEP: POINTER FOR NUMBER OF STEPS
C NSEMB: NUMBER OF STEPS IN THE EMBEDDING
C LEV: NUMBER OF THE CONTOUR AT EMBEDDING LEVEL
C LESS: LAST CONTOUR TO BE MEDELLED INSTEPAD OF THE BASIC-CONTOUR
C NRNOD: TOTAL NUMBER OF NODES AND POINTER FOR THE NODE NUMBERS
C NSTEM: NUMBER OF THE LAST NODE OF THE STEM
C NTOP: NUMBER OF THE LAST NODE OF THE TOP (NECK)
C NEMBE: NUMBER OF THE LAST NODE OF THE EMBEDDING
C NBLOCK: POINTER ON NUMBER OF PAFBLOCKS
C FORCE: VALUE OF THE LOAD IN DIN-TEST
C POINTS: ARRAY CONTAINING THE COORDINATES OF THE GEOMETRY POINTS
C NSCN(4): NUMBER OF THE BASIC STEP
C NSCN(5): NUMBER OF THE LAST 8MM-PART
C RKMP: ARRAY CONTAINING THE MIDPOINT OF PROSTHESIS HEAD
C CKMP: ARRAY CONTAINING POINT ON THE NECK AXIS 25MM FROM RKMP
C SKMP: ARRAY CONTAINING POINTS OF THE NECK (FOUR IN EACH LEVEL)
C CMID: ARRAY CONTAINING THE COORDINATES OF THE CENTRE POINT OF
C       EACH CONTOUR (CALCULATED IN SUBROUTINE SECT)
C
C
C READ INPUT FILE
C
      CALL READAT(POINTS,NSCN,RKMP,STEP,L1,L2,UEB,CCD,NSCH,NPPS,NAME)
C
C CALCULATE CENTRE POINTS AN REDEFINE COORDINATES OF THE POINTS
C OF THE CONTOURS
C
      CALL SECT(POINTS,NPPS,NSCN(4),CMID)
C
C SET VALUES OF PARAMETERS
C
      J=0
      LESS=4
C LESS IS ALSO USED IN SUBROUTINE DEFEMBE
```

```
      NSCN(4)=NSCN(4)-LESS
      ISHOR=24
      IDIV=2
      FORCE=-1.0
C
C OPEN OUTPUT FILE
C
      OPEN(12,FILE='PROEM.DAT')
C
C WRITE TITLE
C
      WRITE(12,1)'TITLE FEM-APPROX PROT ',NAME
C
C WRITE CONTROL MODULE
C
      WRITE(12,1)'CONTROL'
      WRITE(12,1)'FULL.CONTROL'
      WRITE(12,1)'REDUCED OUTPUT'
      WRITE(12,1)'STRESS'
      WRITE(12,1)'PHASE=1,2,4,6,7,9'
      WRITE(12,1)'CONTROL.END'
C
C WRITE NODES
C
      WRITE(12,1)'NODES'
      WRITE(12,1)'NODE.NUMBER X Y Z '
1     FORMAT(80A)
C
C DEFINE NODES OF THE STEM
C
      CALL DEFSTE(POINTS,NSCN(4),NSCN(5),NPPS,ISHOR,CMID,J,NRNOD,NSTEM
     +,NSTEP,IDIV)
C
C DEFINE NODES OF THE NECK (TOP)
C
      CALL DEFTOP(RKMP,CKMP,SKMP,NPPS,CCD,NRNOD,ISHOR,J,IDIV,NTOP)
C
C DEFINE NODES OF THE EMBEDDING
C
      CALL DEFEMB(POINTS,NSCN(4),NSCN(5),NPPS,ISHOR,CMID,
     +J,NRNOD,NEMBE,NSEMB,LEV,IDIV,LESS)
C
C WRITE PAFBLOCKS (ELEMENTS)
C
      WRITE(12,1)'PAFBLOCKS'
      WRITE(12,1)'BLOCK.NUMBER TYPE ELEMENT.TYPE PROPERTIES
     + N1 N2 N5 TOPOLOGY'
      NBLOCK=0
C
C DEFINE PAFBLOCKS OF THE STEM AND THE NECK
C
      CALL PROTE(NBLOCK,NSTEM,NSTEP,NRNOD,ISHOR,IDIV)
C
C DEFINE PAFBLOCKS OF THE EMBEDDING
C
      CALL EMBE(NBLOCK,NEMBE,NSEMB,NRNOD,ISHOR,NTOP,IDIV)
C
C DEFINE MESH AND REFERENCE SPACING LIST
C
      WRITE(12,1)'MESH'
      WRITE(12,1)'REFERENCE SPACING.LIST'
      WRITE(12,1)'1 1'
```

```
      WRITE(12,1)'2 2'
      WRITE(12,1)'3 3'
      WRITE(12,1)'4 5'
      WRITE(12,1)'5 5'
      WRITE(12,1)'6 6'
      WRITE(12,1)'7 10'
C
C DEFINE GAPS AS INTERFACE ELEMENTS
C (COMMENTED OUT)
C
C      CALL GAPS(NTOP,ISHOR,NSEMB,IDIV)
C
C WRITE RESTRAINTS
C
      WRITE(12,1)'RESTRAINTS'
      WRITE(12,1)'NODE.NUMBER DIRECTION'
C
C DEFINE NODES TO BE CONSTRAINED
C
      CALL BOND(NRNOD,NSTEM,ISHOR,IDIV,NTOP,NSEMB)
C
C SPECIFY MATERIAL PROPERTIES
C
      WRITE(12,1)'MATERIAL'
      WRITE(12,1)'MATERIAL.NUMBER E NU'
C
C TITANIUM
C E MUST BE GIVEN IN N/MM2
C
      WRITE(12,1)'20 110000 0.35'
C
C PMMA
C E MUST BE GIVEN IN N/MM2
C
      WRITE(12,1)'21 3500 0.35'
C
C WRITE LOADING
C
      WRITE(12,1)'LOADS'
      WRITE(12,1)'NODE.NUMBER DIRECTIONS.OF.LOAD VALUE.OF.LOAD'
C
C DEFINE LOADS
C
      CALL DEFLOD(FORCE,CCD,NTOP,ISHOR,IDIV,RKMP,CKMP)
C
C WRITE END OF DATA
C
      WRITE(12,1)'END.OF.DATA'
C
      STOP
      END
C
C END OF MAIN PROGRAM PROEM
C
C
C
C
      SUBROUTINE READAT(POINTS,NSCN,RKMP,STEP,L1,L2,UEB,CCD,NSCH,
     +NPPS,NAME)
      CHARACTER NAME*10
      DIMENSION POINTS(30,73,3)
      DIMENSION NSCN(5)
```

```
      DIMENSION RKMP(3)
      INTEGER STEP,NSCH,NPPS,CCD,L1,L2,UEB,I,J,K
C
C THIS SUBROUTINE READS THE INPUT FILE
C
      PRINT*,'ENTER THE NAME OF THE INPUT-FILE NAME '
      PRINT*,'                        (UP TO 10 CHARACTERS)'
      READ(*,7) NAME
7     FORMAT(A10)
      OPEN(10,FILE=NAME,STATUS='OLD')
C
C READ PARAMETERS
C
      READ(10,*) STEP
      DO 100 I=1,5
       READ(10,*) NSCN(I)
100   CONTINUE
      READ(10,*) NSCH
      READ(10,*) NPPS
      READ(10,*) CCD
      READ(10,*) L1
      READ(10,*) L2
      READ(10,*) UEB
C
C READ THE COORDINATES OF POINTS
C
      DO 110 I=1,NSCH
       DO 120 J=1,NPPS
        DO 130 K=1,3
         READ(10,*) POINTS(I,J,K)
130     CONTINUE
120    CONTINUE
110   CONTINUE
C
C READ THE COORDINATES OF THE CENTRE OF THE HEAD
C
      DO 140 I=1,3
       READ(10,*) RKMP(I)
140   CONTINUE
C
      PRINT*,'DATA SUCESSFULLY READ'
C
      CLOSE(10)
      RETURN
      END
C
C END READAT
C
C
C
      SUBROUTINE SECT(POINTS,NPPS,NSCN4,CMID)
      INTEGER I,J,K,L,OFFSET,NSCN4
      REAL PI,X,X1,X2,Y,Y1,Y2,A1,A2,B1,B2,D1,D2,W,ALPHA
      DIMENSION POINTS(30,73,3),SEC(0:72,3),CMID(30,3),WEIG(73)
C
C THIS SUBROUTINE CALCULATES 'CENTRES' OF THE SECTIONS
C AND NEW POINTS AROUND THESE
C
C CMID: COORDINATES OF THE CENTRE POINTS
C
      DO 100 I=1,NSCN4
       DO 110 J=1,NPPS
```

```
        DO 120 K=1,3
         SEC(J-1,K)=POINTS(I,J,K)
120     CONTINUE
110    CONTINUE
C
C COMPUTING WEIGHTS
C
       W=0.0
       DO 130 J=0,NPPS-1
        D1=SQRT((SEC(J,1)-SEC(MOD(J+1,NPPS-1),1))**2+(SEC(J,2)-
     +  SEC(MOD(J+1,NPPS-1),2))**2)
        D2=SQRT((SEC(J,1)-SEC(MOD(J+NPPS-2,NPPS-1),1))**2+(SEC(J,2)-
     +  SEC(MOD(J+NPPS-2,NPPS-1),2))**2)
        WEIG(J+1)=(D1+D2)/2
        W=W+WEIG(J+1)
130    CONTINUE
C
C CALCULATE THE COORDINATES OF THE CENTRE POINTS
C
       DO 140 K=1,3
        CMID(I,K)=0.0
140    CONTINUE
       DO 145 J=1,NPPS
        CMID(I,1)=WEIG(J)*SEC(J-1,1)+CMID(I,1)
        CMID(I,2)=WEIG(J)*SEC(J-1,2)+CMID(I,2)
145    CONTINUE
       CMID(I,1)=CMID(I,1)/W
       CMID(I,2)=CMID(I,2)/W
       CMID(I,3)=SEC(1,3)
C
C FIND NEW POINTS ON THE CONTOUT IN ALPHA DEGREES STEPS
C AROUND CMIDS STARTING WITH THE POINT NO. 1
C
       PI=4*ATAN(1.0)
       ALPHA=2*PI/(NPPS-1)
        SEC(0,1)=POINTS(I,1,1)
        SEC(0,2)=POINTS(I,1,2)
       K=1
       DO 210 J=0,NPPS-2
        X1=SEC(J,1)-CMID(I,1)
        Y1=SEC(J,2)-CMID(I,2)
        D1=SQRT(X1**2+Y1**2)
99      X2=POINTS(I,MOD(K,NPPS)+1,1)-CMID(I,1)
        Y2=POINTS(I,MOD(K,NPPS)+1,2)-CMID(I,2)
        D2=SQRT(X2**2+Y2**2)
       IF(((X1*X2+Y1*Y2)/(D1*D2)).GT.(COS(ALPHA)))THEN
         K=K+1
         GOTO 99
        END IF
        X=X1*COS(ALPHA)-Y1*SIN(ALPHA)
        Y=X1*SIN(ALPHA)+Y1*COS(ALPHA)
        X=X+CMID(I,1)
        Y=Y+CMID(I,2)
C
        IF((ABS(X-CMID(I,1)).GT.1E-12))THEN
         A1=(Y-CMID(I,2))/(X-CMID(I,1))
         B1=Y-A1*X
         IF((ABS(POINTS(I,MOD(K,NPPS)+1,1)-SEC(J,1)).GT.1E-12))THEN
      A2=(POINTS(I,MOD(K,NPPS)+1,2)-SEC(J,2))/(POINTS(I,MOD(K,NPPS)+
     +    1,1)-SEC(J,1))
         B2=POINTS(I,MOD(K,NPPS)+1,2)-A2*POINTS(I,MOD(K,NPPS)+1,1)
        SEC(J+1,1)=(B2-B1)/(A1-A2)
```

```
          SEC(J+1,2)=A1*SEC(J+1,1)+B1
          ELSE
           A2=0.0
           B2=POINTS(I,MOD(K,NPPS)+1,2)
           SEC(J+1,1)=(B2-B1)/A1
           SEC(J+1,2)=B2
          END IF
         ELSE
          A1=0.0
          B1=Y
          IF((ABS(POINTS(I,MOD(K,NPPS)+1,1)-SEC(J,1)).GT.1E-12))THEN
       A2=(POINTS(I,MOD(K,NPPS)+1,2)-SEC(J,2))/(POINTS(I,MOD(K,NPPS)+
      +    1,1)-SEC(J,1))
          B2=POINTS(I,MOD(K,NPPS)+1,2)-A2*POINTS(I,MOD(K,NPPS)+1,1)
          SEC(J+1,1)=(B1-B2)/A2
          SEC(J+1,2)=B1
         ELSE
          SEC(J+1,1)=POINTS(I,MOD(K,NPPS)+1,1)
          SEC(J+1,2)=POINTS(I,MOD(K,NPPS)+1,2)
         END IF
        END IF
210     CONTINUE
C
C CALCULATE OFFSET, I.E., THE NUMBER OF THE POINT, THE Y COORDINATE
C OF WHICH IS CLOSEST OT THE Y-COORDINATE OF THE CENTRE POINT
C
       L=IFIX((PI/4/ALPHA))
       OFFSET=0
       D1=ABS(CMID(I,2)-SEC(0,2))
       DO 205 J=-L,L
        Y=SEC(MOD(NPPS+L,NPPS),2)
        D2=ABS(CMID(I,2)-Y)
        IF(D2.LT.D1) THEN
         OFFSET=MOD(NPPS+L,NPPS)
        ENDIF
205    CONTINUE
C
C WRITE THE COORDINATES OF NEW POINTS
C
        DO 220 J=1,NPPS
         DO 230 K=1,2
          POINTS(I,J,K)=SEC(MOD(J-1+OFFSET,NPPS),K)
230      CONTINUE
220     CONTINUE
100    CONTINUE
C
       RETURN
       END
C
C END SECT
C
C
C
       SUBROUTINE DEFSTE(POINTS,NSCN4,NSCN5,NPPS,ISHOR,CMID,J
      +,NRNOD,NSTEM,NSTEP,IDIV)
       INTEGER NSCN4,NSCN5,ISHOR,NPPS,J,NRNOD,NSTEM
       INTEGER I,K,L,M,NSTEP,IDIV,INT
       REAL PI
       DIMENSION POINTS(30,73,3),CMID(30,3)
       DIMENSION ARRAY(3)
C
C THIS SUBROPUTINE DEFINES NODES OF THE STEM.
```

```
C
      PI=ATAN(1.0)*4
      INT=(NPPS-1)/ISHOR
      NRNOD=0
C
C INT IS THE INTERVAL USED IN ORDER TO PICK OUT THE POINTS
C
C DEFINE ZERO LEVEL
C
      WRITE(12,2)'C DEFINE POINTS ON ZERO LEVEL'
      DO 100 I=0,IDIV-1
       WRITE(12,2)'C STEM SECTION'
       DO 110 K=0,ISHOR-1
        ARRAY(1)=POINTS(1,MOD(K*INT+J,NPPS-2)+1,1)
        ARRAY(2)=POINTS(1,MOD(K*INT+J,NPPS-2)+1,2)
        ARRAY(3)=0.0
        ARRAY(1)=ARRAY(1)-(ARRAY(1)-CMID(1,1))/IDIV*I
        ARRAY(2)=ARRAY(2)-(ARRAY(2)-CMID(1,2))/IDIV*I
        NRNOD=NRNOD+1
        WRITE(12,1)NRNOD,(ARRAY(L),L=1,3)
1       FORMAT(I5,1X,E15.7,1X,E15.7,1X,E15.7)
110    CONTINUE
2      FORMAT(80A)
100   CONTINUE
C
      NRNOD=NRNOD+1
      WRITE(12,2)'C MIDPOINT ZERO LEVEL'
      WRITE(12,1)NRNOD,CMID(1,1),CMID(1,2),0.0
C
      WRITE(12,2)'C BETWEEN SECTIONS AT Z=4 MM'
      DO 111 K=0,ISHOR-1
       IF((MOD(K,2)).EQ.1)GOTO 111
       ARRAY(1)=POINTS(1,MOD(K*INT+J,NPPS-2)+1,1)
       ARRAY(2)=POINTS(1,MOD(K*INT+J,NPPS-2)+1,2)
       ARRAY(1)=(ARRAY(1)+POINTS(2,MOD(K*INT+J,NPPS-2)+1,1))/2
       ARRAY(2)=(ARRAY(2)+POINTS(2,MOD(K*INT+J,NPPS-2)+1,2))/2
       ARRAY(3)=4.0
       NRNOD=NRNOD+1
       WRITE(12,1)NRNOD,(ARRAY(L),L=1,3)
111   CONTINUE
C
C DEFINE REST OF THE STEM
C
      NSTEP=0
      DO 115 M=1,NSCN4
       IF(M.GT.NSCN5)THEN
        IF(MOD(M-NSCN5,2).GT.0)GOTO 114
       ENDIF
       NSTEP=NSTEP+1
      DO 120 I=0,IDIV-1
       WRITE(12,2)'C STEM SECTION ', NSTEP*8, ' MM'
       DO 130 K=0,ISHOR-1
        ARRAY(1)=POINTS(M,MOD(K*INT+J,NPPS-2)+1,1)
        ARRAY(2)=POINTS(M,MOD(K*INT+J,NPPS-2)+1,2)
        ARRAY(3)=POINTS(M,1,3)
        ARRAY(1)=ARRAY(1)-(ARRAY(1)-CMID(M,1))/IDIV*I
        ARRAY(2)=ARRAY(2)-(ARRAY(2)-CMID(M,2))/IDIV*I
        NRNOD=NRNOD+1
        WRITE(12,1)NRNOD,(ARRAY(L),L=1,3)
130    CONTINUE
120   CONTINUE
C
```

```
        NRNOD=NRNOD+1
        WRITE(12,2)'C MIDPOINT'
        WRITE(12,1)NRNOD,CMID(M,1),CMID(M,2),CMID(M,3)
C
       WRITE(12,2)'C BETWEEN SECTIONS', NSTEP*8+4,' MM'
        DO 125 K=0,ISHOR-1
         IF(((MOD(K,2)).EQ.1).OR.(M.GE.(NSCN4-1)))GOTO 125
         ARRAY(1)=POINTS(M,MOD(K*INT+J,NPPS-2)+1,1)
         ARRAY(2)=POINTS(M,MOD(K*INT+J,NPPS-2)+1,2)
         ARRAY(1)=(ARRAY(1)+POINTS(M+1,MOD(K*INT+J,NPPS-2)+1,1))/2
         ARRAY(2)=(ARRAY(2)+POINTS(M+1,MOD(K*INT+J,NPPS-2)+1,2))/2
         ARRAY(3)=POINTS(M,1,3)+4.0
         NRNOD=NRNOD+1
         WRITE(12,1)NRNOD,(ARRAY(L),L=1,3)
125     CONTINUE
114    CONTINUE
115    CONTINUE
C
       WRITE(12,2)'C END STEM'
       NSTEM=NRNOD
C
       RETURN
       END
C
C END DEFSTE
C
C
C
       SUBROUTINE DEFTOP(RKMP,CKMP,SKMP,NPPS,CCD,NRNOD,ISHOR,JJ,
      +IDIV,NTOP)
       INTEGER NPPS,CCD,NRNOD,ISHOR,IDIV,NTOP
       INTEGER I,J,JJ,K,L,LEN
       REAL PI,R
       DIMENSION RKMP(3),CKMP(3),SKMP(73,3)
C
C THIS SUBROUTINE DEFINES NODES OF THE NECK PART.
C THE NECK PART IS A SIMPLIFIED STANDARD. IN ORDER TO HAVE A
C SIMPLER MODEL, THE NECK IS SHORTENED ALONG THE NECK AXIS BY
C THE VALUE OF LEN.
C HERE:
C LEN= 25MM
C LEN IS ALSO USED IN SUBROUTINE DEFLOAD IN ORDER TO DEFINE
C THE LOADS
C THE DIAMETER OF THE NECK IS SET TO 2R= 16 MM
C
       LEN=25
       R=8
       J=0
C
C CALCULATE THE CENTRE NODE ALONG THE NECK AXIS AND BY
C LEN DISTANT FROM THE CENTRE OF THE HEAD OF THE PROSTHESIS
C
       PI=4*ATAN(1.0)
       IF(RKMP(1).GT.0.0)THEN
        CKMP(1)=RKMP(1)-LEN*COS((CCD-90)*PI/180)
       ELSE
        CKMP(1)=RKMP(1)+LEN*COS((CCD-90)*PI/180)
       END IF
       CKMP(2)=RKMP(2)
       CKMP(3)=RKMP(3)-LEN*SIN((CCD-90)*PI/180)
C
C CALCULATE NODES AROUND CKMP
```

```
C
      WRITE(12,2)'C DEFINE TOP PART'
      DO 90 K=0,IDIV-1
       WRITE(12,2)'C TOP SECTION'
      DO 100 I=0,ISHOR-1
       SKMP(I+1,1)=CKMP(1)+R*COS(I*2*PI/ISHOR+J*2*PI/(NPPS-1))
       SKMP(I+1,2)=CKMP(2)+R*SIN(I*2*PI/ISHOR+J*2*PI/(NPPS-1))
       SKMP(I+1,3)=CKMP(3)
       SKMP(I+1,1)=SKMP(I+1,1)-(SKMP(I+1,1)-CKMP(1))/IDIV*K
       SKMP(I+1,2)=SKMP(I+1,2)-(SKMP(I+1,2)-CKMP(2))/IDIV*K
       NRNOD=NRNOD+1
       WRITE(12,1)NRNOD,(SKMP(I+1,L),L=1,3)
1      FORMAT(I5,3(1X,E15.7))
100   CONTINUE
2      FORMAT(80A)
90    CONTINUE
      WRITE(12,2)'C MIDPOINT TOP'
C
      NRNOD=NRNOD+1
      WRITE(12,1)NRNOD,(CKMP(L),L=1,3)
C
      WRITE(12,2)'C END TOP'
      NTOP=NRNOD
C
      RETURN
      END
C
C END DEFTOP
C
C
C
      SUBROUTINE DEFEMB(POINTS,NSCN4,NSCN5,NPPS,ISHOR,CMID,J
     +,NRNOD,NEMBE,NSEMB,LEV,IDIV,LESS)
      INTEGER NSCN4,NSCN5,ISHOR,NPPS,J,NRNOD,NEMBE,LESS,IDIV
      INTEGER I,K,L,M,LEV,HIGH,NSEMB,INT
      REAL PI,DICK,ALPHA
      DIMENSION POINTS(30,73,3),CMID(30,3)
      DIMENSION ARRAY(3)
C
C THIS SUBROUTINE DEFINES NODES OF THE EMBEDDING.
C
      PI=ATAN(1.0)*4
      ALPHA=2*PI/(NPPS-1)
      INT=(NPPS-1)/ISHOR
C
C INT IS THE INTERVAL REQUIRED FOR PICKING OUT THE POINTS
C
C DICK IS THE THICKNESS OF THE EMBEDDING
C
      DICK=20
C
C CALCULATE THE EMBEDDING LEVEL: LEV
C
      HIGH=NSCN5*8+(NSCN4+LESS-NSCN5)*4-50
      IF((NSCN5*8).GT.HIGH)THEN
       LEV=HIGH/8-1
      ELSE
       LEV=NSCN5+(HIGH-NSCN5*8)/4-1
      END IF
      LEV=LEV+2
C
C DEFINE LOWER PART OF THE EMBEDDING AROUND THE TIP OF THE PROSTHESIS
```

```
C
      WRITE(12,2)'C DEFINE DISTAL PART OF THE EMBEDDING'
2     FORMAT(80A)
      DO 90 M=0,1
       WRITE(12,2)'C DISTAL SECTION EMBEDDING SURFACE'
       DO 110 K=0,ISHOR-1
        ARRAY(3)=0.0+(M-1)*8
        ARRAY(1)=DICK*COS(MOD(K*INT+J,NPPS-2)*ALPHA)
        ARRAY(2)=DICK*SIN(MOD(K*INT+J,NPPS-2)*ALPHA)
        NRNOD=NRNOD+1
        WRITE(12,1)NRNOD,(ARRAY(L),L=1,3)
1       FORMAT(I5,1X,E15.7,1X,E15.7,1X,E15.7)
110    CONTINUE
       DO 111 I=0,IDIV-1
       WRITE(12,2)'C EMBEDDING SECTION'
        DO 112 K=0,ISHOR-1
         ARRAY(1)=POINTS(1,MOD(K*INT+J,NPPS-2)+1,1)
         ARRAY(2)=POINTS(1,MOD(K*INT+J,NPPS-2)+1,2)
         ARRAY(3)=0.0+(M-1)*8
         ARRAY(1)=ARRAY(1)-(ARRAY(1)-CMID(1,1))/IDIV*I
         ARRAY(2)=ARRAY(2)-(ARRAY(2)-CMID(1,2))/IDIV*I
         NRNOD=NRNOD+1
         WRITE(12,1)NRNOD,(ARRAY(L),L=1,3)
112     CONTINUE
111    CONTINUE
      WRITE(12,2)'C MIDPOINT'
      NRNOD=NRNOD+1
      WRITE(12,1)NRNOD,CMID(1,1),CMID(1,2),ARRAY(3)
90    CONTINUE
C
C
C DEFINE ZERO LEVEL
C
       DO 120 I=0,1
       WRITE(12,2)'C EMBEDDING SECTION '
       DO 130 K=0,ISHOR-1
        ARRAY(1)=POINTS(1,MOD(K*INT+J,NPPS-2)+1,1)
        ARRAY(2)=POINTS(1,MOD(K*INT+J,NPPS-2)+1,2)
        ARRAY(3)=0.0
        ARRAY(1)=ARRAY(1)*I+DICK*COS(MOD(K*INT+J,NPPS-2)*ALPHA)*(1-I)
        ARRAY(2)=ARRAY(2)*I+DICK*SIN(MOD(K*INT+J,NPPS-2)*ALPHA)*(1-I)
        NRNOD=NRNOD+1
        WRITE(12,1)NRNOD,(ARRAY(L),L=1,3)
130    CONTINUE
120    CONTINUE
       WRITE(12,2)'C BETWEEN'
       DO 135 K=0,ISHOR-1
       IF((MOD(K,2)).EQ.1) GOTO 135
        ARRAY(1)=POINTS(1,MOD(K*INT+J,NPPS-2)+1,1)
        ARRAY(2)=POINTS(1,MOD(K*INT+J,NPPS-2)+1,2)
        ARRAY(1)=(ARRAY(1)+POINTS(2,MOD(K*INT+J,NPPS-2)+1,1))/2
        ARRAY(2)=(ARRAY(2)+POINTS(2,MOD(K*INT+J,NPPS-2)+1,2))/2
        ARRAY(3)=4.0
        NRNOD=NRNOD+1
        WRITE(12,1)NRNOD,(ARRAY(L),L=1,3)
135    CONTINUE
C
C
C DEFINE REMAINING NODES OF THE EMBEDDING
C
      NSEMB=1
      DO 115 M=1,LEV
```

```
      WRITE(12,2)'C EMBEDDING'
       IF(M.GT.NSCN5)THEN
        IF(MOD(M-NSCN5,2).GT.0)GOTO 114
       ENDIF
       NSEMB=NSEMB+1
       DO 140 I=0,1
       WRITE(12,2)'C EMBEDDING SECTION '
       DO 150 K=0,ISHOR-1
        ARRAY(1)=POINTS(M,MOD(K*INT+J,NPPS-2)+1,1)
        ARRAY(2)=POINTS(M,MOD(K*INT+J,NPPS-2)+1,2)
        ARRAY(3)=POINTS(M,1,3)
        ARRAY(1)=ARRAY(1)*I+DICK*COS(MOD(K*INT+J,NPPS-2)*ALPHA)*(1-I)
        ARRAY(2)=ARRAY(2)*I+DICK*SIN(MOD(K*INT+J,NPPS-2)*ALPHA)*(1-I)
        NRNOD=NRNOD+1
        WRITE(12,1)NRNOD,(ARRAY(L),L=1,3)
150    CONTINUE
140    CONTINUE
       WRITE(12,2)'C BETWEEN'
       DO 160 K=0,ISHOR-1
        IF(((MOD(K,2)).EQ.1).OR.(M.EQ.LEV))GOTO 160
        ARRAY(1)=POINTS(M,MOD(K*INT+J,NPPS-2)+1,1)
        ARRAY(2)=POINTS(M,MOD(K*INT+J,NPPS-2)+1,2)
        ARRAY(1)=(ARRAY(1)+POINTS(M+1,MOD(K*INT+J,NPPS-2)+1,1))/2
        ARRAY(2)=(ARRAY(2)+POINTS(M+1,MOD(K*INT+J,NPPS-2)+1,2))/2
        ARRAY(3)=POINTS(M,1,3)+4.0
        NRNOD=NRNOD+1
        WRITE(12,1)NRNOD,(ARRAY(L),L=1,3)
160    CONTINUE
C
114    CONTINUE
115    CONTINUE
C
       WRITE(12,2)'C END EMBEDDING'
       NEMBE=NRNOD
C
       RETURN
       END
C
C END DEFEMB
C
C
C
C
C
       SUBROUTINE PROTE(NBLOCK,NSTEM,NSTEP,NTOP,ISHOR,IDIV)
       INTEGER I,J,K,L,NBLOCK,NSTEM,NSTEP,NTOP,ISHOR,IDIV
       INTEGER INT,EL,TYP,N1,N2,N5,PROP,MID
       INTEGER*4 ETYP
       DIMENSION EL(20)
C
C THIS SUBROUTINE DEFINES PAFBLOCKS (ELEMENTS) OF THE STEM
C
       DO 100 I=1,20
        EL(I)=0
100    CONTINUE
C
C PROPERTY 20 = TITANIUM
C
       PROP=20
C
C PARAMETERS FOR SPECIFYING THE DIVISIONS IN PAFBLOCKS
C
```

```
      N1=1
      N2=1
      N5=1
      DO 110 K=1,NSTEP+1
       WRITE(12,2)'C STEM '
2      FORMAT(80A)
       MID=K*(IDIV*ISHOR+1)+(K-1)*ISHOR/2
       INT=(K-1)*(IDIV*ISHOR+ISHOR/2+1)
      IF(K.EQ.(NSTEP+1))THEN
       WRITE(12,2)'C '
C
C DIVISION OF THE NECK PART
C
       N5=4
      END IF
       DO 120 J=1,IDIV
        IF(J.EQ.IDIV)THEN
         TYP=4
C
C WEDGE ELEMENT
C
         ETYP=37210
         DO 130 I=0,ISHOR/2-1
          EL(1)=MID
          EL(2)=INT+1+I*2
          EL(3)=INT+1+MOD((I+1)*2,ISHOR)
          EL(4)=EL(1)+IDIV*ISHOR+1+ISHOR/2
          EL(5)=EL(2)+IDIV*ISHOR+1+ISHOR/2
          EL(6)=EL(3)+IDIV*ISHOR+1+ISHOR/2
          EL(7)=0
          EL(8)=EL(2)+1
          EL(9)=0
          EL(10)=0
          EL(11)=0
          EL(12)=0
          EL(13)=0
          EL(14)=EL(8)+IDIV*ISHOR+1+ISHOR/2
          EL(15)=0
         IF(K.EQ.(NSTEP+1))THEN
          EL(4)=EL(4)-ISHOR/2
          EL(5)=EL(5)-ISHOR/2
          EL(6)=EL(6)-ISHOR/2
          EL(14)=EL(14)-ISHOR/2
         END IF
         NBLOCK=NBLOCK+1
         WRITE(12,1)NBLOCK,TYP,ETYP,PROP,N1,N2,N5,(EL(L),L=1,15)
130     CONTINUE
         ELSE
       WRITE(12,2)'C '
      TYP=1
C
C BRICK ELEMENT
C
      ETYP=37110
      DO 140 I=0,ISHOR/2-1
       EL(1)=1+I*2+INT
       EL(2)=INT+1+MOD((I+1)*2,ISHOR)
       EL(3)=EL(1)+IDIV*ISHOR+1+ISHOR/2
       EL(4)=EL(2)+IDIV*ISHOR+1+ISHOR/2
       EL(5)=EL(1)+ISHOR
       EL(6)=EL(2)+ISHOR
       EL(7)=EL(5)+IDIV*ISHOR+1+ISHOR/2
```

```
        EL(8)=EL(6)+IDIV*ISHOR+1+ISHOR/2
        EL(9)=EL(1)+1
        EL(12)=EL(3)+1
        EL(17)=EL(5)+1
        EL(20)=EL(7)+1
        EL(10)=0
        EL(11)=0
        EL(13)=0
        EL(14)=0
        EL(15)=0
        EL(16)=0
        EL(18)=0
        EL(19)=0
        IF(J.EQ.1)THEN
         EL(10)=MID+I+1
         EL(11)=MID+MOD(I+1,ISHOR/2)+1
        ENDIF
        IF(K.EQ.(NSTEP+1))THEN
         EL(3)=EL(3)-ISHOR/2
         EL(4)=EL(4)-ISHOR/2
         EL(7)=EL(7)-ISHOR/2
         EL(8)=EL(8)-ISHOR/2
         EL(10)=0
         EL(11)=0
         EL(12)=EL(12)-ISHOR/2
         EL(20)=EL(20)-ISHOR/2
        END IF
       NBLOCK=NBLOCK+1
       WRITE(12,1)NBLOCK,TYP,ETYP,PROP,N1,N2,N5,(EL(L),L=1,20)
1      FORMAT(7(I6,1X)/'* ',10(I5,1X)/'* ',10(I5,1X))
140    CONTINUE
       END IF
        INT=INT+ISHOR
120    CONTINUE
110    CONTINUE
C
       RETURN
       END
C
C
C END PROTE
C
C
C
       SUBROUTINE EMBE(NBLOCK,NEMBE,NSEMB,NRNOD,ISHOR,NTOP,IDIV)
       INTEGER I,J,K,L,NBLOCK,NEMBE,NSEMB,NRNOD,ISHOR,NTOP,IDIV
       INTEGER INT,MID,TYP,N1,N2,N5,PROP,EL
       INTEGER*4 ETYP
       DIMENSION EL(20)
C
C THIS SUBROUTINE DEFINES PAFBLOCKS (ELEMENTS) OF THE EMBEDDING
C
       DO 100 I=1,20
        EL(I)=0
100    CONTINUE
C
C PROPERTY 21 = PMMA
C
       PROP=21
       N1=1
C
C N2=2 OR N2 =1. IF THE DISTAL PART OF
```

```
C THE EMBEDDING (JUST BELOW THE STEM) IS MODELLED, N2=1. IF
C THE EMBEDDING AROUND THE STEM IS MODELLED, N2=2
C
      N5=1
C
       WRITE(12,2)'C EMBEDDING SECTION'
2      FORMAT(80A)
       MID=ISHOR*(IDIV+1)+NTOP+1
       INT=NTOP
       DO 77 J=1,IDIV+1
       WRITE(12,2)'C EMBEDDING'
        IF(J.EQ.(IDIV+1))THEN
         TYP=4
         N2=1
         ETYP=37210
         DO 78 I=0,ISHOR/2-1
          EL(1)=MID
          EL(2)=INT+1+I*2
          EL(3)=INT+1+MOD((I+1)*2,ISHOR)
          EL(4)=EL(1)+(IDIV+1)*ISHOR+1
          EL(5)=EL(2)+(IDIV+1)*ISHOR+1
          EL(6)=EL(3)+(IDIV+1)*ISHOR+1
          EL(7)=0
          EL(8)=EL(2)+1
          EL(9)=0
          EL(10)=0
          EL(11)=0
          EL(12)=0
          EL(13)=0
          EL(14)=EL(8)+(IDIV+1)*ISHOR+1
          EL(15)=0
         NBLOCK=NBLOCK+1
         WRITE(12,1)NBLOCK,TYP,ETYP,PROP,N1,N2,N5,(EL(L),L=1,15)
78      CONTINUE
          ELSE
      TYP=1
      N2=2
      ETYP=37110
      DO 79 I=0,ISHOR/2-1
       EL(1)=1+I*2+INT
       EL(2)=INT+1+MOD((I+1)*2,ISHOR)
       EL(3)=EL(1)+(IDIV+1)*ISHOR+1
       EL(4)=EL(2)+(IDIV+1)*ISHOR+1
       EL(5)=EL(1)+ISHOR
       EL(6)=EL(2)+ISHOR
       EL(7)=EL(5)+(IDIV+1)*ISHOR+1
       EL(8)=EL(6)+(IDIV+1)*ISHOR+1
       EL(9)=EL(1)+1
       EL(12)=EL(3)+1
       EL(17)=EL(5)+1
       EL(20)=EL(7)+1
       EL(10)=0
       EL(11)=0
       EL(13)=0
       EL(14)=0
       EL(15)=0
       EL(16)=0
       EL(18)=0
       EL(19)=0
      NBLOCK=NBLOCK+1
      WRITE(12,1)NBLOCK,TYP,ETYP,PROP,N1,N2,N5,(EL(L),L=1,20)
1     FORMAT(7(I6,1X)/'* ',10(I5,1X)/'* ',10(I5,1X))
```

```
79     CONTINUE
       END IF
        INT=INT+ISHOR
77     CONTINUE
C
       DO 110 K=1,NSEMB-1
        WRITE(12,2)'C EMBEDDING SECTION'
       TYP=1
        N2=2
       ETYP=37110
          INT=NTOP+2*ISHOR*(IDIV+1)+2+(K-1)*(2*ISHOR+ISHOR/2)
       DO 140 I=0,ISHOR/2-1
        EL(1)=1+I*2+INT
        EL(2)=INT+1+MOD((I+1)*2,ISHOR)
       EL(3)=EL(1)+2*ISHOR+ISHOR/2
       EL(4)=EL(2)+2*ISHOR+ISHOR/2
       EL(5)=EL(1)+ISHOR
       EL(6)=EL(2)+ISHOR
       EL(7)=EL(5)+2*ISHOR+ISHOR/2
       EL(8)=EL(6)+2*ISHOR+ISHOR/2
       EL(9)=EL(1)+1
       EL(12)=EL(3)+1
       EL(17)=EL(5)+1
       EL(20)=EL(7)+1
        EL(10)=0
        EL(11)=0
        EL(13)=0
        EL(14)=0
        EL(15)=0
        EL(16)=0
        EL(18)=INT+2*ISHOR+I+1
        EL(19)=INT+2*ISHOR+MOD(I+1,ISHOR/2)+1
       NBLOCK=NBLOCK+1
       WRITE(12,1)NBLOCK,TYP,ETYP,PROP,N1,N2,N5,(EL(L),L=1,20)
140    CONTINUE
110    CONTINUE
C
       RETURN
       END

C
C
C

       SUBROUTINE BOND(NRNOD,NSTEM,ISHOR,IDIV,NTOP,NSEMB)
       INTEGER NRNOD,NSTEM,I,J,NSEMB,ISHOR,IDIV,NTOP,L1,L2
C
C THIS SUBROUTINE DEFINES CONSTRAINTS ON THE EMBEDDING
C
       DO 100 I=NTOP+1,NTOP+ISHOR*(IDIV+1)+1
        WRITE(12,2)I,0
100    CONTINUE
       WRITE(12,1)'C'
1      FORMAT(80A)
       DO 105 I=NTOP+ISHOR*(IDIV+1)+2,NTOP+ISHOR*(IDIV+1)+1+ISHOR
        WRITE(12,2)I,0
105    CONTINUE
       WRITE(12,1)'C'
       L1=NTOP+2*ISHOR*(IDIV+1)+2
       L2=2*ISHOR+ISHOR/2
       DO 110 I=1,NSEMB
        WRITE(12,1)'C'
```

```
       DO 120 J=L1+1,L1+ISHOR
       WRITE(12,2)J,0
2      FORMAT(2(I6,1X))
120    CONTINUE
       L1=L1+L2
110    CONTINUE
       RETURN
       END
C
C END BOND
C
C
C
C
       SUBROUTINE DEFLOD(FORCE,CCD,NTOP,ISHOR,IDIV,RKMP,CKMP)
       INTEGER NTOP,CCD,ISHOR,IDIV
       REAL FORCE,PI,LEN,M1,M2,FX,FZ
       DIMENSION RKMP(3),CKMP(3)
C
C THIS SUBROUTINE CALCULATES THE LOADS ON THE PROSTHESIS
C
C LEN IS THE AMOUNT BY WHICH THE NECK WAS SHORTENED
C IN THE MODEL. LEN WAS ALSO USED IN SUBROUTINE DEFTOP
C WHERE IT WAS SPECIFIED AS LEN=25 MM
C
       LEN=25
       PI=4*ATAN(1.0)
C
C (PI/18) = 10 DEGREES => TILT OF THE PROSTHESIS IN DIN-TEST
C
       FX=FORCE*SIN(PI/18)
       IF(RKMP(1).LT.0.0)THEN
        FX=-FX
       END IF
C
C CALCULATE LOADS WHICH RESULTS FROM THE SHORTENING OF THE NECK
C
       M1=FORCE*(RKMP(1)-CKMP(1))*COS(PI/18)
       M2=FORCE*(RKMP(3)-CKMP(3))*SIN(PI/18)
       M1=M1-M2
       FZ=M1/16
       WRITE(12,2)NTOP,3,FORCE*COS(PI/18)
       WRITE(12,2)NTOP,1,FX
       WRITE(12,2)NTOP-IDIV*ISHOR,3,FZ
       WRITE(12,2)NTOP-IDIV*ISHOR+ISHOR/2,3,-FZ
2      FORMAT(I6,1X,I6,1X,E15.7)
       RETURN
       END
C
C
C
C
       SUBROUTINE GAPS(NTOP,ISHOR,NSEMB,IDIV)
       INTEGER NTOP,ISHOR,NSEMB,IDIV
       INTEGER I,J,K,N1,N2
       K=0
C
C THIS SUBROUTINE SPECIFIES GAPS ELEMENTS ON THE EMBEDDING/
C PROSTHESIS INTERFACE AS AN OPTION TO THE CONTINUOUS
C INTERFACE CONDITIONS ASSUMED IN FEM-ANALYSIS.
C
C (COMMENTED OUT IN THE MAIN PROGRAM)
```

```
C
      WRITE(12,1)'GAPS'
      WRITE(12,1)'N1 N2 AXIS.NUMB DIRE TYPE'
1     FORMAT(80A)
      DO 100 I=1,NSEMB
       DO 110 J=1,ISHOR
        N1=NTOP+J+I*2*ISHOR+ISHOR+K
        N2=J+(I-1)*(IDIV*ISHOR+1)+K
        WRITE(12,2)N1,N2,1,1,1
2       FORMAT(5(I5,1X))
110    CONTINUE
       DO 120 J=1,ISHOR/2
       IF(I.EQ.NSEMB)GOTO 120
        N1=NTOP+(I+1)*2*ISHOR+J+K
        N2=I*(IDIV*ISHOR+1)+J+K
        WRITE(12,2)N1,N2,1,1,1
120    CONTINUE
       K=K+ISHOR/2
100   CONTINUE
      RETURN
      END
C
C END GAPS
```

```
Program COSCF

C THIS PROGRAMM ENABLES THE ESTIMATION OF THE STRESS CONCENTRATION
C FACTORS IN ONE STEP OF THE PROSTHESIS
C
      PROGRAM COSCF
      CHARACTER NAME*10
      INTEGER STEP,L1,L2,CCD,UEB,NPPS,NSCH
      INTEGER NST,ISHOR,NN0,NN1,NN2,NN3
      REAL RAD,XP,YP,ZP
      DIMENSION NSCN(5)
      DIMENSION X(73,30),Y(73,30),Z(73,30)
      DIMENSION XBAR(30),YBAR(30)
      DIMENSION XT(73,30),YT(73,30)
C
C NST: NUMBER OF STEP TO MODELLED
C NN0,NN1,NN2,NN3: TOTAL NUMBER OF NODES IN EACH MODEL
C XP,YP,ZP: COORDINATES OF THE CENTRE OF THE HEAD OF THE PROSTHESIS
C RAD: VALUE OF THE TRANSITION RADIUS
C X,Y,Z: ARRAYS CONTAINING THE COORDINATES OF POINTS
C        OF THE PROSTHESIS
C XBAR, YBAR: COORDINATES OF POINTS OF GRAVITY (CENTROIDS) OF
C             EACH SECTION (CALCULATED IN SUBROUTINE SECT)
C XT,YT: CHANGED COORDINATES OF POINTS OF THE PROSTHESIS
C        AROUND CENTROIDS (SUBROUTINE SECT)
C
      RAD=2.0
      ISHOR=24
      NN0=0
      NN1=0
      NN2=0
      NN3=0
C
C READ INPUT FILE
C
      CALL READAT(X,Y,Z,NSCN,XP,YP,ZP,STEP,L1,L2,UEB,CCD,NSCH,
     +NPPS,NAME)
C
C CALCULATE CENTROIDS AND NEW POINT COORDINATE AROUND CENTROIDS
C
      CALL SECT(X,Y,Z,NPPS,NSCN(4),XBAR,YBAR,XT,YT)
C
      PRINT*,'NUMBER OF THE STEP TO BE MODELLED ?'
      READ(*,*)NST
C
      OPEN(3,FILE='COSF0.DAT')
      OPEN(4,FILE='COSF1.DAT')
      OPEN(5,FILE='COSF2.DAT')
      OPEN(6,FILE='COSF3.DAT')
C
C DEFINE NODES (GEOMETRIES) OF THE MODELS
C
      CALL GEOM0(XT,YT,NAME,NST,RAD,ISHOR,NPPS,NN0)
      CALL GEOM1(XT,YT,NAME,NST,RAD,ISHOR,NPPS,NN1)
      CALL GEOM2(XT,YT,NAME,NST,RAD,ISHOR,NPPS,NN2)
      CALL GEOM3(XT,YT,NAME,NST,RAD,ISHOR,NPPS,NN3)
C
C DEFINE PAFBLOCKS (ELEMENTS)
C
      CALL PABLO0(ISHOR)
      CALL PABLO1(ISHOR)
      CALL PABLO2(ISHOR)
```

```
      CALL PABLO3(ISHOR)
C
C DEFINE RESTRAINTS
C
      CALL RES0(ISHOR,NN0)
      CALL RES1(ISHOR,NN1)
      CALL RES2(ISHOR,NN2)
      CALL RES3(ISHOR,NN3)
C
C DEFINE LOADS
C
      CALL LOA0(XT,YT,NST,XP,ISHOR,NPPS)
      CALL LOA1(XT,YT,NST,XP,ISHOR,NPPS)
      CALL LOA2(XT,YT,NST,XP,ISHOR,NPPS)
      CALL LOA3(XT,YT,NST,XP,ISHOR,NPPS)
C
C
C
      STOP
      END
C
C
      SUBROUTINE READAT(X,Y,Z,NSCN,XP,YP,ZP,STEP,L1,L2,UEB,CCD,NSCH,
     +NPPS,NAME)
      CHARACTER NAME*10
      DIMENSION X(73,30),Y(73,30),Z(73,30)
      DIMENSION NSCN(5)
      INTEGER STEP,NSCH,NPPS,CCD,L1,L2,UEB,I,J,K
      REAL XP,YP,ZP
C
C THIS SUBROUTINE READS THE INPUT FILE
C
      PRINT*,'NAME OF THE DATA FILE'
      READ(*,55)NAME
55    FORMAT(A10)
      OPEN(2,FILE=NAME,STATUS='OLD')
      READ(2,*)STEP
C      PRINT*,STEP
      DO 100 I=1,5
       READ(2,*) NSCN(I)
C           PRINT*,NSCN(I)
100   CONTINUE
      READ(2,*) NSCH
      READ(2,*) NPPS
      READ(2,*) CCD
      READ(2,*) L1
      READ(2,*) L2
      READ(2,*) UEB
      DO 110 I=1,NSCH
       DO 120 J=1,NPPS
        READ(2,*)X(J,I)
        READ(2,*)Y(J,I)
        READ(2,*)Z(J,I)
130    CONTINUE
120    CONTINUE
110   CONTINUE
       READ(2,*)XP
       READ(2,*)YP
       READ(2,*)ZP
      CLOSE(10)
      RETURN
      END
```

```
C
C END READAT
C
C
C
      SUBROUTINE SECT(X,Y,Z,NPPS,NSCN4,XBAR,YBAR,XT,YT)
      INTEGER I,J,K,NSCN4,M,II,IQADR
      REAL B,DELX,H1,H2,YMIN,YMAX
      REAL PI,X3,X1,X2,Y3,Y1,Y2,A1,A2,B1,B2,D1,D2,ALPHA
      DIMENSION A(30,3),AX(30,3),AY(30,3),XMAX(4,30),IPOINT(4,30)
      DIMENSION X(73,30),Y(73,30),Z(73,30),SEC(0:72,3),CMID(30,3)
      DIMENSION XBAR(30),YBAR(30),XT(73,30),YT(73,30)
C
C THIS SUBROUTINE CALCULATES THE CENTROIDS OF EACH SECTION
C AND NEW COORDINATES OF POINT AROUND THE CENTROIDS
C
C CALCULATION OF THE CENTROIDES FOR EACH CONTOUR
C
C FIND MAXIMUM X FOR EACH QUADRANT OF THE CONTOUR
C
      IQADR=(NPPS-1)/4
      DO 10 M=1,NSCN4
       I=0
       DO 11 II=1,4
        XMAX(II,M)=0.
        DO 12 J=1,IQADR
         I=I+1
         IF(ABS(X(I,M)).GT.ABS(XMAX(II,M)))THEN
          XMAX(II,M)=X(I,M)
          IPOINT(II,M)=I
         END IF
12      CONTINUE
11     CONTINUE
10    CONTINUE
C
C LOOP ON NUMBER OF STEPS
C
      DO 20 M=1,NSCN4
C
C LOOP ON NUMBER OF POINTS AROUND PERIPHERY OF STEP
C
C CALCULATION OF STRIP PROPERTIES
C
       DO 21 I=1,NPPS-1
        DELX=ABS(X(I+1,M)-X(I,M))
        B=DELX
C
C 'NEGATIVE' AREAS OF CROSS-SECTION
C
        IF(I.LT.(IQADR+1))THEN
         IF((X(I,M).LT.XMAX(1,M)).AND.(I.LT.IPOINT(1,M)))THEN
          B=-DELX
         END IF
        END IF
        IF((I.GT.(IQADR+1)).AND.(I.LT.(2*IQADR+1)))THEN
         IF((ABS(X(I,M)).LT.ABS(XMAX(2,M))).AND.(I.GT.IPOINT(2,M)))
     +   THEN
          B=-DELX
         END IF
        END IF
        IF((I.GT.(2*IQADR+1)).AND.(I.LT.(3*IQADR+1)))THEN
         IF((ABS(X(I,M)).LT.ABS(XMAX(3,M))).AND.(I.LT.IPOINT(3,M)))
```

```
     +    THEN
           B=-DELX
          END IF
         END IF
         IF(I.GT.(IQADR*3))THEN
          IF((X(I,M).LT.XMAX(4,M)).AND.(I.GT.IPOINT(4,M)))THEN
           B=-DELX
          END IF
         END IF
         IF((ABS(Y(I,M))).LT.ABS(Y(I+1,M))) THEN
          YMIN=Y(I,M)
          YMAX=Y(I+1,M)
         ELSE
          YMIN=Y(I+1,M)
          YMAX=Y(I,M)
         END IF
         H1=ABS(YMIN)
         H2=ABS(YMAX)
         X1=(X(I,M)+X(I+1,M))/2.
         Y1=YMIN/2.
         Y2=YMAX/2.
         A1=B*H1
         A2=B*H2
         X2=X1
C
C CALCULATION OF STEP PROPERTIES, I.E., AREAS AND SECOND MOMENTS
C OF AREAS
C
         A(M,1)=A(M,1)+A1
         A(M,2)=A(M,2)+A2
         AX(M,1)=AX(M,1)+A1*X1
         AX(M,2)=AX(M,2)+A2*X2
         AY(M,1)=AY(M,1)+A1*Y1
         AY(M,2)=AY(M,2)+A2*Y2
21      CONTINUE
C
C AVERAGE MAX AND MIN PROPERTIES
C
        A(M,3)=(A(M,1)+A(M,2))/2.
        AX(M,3)=(AX(M,1)+AX(M,2))/2.
        AY(M,3)=(AY(M,1)+AY(M,2))/2.
C
C CENTROID COORDINATES OF STEP M
C
        XBAR(M)=AX(M,3)/A(M,3)
        YBAR(M)=AY(M,3)/A(M,3)
C
20     CONTINUE
C
C CALCULATE NEW POINTS ON THE CONTOUR IN ALPHA DEGREE STEPS
C AROUND CMIDS STARTING WITH THE POINT NO. 1
C
       PI=4*ATAN(1.0)
       ALPHA=2*PI/(NPPS-1)
       DO 100 I=1,NSCN4
        CMID(I,1)=XBAR(I)
        CMID(I,2)=YBAR(I)
         SEC(0,1)=X(1,I)
         SEC(0,2)=Y(1,I)
        K=1
        DO 210 J=0,NPPS-2
         X1=SEC(J,1)-CMID(I,1)
```

```
          Y1=SEC(J,2)-CMID(I,2)
          D1=SQRT(X1**2+Y1**2)
99        X2=X(MOD(K,NPPS)+1,I)-CMID(I,1)
          Y2=Y(MOD(K,NPPS)+1,I)-CMID(I,2)
          D2=SQRT(X2**2+Y2**2)
          IF(((X1*X2+Y1*Y2)/(D1*D2)).GT.(COS(ALPHA)))THEN
           K=K+1
           GOTO 99
          END IF
          X3=X1*COS(PI/36)-Y1*SIN(ALPHA)
          Y3=X1*SIN(PI/36)+Y1*COS(ALPHA)
          X3=X3+CMID(I,1)
          Y3=Y3+CMID(I,2)
          IF((ABS(X3-CMID(I,1)).GT.1E-12))THEN
           A1=(Y3-CMID(I,2))/(X3-CMID(I,1))
           B1=Y3-A1*X3
           IF((ABS(X(MOD(K,NPPS)+1,I)-SEC(J,1)).GT.1E-12))THEN
            A2=(Y(MOD(K,NPPS)+1,I)-SEC(J,2))/(X(MOD(K,NPPS)+1,I)
     +     -SEC(J,1))
            B2=Y(MOD(K,NPPS)+1,I)-A2*X(MOD(K,NPPS)+1,I)
            SEC(J+1,1)=(B2-B1)/(A1-A2)
            SEC(J+1,2)=A1*SEC(J+1,1)+B1
           ELSE
            A2=0.0
            B2=Y(MOD(K,NPPS)+1,I)
            SEC(J+1,1)=(B2-B1)/A1
            SEC(J+1,2)=B2
           END IF
          ELSE
           A1=0.0
           B1=Y3
           IF((ABS(X(MOD(K,NPPS)+1,I)-SEC(J,1)).GT.1E-12))THEN
            A2=(Y(MOD(K,NPPS)+1,I)-SEC(J,2))/(X(MOD(K,NPPS)+1,I)
     +     -SEC(J,1))
            B2=Y(MOD(K,NPPS)+1,I)-A2*X(MOD(K,NPPS)+1,I)
            SEC(J+1,1)=(B1-B2)/A2
            SEC(J+1,2)=B1
           ELSE
            SEC(J+1,1)=X(MOD(K,NPPS)+1,I)
            SEC(J+1,2)=Y(MOD(K,NPPS)+1,I)
           END IF
          END IF
210      CONTINUE
C
C WRITE NEW POINTS
C
         DO 220 J=1,NPPS
           X(J,I)=SEC(J-1,1)
           Y(J,I)=SEC(J-1,2)
220      CONTINUE
100     CONTINUE
C
C       POINT COORDINATES ABOUT CENTROIDS
C
        DO 300 M=1,NSCN4
         DO 310 I=1,NPPS
          XT(I,M)=X(I,M)-XBAR(M)
          YT(I,M)=Y(I,M)-YBAR(M)
310      CONTINUE
300     CONTINUE
C
C
```

```
      RETURN
      END
C
C END SECT
C
      SUBROUTINE GEOMØ(XT,YT,NAME,NST,RAD,ISHOR,NPPS,NNØ)
      CHARACTER NAME*1Ø
      INTEGER I,J,K,NST,INT,NNØ
      REAL PI,RAD,X,Y
      DIMENSION XT(73,3Ø),YT(73,3Ø)
C
C THIS SUBROUTINE DEFINES NODES FOR A 'SMOOTH' STEP
C
      PI=4.ØØ*ATAN(1.Ø)
      INT=(NPPS-1)/ISHOR
      WRITE(3,1)'TITLE COSFØ - STEP   PROT ',NAME
      WRITE(3,3)'C  STEP NUMBER ',NST
3     FORMAT(A14,I6)
      WRITE(3,1)'CONTROL'
      WRITE(3,1)'FULL.CONTROL'
      WRITE(3,1)'PHASE=1,2,4,6,7,9'
      WRITE(3,1)'CLEAR.FILES'
      WRITE(3,1)'CONTROL.END'
C
      WRITE(3,1)'NODES'
      WRITE(3,1)'NODE.NUMB  X   Y   Z'
1     FORMAT(8ØA)
C
      J=Ø
      WRITE(3,1)'C C FIRST CONTOUR, EDGE'
      DO 1Ø I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)
       Y=YT(K,NST)
       WRITE(3,2)J,X,Y,Ø.Ø
2      FORMAT(I5,1X,E15.7,1X,E15.7,1X,E15.7)
1Ø    CONTINUE
C
      WRITE(3,1)'C FIRST CONTOUR , INSIDE'
      DO 2Ø I=1,ISHOR
        J=J+1
       K=1+(I-1)*INT
        X=XT(K,NST)/2
        Y=YT(K,NST)/2
        WRITE(3,2)J,X,Y,Ø.Ø
2Ø    CONTINUE
      WRITE(3,1)'C MIDPOINT'
      J=J+1
      WRITE(3,2)J,Ø.Ø,Ø.Ø,Ø.Ø
C
C
      WRITE(3,1)'C SECOND CONTOUR, EDGE'
      DO 3Ø I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)
       Y=YT(K,NST)
       WRITE(3,2)J,X,Y,16.Ø
3Ø    CONTINUE
      WRITE(3,1)'C SECOND CONTOUR, INSIDE'
      DO 4Ø I=1,ISHOR
```

```
        J=J+1
        K=1+(I-1)*INT
        X=XT(K,NST)/2
        Y=YT(K,NST)/2
        WRITE(3,2)J,X,Y,16.0
40      CONTINUE
        WRITE(3,1)'C SECOND CONTOUR, MIDPOINT'
        J=J+1
        WRITE(3,2)J,0.0,0.0,16.0
C
        NN0=J
C
C
C
        RETURN
        END
C
C END GEOM0
C
C
        SUBROUTINE PABLO0(ISHOR)
        INTEGER I,J,K,ISHOR
        INTEGER N1,N2,N5,EL,TYP,PROP
        INTEGER*4 ETYP
        DIMENSION EL(20)
C
C THIS SUBROUTINE DEFINES PAFBLOCKS (ELEMENTS) FOR THE SMOOTH STEP
C
        WRITE(3,1)'PAFBLOCCKS'
        WRITE(3,1)'BLOCK.NUMB TYPE ELEM.TYPE PROP N1 N2 N5 TOPO'
1       FORMAT(80A)
C
        WRITE(3,1)'C FIRST OUTSIDE'
        J=0
        N1=1
        N2=1
        N5=5
        ETYP=37110
        TYP=1
        PROP=20
        DO 10 I=1,ISHOR/2
         EL(1)=1+(I-1)*2
         EL(2)=1+MOD(I*2,ISHOR)
         EL(3)=2*ISHOR+1+EL(1)
         EL(4)=2*ISHOR+1+EL(2)
         EL(5)=ISHOR+EL(1)
         EL(6)=ISHOR+EL(2)
         EL(7)=ISHOR+EL(3)
         EL(8)=ISHOR+EL(4)
         EL(9)=EL(1)+1
         EL(10)=0
         EL(11)=0
         EL(12)=EL(3)+1
         EL(13)=0
         EL(14)=0
         EL(15)=0
         EL(16)=0
         EL(17)=EL(5)+1
         EL(18)=0
         EL(19)=0
         EL(20)=EL(7)+1
C
```

```
        J=J+1
        WRITE(3,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
2       FORMAT(7(I6,1X)/'* ',10(I5,1X)/'* ',10(I5,1X))
10      CONTINUE
C
        WRITE(3,1)'C FIRST INSIDE'
        ETYP=37210
        TYP=4
        DO 20 I=1,ISHOR/2
         EL(1)=2*ISHOR+1
         EL(2)=ISHOR+1+(I-1)*2
         EL(3)=ISHOR+1+MOD(I*2,ISHOR)
         EL(4)=EL(1)+2*ISHOR+1
         EL(5)=EL(2)+2*ISHOR+1
         EL(6)=EL(3)+2*ISHOR+1
         EL(7)=0
         EL(8)=EL(2)+1
         EL(9)=0
         EL(10)=0
         EL(11)=0
         EL(12)=0
         EL(13)=0
         EL(14)=EL(5)+1
         EL(15)=0
C
         J=J+1
         WRITE(3,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,15)
20      CONTINUE
C
C
C
        WRITE(3,1)'MESH'
        WRITE(3,1)'REFE SPAC.LIST'
        WRITE(3,1)'1 1'
        WRITE(3,1)'2 2'
        WRITE(3,1)'3 3'
        WRITE(3,1)'4 4'
        WRITE(3,1)'5 5'
        WRITE(3,1)'6 6'
        WRITE(3,1)'7 7'
        WRITE(3,1)'8 8'
        WRITE(3,1)'C '
        WRITE(3,1)'MATERIAL'
        WRITE(3,1)'MATE.NUMB E NU'
        WRITE(3,1)'20 110000 0.35'
C
C
        RETURN
        END
C
C END PABLO0
C
        SUBROUTINE RES0(ISHOR,NN0)
        INTEGER I,ISHOR,NN0
C
C THIS SUBROUTINE WRITES RESTRAINTS OF THE SMOOTH STEP
C
        WRITE(3,1)'RESTRAINTS'
        WRITE(3,1)'NODE.NUMB DIRECTION'
1       FORMAT(80A)
C
        DO 10 I=NN0-2*ISHOR,NN0
```

```
       WRITE(3,2)I,0
2      FORMAT(I6,1X,I6)
10     CONTINUE
C
       RETURN
       END
C
C END RES0
C
       SUBROUTINE LOA0(XT,YT,NST,XP,ISHOR,NPPS)
       INTEGER NST,ISHOR,NPPS,I
       REAL D1,D2,F1,F2,XP
       DIMENSION XT(73,30),YT(73,30)
C
C THIS SUBROUTINE CALCULATES LOADS FOR SMOOTH STEP
C
       D1=SQRT((XT(1,NST))**2+(YT(1,NST))**2)
       D2=SQRT((XT((NPPS-1)/2+1,NST))**2+(YT((NPPS-1)/2+1,NST))**2)
       F1=-0.5/D1
       F2=0.5/D2
       IF(XP.LE.0.0)THEN
        F1=-F1
        F2=-F2
       END IF
C
       WRITE(3,1)'LOADS'
       WRITE(3,1)'NODE.NUMB DIRE.OF.LOAD VALU.OF.LOAD'
1      FORMAT(80A)
C
       WRITE(3,2)1,3,F1
       I=(NPPS-1)/2+1
       WRITE(3,2)I,3,F2
C
2      FORMAT(I6,1X,I6,1X,E15.7)
C
       WRITE(3,1)'END.OF.DATA'
       RETURN
       END
C
C END LOA0
C
       SUBROUTINE GEOM1(XT,YT,NAME,NST,RAD,ISHOR,NPPS,NN1)
       CHARACTER NAME*10
       INTEGER I,J,K,NST,NPPS,ISHOR,INT,NN1
       REAL PI,RAD,AL,X,Y,Z
       DIMENSION XT(73,30),YT(73,30)
C
C THIS SUBROUTINE DEFINES NODES FOR STEP WITH 1/3 OF TRANSITION
C RADIUS
C
       PI=4.0*ATAN(1.0)
       INT=(NPPS-1)/ISHOR
       WRITE(4,1)'TITLE COSF1 - STEP   PROT ',NAME
       WRITE(4,3)'C  STEP NUMBER ',NST
3      FORMAT(A14,I6)
       WRITE(4,1)'CONTROL'
       WRITE(4,1)'FULL.CONTROL'
       WRITE(4,1)'PHASE=1,2,4,6,7,9'
       WRITE(4,1)'CLEAR.FILES'
       WRITE(4,1)'CONTROL.END'
C
       WRITE(4,1)'NODES'
```

```
      WRITE(4,1)'NODE.NUMB  X   Y   Z'
1     FORMAT(80A)
C
      J=0
      WRITE(4,1)'C C FIRST CONTOUR, EDGE'
      DO 10 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)
       Y=YT(K,NST)
       WRITE(4,2)J,X,Y,0.0
2      FORMAT(I5,1X,E15.7,1X,E15.7,1X,E15.7)
10    CONTINUE
C
      WRITE(4,1)'C FIRST CONTOUR , INSIDE'
      DO 20 I=1,ISHOR
        J=J+1
       K=1+(I-1)*INT
        X=XT(K,NST)/2
        Y=YT(K,NST)/2
        WRITE(4,2)J,X,Y,0.0
20    CONTINUE
      WRITE(4,1)'C MIDPOINT'
      J=J+1
      WRITE(4,2)J,0.0,0.0,0.0
C
C
      WRITE(4,1)'C SECOND CONTOUR, EDGE'
      DO 30 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)
       Y=YT(K,NST)
       WRITE(4,2)J,X,Y,8.0
30    CONTINUE
      WRITE(4,1)'C SECOND CONTOUR, INSIDE'
      DO 40 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)/2
       Y=YT(K,NST)/2
       WRITE(4,2)J,X,Y,8.0
40    CONTINUE
      WRITE(4,1)'C SECOND CONTOUR, MIDPOINT'
      J=J+1
      WRITE(4,2)J,0.0,0.0,8.0
C
C
      WRITE(4,1)'C THIRD CONTOUR, STEP'
C
      AL=XT(1,NST)/SQRT(XT(1,NST)**2+YT(1,NST)**2)
      IF(AL.GE.1.0)THEN
       AL=0.0
      ELSE
       AL=ACOS(AL)
      END IF
C
      DO 50 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)+RAD*(1-COS(PI/6))*COS((I-1)*PI/12+AL)
       Y=YT(K,NST)+RAD*(1-COS(PI/6))*SIN((I-1)*PI/12+AL)
```

```
        Z=8.0+RAD*SIN(PI/6)
        WRITE(4,2)J,X,Y,Z
50      CONTINUE
        WRITE(4,1)'C THIRD CONTOUR, INSIDE'
        DO 60 I=1,ISHOR
         J=J+1
         K=1+(I-1)*INT
         X=(XT(K,NST)+RAD*(1-COS(PI/6))*COS((I-1)*PI/12+AL))/2
         Y=(YT(K,NST)+RAD*(1-COS(PI/6))*SIN((I-1)*PI/12+AL))/2
         Z=8.0+RAD*SIN(PI/6)
         WRITE(4,2)J,X,Y,Z
60      CONTINUE
        WRITE(4,1)'C THIRD CONTOUR, MIDPOINT'
        J=J+1
        WRITE(4,2)J,0.0,0.0,8.0+RAD*SIN(PI/6)
C
C
        WRITE(4,1)'C FOURTH CONTOUR, EDGE'
        DO 70 I=1,ISHOR
         J=J+1
         K=1+(I-1)*INT
         X=XT(K,NST)+RAD*(1-COS(PI/6))*COS((I-1)*PI/12+AL)
         Y=YT(K,NST)+RAD*(1-COS(PI/6))*SIN((I-1)*PI/12+AL)
         Z=16.0+RAD*SIN(PI/6)
         WRITE(4,2)J,X,Y,Z
70      CONTINUE
        WRITE(4,1)'C FOURTH CONTOUR, INSIDE'
        DO 80 I=1,ISHOR
         J=J+1
         K=1+(I-1)*INT
         X=(XT(K,NST)+RAD*(1-COS(PI/6))*COS((I-1)*PI/12+AL))/2
         Y=(YT(K,NST)+RAD*(1-COS(PI/6))*SIN((I-1)*PI/12+AL))/2
         Z=16.0+RAD*SIN(PI/6)
         WRITE(4,2)J,X,Y,Z
80      CONTINUE
        WRITE(4,1)'C FOURTH CONTOUR, MIDPOINT'
        J=J+1
        WRITE(4,2)J,0.0,0.0,16.0+RAD*SIN(PI/6)
C
        NN1=J
C
        RETURN
        END
C
C END GEOM1
C
C
        SUBROUTINE PABLO1(ISHOR)
        INTEGER I,J,K,ISHOR
        INTEGER N1,N2,N5,EL,TYP,PROP
        INTEGER*4 ETYP
        DIMENSION EL(20)
C
C THIS SUBROUTINE DEFINES PAFBLOCKS (ELEMENTS) FOR STEP WITH
C 1/3 OF THE TRANSITION RADIUS
C
        WRITE(4,1)'PAFBLOCCKS'
        WRITE(4,1)'BLOCK.NUMB TYPE ELEM.TYPE PROP N1 N2 N5 TOPO'
1       FORMAT(80A)
C
        WRITE(4,1)'C FIRST OUTSIDE'
        J=0
```

```
      N1=1
      N2=1
      N5=5
      ETYP=37110
      TYP=1
      PROP=20
      DO 10 I=1,ISHOR/2
       EL(1)=1+(I-1)*2
       EL(2)=1+MOD(I*2,ISHOR)
       EL(3)=2*ISHOR+1+EL(1)
       EL(4)=2*ISHOR+1+EL(2)
       EL(5)=ISHOR+EL(1)
       EL(6)=ISHOR+EL(2)
       EL(7)=ISHOR+EL(3)
       EL(8)=ISHOR+EL(4)
       EL(9)=EL(1)+1
       EL(10)=0
       EL(11)=0
       EL(12)=EL(3)+1
       EL(13)=0
       EL(14)=0
       EL(15)=0
       EL(16)=0
       EL(17)=EL(5)+1
       EL(18)=0
       EL(19)=0
       EL(20)=EL(7)+1
C
       J=J+1
       WRITE(4,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
2      FORMAT(7(I6,1X)/'* ',10(I5,1X)/'* ',10(I5,1X))
10    CONTINUE
C
      WRITE(4,1)'C FIRST INSIDE'
      ETYP=37210
      TYP=4
      DO 20 I=1,ISHOR/2
       EL(1)=2*ISHOR+1
       EL(2)=ISHOR+1+(I-1)*2
       EL(3)=ISHOR+1+MOD(I*2,ISHOR)
       EL(4)=EL(1)+2*ISHOR+1
       EL(5)=EL(2)+2*ISHOR+1
       EL(6)=EL(3)+2*ISHOR+1
       EL(7)=0
       EL(8)=EL(2)+1
       EL(9)=0
       EL(10)=0
       EL(11)=0
       EL(12)=0
       EL(13)=0
       EL(14)=EL(5)+1
       EL(15)=0
C
       J=J+1
       WRITE(4,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,15)
20    CONTINUE
C
C
      WRITE(4,1)'C SECOND OUTSIDE'
      N5=1
      ETYP=37110
      TYP=1
```

```
      DO 30 I=1,ISHOR/2
       EL(1)=1+(I-1)*2+2*ISHOR+1
       EL(2)=1+MOD(I*2,ISHOR)+2*ISHOR+1
       EL(3)=2*ISHOR+1+EL(1)
       EL(4)=2*ISHOR+1+EL(2)
       EL(5)=ISHOR+EL(1)
       EL(6)=ISHOR+EL(2)
       EL(7)=ISHOR+EL(3)
       EL(8)=ISHOR+EL(4)
       EL(9)=EL(1)+1
       EL(10)=0
       EL(11)=0
       EL(12)=EL(3)+1
       EL(13)=0
       EL(14)=0
       EL(15)=0
       EL(16)=0
       EL(17)=EL(5)+1
       EL(18)=0
       EL(19)=0
       EL(20)=EL(7)+1
C
       J=J+1
       WRITE(4,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
30    CONTINUE
      WRITE(4,1)'C SECOND INSIDE'
      ETYP=37210
      TYP=4
      DO 40 I=1,ISHOR/2
       EL(1)=2*(2*ISHOR+1)
       EL(2)=ISHOR+1+(I-1)*2+2*ISHOR+1
       EL(3)=ISHOR+1+MOD(I*2,ISHOR)+2*ISHOR+1
       EL(4)=EL(1)+2*ISHOR+1
       EL(5)=EL(2)+2*ISHOR+1
       EL(6)=EL(3)+2*ISHOR+1
       EL(7)=0
       EL(8)=EL(2)+1
       EL(9)=0
       EL(10)=0
       EL(11)=0
       EL(12)=0
       EL(13)=0
       EL(14)=EL(5)+1
       EL(15)=0
C
       J=J+1
       WRITE(4,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,15)
40    CONTINUE
C
      WRITE(4,1)'C THIRD OUTSIDE'
      N5=5
      ETYP=37110
      TYP=1
      DO 50 I=1,ISHOR/2
       EL(1)=1+(I-1)*2+2*(2*ISHOR+1)
       EL(2)=1+MOD(I*2,ISHOR)+2*(2*ISHOR+1)
       EL(3)=2*ISHOR+1+EL(1)
       EL(4)=2*ISHOR+1+EL(2)
       EL(5)=ISHOR+EL(1)
       EL(6)=ISHOR+EL(2)
       EL(7)=ISHOR+EL(3)
       EL(8)=ISHOR+EL(4)
```

```
        EL(9)=EL(1)+1
        EL(10)=0
        EL(11)=0
        EL(12)=EL(3)+1
        EL(13)=0
        EL(14)=0
        EL(15)=0
        EL(16)=0
        EL(17)=EL(5)+1
        EL(18)=0
        EL(19)=0
        EL(20)=EL(7)+1
C
        J=J+1
        WRITE(4,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
50      CONTINUE
        WRITE(4,1)'C THIRD INSIDE'
        ETYP=37210
        TYP=4
        DO 60 I=1,ISHOR/2
        EL(1)=3*(2*ISHOR+1)
        EL(2)=ISHOR+1+(I-1)*2+2*(2*ISHOR+1)
        EL(3)=ISHOR+1+MOD(I*2,ISHOR)+2*(2*ISHOR+1)
        EL(4)=EL(1)+2*ISHOR+1
        EL(5)=EL(2)+2*ISHOR+1
        EL(6)=EL(3)+2*ISHOR+1
        EL(7)=0
        EL(8)=EL(2)+1
        EL(9)=0
        EL(10)=0
        EL(11)=0
        EL(12)=0
        EL(13)=0
        EL(14)=EL(5)+1
        EL(15)=0
C
        J=J+1
        WRITE(4,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,15)
60      CONTINUE
C
        WRITE(4,1)'MESH'
        WRITE(4,1)'REFE SPAC.LIST'
        WRITE(4,1)'1 1'
        WRITE(4,1)'2 2'
        WRITE(4,1)'3 3'
        WRITE(4,1)'4 4'
        WRITE(4,1)'5 5'
        WRITE(4,1)'6 6'
        WRITE(4,1)'7 7'
        WRITE(4,1)'8 8'
        WRITE(4,1)'C '
        WRITE(4,1)'MATERIAL'
        WRITE(4,1)'MATE.NUMB E NU'
        WRITE(4,1)'20 110000 0.35'
C
C
        RETURN
        END
C
C END PABLO1
C
        SUBROUTINE RES1(ISHOR,NN1)
```

```
      INTEGER I,ISHOR,NN1
C
C THIS SUBROUTINE INPOSES CONSTRAINTS ON THE MODEL OF STEP
C WITH 1/3 OF THE TRANSITION RADIUS
C
      WRITE(4,1)'RESTRAINTS'
      WRITE(4,1)'NODE.NUMB DIRECTION'
1     FORMAT(80A)
C
      DO 10 I=NN1-2*ISHOR,NN1
       WRITE(4,2)I,0
2      FORMAT(I6,1X,I6)
10    CONTINUE
C
      RETURN
      END
C
C END RES1
C
      SUBROUTINE LOA1(XT,YT,NST,XP,ISHOR,NPPS)
      INTEGER NST,ISHOR,NPPS,I
      REAL D1,D2,F1,F2,XP,X,Y,PI
      DIMENSION XT(73,30),YT(73,30)
C
C THIS SUBROUTINE CALCULATES LOADS FOR STEP WITH 1/3 OF
C TRANSITION RADIUS
C
      PI=4.0*ATAN(1.0)
      X=XT(1,NST)
      Y=YT(1,NST)
      D1=SQRT(X**2+Y**2)
C
      X=XT((NPPS-1)/2+1,NST)
      Y=YT((NPPS-1)/2+1,NST)
      D2=SQRT(X**2+Y**2)
C
      F1=-0.5/D1
      F2=0.5/D2
      IF(XP.LE.0.0)THEN
       F1=-F1
       F2=-F2
      END IF
C
      WRITE(4,1)'LOADS'
      WRITE(4,1)'NODE.NUMB DIRE.OF.LOAD VALU.OF.LOAD'
1     FORMAT(80A)
C
      WRITE(4,2)1,3,F1
      I=(NPPS-1)/2+1
      WRITE(4,2)I,3,F2
C
2     FORMAT(I6,1X,I6,1X,E15.7)
C
      WRITE(4,1)'END.OF.DATA'
      RETURN
      END
C
C END LOA1
C
      SUBROUTINE GEOM2(XT,YT,NAME,NST,RAD,ISHOR,NPPS,NN2)
      CHARACTER NAME*10
      INTEGER I,J,K,NST,ISHOR,NPPS,INT,NN2
```

```
      REAL PI,RAD,AL,X,Y,Z
      DIMENSION XT(73,30),YT(73,30)
C
C THIS SUBROUTINE DEFINES NODES FOR STEP WITH 2/3 OF THE
C TRANSITION RADIUS
C
      PI=4.0*ATAN(1.0)
      INT=(NPPS-1)/ISHOR
      WRITE(5,1)'TITLE COSF2 - STEP   PROT ',NAME
      WRITE(5,3)'C  STEP NUMBER ',NST
3     FORMAT(A14,I6)
      WRITE(5,1)'CONTROL'
      WRITE(5,1)'FULL.CONTROL'
      WRITE(5,1)'PHASE=1,2,4,6,7,9'
      WRITE(5,1)'CLEAR.FILES'
      WRITE(5,1)'CONTROL.END'
C
      WRITE(5,1)'NODES'
      WRITE(5,1)'NODE.NUMB  X   Y   Z'
1     FORMAT(80A)
C
      J=0
      WRITE(5,1)'C C FIRST CONTOUR, EDGE'
      DO 10 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       WRITE(5,2)J,XT(K,NST),YT(K,NST),0.0
2      FORMAT(I5,1X,E15.7,1X,E15.7,1X,E15.7)
10    CONTINUE
C
      AL=XT(1,NST)/SQRT(XT(1,NST)**2+YT(1,NST)**2)
      IF(AL.GE.1.0)THEN
       AL=0.0
      ELSE
       AL=ACOS(AL)
      END IF
C
      WRITE(5,1)'C FIRST CONTOUR , 1ST INSIDE'
      DO 15 I=1,ISHOR
        J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2
       Y=YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2
       WRITE(5,2)J,X,Y,0.0
15    CONTINUE
C
      WRITE(5,1)'C FIRST CONTOUR , 2ND INSIDE'
      DO 20 I=1,ISHOR
        J=J+1
       K=1+(I-1)*INT
       X=(XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2)/2
       Y=(YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2)/2
       WRITE(5,2)J,X,Y,0.0
20    CONTINUE
      WRITE(5,1)'C MIDPOINT'
      J=J+1
      WRITE(5,2)J,0.0,0.0,0.0
C
C
      WRITE(5,1)'C SECOND CONTOUR, EDGE'
      DO 30 I=1,ISHOR
       J=J+1
```

```
        K=1+(I-1)*INT
        WRITE(5,2)J,XT(K,NST),YT(K,NST),8.0
30      CONTINUE
C
        WRITE(5,1)'C SECOND CONTOUR, 1ST INSIDE'
        DO 35 I=1,ISHOR
        J=J+1
        K=1+(I-1)*INT
        X=XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2
        Y=YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2
        WRITE(5,2)J,X,Y,8.0
35      CONTINUE
C
        WRITE(5,1)'C SECOND CONTOUR, 2ND INSIDE'
        DO 40 I=1,ISHOR
        J=J+1
        K=1+(I-1)*INT
        X=(XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2)/2
        Y=(YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2)/2
        WRITE(5,2)J,X,Y,8.0
40      CONTINUE
        WRITE(5,1)'C SECOND CONTOUR, MIDPOINT'
        J=J+1
        WRITE(5,2)J,0.0,0.0,8.0
C
C
        WRITE(5,1)'C THIRD CONTOUR, STEP'
        DO 50 I=1,ISHOR
        J=J+1
        K=1+(I-1)*INT
        X=XT(K,NST)+RAD*(1-COS(PI/6))*COS((I-1)*PI/12+AL)
        Y=YT(K,NST)+RAD*(1-COS(PI/6))*SIN((I-1)*PI/12+AL)
        Z=8.0+RAD*SIN(PI/6)
        WRITE(5,2)J,X,Y,Z
50      CONTINUE
C
        WRITE(5,1)'C THIRD CONTOUR, 1ST INSIDE'
        DO 55 I=1,ISHOR
        J=J+1
        K=1+(I-1)*INT
        X=XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2
        Y=YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2
        Z=8.0+RAD*SIN(PI/INT)
        WRITE(5,2)J,XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2
     1,YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2
     2,8.0+RAD*SIN(PI/INT)
55      CONTINUE
C
        WRITE(5,1)'C THIRD CONTOUR, 2ND INSIDE'
        DO 60 I=1,ISHOR
        J=J+1
        K=1+(I-1)*INT
        X=(XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2)/2
        Y=(YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2)/2
        Z=8.0+RAD*SIN(PI/3)
        WRITE(5,2)J,X,Y,Z
60      CONTINUE
        WRITE(5,1)'C THIRD CONTOUR, MIDPOINT'
        J=J+1
        Z=8.0+RAD*SIN(PI/3)
        WRITE(5,2)J,0.0,0.0,Z
C
```

```
      WRITE(5,1)'C FOURTH CONTOUR, STEP'
      DO 70 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)+RAD*(1-COS(PI/3))*COS((I-1)*PI/12+AL)
       Y=YT(K,NST)+RAD*(1-COS(PI/3))*SIN((I-1)*PI/12+AL)
       Z=8.0+RAD*SIN(PI/3)
       WRITE(5,2)J,X,Y,Z
70    CONTINUE
C
      WRITE(5,1)'C FOURTH CONTOUR, 1ST INSIDE'
      DO 75 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2
       Y=YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2
       Z=8.0+RAD*SIN(PI/3)
       WRITE(5,2)J,X,Y,Z
75    CONTINUE
C
      WRITE(5,1)'C FOURTH CONTOUR, 2ND INSIDE'
      DO 80 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=(XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2)/2
       Y=(YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2)/2
       Z=8.0+RAD*SIN(PI/3)
       WRITE(5,2)J,X,Y,Z
80    CONTINUE
      WRITE(5,1)'C FOURTH CONTOUR, MIDPOINT'
      J=J+1
      Z=8.0+RAD*SIN(PI/3)
      WRITE(5,2)J,0.0,0.0,Z
C
      WRITE(5,1)'C FIFTH CONTOUR, EDGE'
      DO 90 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)+RAD*(1-COS(PI/3))*COS((I-1)*PI/12+AL)
       Y=YT(K,NST)+RAD*(1-COS(PI/3))*SIN((I-1)*PI/12+AL)
       Z=8.0+RAD*SIN(PI/3)
       WRITE(5,2)J,X,Y,Z
90    CONTINUE
C
      WRITE(5,1)'C FIFTH CONTOUR, 1ST INSIDE'
      DO 95 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2
       Y=YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2
       Z=16.0+RAD*SIN(PI/3)
       WRITE(5,2)J,X,Y,Z
95    CONTINUE
C
      WRITE(5,1)'C FIFTH CONTOUR, 2ND INSIDE'
      DO 100 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=(XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2)/2
       Y=(YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2)/2
       Z=16.0+RAD*SIN(PI/3)
       WRITE(5,2)J,X,Y,Z
```

```
100   CONTINUE
      WRITE(5,1)'C FITH CONTOUR, MIDPOINT'
      J=J+1
      Z=16.0+RAD*SIN(PI/3)
      WRITE(5,2)J,0.0,0.0,Z
C
      NN2=J
C
      RETURN
      END
C
C END GEOM2
C
C
      SUBROUTINE PABLO2(ISHOR)
      INTEGER I,J,K,ISHOR
      INTEGER N1,N2,N5,EL,TYP,PROP
      INTEGER*4 ETYP
      DIMENSION EL(20)
C
C THIS SUBROUTINE DEFINES PAFBLOCKS (ELEMENTS) FOR STEP WITH
C 2/3 OF TRANSITION RADIUS
C
      WRITE(5,1)'PAFBLOCCKS'
      WRITE(5,1)'BLOCK.NUMB TYPE ELEM.TYPE PROP N1 N2 N5 TOPO'
1     FORMAT(80A)
C
      WRITE(5,1)'C FIRST OUTSIDE'
      J=0
      N1=1
      N2=1
      N5=5
      ETYP=37110
      TYP=1
      PROP=20
      DO 10 I=1,ISHOR/2
       EL(1)=1+(I-1)*2
       EL(2)=1+MOD(I*2,ISHOR)
       EL(3)=3*ISHOR+1+EL(1)
       EL(4)=3*ISHOR+1+EL(2)
       EL(5)=ISHOR+EL(1)
       EL(6)=ISHOR+EL(2)
       EL(7)=ISHOR+EL(3)
       EL(8)=ISHOR+EL(4)
       EL(9)=EL(1)+1
       EL(10)=0
       EL(11)=0
       EL(12)=EL(3)+1
       EL(13)=0
       EL(14)=0
       EL(15)=0
       EL(16)=0
       EL(17)=EL(5)+1
       EL(18)=0
       EL(19)=0
       EL(20)=EL(7)+1
C
       J=J+1
       WRITE(5,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
2      FORMAT(7(I6,1X)/'* ',10(I5,1X)/'* ',10(I5,1X))
10    CONTINUE
C
```

```
      WRITE(5,1)'C FIRST INSIDE, 1ST'
      ETYP=37110
      TYP=1
      PROP=20
      DO 15 I=1,ISHOR/2
       EL(1)=1+(I-1)*2+ISHOR
       EL(2)=1+MOD(I*2,ISHOR)+ISHOR
       EL(3)=3*ISHOR+1+EL(1)
       EL(4)=3*ISHOR+1+EL(2)
       EL(5)=ISHOR+EL(1)
       EL(6)=ISHOR+EL(2)
       EL(7)=ISHOR+EL(3)
       EL(8)=ISHOR+EL(4)
       EL(9)=EL(1)+1
       EL(10)=0
       EL(11)=0
       EL(12)=EL(3)+1
       EL(13)=0
       EL(14)=0
       EL(15)=0
       EL(16)=0
       EL(17)=EL(5)+1
       EL(18)=0
       EL(19)=0
       EL(20)=EL(7)+1
C
       J=J+1
       WRITE(5,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
15    CONTINUE
C
      WRITE(5,1)'C FIRST INSIDE, 2ND'
      ETYP=37210
      TYP=4
      DO 20 I=1,ISHOR/2
       EL(1)=3*ISHOR+1
       EL(2)=2*ISHOR+1+(I-1)*2
       EL(3)=2*ISHOR+1+MOD(I*2,ISHOR)
       EL(4)=EL(1)+3*ISHOR+1
       EL(5)=EL(2)+3*ISHOR+1
       EL(6)=EL(3)+3*ISHOR+1
       EL(7)=0
       EL(8)=EL(2)+1
       EL(9)=0
       EL(10)=0
       EL(11)=0
       EL(12)=0
       EL(13)=0
       EL(14)=EL(5)+1
       EL(15)=0
C
       J=J+1
       WRITE(5,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,15)
20    CONTINUE
C
C
      WRITE(5,1)'C SECOND OUTSIDE'
      N5=1
      ETYP=37110
      TYP=1
      DO 30 I=1,ISHOR/2
       EL(1)=1+(I-1)*2+3*ISHOR+1
       EL(2)=1+MOD(I*2,ISHOR)+3*ISHOR+1
```

```
      EL(3)=3*ISHOR+1+EL(1)
      EL(4)=3*ISHOR+1+EL(2)
      EL(5)=ISHOR+EL(1)
      EL(6)=ISHOR+EL(2)
      EL(7)=ISHOR+EL(3)
      EL(8)=ISHOR+EL(4)
      EL(9)=EL(1)+1
      EL(10)=0
      EL(11)=0
      EL(12)=EL(3)+1
      EL(13)=0
      EL(14)=0
      EL(15)=0
      EL(16)=0
      EL(17)=EL(5)+1
      EL(18)=0
      EL(19)=0
      EL(20)=EL(7)+1
C
      J=J+1
      WRITE(5,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
30    CONTINUE
C
      WRITE(5,1)'C SECOND INSIDE, 1ST'
      ETYP=37110
      TYP=1
      DO 35 I=1,ISHOR/2
      EL(1)=1+(I-1)*2+ISHOR+3*ISHOR+1
      EL(2)=1+MOD(I*2,ISHOR)+ISHOR+3*ISHOR+1
      EL(3)=3*ISHOR+1+EL(1)
      EL(4)=3*ISHOR+1+EL(2)
      EL(5)=ISHOR+EL(1)
      EL(6)=ISHOR+EL(2)
      EL(7)=ISHOR+EL(3)
      EL(8)=ISHOR+EL(4)
      EL(9)=EL(1)+1
      EL(10)=0
      EL(11)=0
      EL(12)=EL(3)+1
      EL(13)=0
      EL(14)=0
      EL(15)=0
      EL(16)=0
      EL(17)=EL(5)+1
      EL(18)=0
      EL(19)=0
      EL(20)=EL(7)+1
C
      J=J+1
      WRITE(5,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
35    CONTINUE
C
      WRITE(5,1)'C SECOND INSIDE, 2ND'
      ETYP=37210
      TYP=4
      DO 40 I=1,ISHOR/2
      EL(1)=2*(3*ISHOR+1)
      EL(2)=2*ISHOR+1+(I-1)*2+3*ISHOR+1
      EL(3)=2*ISHOR+1+MOD(I*2,ISHOR)+3*ISHOR+1
      EL(4)=EL(1)+3*ISHOR+1
      EL(5)=EL(2)+3*ISHOR+1
      EL(6)=EL(3)+3*ISHOR+1
```

```
        EL(7)=0
        EL(8)=EL(2)+1
        EL(9)=0
        EL(10)=0
        EL(11)=0
        EL(12)=0
        EL(13)=0
        EL(14)=EL(5)+1
        EL(15)=0
C
        J=J+1
        WRITE(5,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,15)
40      CONTINUE
C
        WRITE(5,1)'C WEDGE IN STEP'
        N5=1
        ETYP=37210
        TYP=4
        DO 50 I=1,ISHOR/2
        EL(1)=2*(3*ISHOR+1)+1+(I-1)*2
        EL(2)=EL(1)+ISHOR
        EL(3)=3*(3*ISHOR+1)+1+(I-1)*2
        EL(4)=2*(3*ISHOR+1)+1+MOD(I*2,ISHOR)
        EL(5)=EL(4)+ISHOR
        EL(6)=3*(3*ISHOR+1)+1+MOD(I*2,ISHOR)
        EL(7)=0
        EL(8)=0
        EL(9)=0
        EL(10)=EL(1)+1
        EL(11)=EL(2)+1
        EL(12)=EL(3)+1
        EL(13)=0
        EL(14)=0
        EL(15)=0
C
        J=J+1
        WRITE(5,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,15)
50      CONTINUE
C
        WRITE(5,1)'C THIRD OUTSIDE'
        N5=5
        ETYP=37110
        TYP=1
        DO 60 I=1,ISHOR/2
        EL(1)=1+(I-1)*2+3*(3*ISHOR+1)
        EL(2)=1+MOD(I*2,ISHOR)+3*(3*ISHOR+1)
        EL(3)=3*ISHOR+1+EL(1)
        EL(4)=3*ISHOR+1+EL(2)
        EL(5)=ISHOR+EL(1)
        EL(6)=ISHOR+EL(2)
        EL(7)=ISHOR+EL(3)
        EL(8)=ISHOR+EL(4)
        EL(9)=EL(1)+1
        EL(10)=0
        EL(11)=0
        EL(12)=EL(3)+1
        EL(13)=0
        EL(14)=0
        EL(15)=0
        EL(16)=0
        EL(17)=EL(5)+1
        EL(18)=0
```

```
        EL(19)=0
        EL(20)=EL(7)+1
C
        J=J+1
        WRITE(5,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
60      CONTINUE
C
        WRITE(5,1)'C THIRD INSIDE, 1ST'
        ETYP=37110
        TYP=1
        DO 65 I=1,ISHOR/2
        EL(1)=1+(I-1)*2+ISHOR+3*(3*ISHOR+1)
        EL(2)=1+MOD(I*2,ISHOR)+ISHOR+3*(3*ISHOR+1)
        EL(3)=3*ISHOR+1+EL(1)
        EL(4)=3*ISHOR+1+EL(2)
        EL(5)=ISHOR+EL(1)
        EL(6)=ISHOR+EL(2)
        EL(7)=ISHOR+EL(3)
        EL(8)=ISHOR+EL(4)
        EL(9)=EL(1)+1
        EL(10)=0
        EL(11)=0
        EL(12)=EL(3)+1
        EL(13)=0
        EL(14)=0
        EL(15)=0
        EL(16)=0
        EL(17)=EL(5)+1
        EL(18)=0
        EL(19)=0
        EL(20)=EL(7)+1
C
        J=J+1
        WRITE(5,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
65      CONTINUE
C
        WRITE(5,1)'C THIRD INSIDE, 2ND'
        ETYP=37210
        TYP=4
        DO 70 I=1,ISHOR/2
        EL(1)=4*(3*ISHOR+1)
        EL(2)=2*ISHOR+1+(I-1)*2+3*(3*ISHOR+1)
        EL(3)=2*ISHOR+1+MOD(I*2,ISHOR)+3*(3*ISHOR+1)
        EL(4)=EL(1)+3*ISHOR+1
        EL(5)=EL(2)+3*ISHOR+1
        EL(6)=EL(3)+3*ISHOR+1
        EL(7)=0
        EL(8)=EL(2)+1
        EL(9)=0
        EL(10)=0
        EL(11)=0
        EL(12)=0
        EL(13)=0
        EL(14)=EL(5)+1
        EL(15)=0
C
        J=J+1
        WRITE(5,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,15)
70      CONTINUE
C
C
        WRITE(5,1)'MESH'
```

```
      WRITE(5,1)'REFE SPAC.LIST'
      WRITE(5,1)'1 1'
      WRITE(5,1)'2 2'
      WRITE(5,1)'3 3'
      WRITE(5,1)'4 4'
      WRITE(5,1)'5 5'
      WRITE(5,1)'6 6'
      WRITE(5,1)'7 7'
      WRITE(5,1)'8 8'
      WRITE(5,1)'C '
      WRITE(5,1)'MATERIAL'
      WRITE(5,1)'MATE.NUMB E NU'
      WRITE(5,1)'20 110000 0.35'
C
C
      RETURN
      END
C
C END PABLO2
C
      SUBROUTINE RES2(ISHOR,NN2)
      INTEGER I,ISHOR,NN2
C
C THIS SUBROUTINE IMPOSES CONSTRAINTS ON THE MODEL OF STEP
C WITH 2/3 OF TRANSITION RADIUS
C
      WRITE(5,1)'RESTRAINTS'
      WRITE(5,1)'NODE.NUMB DIRECTION'
1     FORMAT(80A)
C
      DO 10 I=NN2-3*ISHOR,NN2
       WRITE(5,2)I,0
2      FORMAT(I6,1X,I6)
10    CONTINUE
C
      RETURN
      END
C
C END RES2
C
      SUBROUTINE LOA2(XT,YT,NST,XP,ISHOR,NPPS)
      INTEGER NST,ISHOR,NPPS,I
      REAL D1,D2,F1,F2,XP,X,Y,PI
      DIMENSION XT(73,30),YT(73,30)
C
C THIS SUBROUTINE CALCULATES LOAD FOR STEP WITH
C 2/3 OF TRANSITION RADIUS
C
      PI=4.0*ATAN(1.0)
      X=XT(1,NST)
      Y=YT(1,NST)
      D1=SQRT(X**2+Y**2)
C
      X=XT((NPPS-1)/2+1,NST)
      Y=YT((NPPS-1)/2+1,NST)
      D2=SQRT(X**2+Y**2)
C
C
      F1=-0.5/D1
      F2=0.5/D2
      IF(XP.LE.0.0)THEN
       F1=-F1
```

```
       F2=-F2
      END IF
C
      WRITE(5,1)'LOADS'
      WRITE(5,1)'NODE.NUMB DIRE.OF.LOAD VALU.OF.LOAD'
1     FORMAT(80A)
C
      WRITE(5,2)1,3,F1
      I=(NPPS-1)/2+1
      WRITE(5,2)I,3,F2
C
2     FORMAT(I6,1X,I6,1X,E15.7)
C
      WRITE(5,1)'END.OF.DATA'
      RETURN
      END
C
C END LOA2
C
      SUBROUTINE GEOM3(XT,YT,NAME,NST,RAD,ISHOR,NPPS,NN3)
      CHARACTER NAME*10
      INTEGER I,J,K,NST,ISHOR,NPPS,INT,NN3
      REAL PI,RAD,AL,X,Y,Z
      DIMENSION XT(73,30),YT(73,30)
C
C THIS SUBROUTINE DEFINES NODES FOR STEP WITH WHOLE
C TRANSITION RADIUS
C
      PI=4.0*ATAN(1.0)
      INT=(NPPS-1)/ISHOR
      WRITE(6,1)'TITLE COSF3 - STEP   PROT ',NAME
      WRITE(6,3)'C  STEP NUMBER ',NST
3     FORMAT(A14,I6)
      WRITE(6,1)'CONTROL'
      WRITE(6,1)'FULL.CONTROL'
      WRITE(6,1)'PHASE=1,2,4,6,7,9'
      WRITE(6,1)'CLEAR.FILES'
      WRITE(6,1)'CONTROL.END'
C
      WRITE(6,1)'NODES'
      WRITE(6,1)'NODE.NUMB  X   Y   Z'
1     FORMAT(80A)
C
      AL=XT(1,NST)/SQRT(XT(1,NST)**2+YT(1,NST)**2)
      IF(AL.GE.1.0)THEN
       AL=0.0
      ELSE
       AL=ACOS(AL)
      END IF
C
      J=0
      WRITE(6,1)'C C FIRST CONTOUR, EDGE'
      DO 10 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       WRITE(6,2)J,XT(K,NST),YT(K,NST),0.0
2      FORMAT(I5,1X,E15.7,1X,E15.7,1X,E15.7)
10    CONTINUE
C
      WRITE(6,1)'C FIRST CONTOUR , 1ST INSIDE'
      DO 15 I=1,ISHOR
        J=J+1
```

```
       K=1+(I-1)*INT
       X=XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2
       Y=YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2
       WRITE(6,2)J,X,Y,0.0
15     CONTINUE
C
       WRITE(6,1)'C FIRST CONTOUR , 2ND INSIDE'
       DO 20 I=1,ISHOR
         J=J+1
       K=1+(I-1)*INT
       X=(XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2)/2
       Y=(YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2)/2
       WRITE(6,2)J,X,Y,0.0
20     CONTINUE
       WRITE(6,1)'C MIDPOINT'
       J=J+1
       WRITE(6,2)J,0.0,0.0,0.0
C
C
       WRITE(6,1)'C SECOND CONTOUR, EDGE'
       DO 30 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       WRITE(6,2)J,XT(K,NST),YT(K,NST),8.0
30     CONTINUE
C
       WRITE(6,1)'C SECOND CONTOUR, 1ST INSIDE'
       DO 35 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2
       Y=YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2
       WRITE(6,2)J,X,Y,8.0
35     CONTINUE
C
       WRITE(6,1)'C SECOND CONTOUR, 2ND INSIDE'
       DO 40 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=(XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2)/2
       Y=(YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2)/2
       WRITE(6,2)J,X,Y,8.0
40     CONTINUE
       WRITE(6,1)'C SECOND CONTOUR, MIDPOINT'
       J=J+1
       WRITE(6,2)J,0.0,0.0,8.0
C
C
       WRITE(6,1)'C THIRD CONTOUR, STEP'
C
       DO 50 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)+RAD*(1-COS(PI/6))*COS((I-1)*PI/12+AL)
       Y=YT(K,NST)+RAD*(1-COS(PI/6))*SIN((I-1)*PI/12+AL)
       Z=8.0+RAD*SIN(PI/6)
       WRITE(6,2)J,X,Y,Z
50     CONTINUE
C
       WRITE(6,1)'C THIRD CONTOUR, 1ST INSIDE'
       DO 55 I=1,ISHOR
       J=J+1
```

```
       K=1+(I-1)*INT
       X=XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2
       Y=YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2
       Z=8.Ø+RAD*SIN(PI/3)
       WRITE(6,2)J,X,Y,Z
55     CONTINUE
C
       WRITE(6,1)'C THIRD CONTOUR, 2ND INSIDE'
       DO 6Ø I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=(XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2)/2
       Y=(YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2)/2
       Z=8.Ø+RAD*SIN(PI/3)
       WRITE(6,2)J,X,Y,Z
6Ø     CONTINUE
       WRITE(6,1)'C THIRD CONTOUR, MIDPOINT'
       J=J+1
       Z=8.Ø+RAD*SIN(PI/3)
       WRITE(6,2)J,Ø.Ø,Ø.Ø,Z
C
       WRITE(6,1)'C FOURTH CONTOUR, STEP'
       DO 7Ø I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)+RAD*(1-COS(PI/3))*COS((I-1)*PI/12+AL)
       Y=YT(K,NST)+RAD*(1-COS(PI/3))*SIN((I-1)*PI/12+AL)
       Z=8.Ø+RAD*SIN(PI/3)
       WRITE(6,2)J,X,Y,Z
7Ø     CONTINUE
C
       WRITE(6,1)'C FOURTH CONTOUR, 1ST INSIDE'
       DO 75 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       Z=8.Ø+RAD*3/2
       WRITE(6,2)J,XT(K,NST),YT(K,NST),Z
75     CONTINUE
C
       WRITE(6,1)'C FOURTH CONTOUR, 2ND INSIDE'
       DO 78 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2
       Y=YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2
       Z=8.Ø+RAD*3/2
       WRITE(6,2)J,X,Y,Z
78     CONTINUE
C
       WRITE(6,1)'C FOURTH CONTOUR, 3RD INSIDE'
       DO 8Ø I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=(XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2)/2
       Y=(YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2)/2
       Z=8.Ø+RAD*3/2
       WRITE(6,2)J,X,Y,Z
8Ø     CONTINUE
C
       WRITE(6,1)'C FOURTH CONTOUR, MIDPOINT'
       J=J+1
       Z=8.Ø+RAD*3/2
```

```
      WRITE(6,2)J,0.0,0.0,Z
C
      WRITE(6,1)'C FOURTH CONTOUR, EXTRA'
      DO 85 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)+RAD*COS((I-1)*PI/12+AL)
       Y=YT(K,NST)+RAD*SIN((I-1)*PI/12+AL)
       Z=8.0+RAD
       WRITE(6,2)J,X,Y,Z
85    CONTINUE
C
      WRITE(6,1)'C FIFTH CONTOUR, EDGE'
      DO 90 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)+RAD*COS((I-1)*PI/12+AL)
       Y=YT(K,NST)+RAD*SIN((I-1)*PI/12+AL)
       Z=8.0+RAD*3/2
       WRITE(6,2)J,X,Y,Z
90    CONTINUE
C
      WRITE(6,1)'C FITH CONTOUR, 1ST INSIDE'
      DO 92 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       Z=8.0+RAD*3/2
       WRITE(6,2)J,XT(K,NST),YT(K,NST),Z
92    CONTINUE
C
      WRITE(6,1)'C FIFTH CONTOUR, 2ND INSIDE'
      DO 95 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2
       Y=YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2
       Z=8.0+RAD*3/2
       WRITE(6,2)J,X,Y,Z
95    CONTINUE
C
      WRITE(6,1)'C FIFTH CONTOUR, 3RD INSIDE'
      DO 100 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=(XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2)/2
       Y=(YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2)/2
       Z=8.0+RAD*3/2
       WRITE(6,2)J,X,Y,Z
100   CONTINUE
      WRITE(6,1)'C FITH CONTOUR, MIDPOINT'
      J=J+1
      Z=8.0+RAD*3/2
      WRITE(6,2)J,0.0,0.0,Z
C
      WRITE(6,1)'C SIXTH CONTOUR, EDGE'
      DO 110 I=1,ISHOR
       J=J+1
       K=1+(I-1)*INT
       X=XT(K,NST)+RAD*COS((I-1)*PI/12+AL)
       Y=YT(K,NST)+RAD*SIN((I-1)*PI/12+AL)
       Z=16.0+RAD*3/2
       WRITE(6,2)J,X,Y,Z
```

```
110     CONTINUE
C
       WRITE(6,1)'C SIXTH CONTOUR, 1ST INSIDE'
       DO 112 I=1,ISHOR
        J=J+1
        K=1+(I-1)*INT
        Z=16.0+RAD*3/2
        WRITE(6,2)J,XT(K,NST),YT(K,NST),Z
112     CONTINUE
C
       WRITE(6,1)'C SIXTH CONTOUR, 2ND INSIDE'
       DO 115 I=1,ISHOR
        J=J+1
        K=1+(I-1)*INT
        X=XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2
        Y=YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2
        Z=16.0+RAD*3/2
        WRITE(6,2)J,X,Y,Z
115     CONTINUE
C
       WRITE(6,1)'C SIXTH CONTOUR, 3RD INSIDE'
       DO 120 I=1,ISHOR
        J=J+1
        K=1+(I-1)*INT
        X=(XT(K,NST)-RAD*COS((I-1)*PI/12+AL)/2)/2
        Y=(YT(K,NST)-RAD*SIN((I-1)*PI/12+AL)/2)/2
        Z=16.0+RAD*3/2
        WRITE(6,2)J,X,Y,Z
120    CONTINUE
       WRITE(6,1)'C FITH CONTOUR, MIDPOINT'
       J=J+1
       Z=16.0+RAD*3/2
       WRITE(6,2)J,0.0,0.0,Z
C
       NN3=J
C
       RETURN
       END
C
C END GEOM3
C
C
       SUBROUTINE PABLO3(ISHOR)
       INTEGER I,J,K,ISHOR
       INTEGER N1,N2,N5,EL,TYP,PROP
       INTEGER*4 ETYP
       DIMENSION EL(20)
C
C THIS SUBROUTINE DEFINES PAFBLOCKS (ELEMENTS) FOR STEP WITH
C WHOLE TRANSITION RADIUS
C
       WRITE(6,1)'PAFBLOCCKS'
       WRITE(6,1)'BLOCK.NUMB TYPE ELEM.TYPE PROP N1 N2 N5 TOPO'
1      FORMAT(80A)
C
       WRITE(6,1)'C FIRST OUTSIDE'
       J=0
       N1=1
       N2=1
       N5=5
       ETYP=37110
       TYP=1
```

```
      PROP=20
      DO 10 I=1,ISHOR/2
       EL(1)=1+(I-1)*2
       EL(2)=1+MOD(I*2,ISHOR)
       EL(3)=3*ISHOR+1+EL(1)
       EL(4)=3*ISHOR+1+EL(2)
       EL(5)=ISHOR+EL(1)
       EL(6)=ISHOR+EL(2)
       EL(7)=ISHOR+EL(3)
       EL(8)=ISHOR+EL(4)
       EL(9)=EL(1)+1
       EL(10)=0
       EL(11)=0
       EL(12)=EL(3)+1
       EL(13)=0
       EL(14)=0
       EL(15)=0
       EL(16)=0
       EL(17)=EL(5)+1
       EL(18)=0
       EL(19)=0
       EL(20)=EL(7)+1
C
       J=J+1
       WRITE(6,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
2      FORMAT(7(I6,1X)/'* ',10(I5,1X)/'* ',10(I5,1X))
10    CONTINUE
C
      WRITE(6,1)'C FIRST INSIDE, 1ST'
      ETYP=37110
      TYP=1
      PROP=20
      DO 15 I=1,ISHOR/2
       EL(1)=1+(I-1)*2+ISHOR
       EL(2)=1+MOD(I*2,ISHOR)+ISHOR
       EL(3)=3*ISHOR+1+EL(1)
       EL(4)=3*ISHOR+1+EL(2)
       EL(5)=ISHOR+EL(1)
       EL(6)=ISHOR+EL(2)
       EL(7)=ISHOR+EL(3)
       EL(8)=ISHOR+EL(4)
       EL(9)=EL(1)+1
       EL(10)=0
       EL(11)=0
       EL(12)=EL(3)+1
       EL(13)=0
       EL(14)=0
       EL(15)=0
       EL(16)=0
       EL(17)=EL(5)+1
       EL(18)=0
       EL(19)=0
       EL(20)=EL(7)+1
C
       J=J+1
       WRITE(6,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
15    CONTINUE
C
      WRITE(6,1)'C FIRST INSIDE, 2ND'
      ETYP=37210
      TYP=4
      DO 20 I=1,ISHOR/2
```

```
        EL(1)=3*ISHOR+1
        EL(2)=2*ISHOR+1+(I-1)*2
        EL(3)=2*ISHOR+1+MOD(I*2,ISHOR)
        EL(4)=EL(1)+3*ISHOR+1
        EL(5)=EL(2)+3*ISHOR+1
        EL(6)=EL(3)+3*ISHOR+1
        EL(7)=0
        EL(8)=EL(2)+1
        EL(9)=0
        EL(10)=0
        EL(11)=0
        EL(12)=0
        EL(13)=0
        EL(14)=EL(5)+1
        EL(15)=0
C
        J=J+1
        WRITE(6,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,15)
20    CONTINUE
C
C
      WRITE(6,1)'C SECOND OUTSIDE'
      N5=1
      ETYP=37110
      TYP=1
      DO 30 I=1,ISHOR/2
       EL(1)=1+(I-1)*2+3*ISHOR+1
       EL(2)=1+MOD(I*2,ISHOR)+3*ISHOR+1
       EL(3)=3*ISHOR+1+EL(1)
       EL(4)=3*ISHOR+1+EL(2)
       EL(5)=ISHOR+EL(1)
       EL(6)=ISHOR+EL(2)
       EL(7)=ISHOR+EL(3)
       EL(8)=ISHOR+EL(4)
       EL(9)=EL(1)+1
       EL(10)=0
       EL(11)=0
       EL(12)=EL(3)+1
       EL(13)=0
       EL(14)=0
       EL(15)=0
       EL(16)=0
       EL(17)=EL(5)+1
       EL(18)=0
       EL(19)=0
       EL(20)=EL(7)+1
C
       J=J+1
       WRITE(6,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
30    CONTINUE
C
      WRITE(6,1)'C SECOND INSIDE, 1ST'
      ETYP=37110
      TYP=1
      DO 35 I=1,ISHOR/2
       EL(1)=1+(I-1)*2+ISHOR+3*ISHOR+1
       EL(2)=1+MOD(I*2,ISHOR)+ISHOR+3*ISHOR+1
       EL(3)=3*ISHOR+1+EL(1)
       EL(4)=3*ISHOR+1+EL(2)
       EL(5)=ISHOR+EL(1)
       EL(6)=ISHOR+EL(2)
       EL(7)=ISHOR+EL(3)
```

```
        EL(8)=ISHOR+EL(4)
        EL(9)=EL(1)+1
        EL(10)=0
        EL(11)=0
        EL(12)=EL(3)+1
        EL(13)=0
        EL(14)=0
        EL(15)=0
        EL(16)=0
        EL(17)=EL(5)+1
        EL(18)=0
        EL(19)=0
        EL(20)=EL(7)+1
C
        J=J+1
        WRITE(6,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
35      CONTINUE
C
        WRITE(6,1)'C SECOND INSIDE, 2ND'
        ETYP=37210
        TYP=4
        DO 40 I=1,ISHOR/2
        EL(1)=2*(3*ISHOR+1)
        EL(2)=2*ISHOR+1+(I-1)*2+3*ISHOR+1
        EL(3)=2*ISHOR+1+MOD(I*2,ISHOR)+3*ISHOR+1
        EL(4)=EL(1)+3*ISHOR+1
        EL(5)=EL(2)+3*ISHOR+1
        EL(6)=EL(3)+3*ISHOR+1
        EL(7)=0
        EL(8)=EL(2)+1
        EL(9)=0
        EL(10)=0
        EL(11)=0
        EL(12)=0
        EL(13)=0
        EL(14)=EL(5)+1
        EL(15)=0
C
        J=J+1
        WRITE(6,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,15)
40      CONTINUE
C
        WRITE(6,1)'C THIRD OUTSIDE'
        N5=1
        ETYP=37110
        TYP=1
        DO 60 I=1,ISHOR/2
        EL(1)=1+(I-1)*2+2*(3*ISHOR+1)
        EL(2)=1+MOD(I*2,ISHOR)+2*(3*ISHOR+1)
        EL(3)=3*ISHOR+1+EL(1)
        EL(4)=3*ISHOR+1+EL(2)
        EL(5)=ISHOR+EL(1)
        EL(6)=ISHOR+EL(2)
        EL(7)=ISHOR+EL(3)
        EL(8)=ISHOR+EL(4)
        EL(9)=EL(1)+1
        EL(10)=0
        EL(11)=0
        EL(12)=EL(3)+1
        EL(13)=0
        EL(14)=0
        EL(15)=0
```

```
      EL(16)=0
      EL(17)=EL(5)+1
      EL(18)=0
      EL(19)=0
      EL(20)=EL(7)+1
C
      J=J+1
      WRITE(6,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
60    CONTINUE
C
      WRITE(6,1)'C THIRD INSIDE WEDGE'
      ETYP=37210
      TYP=4
      DO 62 I=1,ISHOR/2
      EL(1)=2*(3*ISHOR+1)+1+(I-1)*2+ISHOR
      EL(2)=3*(3*ISHOR+1)+1+(I-1)*2+2*ISHOR
      EL(3)=3*(3*ISHOR+1)+1+(I-1)*2+ISHOR
      EL(4)=2*(3*ISHOR+1)+1+MOD(I*2,ISHOR)+ISHOR
      EL(5)=3*(3*ISHOR+1)+1+MOD(I*2,ISHOR)+2*ISHOR
      EL(6)=3*(3*ISHOR+1)+1+MOD(I*2,ISHOR)+ISHOR
      EL(7)=0
      EL(8)=0
      EL(9)=0
      EL(10)=EL(1)+1
      EL(11)=EL(2)+1
      EL(12)=EL(3)+1
C
      J=J+1
      WRITE(6,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,15)
62    CONTINUE
C
      WRITE(6,1)'C THIRD INSIDE, 1ST'
      ETYP=37110
      TYP=1
      DO 65 I=1,ISHOR/2
      EL(1)=1+(I-1)*2+ISHOR+2*(3*ISHOR+1)
      EL(2)=1+MOD(I*2,ISHOR)+ISHOR+2*(3*ISHOR+1)
      EL(3)=4*ISHOR+1+EL(1)
      EL(4)=4*ISHOR+1+EL(2)
      EL(5)=ISHOR+EL(1)
      EL(6)=ISHOR+EL(2)
      EL(7)=ISHOR+EL(3)
      EL(8)=ISHOR+EL(4)
      EL(9)=EL(1)+1
      EL(10)=0
      EL(11)=0
      EL(12)=EL(3)+1
      EL(13)=0
      EL(14)=0
      EL(15)=0
      EL(16)=0
      EL(17)=EL(5)+1
      EL(18)=0
      EL(19)=0
      EL(20)=EL(7)+1
C
      J=J+1
      WRITE(6,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
65    CONTINUE
C
      WRITE(6,1)'C THIRD INSIDE, 2ND'
      ETYP=37210
```

```
      TYP=4
      DO 70 I=1,ISHOR/2
       EL(1)=3*(3*ISHOR+1)
       EL(2)=2*ISHOR+1+(I-1)*2+2*(3*ISHOR+1)
       EL(3)=2*ISHOR+1+MOD(I*2,ISHOR)+2*(3*ISHOR+1)
       EL(4)=EL(1)+4*ISHOR+1
       EL(5)=EL(2)+4*ISHOR+1
       EL(6)=EL(3)+4*ISHOR+1
       EL(7)=0
       EL(8)=EL(2)+1
       EL(9)=0
       EL(10)=0
       EL(11)=0
       EL(12)=0
       EL(13)=0
       EL(14)=EL(5)+1
       EL(15)=0
C
       J=J+1
       WRITE(6,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,15)
70    CONTINUE
C
      WRITE(6,1)'C EXTRA IN STEP'
      ETYP=37110
      TYP=1
      DO 80 I=1,ISHOR/2
       EL(1)=3*(3*ISHOR+1)+4*ISHOR+1+1+(I-1)*2
       EL(2)=3*(3*ISHOR+1)+4*ISHOR+1+1+MOD(I*2,ISHOR)
       EL(3)=EL(1)+ISHOR
       EL(4)=EL(2)+ISHOR
       EL(5)=3*(3*ISHOR+1)+1+(I-1)*2
       EL(6)=3*(3*ISHOR+1)+1+MOD(I*2,ISHOR)
       EL(7)=EL(5)+ISHOR
       EL(8)=EL(6)+ISHOR
       EL(9)=EL(1)+1
       EL(10)=0
       EL(11)=0
       EL(12)=EL(3)+1
       EL(13)=0
       EL(14)=0
       EL(15)=0
       EL(16)=0
       EL(17)=EL(5)+1
       EL(18)=0
       EL(19)=0
       EL(20)=EL(7)+1
C
       J=J+1
       WRITE(6,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
80    CONTINUE
C
      WRITE(6,1)'C FOURTH OUTSIDE'
      N5=5
      ETYP=37110
      TYP=1
      DO 90 I=1,ISHOR/2
       EL(1)=3*(3*ISHOR+1)+5*ISHOR+1+1+(I-1)*2
       EL(2)=3*(3*ISHOR+1)+5*ISHOR+1+1+MOD(I*2,ISHOR)
       EL(3)=4*ISHOR+1+EL(1)
       EL(4)=4*ISHOR+1+EL(2)
       EL(5)=ISHOR+EL(1)
       EL(6)=ISHOR+EL(2)
```

```
      EL(7)=ISHOR+EL(3)
      EL(8)=ISHOR+EL(4)
      EL(9)=EL(1)+1
      EL(10)=0
      EL(11)=0
      EL(12)=EL(3)+1
      EL(13)=0
      EL(14)=0
      EL(15)=0
      EL(16)=0
      EL(17)=EL(5)+1
      EL(18)=0
      EL(19)=0
      EL(20)=EL(7)+1
C
      J=J+1
      WRITE(6,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
90    CONTINUE
C
      WRITE(6,1)'C FOURTH 1ST INSIDE'
      ETYP=37110
      TYP=1
      DO 100 I=1,ISHOR/2
      EL(1)=3*(3*ISHOR+1)+6*ISHOR+1+1+(I-1)*2
      EL(2)=3*(3*ISHOR+1)+6*ISHOR+1+1+MOD(I*2,ISHOR)
      EL(3)=4*ISHOR+1+EL(1)
      EL(4)=4*ISHOR+1+EL(2)
      EL(5)=ISHOR+EL(1)
      EL(6)=ISHOR+EL(2)
      EL(7)=ISHOR+EL(3)
      EL(8)=ISHOR+EL(4)
      EL(9)=EL(1)+1
      EL(10)=0
      EL(11)=0
      EL(12)=EL(3)+1
      EL(13)=0
      EL(14)=0
      EL(15)=0
      EL(16)=0
      EL(17)=EL(5)+1
      EL(18)=0
      EL(19)=0
      EL(20)=EL(7)+1
C
      J=J+1
      WRITE(6,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
100   CONTINUE
C
      WRITE(6,1)'C FOURTH 2ND INSIDE'
      ETYP=37110
      TYP=1
      DO 110 I=1,ISHOR/2
      EL(1)=3*(3*ISHOR+1)+7*ISHOR+1+1+(I-1)*2
      EL(2)=3*(3*ISHOR+1)+7*ISHOR+1+1+MOD(I*2,ISHOR)
      EL(3)=4*ISHOR+1+EL(1)
      EL(4)=4*ISHOR+1+EL(2)
      EL(5)=ISHOR+EL(1)
      EL(6)=ISHOR+EL(2)
      EL(7)=ISHOR+EL(3)
      EL(8)=ISHOR+EL(4)
      EL(9)=EL(1)+1
      EL(10)=0
```

```
        EL(11)=0
        EL(12)=EL(3)+1
        EL(13)=0
        EL(14)=0
        EL(15)=0
        EL(16)=0
        EL(17)=EL(5)+1
        EL(18)=0
        EL(19)=0
        EL(20)=EL(7)+1
C
        J=J+1
        WRITE(6,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,20)
110     CONTINUE
C
      WRITE(6,1)'C FOURTH, 3RD INSIDE'
      ETYP=37210
      TYP=4
      DO 120 I=1,ISHOR/2
       EL(1)=3*(3*ISHOR+1)+5*ISHOR+1+4*ISHOR+1
       EL(2)=3*(3*ISHOR+1)+8*ISHOR+1+1+(I-1)*2
       EL(3)=3*(3*ISHOR+1)+8*ISHOR+1+1+MOD(I*2,ISHOR)
       EL(4)=EL(1)+4*ISHOR+1
       EL(5)=EL(2)+4*ISHOR+1
       EL(6)=EL(3)+4*ISHOR+1
       EL(7)=0
       EL(8)=EL(2)+1
       EL(9)=0
       EL(10)=0
       EL(11)=0
       EL(12)=0
       EL(13)=0
       EL(14)=EL(5)+1
       EL(15)=0
C
       J=J+1
       WRITE(6,2)J,TYP,ETYP,PROP,N1,N2,N5,(EL(K),K=1,15)
120     CONTINUE
C
C
      WRITE(6,1)'MESH'
      WRITE(6,1)'REFE SPAC.LIST'
      WRITE(6,1)'1 1'
      WRITE(6,1)'2 2'
      WRITE(6,1)'3 3'
      WRITE(6,1)'4 4'
      WRITE(6,1)'5 5'
      WRITE(6,1)'6 6'
      WRITE(6,1)'7 7'
      WRITE(6,1)'8 8'
      WRITE(6,1)'C '
      WRITE(6,1)'MATERIAL'
      WRITE(6,1)'MATE.NUMB E NU'
      WRITE(6,1)'20 110000 0.35'
C
C
      RETURN
      END
C
C END PABLO3
C
      SUBROUTINE RES3(ISHOR,NN3)
```

```
      INTEGER I,ISHOR,NN3
C
C THIS SUBROUTINE IMPOSES CONSTRAINTS ON THE MODEL OF STEP WITH
C WHOLE TRANSITION RADIUS
C
      WRITE(6,1)'RESTRAINTS'
      WRITE(6,1)'NODE.NUMB DIRECTION'
1     FORMAT(80A)
C
      DO 10 I=NN3-4*ISHOR,NN3
       WRITE(6,2)I,0
2      FORMAT(I6,1X,I6)
10    CONTINUE
C
      RETURN
      END
C
C END RES3
C
      SUBROUTINE LOA3(XT,YT,NST,XP,ISHOR,NPPS)
      INTEGER NST,ISHOR,NPPS,I
      REAL D1,D2,F1,F2,XP,X,Y,PI
      DIMENSION XT(73,30),YT(73,30)
C
C THIS SUBROUTINE CALCULATES LOADS FOR STEP WITH WHOLE
C TRANSITION RADIUS
C
      PI=4.0*ATAN(1.0)
      X=XT(1,NST)
      Y=YT(1,NST)
      D1=SQRT(X**2+Y**2)
C
      X=XT((NPPS-1)/2+1,NST)
      Y=YT((NPPS-1)/2+1,NST)
      D2=SQRT(X**2+Y**2)
C
C
      F1=-0.5/D1
      F2=0.5/D2
      IF(XP.LE.0.0)THEN
       F1=-F1
       F2=-F2
      END IF
C
      WRITE(6,1)'LOADS'
      WRITE(6,1)'NODE.NUMB DIRE.OF.LOAD VALU.OF.LOAD'
1     FORMAT(80A)
C
      WRITE(6,2)1,3,F1
      I=(NPPS-1)/2+1
      WRITE(6,2)I,3,F2
C
2     FORMAT(I6,1X,I6,1X,E15.7)
C
      WRITE(6,1)'END.OF.DATA'
      RETURN
      END
C
C END LOA3
C
```

APPENDIX 3. Simple model. Program PROCO

Program PROCO

```
C     PROGRAM FOR THE CALCULATION OF BENDING STRESSES IN PROSTHESES
C     USING SIMPLE BENDING THEORY
C
C
      PROGRAM PROCO
      CHARACTER NAME*10
      INTEGER STEP,L1,L2,CCD,UEB,NPPS,NSCH
      INTEGER ICHO,LEVEL
      INTEGER M,I
      REAL M1,M2,INAX,INAY,INAXY,XP,YP,ZP,ZPN,XPN
      REAL DELC,BETA,P,RAD,D1,D2,PI,DUM,TORQ
      REAL ES,EC,DB
      DIMENSION NSCN(5),SCF(8,30,2)
      DIMENSION X(73,30),Y(73,30),Z(73,30),XO(73,30),YO(73,30),
     +XT(73,30),YT(73,30)
      DIMENSION XPL(30),YPL(30),ANG(30),CMID(30,3)
      DIMENSION INAX(30),INAY(30),INAXY(30),A(30,3),XBAR(30),YBAR(30)
      DIMENSION M1(30),M2(30)
      DIMENSION SSUM(73,30),FSUM(73,30)
      DIMENSION SMAX(30),SMIN(30),IMAX(30),MMAX(30),IMIN(30),MMIN(30)
     +,SFMAX(30),SFMIN(30),IFMIN(30),MFMIN(30),IFMAX(30),MFMAX(30)
      DIMENSION TMAX(30,3)
C
C
C READ DATA FROM THE GEOMETRY INPUT FILE AND FILE 'SCFDAT'
C WITH STRESS CONCENTRATION FACTORS
C
      CALL READAT(X,Y,Z,NSCN,XP,YP,ZP,STEP,L1,L2,UEB,CCD,NSCH,
     +NPPS,NAME,SCF)
C
C
      OPEN(9,FILE='STRESS')
3333  CONTINUE
C
C
C BETA: TILT ANGEL OF THE PROSTHESIS 10 DEGREES
C
      PI=4*ATAN(1.0)
      PRINT*,'TILTANGLE OF PROSTHESIS IN DEG ? '
      READ(*,*)BETA
      BETA=BETA*PI/180
C
C P: LOAD OF THE PROSTHESIS IN NEWTONS
C
      PRINT*,'ABSOLUTE VALUE OF THE LOAD IN [N] ?'
      READ(*,*)P
C
C RAD: VALUE OF THE TRANSITION RADIUS
C
      PRINT*,'VALUE OF THE TRANSITION RADIUS [MM] ?'
      READ(*,*)RAD
C
C
C LEVEL: NUMBER OF THE CONTOUR AT EMBEDDING LEVEL
```

```
C
      PRINT*,' NUMBER OF CONTOUR AT THE EMBEDDING LEVEL ?'
      READ(*,*)LEVEL
      IF(LEVEL.GT.NSCN(4)) THEN
       LEVEL=NSCN(4)
      END IF
      IF(LEVEL.LT.0) THEN
       LEVEL=0
      END IF
C
C
      PRINT*,'ESTEM = ?     '
      READ(*,*)ES
      PRINT*,'ECEMENT = ?    '
      READ(*,*)EC
      PRINT*,'DIAMETER OF EMBEDDING = ?    '
      READ(*,*)DB
C
C DEFINE NEW POINT COORDINATES AROUND CENTRE POINTS
C
      CALL SECT(X,Y,XO,YO,Z,NPPS,NSCN(4),CMID,ANG,XPL,YPL,XP,YP)
C
C CALCULATE COORDINATES OF CENTROIDS (POINTS OF GRAVITY) AND
C SECOND MOMENTS OF AREA ABOUT XX AND YY
C
      CALL CENTRO(NSCN(4),X,Y,NPPS,XBAR,YBAR,INAX,INAY,INAXY,XT,YT,
     +XO,YO,A)
C
C
C CALCULATE APPLIED MOMENTS M1,M2 FOR PROSTHESIS WITHOUT EMBEDDING
C
      CALL MPROT(NSCN(4),XPL,YPL,ZP,P,XBAR,YBAR,Z,BETA,ANG,
     +M1,M2,INAX,INAY,INAXY)
C
C CALCULATE MOMENTS FOR EMBEDDED PROSTHESIS
C
      CALL MPROEM(NSCN(4),LEVEL,XPL(LEVEL),P,XBAR(LEVEL),
     +Z,BETA,ANG(LEVEL),M1,M2,INAX,INAY,X,Y,ES,EC,DB)
C
C
C CALCULATE STRESSES DUE TO MOMENTS AND SHEAR FORCE
C
      CALL COSTRE(NSCN(4),ANG,LEVEL,BETA,INAX,
     +INAY,M1,M2,SSUM,A,P,XT,YT)
C
C CALCULATE STRESS CONCENTRATION FACTOR AND MULTIPLY STRESSES
C WITH THESE
C
      CALL CONCTR(NSCN(4),XBAR,YBAR,ANG,CMID,XO,YO,SSUM,FSUM,SCF,
     +RAD)
C
C FIND MINIMUM AND MAXIMUM STRESSES AND FACTORED STRESSES ALONG
C EACH CONTOUR
C
      CALL SSORT(NSCN(4),NPPS,SSUM,FSUM,IMIN,MMIN,IMAX,MMAX,SMAX,
     +SMIN,IFMIN,MFMIN,IFMAX,MFMAX,SFMAX,SFMIN)
C
C
      WRITE(9,1000)' ANALYSIS OF PROSTHESIS   ',NAME
1000  FORMAT(80A)
C
      PRINT*,' ANALYSIS OF ',NAME
```

```
C
C
      WRITE(9,1100)
      WRITE(*,1100)
1100  FORMAT(//'      MAXIMUM AND MINIMUM STRESSES          '/)
      WRITE(9,1200)
      WRITE(*,1200)
1200  FORMAT(/'            STEP POINT MAXSTRESS '/)
C
      DO 10 M=1,NSCN(4)-1
       WRITE(9,1300)  MMAX(M),IMAX(M),SMAX(M)
       WRITE(*,1300) MMAX(M),IMAX(M),SMAX(M)
1300  FORMAT(8X,2I5,F10.3)
10    CONTINUE
      WRITE(9,1400)
      WRITE(*,1400)
1400  FORMAT(/'            STEP POINT MINSTRESS '/)
      DO 20 M=1,NSCN(4)-1
       WRITE(9,1300)  MMIN(M),IMIN(M),SMIN(M)
       WRITE(*,1300) MMIN(M),IMIN(M),SMIN(M)
20    CONTINUE
C
      WRITE(9,1500)
      WRITE(*,1500)
1500  FORMAT(//'      MAXIMUM AND MINIMUM FACTORED STRESSES  '/)
      WRITE(9,1200)
      WRITE(*,1200)
C
      DO 30 M=1,NSCN(4)-1
       WRITE(9,1300)  MFMAX(M),IFMAX(M),SFMAX(M)
       WRITE(*,1300) MFMAX(M),IFMAX(M),SFMAX(M)
30    CONTINUE
      WRITE(9,1400)
      WRITE(*,1400)
      DO 40 M=1,NSCN(4)-1
       WRITE(9,1300)  MFMIN(M),IFMIN(M),SFMIN(M)
       WRITE(*,1300) MFMIN(M),IFMIN(M),SFMIN(M)
40    CONTINUE
C
C
      PRINT*,'DEFAULTS FOR THIS RUN WERE:'
      PRINT*,'TILTANGLE    ',BETA*45/ATAN(1.0)
      PRINT*,'LOAD         ',P
      PRINT*,'RADIUS       ',RAD
      PRINT*,'EMB. LEVEL   ',LEVEL
C
      WRITE(9,1000)' '
      WRITE(9,1000)'  LOAD       RADIUS      TILTANGLE      EMB.LEV'
      WRITE(9,1000)' '
      WRITE(9,1700)P,RAD,BETA*45/ATAN(1.0),LEVEL
1700  FORMAT(F10.4,4X,F10.4,4X,F10.4,4X,I3)
C
C
C TORSION COMPUTATION
C
      DO 56 M=1,NSCN(4)-1
       CALL TORO(X,Y,M,TMAX)
C
C CALCULATE TORSION MOMENTS
C
      TORQ=P*SIN(BETA)*(YP-CMID(M,2))
      TMAX(M,2)=TMAX(M,2)*TORQ
```

```
C
56    CONTINUE
C
      WRITE(9,1800)
      WRITE(*,1800)
1800  FORMAT(//'            STEP  POINT  TOR.STRESS              '/)
      DO 57 M=1,NSCN(4)-1
       WRITE(9,1900)M,TMAX(M,1),TMAX(M,2)
       WRITE(*,1900)M,TMAX(M,1),TMAX(M,2)
1900   FORMAT(4X,I5,4X,F10.3,4X,F10.3)
C
57    CONTINUE
C
C
C REDESIGN
C
C
C
      PRINT*,'REDESIGN OPTIONS'
      PRINT*,'  1        INCREASE CCD ANGLE'
      PRINT*,'  2        MAKE CONE SHORTER'
      PRINT*,'  3        NEW PARAMETERS'
      PRINT*,'  4        STOP'
      PRINT*,'ENTER OPTION NUMBER'
      READ(*,*)ICHO
      IF(ICHO.EQ.1)THEN
       PRINT*,'ENTER INCREASE IN CCD-ANGLE IN DEGREES'
       READ(*,*)DELC
       DELC=DELC*PI/180
       ZPN=ABS(XP)*TAN((CCD-90)*PI/180)
       DUM=ZP-ZPN
       IF(XP.GT.0.0)THEN
        XPN=XP*COS(DELC)-ZPN*SIN(DELC)
        ZP=XP*SIN(DELC)+ZPN*COS(DELC)
        ZP=ZP+DUM
       ELSE
        XPN=XP*COS(DELC)+ZPN*SIN(DELC)
        ZP=-XP*SIN(DELC)+ZPN*COS(DELC)
        ZP=ZP+DUM
       END IF
       XP=XPN
        DO 50 I=1,NSCN(4)
         XPL(I)=XP-CMID(I,1)
         YPL(I)=YP-CMID(I,2)
         D1=YPL(I)*COS(ANG(I))-XPL(I)*SIN(ANG(I))
         D2=YPL(I)*SIN(ANG(I))+XPL(I)*COS(ANG(I))
         XPL(I)=D2
         YPL(I)=D1
50      CONTINUE
       GOTO 3333
      END IF
      IF(ICHO.EQ.2)THEN
       PRINT*,'ENTER CHANGE IN THE CONE LENGHT IN MM'
       READ(*,*)DELC
       ZPN=ZP-DELC*SIN((180-CCD)*PI/180)
       IF(XP.GT.0.0)THEN
        XPN=XP-DELC*COS((180-CCD)*PI/180)
       ELSE
        XPN=XP+DELC*COS((180-CCD)*PI/180)
       END IF
       XP=XPN
       ZP=ZPN
```

```
          DO 60 I=1,NSCN(4)
           XPL(I)=XP-CMID(I,1)
           YPL(I)=YP-CMID(I,2)
           D1=YPL(I)*COS(ANG(I))-XPL(I)*SIN(ANG(I))
           D2=YPL(I)*SIN(ANG(I))+XPL(I)*COS(ANG(I))
           XPL(I)=D2
           YPL(I)=D1
60        CONTINUE
         GOTO 3333
        END IF
C
        IF(ICHO.EQ.3)THEN
         GOTO 3333
        END IF
C
C
        STOP
        END
C
C END MAIN PROGRAM
C
C
C
        SUBROUTINE READAT(X,Y,Z,NSCN,XP,YP,ZP,STEP,L1,L2,UEB,CCD,NSCH,
       +NPPS,NAME,SCF)
        CHARACTER NAME*10
        INTEGER STEP,NSCH,NPPS,CCD,L1,L2,UEB,I,J,M
        REAL XP,YP,ZP
        DIMENSION X(73,30),Y(73,30),Z(73,30)
        DIMENSION NSCN(5),SCF(8,30,2)
C
        PRINT*,'NAME OF THE INPUT FILE'
        READ(*,55)NAME
55      FORMAT(A10)
        OPEN(2,FILE=NAME)
        READ(2,*)STEP
        DO 100 I=1,5
         READ(2,*) NSCN(I)
100     CONTINUE
        READ(2,*) NSCH
        READ(2,*) NPPS
        READ(2,*) CCD
        READ(2,*) L1
        READ(2,*) L2
        READ(2,*) UEB
        DO 110 I=1,NSCH
         DO 120 J=1,NPPS
          READ(2,*)X(J,I)
          READ(2,*)Y(J,I)
          READ(2,*)Z(J,I)
120      CONTINUE
110     CONTINUE
C
        READ(2,*)XP
        READ(2,*)YP
        READ(2,*)ZP
        PRINT*,'CENTRE OF THE HEAD'
        PRINT*,XP,YP,ZP
C
        CLOSE(2)
C
C       READ DATA FROM SCFDAT: STRESS CONCENTRATION FACTORS
```

```
C
      OPEN(3,FILE='SCFDAT')
      DO 300 I=1,8
       DO 310 J=1,30
        READ(3,*)(SCF(I,J,M),M=1,2)
310    CONTINUE
300   CONTINUE
C
      CLOSE(3)
C
      RETURN
      END
C
C END READAT
C
C
C
      SUBROUTINE SECT(X,Y,XO,YO,Z,NPPS,NSCN4,CMID,ANG,XPL,YPL,XP,
     +YP)
      INTEGER I,J,K,NSCN4
      REAL PI,X3,X1,X2,Y3,Y1,Y2,A1,A2,B1,B2,D1,D2,W,XP,YP
      DIMENSION X(73,30),Y(73,30),Z(73,30),SEC(0:72,3),CMID(30,3)
      DIMENSION XO(73,30),YO(73,30)
      DIMENSION WEIG(73),ANG(30),XPL(30),YPL(30)
C
C
      DO 100 I=1,NSCN4
       DO 110 J=1,NPPS
         SEC(J-1,1)=X(J,I)
         SEC(J-1,2)=Y(J,I)
         SEC(J-1,3)=Z(J,I)
         XO(J,I)=X(J,I)
         YO(J,I)=Y(J,I)
110    CONTINUE
C COMPUTING WEIGHTS
       W=0.0
       DO 130 J=0,NPPS-1
        D1=SQRT((SEC(J,1)-SEC(MOD(J+1,NPPS-1),1))**2+(SEC(J,2)-
     +  SEC(MOD(J+1,NPPS-1),2))**2)
        D2=SQRT((SEC(J,1)-SEC(MOD(J+NPPS-2,NPPS-1),1))**2+(SEC(J,2)-
     +  SEC(MOD(J+NPPS-2,NPPS-1),2))**2)
        WEIG(J+1)=(D1+D2)/2
        W=W+WEIG(J+1)
130    CONTINUE
C COMPUTE THE MIDPOINTS
       DO 140 K=1,3
        CMID(I,K)=0.0
140    CONTINUE
       DO 145 J=1,NPPS
        CMID(I,1)=WEIG(J)*SEC(J-1,1)+CMID(I,1)
        CMID(I,2)=WEIG(J)*SEC(J-1,2)+CMID(I,2)
145    CONTINUE
       CMID(I,1)=CMID(I,1)/W
       CMID(I,2)=CMID(I,2)/W
       CMID(I,3)=SEC(1,3)
C
C FIND NEW POINTS ON THE CONTOUR IN 5 DEGREE STEPS
C AROUND CMIDS STARTING WITH THE POINT NO. 1
       PI=4*ATAN(1.0)
        SEC(0,1)=X(1,I)
        SEC(0,2)=Y(1,I)
       K=1
```

```
        DO 210 J=0,NPPS-2
         X1=SEC(J,1)-CMID(I,1)
         Y1=SEC(J,2)-CMID(I,2)
         D1=SQRT(X1**2+Y1**2)
99       X2=X(MOD(K,NPPS)+1,I)-CMID(I,1)
         Y2=Y(MOD(K,NPPS)+1,I)-CMID(I,2)
         D2=SQRT(X2**2+Y2**2)
        IF(((X1*X2+Y1*Y2)/(D1*D2)).GT.(COS(PI/36)))THEN
          K=K+1
          GOTO 99
         END IF
         X3=X1*COS(PI/36)-Y1*SIN(PI/36)
         Y3=X1*SIN(PI/36)+Y1*COS(PI/36)
         X3=X3+CMID(I,1)
         Y3=Y3+CMID(I,2)
         IF((ABS(X3-CMID(I,1)).GT.1E-12))THEN
          A1=(Y3-CMID(I,2))/(X3-CMID(I,1))
          B1=Y3-A1*X3
          IF((ABS(X(MOD(K,NPPS)+1,I)-SEC(J,1)).GT.1E-12))THEN
      A2=(Y(MOD(K,NPPS)+1,I)-SEC(J,2))/(X(MOD(K,NPPS)+1,I)
     +     -SEC(J,1))
          B2=Y(MOD(K,NPPS)+1,I)-A2*X(MOD(K,NPPS)+1,I)
         SEC(J+1,1)=(B2-B1)/(A1-A2)
         SEC(J+1,2)=A1*SEC(J+1,1)+B1
         ELSE
          A2=0.0
          B2=Y(MOD(K,NPPS)+1,I)
          SEC(J+1,1)=(B2-B1)/A1
          SEC(J+1,2)=B2
         END IF
        ELSE
         A1=0.0
         B1=Y3
         IF((ABS(X(MOD(K,NPPS)+1,I)-SEC(J,1)).GT.1E-12))THEN
      A2=(Y(MOD(K,NPPS)+1,I)-SEC(J,2))/(X(MOD(K,NPPS)+1,I)
     +     -SEC(J,1))
          B2=Y(MOD(K,NPPS)+1,I)-A2*X(MOD(K,NPPS)+1,I)
          SEC(J+1,1)=(B1-B2)/A2
          SEC(J+1,2)=B1
         ELSE
          SEC(J+1,1)=X(MOD(K,NPPS)+1,I)
          SEC(J+1,2)=Y(MOD(K,NPPS)+1,I)
         END IF
        END IF
210     CONTINUE
C WRITE NEW POINTS
        DO 220 J=1,NPPS
          X(J,I)=SEC(J-1,1)-CMID(I,1)
          Y(J,I)=SEC(J-1,2)-CMID(I,2)
          XO(J,I)=XO(J,I)-CMID(I,1)
          YO(J,I)=YO(J,I)-CMID(I,2)
220     CONTINUE
C
        XPL(I)=XP-CMID(I,1)
        YPL(I)=YP-CMID(I,2)
C
C ROTATE SO THAT THE FIRST POINT IS ON Y=0
C
C CALCULATE ANGLE OF ROTATION ANG
C
        D1=X(1,I)/(SQRT(X(1,I)**2+Y(1,I)**2))
        D2=Y(1,I)/(SQRT(X(1,I)**2+Y(1,I)**2))
```

```
        IF(D1.GE.0.)THEN
         IF(D2.GE.0.)THEN
          ANG(I)=ACOS(D1)
         ELSE
          ANG(I)=-ABS(ACOS(D1))
         END IF
        ELSE
         IF(D2.GE.0)THEN
          ANG(I)=ABS(ACOS(D1))
         ELSE
          ANG(I)=-ABS(ACOS(D1))
         END IF
        END IF
C
C ROTATE BY ANG
C
        DO 230 J=1,NPPS
         D1=Y(J,I)*COS(ANG(I))-X(J,I)*SIN(ANG(I))
         D2=Y(J,I)*SIN(ANG(I))+X(J,I)*COS(ANG(I))
         X(J,I)=D2
         Y(J,I)=D1
         D1=YO(J,I)*COS(ANG(I))-XO(J,I)*SIN(ANG(I))
         D2=YO(J,I)*SIN(ANG(I))+XO(J,I)*COS(ANG(I))
         XO(J,I)=D2
         YO(J,I)=D1
230     CONTINUE
C
C NEW COORDINATES OF THE CENTR OF THE HEAD: XPL, YPL
C
        D1=YPL(I)*COS(ANG(I))-XPL(I)*SIN(ANG(I))
        D2=YPL(I)*SIN(ANG(I))+XPL(I)*COS(ANG(I))
        XPL(I)=D2
        YPL(I)=D1
C
100    CONTINUE
       RETURN
       END
C
C END SECT
C
C
C
       SUBROUTINE CENTRO(NSCN4,X,Y,NPPS,XBAR,YBAR,INAX,INAY,INAXY,XT,
      +YT,XO,YO,A)
       INTEGER NSCN4,NPPS
       INTEGER I,II,J,M
       REAL DELX,B,YMIN,YMAX,H1,H2,X1,X2,Y1,Y2,A1,A2,IX1,IX2,IY1,IY2
       REAL IXY1,IXY2,IXX,IYY,IXYXY,INAX,INAY,INAXY
       DIMENSION X(73,30),Y(73,30),XBAR(30),YBAR(30),INAX(30),INAY(30)
      +,INAXY(30),XT(73,30),YT(73,30),XO(73,30),YO(73,30)
       DIMENSION A(30,3),AX(30,3),AY(30,3),IXX(30,3),IYY(30,3),
      +IXYXY(30,3),XMAX(4,30),IPOINT(4,30)
C
C      FIND MAXIMUM X FOR EACH QUADRANT
C
C
        DO 10 M=1,NSCN4
         I=0
         DO 11 II=1,4
          XMAX(II,M)=0.
          DO 12 J=1,18
           I=I+1
```

```
          IF(ABS(X(I,M)).GT.ABS(XMAX(II,M))) THEN
          XMAX(II,M)=X(I,M)
          IPOINT(II,M)=I
          END IF
12       CONTINUE
11      CONTINUE
10    CONTINUE
C
C LOOP ON NUMBER OF STEPS
C
      DO 20 M=1,NSCN4
C
C LOOP ON NUMBER OF POINTS AROUND PERIPHERY OF STEP
C
C CALCULATION OF STRIP PROPERTIES
C
       DO 21 I=1,NPPS-1
        DELX=ABS(X(I+1,M)-X(I,M))
        B=DELX
C
C 'NEGATIVE' AREAS OF CROSS-SECTION
C
        IF(I.LT.19) THEN
         IF((X(I,M).LT.XMAX(1,M)).AND.(I.LT.IPOINT(1,M))) THEN
          B=-DELX
         END IF
        END IF
        IF((I.GT.19).AND.(I.LT.37)) THEN
         IF((ABS(X(I,M)).LT.ABS(XMAX(2,M))).AND.(I.GT.IPOINT(2,M))) THEN
          B=-DELX
         END IF
        END IF
        IF((I.GT.37).AND.(I.LT.55)) THEN
         IF((ABS(X(I,M)).LT.ABS(XMAX(3,M))).AND.(I.LT.IPOINT(3,M))) THEN
          B=-DELX
         END IF
        END IF
        IF(I.GT.55) THEN
         IF((X(I,M).LT.XMAX(4,M)).AND.(I.GT.IPOINT(4,M))) THEN
          B=-DELX
         END IF
        END IF
        IF((ABS(Y(I,M))).LT.ABS(Y(I+1,M))) THEN
         YMIN=Y(I,M)
         YMAX=Y(I+1,M)
        ELSE
         YMIN=Y(I+1,M)
         YMAX=Y(I,M)
        END IF
        H1=ABS(YMIN)
        H2=ABS(YMAX)
        X1=(X(I,M)+X(I+1,M))/2.
        Y1=YMIN/2.
        Y2=YMAX/2.
        A1=B*H1
        A2=B*H2
        X2=X1
        IX1=(B*H1**3)/3.
        IX2=(B*H2**3)/3.
        IY1=(H1*B**3)/12. + A1*X1**2
        IY2=(H2*B**3)/12. + A2*X2**2
        IXY1=A1*X1*Y1
```

```
        IXY2=A2*X2*Y2
C
C CALCULATION OF STEP PROPERTIES
C
        A(M,1)=A(M,1)+A1
        A(M,2)=A(M,2)+A2
        AX(M,1)=AX(M,1)+A1*X1
        AX(M,2)=AX(M,2)+A2*X2
        AY(M,1)=AY(M,1)+A1*Y1
        AY(M,2)=AY(M,2)+A2*Y2
        IXX(M,1)=IXX(M,1)+IX1
        IXX(M,2)=IXX(M,2)+IX2
        IYY(M,1)=IYY(M,1)+IY1
        IYY(M,2)=IYY(M,2)+IY2
        IXYXY(M,1)=IXYXY(M,1)+IXY1
        IXYXY(M,2)=IXYXY(M,2)+IXY2
21      CONTINUE
C
C AVERAGE MAX AND MIN PROPERTIES
C
        A(M,3)=(A(M,1)+A(M,2))/2.
        AX(M,3)=(AX(M,1)+AX(M,2))/2.
        AY(M,3)=(AY(M,1)+AY(M,2))/2.
        IXX(M,3)=(IXX(M,1)+IXX(M,2))/2.
        IYY(M,3)=(IYY(M,1)+IYY(M,2))/2.
        IXYXY(M,3)=(IXYXY(M,1)+IXYXY(M,2))/2.
C
C CENTROID POSITION
C
        XBAR(M)=AX(M,3)/A(M,3)
        YBAR(M)=AY(M,3)/A(M,3)
C
C NOMINAL MOMENTS OF AREA
C
        INAX(M)=IXX(M,3)-A(M,3)*YBAR(M)**2
        INAY(M)=IYY(M,3)-A(M,3)*XBAR(M)**2
        INAXY(M)=IXYXY(M,3)-A(M,3)*XBAR(M)*YBAR(M)
20      CONTINUE
C
C POINT COORDINATES ABOUT PRINCIPAL AXES
C
       DO 30 M=1,NSCN4
        DO 31 I=1,NPPS
         XT(I,M)=XO(I,M)-XBAR(M)
         YT(I,M)=YO(I,M)-YBAR(M)
31      CONTINUE
30     CONTINUE
C
       RETURN
       END
C
C END CENTRO
C
C
C
       SUBROUTINE MPROT(NSCN4,XPL,YPL,ZP,P,XBAR,YBAR,Z,BETA,
      +ANG,M1,M2,INAX,INAY,INAXY)
       INTEGER NSCN4,M
       REAL ZP,P,BETA,M1,M2,INAX,INAY,INAXY
       REAL MX,MY,XL,YL,ZL
       DIMENSION XPL(30),YPL(30),XBAR(30),YBAR(30),Z(73,30),
      +ANG(30),M1(30),M2(30),INAX(30),INAY(30),INAXY(30)
```

```
C
      DO 10 M=1,NSCN4
       XL=XPL(M)-XBAR(M)
       YL=YPL(M)-YBAR(M)
       ZL=ZP-Z(1,M)
C
       IF(XL.LT.0.0)THEN
        XL=XL*COS(BETA)+ZL*SIN(BETA)*COS(ANG(M))
        YL=-YL*COS(BETA)+ZL*SIN(BETA)*SIN(ANG(M))
       ELSE
        XL=XL*COS(BETA)-ZL*SIN(BETA)*COS(ANG(M))
        YL=-YL*COS(BETA)-ZL*SIN(BETA)*SIN(ANG(M))
       END IF
C
       MX=P*YL
       MY=P*XL
C
       M1(M)=(MX+INAXY(M)*MY/INAY(M))/(1-INAXY(M)**2/INAX(M)/INAY(M))
       M2(M)=(MY+INAXY(M)*MX/INAX(M))/(1-INAXY(M)**2/INAX(M)/INAY(M))
10    CONTINUE
C
      RETURN
      END
C
C END MPROT
C
C
      SUBROUTINE MPROEM(NSCN4,LEVEL,XPLLEV,P,
     +XBARLV,Z,BETA,ANGLEV,M1,M2,INAX,INAY,X,Y,ES,EC,DB)
      INTEGER NSCN4,LEVEL,M
      REAL XPLLEV,P,XBARLV,BETA,M1,M2,ES,EC,DB
      REAL INAX,INAY,ANGLEV
      REAL SX,SY,XL,YL,ZL
      DIMENSION Z(73,30),X(73,30),Y(73,30)
      DIMENSION M1(30),M2(30),INAX(30),INAY(30),INAXY(30)
C
C     FINITE DIFFERENCE METHOD FOR BEAMS ON EL. FOUNDATOIN
C
      IF(LEVEL.GT.0)THEN
       XL=XPLLEV-XBARLV
       DO 10 M=1,LEVEL
C
        IF(XL.LT.0.0)THEN
          SX=P*SIN(BETA)*COS(ANGLEV)
          SY=P*SIN(BETA)*SIN(ANGLEV)
        ELSE
          SX=-P*SIN(BETA)*COS(ANGLEV)
          SY=-P*SIN(BETA)*SIN(ANGLEV)
        END IF
        CALL ELBEAM(X,Y,Z,LEVEL,INAX,INAY,M1,M2,SX,SY,ES,EC,DB)
10     CONTINUE
      END IF
C
      RETURN
      END
C
C END MPROEM
C
C
      SUBROUTINE COSTRE(NSCN4,ANG,LEVEL,BETA,
     +INAX,INAY,M1,M2,SSUM,A,P,XT,YT)
      INTEGER NSCN4,LEVEL,M,I
```

```
      REAL P,BETA,M1,M2,INAX,INAY
      DIMENSION ANG(30),INAX(30),INAY(30),M1(30),M2(30)
      DIMENSION SSUM(73,30),A(30,3),XT(73,30),YT(73,30)
      DIMENSION S1(73,30),S2(73,30),SL(30)
C
C     STRESSES DUE TO M1 AND M2 SEPARATELY
C
      DO 10 M=1,NSCN4
       DO 20 I=1,72
        IF(M.GE.LEVEL)THEN
         S1(I,M)=M1(M)*YT(I,M)/INAX(M)
         S2(I,M)=-M2(M)*XT(I,M)/INAY(M)
         SL(M)=-P/A(M,3)*COS(BETA)
         SSUM(I,M)=S1(I,M)+S2(I,M)+SL(M)
        ELSE
         S1(I,M)=M1(M)*YT(I,M)/INAX(M)
         S2(I,M)=-M2(M)*XT(I,M)/INAY(M)
         SL(M)=0.0
         SSUM(I,M)=S1(I,M)+S2(I,M)+SL(M)
        END IF
20     CONTINUE
10    CONTINUE
      RETURN
      END
C
C END COSTRE
C
C
C
      SUBROUTINE CONCTR(NSCN4,XBAR,YBAR,ANG,CMID,XO,YO,
     +SSUM,FSUM,SCF,RAD)
      INTEGER NSCN4
      INTEGER M,I,MSTEP,IPNT
      REAL XCEN,YCEN,RAT1,RAT2,DMIN,DMAX,K,RAD
      DIMENSION XBAR(30),YBAR(30),ANG(30),CMID(30,3),SCF(8,30,2)
      DIMENSION XO(73,30),YO(73,30),SSUM(73,30),FSUM(73,30)
      DIMENSION C1(73,30),C2(73,30),K(73,30)
C
      DO 10 M=1,NSCN4-1
       XCEN=XBAR(M)*COS(ANG(M))-YBAR(M)*SIN(ANG(M))
       YCEN=XBAR(M)*SIN(ANG(M))+YBAR(M)*COS(ANG(M))
       XCEN=XCEN+CMID(M,1)
       YCEN=YCEN+CMID(M,2)
       DO 20 I=1,72
      C1(I,M+1)=XO(I,M+1)*COS(ANG(M+1))-YO(I,M+1)*SIN(ANG(M+1))
      C2(I,M+1)=XO(I,M+1)*SIN(ANG(M+1))+YO(I,M+1)*COS(ANG(M+1))
      C1(I,M)=XO(I,M)*COS(ANG(M))-YO(I,M)*SIN(ANG(M))
      C2(I,M)=XO(I,M)*SIN(ANG(M))+YO(I,M)*COS(ANG(M))
      C1(I,M+1)=C1(I,M+1)+CMID(M+1,1)
      C2(I,M+1)=C2(I,M+1)+CMID(M+1,2)
      C1(I,M)=C1(I,M)+CMID(M,1)
      C2(I,M)=C2(I,M)+CMID(M,2)
C
        DMAX=2.0*SQRT((C1(I,M+1)-XCEN)**2+(C2(I,M+1)-YCEN)**2)
        DMIN=2.0*SQRT((C1(I,M)-XCEN)**2+(C2(I,M)-YCEN)**2)
        RAT1=DMAX/DMIN
        RAT2=RAD/DMIN
C
        IPNT=I
        MSTEP=M
        CALL SCFINT(RAT1,RAT2,IPNT,MSTEP,K,SCF)
        FSUM(I,M)=K(I,M)*SSUM(I,M)
```

```
20      CONTINUE
10     CONTINUE
       RETURN
       END
C
C END CONCTR
C
C
       SUBROUTINE SSORT(NSCN4,NPPS,SSUM,FSUM,IMIN,MMIN,IMAX,MMAX,SMAX,
      +SMIN,IFMIN,MFMIN,IFMAX,MFMAX,SFMAX,SFMIN)
       INTEGER NSCN4,NPPS,M,I
       REAL LOW,HIGH
       DIMENSION SSUM(73,30),FSUM(73,30)
       DIMENSION IMIN(30),MMIN(30),IMAX(30),MMAX(30),SMAX(30),SMIN(30)
       DIMENSION IFMIN(30),MFMIN(30),IFMAX(30),MFMAX(30),SFMAX(30),
      +SFMIN(30)
C
C     MAXIMUM AND MINIMUM STRESSES
C
       HIGH=0.
       LOW=0.
       DO 11 M=1,NSCN4-1
        IMIN(M)=1
        MMIN(M)=M
        IMAX(M)=1
        MMAX(M)=M
        SMAX(M)=SSUM(1,M)
        SMIN(M)=SSUM(1,M)
        DO 12 I=1,NPPS-1
         IF(SSUM(I,M).LT.SMIN(M)) THEN
          SMIN(M)=SSUM(I,M)
          IMIN(M)=I
         END IF
         IF(SSUM(I,M).GT.SMAX(M)) THEN
          SMAX(M)=SSUM(I,M)
          IMAX(M)=I
         END IF
12      CONTINUE
        IF(SMAX(M).GT.HIGH) THEN
         HIGH=SMAX(M)
        END IF
        IF(SMIN(M).LT.LOW) THEN
         LOW=SMIN(M)
        END IF
11     CONTINUE
C
C      MAXIMUM AND MINIMUM FACTORED STRESSES
C
       HIGH=0.
       LOW=0.
       DO 15 M=1,NSCN4-1
        IFMIN(M)=1
        IFMAX(M)=1
        MFMIN(M)=M
        MFMAX(M)=M
        SFMAX(M)=FSUM(1,M)
        SFMIN(M)=FSUM(1,M)
        DO 16 I=1,NPPS-1
         IF(FSUM(I,M).LT.SFMIN(M)) THEN
          SFMIN(M)=FSUM(I,M)
          IFMIN(M)=I
         END IF
```

```
        IF(FSUM(I,M).GT.SFMAX(M)) THEN
         SFMAX(M)=FSUM(I,M)
         IFMAX(M)=I
        END IF
16     CONTINUE
       IF(SFMAX(M).GT.HIGH) THEN
        HIGH=SFMAX(M)
       END IF
       IF(SFMIN(M).LT.LOW) THEN
        LOW=SFMIN(M)
       END IF
15     CONTINUE
C
      RETURN
      END
C
C END SSORT
C
C
C
      SUBROUTINE TORO(X,Y,M,TMAX)
      INTEGER M,FAULT
      DIMENSION X(73,30),Y(73,30),TMAX(30,2)
C
C FINITE ELEMENT STRESS ANALYSIS
C TORSION OF A BAR WITH NONCIRCULAR CROSS SECTION
C ADOPTED FROM
C 'ENGINEERING STRESS ANALYSIS'
C BY D.N.FENNER (ELLIS HORWOOD 1987)
C
      CALL SPEC(X,Y,M)
      CALL ACTIVE
      CALL BWIDTH
      CALL ASSEM
      CALL SOLVE(FAULT)
      IF(FAULT.GT.0) GOTO 10
      CALL STRESS
      CALL OUTPUT(M,TMAX)
10    RETURN
      END
C
C
C
      SUBROUTINE SPEC(X,Y,M)
C
C PROBLEM SPECIFICATION
C
      INTEGER I,M,K
      DIMENSION X(73,30),Y(73,30)
C
      COMMON/ELEM/IETYPE,NOE,IJ(100,3),MAT(100),NODN,NONE,NODE,NOST
      COMMON/NODE/NONS,XY(100,2),NOCO
      COMMON/EQNS/NODS,NOEQ,IBW,IDA(100,6),SS(200,50),FS(200)
      COMMON/DBCS/NOPD,IPD(50,2),PD(50)
      COMMON/MATS/NOPROP,NOSET,EDAT(5,5)
C IETYPE: ELEMENT TYPE
C NOE: NUMBER OF ELEMENTS IN THE SYSTEM
C IJ(IE,NE): NODAL POINT NE FOR ELEMENT IE
C MAT(IE): MATERIAL SET NUMBER FOR ELEMENT IE
C NODN: NUMBER OF NODAL DEGREES OF FREEDOM
C NONE: NUMBER OF NODES PER ELEMENT
C NODE: NUMBER OF ELEMENT DEGREES OF FREEDOM
```

```
C NOST: NUMBER OF ELEMENT STRESS COMPONENTS
C NONS: NUMBER OF NODES IN THE SYSTEM
C XY(NS,ICO): COORDINATE ICO FOR NODAL POINT NS
C NOCO: NUMBER OF COORDINATES DEFINING ELEMENT GEOMETRY
C NODS: NUMBER OF SYSTEM DEGREES OF FREEDOM
C NOEQ: NUMBER OF ACTIVE EQUATIONS
C IBW: SEMI-BANDWIDTH OF SYSTEM STIFFNESS MATRIX
C IDA(NS,IDN): ACTIVE EQATION NUMBER CORRESPONDING TO DEGREE OF
C              FREEDOM IDN OF NODE NS
C SS(IRE,ICE): SYSTEM STIFFNESS, AND ELEMENT STRESSES:
C              SS(IRS,ICS), SYSTEM STIFFNESS COEFFICIENT;
C              SS(IE,IST), STRESS COMPONENT IST FOR ELEMENT IE.
C FS(IDS): PRESCRIBED FORCE FOR SYSTEM DEGREE OF FREEDOM IDS
C NOPD: NUMBER OF PRESCRIBED DISPLACEMENTS
C IPD: POINTERS FOR DISPLACEMENT BOUNDARY CONDITION IBC:
C      IPD(IBC,1),NODE; IPD(IBC,2), DEGREE OF FREEDOM
C PD(IBC): PRESCRIBED DISPLACEMENT FOR BOUNDARY CONDITION IBC
C NOPROP: NUMBER OF PROPERTIES PER SET OF MATERIAL DATA
C NOSET: NUMBER OF SETS OF MATERIAL DATA
C EDAT(IPROP,ISET): MATERIAL PROPERTY IPROP FOR DATA SET ISET
C
      IETYPE=5
      NODN=1
      NONE=3
      NOCO=2
      NOST=2
      NOPROP=0
      NODE=NODN*NONE
C
C DEFINE NODES
C
      NONS=61
C
      DO 10 I=1,24
       XY(I,1)=X(1+(I-1)*3,M)
       XY(I,2)=Y(1+(I-1)*3,M)
10    CONTINUE
C
      DO 20 I=1,18
       XY(I+24,1)=X(1+(I-1)*4,M)*.75
       XY(I+24,2)=Y(1+(I-1)*4,M)*.75
20    CONTINUE
C
      DO 30 I=1,12
       XY(I+42,1)=X(1+(I-1)*6,M)*.5
       XY(I+42,2)=Y(1+(I-1)*6,M)*.5
30    CONTINUE
C
      DO 35 I=1,6
       XY(I+54,1)=X(1+(I-1)*12,M)*.25
       XY(I+54,2)=Y(1+(I-1)*12,M)*.25
35    CONTINUE
C
C
      XY(61,1)=0
      XY(61,2)=0
C
C DEFINE ELEMENTS
C
      NOE=96
C
      K=0
```

```
      DO 40 I=1,24
       IJ(I,1)=I
       IJ(I,2)=MOD(I,24)+1
       IJ(I,3)=24+I+K
       IF(MOD(I,4).EQ.0)THEN
        K=K-1
       END IF
       IF(I.EQ.24)THEN
        IJ(I,3)=25
       END IF
40    CONTINUE
C
      K=0
      DO 50 I=1,18
       IJ(24+I,1)=MOD(I+K,24)+1
       IF(MOD(I,3).EQ.0)THEN
        K=K+1
       END IF
       IJ(24+I,2)=MOD(I,18)+25
       IJ(24+I,3)=24+I
50    CONTINUE
C
      K=0
      DO 55 I=1,18
       IJ(42+I,1)=24+I
       IJ(42+I,2)=MOD(I,18)+25
       IJ(42+I,3)=42+I+K
       IF(MOD(I,3).EQ.0)THEN
        K=K-1
       END IF
       IF(I.EQ.18)THEN
        IJ(I+42,3)=43
       END IF
55    CONTINUE
C
      K=0
      DO 60 I=1,12
       IJ(I+60,1)=MOD(I+K,18)+25
       IF(MOD(I,2).EQ.0)THEN
        K=K+1
       END IF
       IJ(I+60,2)=MOD(I,12)+43
       IJ(I+60,3)=42+I
60    CONTINUE
C
      K=0
      DO 65 I=1,12
       IJ(I+72,1)=42+I
       IJ(I+72,2)=MOD(I,12)+43
       IJ(I+72,3)=54+I+K
       IF(MOD(I,2).EQ.0)THEN
        K=K-1
       END IF
       IF(I.EQ.12)THEN
        IJ(I+72,3)=55
       END IF
65    CONTINUE
C
      K=0
      DO 70 I=1,6
       IJ(I+84,1)=MOD(I+K,12)+43
       K=K+1
```

```
        IJ(I+84,2)=MOD(I,6)+55
        IJ(I+84,3)=54+I
70      CONTINUE
C
        DO 80 I=1,6
        IJ(90+I,1)=I+54
        IJ(90+I,2)=MOD(I,6)+55
        IJ(90+I,3)=61
80      CONTINUE
C
C
        NODS=NONS*NODN
C MATERIAL DATA
        NOSET=0
        DO 90 I=1,NOE
        MAT(I)=0
90      CONTINUE
C BOUNDARY CONDITIONS
        NOPD=1
        IPD(1,1)=61
        IPD(1,2)=1
        PD(1)=0
C
C
        RETURN
        END
C
C
C
        SUBROUTINE ACTIVE
C IDENTIFY ACTIVE EQUATIONS
        INTEGER NS,IDN,IBC
        COMMON/ELEM/IETYPE,NOE,IJ(100,3),MAT(100),NODN,NONE,NODE,NOST
        COMMON/NODE/NONS,XY(100,2),NOCO
        COMMON/EQNS/NODS,NOEQ,IBW,IDA(100,6),SS(200,50),FS(200)
        COMMON/DBCS/NOPD,IPD(50,2),PD(50)
C NS: SYSTEM NODE(=1,.....,NONS)
C IDN: NODAL DEGREE OF FREEDOM(=1,...,NODN)
C IBC: BOUNDARY CONDITION NUMBER
C
        DO 10 NS=1,NONS
        DO 15 IDN=1,NODN
        IDA(NS,IDN)=1
15      CONTINUE
10      CONTINUE
        DO 20 IBC=1,NOPD
        IF(ABS(PD(IBC)).GT.0.000001) GO TO 20
        NS=IPD(IBC,1)
        IDN=IPD(IBC,2)
        IDA(NS,IDN)=0
20      CONTINUE
        NOEQ=0
        DO 30 NS=1,NONS
        DO 30 IDN=1,NODN
        IF(IDA(NS,IDN).LT.0.000001) GOTO 30
        NOEQ=NOEQ+1
        IDA(NS,IDN)=NOEQ
30      CONTINUE
        RETURN
        END
C
C
```

```
C
      SUBROUTINE BWIDTH
C CALCULATE SEMI-BANDWIDTH
      INTEGER IE,NE,NS,IDN,IDE,IRE,IRS,ICE,ICS
      COMMON/ELEM/IETYPE,NOE,IJ(100,3),MAT(100),NODN,NONE,NODE,NOST
      COMMON/EQNS/NODS,NOEQ,IBW,IDA(100,6),SS(200,50),FS(200)
      DIMENSION IRC(6)
C IE: ELEMENT NUMBER(=1,...,NOE)
C NE: ELEMENT NODE (=1,...,NONE)
C IDE: ELEMENT DEGREE OF FREEDOM (=1,...,NODE)
C IRE: ROW POSITION IN ELEMENT STIFFNESS MATRIX
C IRS: ROW POSITION IN SYSTEM STIFFNESS MATRIX
C ICE: COLUMN POSITION IN ELEMENT STIFFNESS MATRIX
C ICS: COLUMN POSITION IN SYSTEM STIFFNESS MATRIX
C IRC(IDE): ROW OR COLUMN POSITION IN SYSTEM MATRIX EQUIVALENT
C           TO ROW OR COLUMN POSITION IDE IN ELEMENT STIFFNESS
C           MATRIX
C
      IBW=0
      DO 40 IE=1,NOE
       DO 10 NE=1,NONE
        NS=IJ(IE,NE)
        DO 15 IDN=1,NODN
         IDE=NODN*(NE-1)+IDN
         IRC(IDE)=IDA(NS,IDN)
15      CONTINUE
10     CONTINUE
       DO 30 IRE=1,NODE
        IRS=IRC(IRE)
        IF(IRS.LT.0.000001) GOTO 30
        DO 20 ICE=1,NODE
         ICS=IRC(ICE)-IRS+1
         IF(ICS.LT.1) GO TO 20
         IBW=MAX(IBW,ICS)
20      CONTINUE
30     CONTINUE
40    CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE ASSEM
C ASSEMBLE EQUILIBRIUM EQUATIONS
      COMMON/ELEM/IETYPE,NOE,IJ(100,3),MAT(100),NODN,NONE,NODE,NOST
      COMMON/EQNS/NODS,NOEQ,IBW,IDA(100,6),SS(200,50),FS(200)
      COMMON/DBCS/NOPD,IPD(50,2),PD(50)
      INTEGER IRS,ICS,IE,IDE,NE,NS,IDN,IRE,ICE,IBC,IDS
      DIMENSION IRC(6),SE(6,6),FE(6)
C IDS: SYSTEM DEGREE OF FREEDOM (=1,...,NODS)
C SE(IRE,ICE): STIFFNESS COEFFICIENT IN ROW IRE AND COLUMN ICE OF
C              ELEMENT STIFFNESS MATRIX
C FE(IDE): PRESCRIBED FORCE FOR ELEMENT DEGREE OF FREEDOM IDE
C
C ZERO ARRAYS
      DO 10 IRS=1,NOEQ
       FS(IRS)=0.
       DO 15 ICS=1,IBW
        SS(IRS,ICS)=0.
15     CONTINUE
10    CONTINUE
C SUM OVER ALL ELEMENTS IN SYSTEM
```

```
      DO 50 IE=1,NOE
       DO 16 IDE=1,NODE
        FE(IDE)=0.
16     CONTINUE
       CALL ELEMNT(IE,SE,FE)
       DO 20 NE=1,NONE
        NS=IJ(IE,NE)
        DO 25 IDN=1,NODN
         IDE=NODN*(NE-1)+IDN
         IRC(IDE)=IDA(NS,IDN)
25      CONTINUE
20     CONTINUE
       DO 40 IRE=1,NODE
        IRS=IRC(IRE)
        IF(IRS.EQ.0) GOTO 40
        FS(IRS)=FS(IRS)+FE(IRE)
        DO 30 ICE=1,NODE
         ICS=IRC(ICE)-IRS+1
         IF(ICS.LT.1) GOTO 30
         SS(IRS,ICS)=SS(IRS,ICS)+SE(IRE,ICE)
30      CONTINUE
40     CONTINUE
50    CONTINUE
C DISPLACEMENT BOUNDARY CONDITIONS
      IF(NOEQ.GT.(NODS-NOPD)) THEN
      DO 70 IBC=1,NOPD
       IF(ABS(PD(IBC)).LT.1E-12) GOTO 70
       NS=IPD(IBC,1)
       IDN=IPD(IBC,2)
       IDS=IDA(NS,IDN)
       SS(IDS,1)=SS(IDS,1)*1.E20
       FS(IDS)=PD(IBC)*SS(IDS,1)
70    CONTINUE
      ENDIF
      RETURN
      END
C
C
C
      SUBROUTINE SOLVE(FAULT)
C SOLVE SET OF BANDED SYMMETRIC EQUILIBRIUM EQUATIONS
C USING GAUSSIAN ELIMINATION
      COMMON/EQNS/NODS,NOEQ,IBW,IDA(100,6),SS(200,50),FS(200)
      INTEGER IEQ,I,IR,IC,FAULT
      REAL R,CN
      DIMENSION DS(200)
C IEQ: ACTIVE EQUATION NUMBER (=1,...,NOEQ)
C I: FORWARD REDUCTION STEP NUMBER
C IR: ROW POSITION IS SYSTEM STIFFNESS MATRIX
C IC: COLUMN POSITION IN SYSTEM STIFFNESS MATRIX
C R: ELIMINATION RATIO FOR FORWARD REDUCTION
C DS: ARRAY STORING DIAGONAL COEFFICIANTS OF SYSTEM MATRIX
C
      FAULT=0
C
C STORE DIAGONAL COEFFICIENTS
C
      DO 10 IEQ=1,NOEQ
       DS(IEQ)=SS(IEQ,1)
10    CONTINUE
C FORWARD REDUCTION
      DO 50 I=1,NOEQ-1
```

```
       DO 40 IR=I+1,I+IBW-1
        IF(ABS(SS(I,IR-I+1)).LT.1E-12) GO TO 40
        R=SS(I,IR-I+1)/SS(I,1)
        DO 20 IC=IR,I+IBW-1
         IF(IC.GT.NOEQ) GO TO 30
         IF(ABS(SS(I,IC-I+1)).LT.1E-12) GO TO 20
         SS(IR,IC-IR+1)=SS(IR,IC-IR+1)-R*SS(I,IC-I+1)
20      CONTINUE
30      IF(ABS(FS(I)).LT.1E-12) GO TO 40
        FS(IR)=FS(IR)-R*FS(I)
40     CONTINUE
       CN=SS(I+1,1)/DS(I+1)
       IF(CN.LT.1E-3) GOTO 80
50    CONTINUE
C BACK SUBSTITUTION
      FS(NOEQ)=FS(NOEQ)/SS(NOEQ,1)
      DO 70 IR=NOEQ-1,1,-1
       DO 60 IC=IR+1,IR+IBW-1
        IF(IC.GT.NOEQ) GO TO 70
      IF(ABS(SS(IR,IC-IR+1)).LT.1E-12.OR.ABS(FS(IC)).LT.1E-12) GOTO 60
        FS(IR)=FS(IR)-SS(IR,IC-IR+1)*FS(IC)
60     CONTINUE
       FS(IR)=FS(IR)/SS(IR,1)
70    CONTINUE
      CALL RENUM
      RETURN
80    WRITE(9,600)
600   FORMAT(//' ***** ILL-CONDITIONED EQUATIONS ***** ')
      FAULT=1
      RETURN
      END
C
C
C
      SUBROUTINE RENUM
C RENUMBER DISPLACEMENTS
      COMMON/ELEM/IETYPE,NOE,IJ(100,3),MAT(100),NODN,NONE,NODE,NOST
      COMMON/EQNS/NODS,NOEQ,IBW,IDA(100,6),SS(200,50),FS(200)
      COMMON/NODE/NONS,XY(100,2),NOCO
      INTEGER NS,IDN,IDS
C
      DO 10 NS=NONS,1,-1
       DO 15 IDN=NODN,1,-1
        IDS=NODN*(NS-1)+IDN
        IF(IDA(NS,IDN).EQ.0) THEN
         FS(IDS)=0.
        ELSE
         FS(IDS)=FS(IDA(NS,IDN))
        ENDIF
15     CONTINUE
10    CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE OUTPUT(M,TMAX)
      INTEGER M,IE,I
      REAL TAU
      DIMENSION TMAX(30,2),ARRAY(24)
      COMMON/EQNS/NODS,NOEQ,IBW,IDA(100,6),SS(200,50),FS(200)
C
```

```
C STRESSES IN SURFACE ELEMENTS DUE TO A TORQUE EQUAL TO 1
C
      DO 10 IE=1,24
        ARRAY(IE)=SQRT(SS(IE,1)**2+SS(IE,2)**2)
        ARRAY(IE)=ARRAY(IE)/SS(1,3)
10    CONTINUE
      TMAX(M,1)=1
      TMAX(M,2)=ARRAY(1)
      TAU=ARRAY(1)
      DO 20 I=1,23
       IF(ARRAY(I).GT.TAU)THEN
        TAU=ARRAY(I)
        TMAX(M,1)=(I-1)*3+1
        TMAX(M,2)=TAU
       END IF
20    CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE ELEMNT(IE,SE,FE)
C EQUILIBRIUM EQUATIONS FOR NONCIRCULAR SHAFT ELEMENT
      COMMON/ELEM/IETYPE,NOE,IJ(100,3),MAT(100),NODN,NONE,NODE,NOST
      COMMON/NODE/NONS,XY(100,2),NOCO
      COMMON/MATS/NOPROP,NOSET,EDAT(5,5)
      INTEGER I,J,K
      REAL AI,AJ,AK,AREA,BI,BJ,BK,CI,CJ,CK,XC,YC
      DIMENSION SE(6,6),FE(6)
C ELEMENT STIFFNESS MATRIX [SE]
      I=IJ(IE,1)
      J=IJ(IE,2)
      K=IJ(IE,3)
      AI=XY(J,1)*XY(K,2)-XY(K,1)*XY(J,2)
      AJ=XY(K,1)*XY(I,2)-XY(I,1)*XY(K,2)
      AK=XY(I,1)*XY(J,2)-XY(J,1)*XY(I,2)
      AREA=.5*(AI+AJ+AK)
      BI=XY(J,2)-XY(K,2)
      BJ=XY(K,2)-XY(I,2)
      BK=XY(I,2)-XY(J,2)
      CI=XY(K,1)-XY(J,1)
      CJ=XY(I,1)-XY(K,1)
      CK=XY(J,1)-XY(I,1)
      SE(1,1)=(BI*BI+CI*CI)/(4.*AREA)
      SE(1,2)=(BI*BJ+CI*CJ)/(4.*AREA)
      SE(1,3)=(BI*BK+CI*CK)/(4.*AREA)
      SE(2,2)=(BJ*BJ+CJ*CJ)/(4.*AREA)
      SE(2,3)=(BJ*BK+CJ*CK)/(4.*AREA)
      SE(3,3)=(BK*BK+CK*CK)/(4.*AREA)
      SE(2,1)=SE(1,2)
      SE(3,1)=SE(1,3)
      SE(3,2)=SE(2,3)
C MATRIX [FE]
      XC=(XY(I,1)+XY(J,1)+XY(K,1))/3.
      YC=(XY(I,2)+XY(J,2)+XY(K,2))/3.
      FE(1)=(YC*BI-XC*CI)/2.
      FE(2)=(YC*BJ-XC*CJ)/2.
      FE(3)=(YC*BK-XC*CK)/2.
      RETURN
      END
C
C
```

```
C
      SUBROUTINE STRESS
C STRESSES AND TORSIONAL CONSTANT FOR NONCIRCULAR
      COMMON/EQNS/NODS,NOEQ,IBW,IDA(100,6),SS(200,50),FS(200)
      COMMON/ELEM/IETYPE,NOE,IJ(100,3),MAT(100),NODN,NONE,NODE,NOST
      COMMON/NODE/NONS,XY(100,2),NOCO
      COMMON/MATS/NOPROP,NOSET,EDAT(5,5)
      INTEGER IE,I,J,K
      REAL XI,XJ,XK,YI,YJ,YK,XC,YC,AI,AJ,AK,AREA,BI,BJ,BK,CI,CJ,CK
      REAL F1,F2
C
      SS(1,3)=0.
      DO 10 IE=1,NOE
      I=IJ(IE,1)
      J=IJ(IE,2)
      K=IJ(IE,3)
      XI=XY(I,1)
      XJ=XY(J,1)
      XK=XY(K,1)
      YI=XY(I,2)
      YJ=XY(J,2)
      YK=XY(K,2)
      XC=(XI+XJ+XK)/3.
      YC=(YI+YJ+YK)/3.
      AI=XJ*YK-XK*YJ
      AJ=XK*YI-XI*YK
      AK=XI*YJ-XJ*YI
      AREA=.5*(AI+AJ+AK)
      BI=YJ-YK
      BJ=YK-YI
      BK=YI-YJ
      CI=XK-XJ
      CJ=XI-XK
      CK=XJ-XI
C TORSIONAL CONSTANT
      F1=((XC*CI-YC*BI)*FS(I)+(XC*CJ-YC*BJ)*FS(J)+
     * (XC*CK-YC*BK)*FS(K))/2.
      F2=AREA/6.*(XI*XI+XJ*XJ+XK*XK+YI*YI+YJ*YJ+YK*YK+XI*XJ+XJ*XK+
     * XK*XI+YI*YJ+YJ*YK+YK*YI)
      SS(1,3)=SS(1,3)+F1+F2
C ELEMENT STRESSES
C SS(IE,1)=TAUYZ; SS(IE,2)=TAUZX
      SS(IE,1)=1./(2.*AREA)*(CI*FS(I)+CJ*FS(J)+CK*FS(K))+XC
      SS(IE,2)=1./(2.*AREA)*(BI*FS(I)+BJ*FS(J)+BK*FS(K))-YC
10    CONTINUE
      RETURN
      END
C
C
C END TORO
C
      SUBROUTINE ELBEAM(X,Y,Z,LEVEL,INAX,INAY,MX,MY,SX,SY,ES,EC,DB)
      INTEGER FLAG,LEVEL,LDIV
      REAL B,XRES,D,ES,EC,DB
      REAL INAX,INAY,KN,HN,X,Y,Z,MX,MY,SX,SY,IN
      DIMENSION X(73,30),Y(73,30),Z(73,30),INAX(30),INAY(30)
      DIMENSION MX(30),MY(30),XRES(100),COEF(100,100)
      DIMENSION IN(100),KN(100),HN(100),IPIVOT(100),B(100),D(100)
C
      LDIV=Z(1,LEVEL)/2
C      ES=2000
C      EC=2000
```

```
C       DB=60
C
        FLAG=1
        CALL PARAM(X,Y,Z,LDIV,KN,HN,IN,INAY,FLAG,EC,DB)
        CALL MATRIX(IN,KN,HN,COEF,LDIV+1,ES,MY(LEVEL),SX,B)
        CALL FACTOR(COEF,IPIVOT,D,LDIV+1)
        CALL SUBST(COEF,B,XRES,IPIVOT,LDIV+1)
        CALL MOMENT(MY,SX,XRES,LEVEL,LDIV,HN,Z,ES,IN)
C
C
        FLAG=0
        CALL PARAM(X,Y,Z,LDIV,KN,HN,IN,INAX,FLAG,EC,DB)
        CALL MATRIX(IN,KN,HN,COEF,LDIV+1,ES,MX(LEVEL),SY,B)
        CALL FACTOR(COEF,IPIVOT,D,LDIV+1)
        CALL SUBST(COEF,B,XRES,IPIVOT,LDIV+1)
        CALL MOMENT(MX,SY,XRES,LEVEL,LDIV,HN,Z,ES,IN)
C
C       CALCULATE STRESSES
C
        RETURN
        END
C
C       END ELBEAM
C
C
        SUBROUTINE PARAM(X,Y,Z,LEVEL,KN,HN,IN,INA,FLAG,EC,DB)
        INTEGER I,K,L,LEVEL,FLAG
        REAL X,Y,Z,KN,HN,IN,INA,DB,EC,DS,HSUM
        DIMENSION X(73,30),Y(73,30),Z(73,30),INA(30)
        DIMENSION IN(100),KN(100),HN(100)
C
C       PREAPERING THE PARAMETERS FOR THE CALCULATION
C       DB - BONE DIAMETER (60 MM)
C       EC - CEMENT ELASTICITY
C       KN - STIFFNESSES OF THE CEMENT RINGS
C
        K=FLAG*18
C
        HN(1)=2
        IN(1)=INA(1)
        DS=SQRT(X(1+K,1)**2+Y(1+K,1)**2)+SQRT(X(37+K,1)**2+Y(37+K,1)**2)
        DS=DS/DB
C
C       COMPUTATION OF THE STIFFNESS ACCORDING TO THE APPROXIMATE
C       FORMULA DEVELOPED BY HUISKES
C
        KN(1)=EC*1.9*1.2*(0.24+DS/(1-DS))
        L=1
        HSUM=0.0
C
        DO 10 I=1,LEVEL
         HN(I+1)=2
         HSUM=HSUM+1
         IF(HSUM.GT.Z(1,L))THEN
          L=L+1
         END IF
         IN(I+1)=INA(L)
        DS=SQRT(X(1+K,L)**2+Y(1+K,L)**2)+SQRT(X(37+K,L)**2+Y(37+K,L)**2)
         DS=DS/DB
         KN(I+1)=EC*1.9*1.2*(0.24+DS/(1-DS))
10      CONTINUE
        RETURN
```

```
      END
C
C     END PARAM
C
      SUBROUTINE MATRIX(IN,KN,HN,COEF,N,ES,M,S,B)
C     SUBROUTINE CALCULATES THE MATRIX FOR THE SOLUTION
C     OF THE BEAM ON ELASTIC FOUNDATION WITH FINITE
C     DIFFERENCE METHOD
C
      INTEGER I,J,K,N
      REAL IN,KN,M,S,HN,ES,R
      DIMENSION IN(100),KN(100),HN(10),COEF(100,100),B(100)
C
C     MATRIX COEF INCLUDES THE COEFFICIENTS FOR THE DISPLACEMENTS
C     STARTING WITH THE POINT N TO 1. THE LAST RAW IS THE SHEAR
C     FORCE S AND MOMENT M VECTOR
C
C     ES ELASTICITY MODULUS OF THE STEM
C
C     ZEROIZE THE MATRIX COEF
C
      DO 10 I=1,100
       B(I)=0.0
       DO 20 J=1,100
        COEF(I,J)=0.0
20     CONTINUE
10    CONTINUE
C
C     FIRST LINE IN COEF
C
      COEF(1,1)=1+KN(1)*HN(1)**4/ES/IN(1)/2
      COEF(1,2)=-2
      COEF(1,3)=1
      B(1)=0.0
C
C     SECOND LINE IN COEF
C
      COEF(2,N)=KN(N)*HN(N)/2/KN(1)/HN(1)
      COEF(2,1)=0.5
      B(2)=S/KN(1)/HN(1)
      DO 30 I=2,N-1
       COEF(2,I)=KN(N-I+1)*HN(N-I+1)/KN(1)/HN(1)
30    CONTINUE
C
C     COMPUTE REST OF THE MATRIX COEF
C
      K=1
      DO 40 I=3,N
       R=0.0
       DO 50 J=K,N
        IF(J.EQ.K)THEN
         COEF(I,J)=1
        END IF
        IF(J.EQ.K+1)THEN
         COEF(I,J)=-2
        END IF
        IF(J.EQ.K+2)THEN
         R=R+HN(J)
         IF(J.EQ.N)THEN
          COEF(I,J)=1+KN(J)*HN(J)**4/ES/IN(J)/2
         ELSE
          COEF(I,J)=1+KN(J)*HN(J)**4/ES/IN(J)
```

```
          END IF
         END IF
         IF(J.GT.K+2)THEN
          R=R+HN(J)
          IF(J.EQ.N)THEN
           COEF(I,J)=KN(J)*HN(J)**3/ES/IN(J)*R/2
          ELSE
           COEF(I,J)=KN(J)*HN(J)**3/ES/IN(J)*R
          END IF
C           COEF(I,J)=0.0
         END IF
50      CONTINUE
        K=K+1
        B(I)=(S*R+M)*HN(N)**2/ES/IN(N)
40     CONTINUE
C
       RETURN
       END
C
C      END MATRIX
C
       SUBROUTINE FACTOR(COEF,IPIVOT,D,N)
C      GAUSS REDUCTION WITH PARTIAL PIVOTING
C
       INTEGER I,IPIVOT,NM1,N,J,K,KP1,IP,IPK
       REAL D,ROWMAX,COLMAX,AWIKOV,RATIO
       DIMENSION COEF(100,100),IPIVOT(100),D(100)
C
C      INITIALIZE IPIVOT,D
C
       DO 10 I=1,N
        IPIVOT(I)=I
        ROWMAX=0.0
        DO 9 J=1,N
         ROWMAX=AMAX1(ROWMAX,ABS(COEF(I,J)))
9       CONTINUE
        IF(ROWMAX.LT.1E-10)GOTO 999
        D(I)=ROWMAX
10     CONTINUE
C
C      GAUSS ELIMNATION WITH SCALED PARTIAL PIVOTING
C
       NM1=N-1
       IF(NM1.EQ.0)RETURN
       DO 20 K=1,NM1
        J=K
        KP1=K+1
        IP=IPIVOT(K)
        COLMAX=ABS(COEF(IP,K))/D(IP)
        DO 11 I=KP1,N
         IP=IPIVOT(I)
         AWIKOV=ABS(COEF(IP,K))/D(IP)
         IF(AWIKOV.LE.COLMAX) GOTO 11
         COLMAX=AWIKOV
         J=I
11      CONTINUE
C
        IF(COLMAX.EQ.0.0) GOTO 999
C
        IPK=IPIVOT(J)
        IPIVOT(J)=IPIVOT(K)
        IPIVOT(K)=IPK
```

```
C
       DO 19 I=KP1,N
        IP=IPIVOT(I)
        COEF(IP,K)=COEF(IP,K)/COEF(IPK,K)
        RATIO=-COEF(IP,K)
C
        DO 18 J=KP1,N
         COEF(IP,J)=RATIO*COEF(IPK,J)+COEF(IP,J)
18      CONTINUE
19     CONTINUE
20    CONTINUE
C
      IF(COEF(IP,N).EQ.0.0) GOTO 999
      RETURN
999   PRINT*,'MATRIX COEF IS SINGULAR'
      RETURN
      END
C
C     END FACTOR
C
      SUBROUTINE SUBST(COEF,B,X,IPIVOT,N)
C
C     FORWARD AND BACK SUBSTINUSION FOR GAUSS FACTOR
C
      INTEGER I,J,K,IPIVOT,N,IP,KM1,NP1MK,KP1
      REAL B,X,COEF,SUM
      DIMENSION COEF(100,100),B(100),X(100),IPIVOT(100)
C
      IF(N.GT.1) GOTO 10
      X(1)=B(1)/COEF(1,1)
      RETURN
C
10    IP=IPIVOT(1)
      X(1)=B(IP)
      DO 15 K=2,N
       IP=IPIVOT(K)
       KM1=K-1
       SUM=0.0
       DO 14 J=1,KM1
        SUM=COEF(IP,J)*X(J)+SUM
14     CONTINUE
       X(K)=B(IP)-SUM
15    CONTINUE
C
      X(N)=X(N)/COEF(IP,N)
      K=N
      DO 20 NP1MK=2,N
       KP1=K
       K=K-1
       IP=IPIVOT(K)
       SUM=0.0
       DO 19 J=KP1,N
        SUM=COEF(IP,J)*X(J)+SUM
19     CONTINUE
       X(K)=(X(K)-SUM)/COEF(IP,K)
20    CONTINUE
C
      RETURN
      END
C
C     END SUBST
C
```

```
      SUBROUTINE MOMENT(M,S,X,LEVEL,N,HN,Z,ES,IN)
      INTEGER I,N,L,LEVEL
      REAL M,S,X,HN,R,MSUM,ES,IN
      DIMENSION M(30),X(100),HN(100),Z(73,30),IN(100)
C
      L=1
      DO 10 I=2,N-1
       IF(ABS(I*HN(1)-Z(1,L)).LT.0.001)THEN
       M(L)=ES*IN(I)/HN(I)**2*(X(I+1)-2*X(I)+X(I-1))
       L=L+1
       END IF
10    CONTINUE
C
      RETURN
      END
C
C     END MOMENT
C
C
      SUBROUTINE SCFINT(RAT1,RAT2,IPNT,MSTEP,K,SCF)
      INTEGER IPNT,MSTEP,I,J,M
      REAL K,RAT1,RAT2,A,B
      DIMENSION SCF(8,30,2),K(73,30)
C
C FIND STRESS CONCENTRATION FACTORS
C
      IF(RAT1.LE.1.01)THEN
       I=1
      ELSEIF(RAT1.LE.1.02)THEN
       I=2
      ELSEIF(RAT1.LE.1.05)THEN
       I=3
      ELSEIF(RAT1.LE.1.1)THEN
       I=4
      ELSEIF(RAT1.LE.1.2)THEN
       I=5
      ELSEIF(RAT1.LE.1.5)THEN
       I=6
      ELSEIF(RAT1.LE.2)THEN
       I=7
      ELSEIF(RAT1.LE.3)THEN
       I=8
      ELSE
       I=9
      ENDIF
C
      IF(RAT2.LE.0.01)THEN
       J=0
       ELSEIF(RAT2.GT.0.3)THEN
       J=30
      ELSE
       DO 100 M=1,30
        IF(RAT2.LE.(0.01+M*0.01))THEN
         J=M
         GOTO 110
        END IF
100    CONTINUE
110    CONTINUE
      END IF
C
C LINEAR INTERPOLATION OF STRESS CONCENTRATION FACTORS
C
```

```
      IF((I.EQ.9).OR.(J.EQ.0))THEN
       PRINT*,'***WARNING***'
       PRINT*,'STRESS CONCENTRATION FACTOR OUT OF'
       PRINT*,'INTERPOLATION RANGE'
       PRINT*,'SCF SET TO 4.5'
       PRINT*,'******'
       K(IPNT,MSTEP)=4.5
       GOTO 300
      END IF
      IF(J.EQ.30)THEN
       K(IPNT,MSTEP)=SCF(I,J,2)
       GOTO 300
      END IF
      A=(SCF(I,J,1)-SCF(I,J+1,1))/(SCF(I,J,2)-SCF(I,J+1,2))
      B=SCF(I,J,2)-A*SCF(I,J,1)
      K(IPNT,MSTEP)=A*RAT2+B
300   RETURN
      END
C
C END SCFINT
C
```

File containing values of stress concentration factors from the literature.

File SCFDAT

D/d = 1.01

r/d	K_T
0.010	1.637
0.020	1.437
0.030	1.352
0.040	1.311
0.050	1.285
0.060	1.273
0.070	1.258
0.080	1.247
0.090	1.245
0.100	1.237
0.110	1.236
0.120	1.229
0.130	1.226
0.140	1.224
0.150	1.223
0.160	1.222
0.170	1.220
0.180	1.220
0.190	1.219
0.200	1.218
0.210	1.216
0.220	1.211
0.230	1.211
0.240	1.208
0.250	1.206
0.260	1.205
0.270	1.203
0.280	1.204
0.290	1.203
0.300	1.203

D/d = 1.02

r/d	K_T
0.010	1.916
0.020	1.658
0.030	1.535
0.040	1.454
0.050	1.406
0.060	1.374
0.070	1.353
0.080	1.342
0.090	1.327
0.100	1.313
0.110	1.309
0.120	1.297
0.130	1.294
0.140	1.286
0.150	1.282
0.160	1.272
0.170	1.272
0.180	1.269
0.190	1.260

0.200	1.258
0.210	1.254
0.220	1.246
0.230	1.238
0.240	1.237
0.250	1.231
0.260	1.235
0.270	1.230
0.280	1.231
0.290	1.225
0.300	1.226

D/d = 1.05

r/d	K_T
0.010	2.415
0.020	2.033
0.030	1.835
0.040	1.731
0.050	1.658
0.060	1.596
0.070	1.551
0.080	1.515
0.090	1.498
0.100	1.470
0.110	1.452
0.120	1.435
0.130	1.420
0.140	1.404
0.150	1.389
0.160	1.373
0.170	1.365
0.180	1.351
0.190	1.342
0.200	1.333
0.210	1.324
0.220	1.315
0.230	1.303
0.240	1.299
0.250	1.292
0.260	1.283
0.270	1.275
0.280	1.272
0.290	1.269
0.300	1.270

D/d = 1.1

r/d	K_T
0.010	2.795
0.020	2.333
0.030	2.091
0.040	1.952
0.050	1.855
0.060	1.778
0.070	1.715
0.080	1.672
0.090	1.630
0.100	1.593
0.110	1.567
0.120	1.538
0.130	1.519

0.140	1.498
0.150	1.476
0.160	1.462
0.170	1.444
0.180	1.426
0.190	1.413
0.200	1.399
0.210	1.388
0.220	1.371
0.230	1.362
0.240	1.352
0.250	1.344
0.260	1.335
0.270	1.324
0.280	1.314
0.290	1.311
0.300	1.312

D/d =1.2

r/d	K_T
0.010	3.258
0.020	2.698
0.030	2.389
0.040	2.199
0.050	2.057
0.060	1.949
0.070	1.860
0.080	1.792
0.090	1.738
0.100	1.691
0.110	1.654
0.120	1.621
0.130	1.593
0.140	1.569
0.150	1.542
0.160	1.519
0.170	1.497
0.180	1.475
0.190	1.452
0.200	1.439
0.210	1.423
0.220	1.412
0.230	1.392
0.240	1.382
0.250	1.369
0.260	1.360
0.270	1.355
0.280	1.351
0.290	1.342
0.300	1.340

D/d = 1.5

r/d	K_T
0.010	3.796
0.020	2.996
0.030	2.620
0.040	2.378
0.050	2.209
0.060	2.076
0.070	1.974

0.080	1.895
0.090	1.825
0.100	1.772
0.110	1.720
0.120	1.680
0.130	1.648
0.140	1.617
0.150	1.587
0.160	1.556
0.170	1.540
0.180	1.514
0.190	1.493
0.200	1.474
0.210	1.459
0.220	1.445
0.230	1.430
0.240	1.416
0.250	1.401
0.260	1.392
0.270	1.384
0.280	1.373
0.290	1.365
0.300	1.358

D/d = 2

r/d	K_T
0.010	3.997
0.020	3.134
0.030	2.755
0.040	2.515
0.050	2.333
0.060	2.194
0.070	2.078
0.080	1.983
0.090	1.909
0.100	1.846
0.110	1.792
0.120	1.739
0.130	1.695
0.140	1.657
0.150	1.628
0.160	1.596
0.170	1.564
0.180	1.541
0.190	1.518
0.200	1.503
0.210	1.481
0.220	1.462
0.230	1.447
0.240	1.431
0.250	1.413
0.260	1.403
0.270	1.392
0.280	1.382
0.290	1.379
0.300	1.368

D/d = 3

r/d	K_T
0.010	4.197

0.020	3.238
0.030	2.914
0.040	2.673
0.050	2.483
0.060	2.332
0.070	2.202
0.080	2.095
0.090	1.996
0.100	1.923
0.110	1.853
0.120	1.803
0.130	1.755
0.140	1.707
0.150	1.667
0.160	1.629
0.170	1.592
0.180	1.566
0.190	1.542
0.200	1.521
0.210	1.499
0.220	1.480
0.230	1.456
0.240	1.450
0.250	1.429
0.260	1.417
0.270	1.408
0.280	1.400
0.290	1.394
0.300	1.383

Example of the output from the progrm PROCOM. In this example, the elasticity modulus of the embedding was 32 times lower than that of the prosthesis.

File STRESS

ANALYSIS OF PROSTHESIS F076

LOAD	RADIUS	TILTANGLE	EMB.LEVEL
1000.00	2.0000	10.0000	12

MAXIMUM AND MINIMUM STRESSES

STEP	POINT	MAXSTRESS
1	7	4.042
2	3	5.834
3	6	6.375
4	5	5.796
5	6	6.475
6	3	6.231
7	66	1.815
8	39	9.796
9	38	33.303
10	38	55.817
11	36	81.379
12	36	82.798
13	37	70.156
14	34	61.552
15	37	50.578
16	31	43.583
17	35	35.193
18	36	25.136
19	34	18.817
20	32	14.501
21	31	12.144
22	31	8.848

STEP	POINT	MINSTRESS
1	43	-4.051
2	39	-5.863
3	39	-5.982
4	38	-5.376
5	38	-6.346
6	38	-5.862
7	28	-1.914
8	3	-10.693
9	4	-36.921
10	3	-64.270
11	2	-98.268
12	2	-118.033
13	3	-104.000
14	2	-94.758
15	2	-78.547
16	2	-68.641

17	2	-55.959
18	2	-38.503
19	1	-31.067
20	2	-24.962
21	2	-21.347
22	3	-15.947

MAXIMUM AND MINIMUM FACTORED STRESSES

STEP	POINT	MAXSTRESS
1	7	5.608
2	9	8.412
3	9	9.264
4	7	7.951
5	6	9.232
6	5	8.664
7	66	2.677
8	33	11.960
9	38	40.634
10	35	74.675
11	35	99.246
12	40	118.173
13	35	102.301
14	38	93.424
15	32	76.878
16	31	74.965
17	35	60.533
18	32	41.975
19	31	29.235
20	32	18.236
21	31	15.463
22	31	11.266

STEP	POINT	MINSTRESS
1	39	-5.027
2	39	-7.099
3	38	-7.380
4	38	-6.536
5	38	-7.743
6	38	-7.182
7	29	-2.375
8	6	-15.761
9	4	-56.934
10	2	-93.764
11	2	-149.262
12	2	-191.284
13	2	-150.292
14	2	-150.952
15	2	-125.127
16	2	-111.899
17	2	-100.294
18	2	-71.625
19	1	-60.559
20	2	-51.335
21	2	-43.900
22	3	-32.794

STEP	POINT	TOR.STRESS
1	61	10.815
2	61	9.028
3	61	7.670
4	64	5.285
5	61	4.256
6	58	4.379
7	58	3.405
8	58	3.001
9	58	2.604
10	58	2.431
11	58	2.429
12	58	2.898
13	58	3.359
14	58	3.302
15	58	2.648
16	58	2.401
17	58	2.700
18	58	2.051
19	58	1.876
20	58	1.002
21	58	0.689
22	55	0.300

APPENDIX 4. Program for Indexing of Diffraction Patterns

The camera constant method [110] was used for indexing diffraction patterns. In the first stage, the distances (r) of several spots are measured from the central spot. The camera constant of the electron microscope $\lambda * L$, where λ is the wave length of the electrons and L the camera length, is divided by the values of r. The results are compared against a list of plane spacings (d). One of the spots is arbitrarily indexed for a given combination of (h,k,l)-indices corresponding to d. The remaining spots are indexed using rules of vector addition. The angles (α) between lines from the centre spot to the spots measured and the correct combination of the (h,k,l)-indices are checked using formulas for relevant crystal structures. In the second stage, the beam direction of the electrons (B) is determined. Having indexed two spots in the diffraction pattern, the corresponding reciprocal lattice vectors (g) can be computed. The beam direction can be then calculated from the vector product of the g vectors:

$$B = g_1 \hat{} g_2$$

where B - beam direction,
g_1, g_2 - reciprocal lattice vectors.

The beam direction is parallel to the zone axis [u,v,w] of the reflecting planes.

The procedure outlined above works well in case of one phase material with known crystallographic structure. If different phases are present in the material and if they can have similar diffraction patterns, as in case of a titanium alloy, the indexing procedure becomes rather difficult. One way of avoiding the difficulties is to photograph several diffraction patterns with different beam directions, which is time consuming. A second way, is to plot theoretical diffraction patterns for different crystal structures and compare them with the experimental ones. The second method can be performed quickly on a computer. The computer program shown below performs two tasks: it searches for plane spacings and angles which would match experimental data in four different crystallographic systems, and it plots the respective diffraction patterns on the computer screen. The plotted diffraction pattern can then be immediately compared with the experimental one. A version of the program can be used for plotting theoretical diffraction patterns only by directly inputing (hkl)-indices. This version is not presented here, for it would not introduce any new information. As a concluding remark it should be pointed out that the use of the computer program significantly speed up and increased the accuracy of the indexing procedure.

```
Program DIFFRA

c     The program DIFFRA calculates theoretical diffraction
c     patterns which match the input data measured on a
c     experimentally obtained diffraction pattern for hexagonal,
c     cubic and orthorhombic crystal systems. The input data are:
c     r - distance in mm from the central spot to one of the
c     diffraction spots,
c     t - distance in mm from the central spot to a second
c     diffraction spot,
c     alpha - angle between vectors r and t in degrees.
c
c     The program asks several questions during it is ran. Two
c     files are produced on output: the file DIFOUT contains a
c     report of the last session and the file DRAWDAT includes
c     input data for the program DRAW which produces a graphical
c     representation of the calculated diffraction patterns.
c
c     The meanings of the variables in the program are as follows:
c     i,j,k - integer pointers
c     nfcc - number of planes in the fcc-array for face centred
c     cubic structure
c     nbcc - number of planes in the bcc-array for body centred
c     cubic structure
c     nhcp - number of planes in the hcp-array for hexagonal
c     structure
c     nort - number of planes in the ort-array for orthorhombic
c     structure
c     nr - identification number of the diffraction pattern
c     fccind, bccind, hcpind, ortind - arrays containing (hkl)-
c     indices of planes for comparison with fcc-, bcc-, hcp- and
c     ort-array
c     fcc, bcc, hcp, ort - arrays containing (hkl)-indices of all
c     possible planes according to cubic, hexagonal and orthorhombic
c     crystal symmetry
c     l - camera length of the microscope in mm
c     l1,l2 - lower and upper limit of the camera length
c     afcc - basis vector length of the fcc unit cell in Angstroem
c     abcc - basis vector length of the bcc unit cell in Angstroem
c     ahcp, chcp - basis vector lengths of the hcp unit cell
c     aort, bort, cort - basis vector lengths of the ort unit cell
c     lambda - wave length of the electrons in Angstroem
c     pi - π
c     alpha1, alpha2, r1, r2, t1, t2 - lower and upper error
c     limits of alpha, r and t
c     er1, err, ert, era - measurement errors of l (mm), r and t
c     (%), alpha (degrees)
c     da, db, da1, da2, db1, db2 - plane spacings and their
c     limits calculated from r and t
c
c
      program diffra
c
      integer i,j,k,nfcc,nbcc,nhcp,nort,nr
      integer*1 fccind(3,19),bccind(3,28),hcpind(4,25),
     +ortind(3,100)
      integer*1 fcc(700,3),bcc(700,3),hcp(700,3),ort(700,3)
      real l,l1,l2,afcc,abcc,ahcp,chcp,aort,bort,cort,lambda,pi
      real alpha,alpha1,alpha2,r,r1,r2,t,t1,t2
      real er1,err,ert,era
      real da,da1,da2,db,db1,db2
      common /dat1/ fcc,bcc,hcp,ort,afcc,abcc,ahcp,chcp,alpha1,
```

```
      *alpha2,da1,da2,db1,db2,alpha,nfcc,nbcc,nhcp,l,lambda,nr,
      *nort,aort,bort,cort
       data fccind /1,1,1,2,0,0,2,2,0,3,1,1,2,2,2,
      *4,0,0,3,3,3,4,2,0,4,2,2,5,1,1,3,3,3,4,4,0,
      *5,3,1,6,0,0,4,4,2,6,2,0,5,3,3,6,2,2,4,4,4/
       data bccind /1,1,0,2,0,0,2,1,1,2,2,0,3,1,0,2,2,2,3,2,1,4,0,0,
      *4,1,1,3,3,0,4,2,0,3,3,2,4,2,2,5,1,0,4,3,1,5,2,1,4,4,0,
      *5,3,0,4,3,3,6,0,0,4,4,2,6,1,1,5,3,2,6,2,0,5,4,1,6,2,2,6,3,1,
      *4,4,4/
       data hcpind /0,1,-1,0,0,0,0,2,0,1,-1,1,0,1,-1,2,
      *1,1,-2,0,0,1,-1,3,0,2,-2,0,1,1,-2,2,0,2,-2,1,
      *0,0,0,4,0,2,-2,2,0,1,-1,4,0,2,-2,3,1,2,-3,0,1,2,-3,1,
      *0,0,0,5,1,1,-2,4,1,2,-3,2,0,1,-1,5,0,2,-2,4,0,3,-3,0,
      *1,2,-3,3,0,3,-3,2,0,0,0,6,-1,2,-1,6/
c      data ortind /1,1,0,1,0,1,0,1,1,2,0,0,0,2,0,0,0,2,
c     *2,2,0,2,0,2,0,2,2,2,2,2,3,1,0,3,0,1,0,3,1,1,3,0,1,0,3,0,1,3,
c      *3,2,1,3,1,2,1,3,2,2,3,1,2,1,3,1,2,3,3,3,2,3,2,3,2,3,3,4,0,0,
c      *0,4,0,0,0,4,4,1,1,1,4,1,1,1,4,4,2,0,4,0,2,0,2,4,2,4,0,2,0,4,
c      *0,2,4,4,2,2,2,4,2,2,2,4,4,3,1,4,1,3,1,4,3,3,4,1,3,1,4,1,4,3,
c      *4,3,3,3,4,3,3,3,4,4,4,4,5,1,0,5,0,1,0,5,1,1,5,0,1,0,5,0,1,5,
c      *5,2,1,5,1,2,1,5,2,2,5,1,2,1,5,1,2,5,5,3,0,5,0,3,0,5,3,3,5,0,
c      *3,0,5,0,3,5,5,3,2,5,2,3,2,5,3,3,5,2,3,2,5,2,3,5/
       data ortind /1,0,0,0,1,0,0,0,1,1,1,0,1,0,1,0,1,1,2,0,0,0,2,0,
      *0,0,2,1,1,1,1,2,0,1,0,2,0,1,2,2,1,0,2,0,1,0,2,1,2,2,0,2,0,2,
      *2,1,1,1,2,1,1,1,2,
      *0,2,2,2,2,1,2,1,2,1,2,2,2,2,2,3,0,0,0,3,0,0,0,3,3,1,0,3,0,1,
      *0,3,1,1,3,0,1,0,3,0,1,3,3,1,1,1,3,1,1,1,3,3,2,1,3,1,2,1,3,2,
      *2,3,1,2,1,3,1,2,3,3,2,2,2,3,2,2,2,3,3,3,2,3,2,3,2,3,3,3,3,3,
      *4,0,0,0,4,0,0,0,4,4,1,0,4,0,1,0,4,1,1,4,0,1,0,4,0,1,4,4,1,1,
      *1,4,1,1,1,4,4,2,0,4,0,2,0,2,4,2,4,0,2,0,4,0,2,4,4,2,1,4,1,2,
      *1,4,2,2,4,1,2,1,4,1,2,4,4,2,2,2,4,2,2,2,4,4,3,0,4,0,3,0,4,3,
      *3,4,0,3,0,4,0,3,4,4,3,1,4,1,3,1,4,3,3,4,1,3,1,4,1,3,4,4,3,2,
      *4,2,3,2,4,3,3,4,2,3,2,4,2,3,4,4,3,3,3,4,3,3,3,4,4,4,4/
c
       pi=3.141592654
c
c      Camera lengt in mm
       l=755.3h
c
c      Wave length in A for 100kV
       lambda=.037
c
c      Lattice parameter fcc martensite in A
       afcc=4.5
c
c      Lattice parameter bcc Ti in A
       abcc=3.283
c
c      Lattice parameters hcp Ti in A
       ahcp=2.95
       chcp=4.683
c
c      Lattice parameters orthorhombic
       aort=3.01
       bort=4.9
       cort=4.63
c
c      Error limits
       er1=5
       err=5
       ert=err
       era=3
```

```
c
c     Zeroise arrays
      do 10 i=1,700
       do 20 j=1,3
        fcc(i,j)=0
        bcc(i,j)=0
        hcp(i,j)=0
        ort(i,j)=0
20     continue
10    continue
c
c     Compute all possible plane indices for cubic, hexagonal and
c     orthorhombic system
c
      call cubsym(fcc,fccind,19,nfcc,bcc,bccind,28,nbcc)
      call hexsym(hcp,hcpind,25,nhcp,)
      call ortsym(ort,ortind,100,nort,aort,bort,cort)
c
c     Open output files
c
      open(3,file='b:difout')
      open(4,file='b:drawdat')
c
c     Input of the diffr. pattern data
c
5     print*,'Photo number ?'
      read(*,*)nr
      print*,'Input the SHORTEST vector lenght in mm'
      print*,'A= ?'
      read(*,*)r
      print*,'Input the next shortest vector in anticlockwise'
      print*,'direction from the previous one in mm'
      print*,'B= ?'
      read(*,*)t
      print*,'Input the angle between A & B in degrees,'
      print*,'positive in anticlockwise direction'
      print*,'Alpha= ?'
      read(*,*)alpha
c
      print*,'Do you want to change settings ?  (y=1,n=0)'
      read(*,*)i
      if(i.eq.1)then
       print*,'Default parameter:'
       print*,'Camera lenght in mm l=',l
       print*,'Wave length in A lambda=',lambda
       print*,'Lattice parameters in A :'
       print*,'fcc martensite afcc=',afcc
       print*,'bcc Ti         abcc=',abcc
       print*,'hcp Ti         ahcp=',ahcp
       print*,'               chcp=',chcp
       print*,'orthorhombic   aort=',aort
       print*,'               bort=',bort
       print*,'               cort=',cort
       print*,' '
       print*,'Do you want to change the parameters ? (y=1,n=0)'
       read(*,*)i
        if(i.eq.1)then
         print*,'l= ? mm'
         read(*,*)l
         print*,'lambda= ? A'
         read(*,*)lambda
         print*,'afcc= ? A'
```

```
          read(*,*)afcc
          print*,'abcc= ? A'
          read(*,*)abcc
          print*,'ahcp= ? A'
          read(*,*)ahcp
          print*,'chcp= ? A'
          read(*,*)chcp
          print*,'aort= ? A'
          read(*,*)aort
          print*,'bort= ? A'
          read(*,*)bort
          print*,'cort= ? A'
          read(*,*)cort
        end if
        print*,'Do you want to change the error limits ? (y=1,n=0)'
        read(*,*)i
        if(i.eq.1)then
          print*,'erlenght= ? mm              [erl=',erl,']'
          read(*,*)erl
          print*,'erradius= ? %               [err=',err,']'
          read(*,*)err
          print*,'erangle = ? deg             [era=',era,']'
          read(*,*)era
        end if
      end if
c
c     Compute error limits
c
      l1=l-erl
      l2=l+erl
      r1=r-err*r/100
      r2=r+err*r/100
      t1=t-ert*t/100
      t2=t+ert*t/100
      alpha1=alpha-era
      alpha2=alpha+era
c
c     Change degrees into radians
c
      alpha=alpha*pi/180
      alpha1=alpha1*pi/180
      alpha2=alpha2*pi/180
c
c     Computation of d spacing
c
      da1=l1*lambda/r2
      da2=l2*lambda/r1
      db1=l1*lambda/t2
      db2=l2*lambda/t1
c
c     Write report file
c
      write(3,*)
      write(3,*)'  Photo Nr. = ',nr
      write(3,*)'_____________________________'
      write(3,*)
c
c     Find planes which match the input data
c
      call fccpln(fccind)
      call bccpln(bccind)
      call hcppln(hcpind)
```

```
      call ortpln(ortind)
c
      print*,'New data to be computed ? (yes=1,no=0)'
      read(*,*)k
      if(k.eq.1)goto 5
c
      stop
      end
c
c
c     Subroutine CUBSYM creates all symmetrically possible  plane
c     indices (hkl) for cubic crystal system.
c     ifcc,ibcc - number of (hkl)-indices in fccind and bccind to
c     be permutated
c     dum - dummy array
c
      subroutine cubsym(fcc,fccind,ifcc,nfcc,bcc,bccind,ibcc,nbcc)
        integer h,k,l,m,n,o,i,ii,jj,j,nfcc,nbcc,ifcc,ibcc
        integer*1 fcc(700,3),bcc(700,3),dum(700,3)
        integer*1 fccind(3,19),bccind(3,28)
c
c
      nfcc=0
c
c     Symmetry operations in cubic system:
c     face-centred cubic
c           do 10 i=1,ifcc
        h=fccind(1,j)
        k=fccind(2,j)
        l=fccind(3,j)
        call stor(h,k,l,nfcc,fcc)
c
        m=-h
        n=k
        o=-l
        call stor(m,n,o,nfcc,fcc)
c
        m=-h
        n=-k
        o=l
        call stor(m,n,o,nfcc,fcc)
c
        m=h
        n=-k
        o=-l
        call stor(m,n,o,nfcc,fcc)
c
        m=k
        n=l
        o=h
        call stor(m,n,o,nfcc,fcc)
c
        m=-k
        n=l
        o=-h
        call stor(m,n,o,nfcc,fcc)
c
        m=-k
        n=-l
        o=h
        call stor(m,n,o,nfcc,fcc)
c
```

```
      m=k
      n=-l
      o=-h
      call stor(m,n,o,nfcc,fcc)
c
      m=l
      n=h
      o=k
      call stor(m,n,o,nfcc,fcc)
c
      m=-l
      n=h
      o=-k
      call stor(m,n,o,nfcc,fcc)
c
      m=-l
      n=-h
      o=k
      call stor(m,n,o,nfcc,fcc)
c
      m=l
      n=-h
      o=-k
      call stor(m,n,o,nfcc,fcc)
c
      m=-l
      n=-k
      o=-h
      call stor(m,n,o,nfcc,fcc)
c
      m=l
      n=-k
      o=h
      call stor(m,n,o,nfcc,fcc)
c
      m=l
      n=k
      o=-h
      call stor(m,n,o,nfcc,fcc)
c
      m=-l
      n=k
      o=h
      call stor(m,n,o,nfcc,fcc)
c
      m=-h
      n=-l
      o=-k
      call stor(m,n,o,nfcc,fcc)
c
      m=h
      n=-l
      o=k
      call stor(m,n,o,nfcc,fcc)
c
      m=h
      n=l
      o=-k
      call stor(m,n,o,nfcc,fcc)
c
      m=-h
      n=l
```

```
          o=k
          call stor(m,n,o,nfcc,fcc)
c
          m=-k
          n=-h
          o=-l
          call stor(m,n,o,nfcc,fcc)
c
          m=k
          n=-h
          o=l
          call stor(m,n,o,nfcc,fcc)
c
          m=k
          n=h
          o=-l
          call stor(m,n,o,nfcc,fcc)
c
          m=-k
          n=h
          o=l
          call stor(m,n,o,nfcc,fcc)
c
10      continue
c
c
c       Symmetry operations in cubic system:
c       body-centred cubic
c
        nbcc=0
c
        do 20 i=1,28
          h=bccind(1,j)
          k=bccind(2,j)
          l=bccind(3,j)
          call stor(h,k,l,nbcc,bcc)
c
          m=-h
          n=k
          o=-l
          call stor(m,n,o,nbcc,bcc)
c
          m=-h
          n=-k
          o=l
          call stor(m,n,o,nbcc,bcc)
c
          m=h
          n=-k
          o=-l
          call stor(m,n,o,nbcc,bcc)
c
          m=k
          n=l
          o=h
          call stor(m,n,o,nbcc,bcc)
c
          m=-k
          n=l
          o=-h
          call stor(m,n,o,nbcc,bcc)
c
```

```
      m=-k
      n=-l
      o=h
      call stor(m,n,o,nbcc,bcc)
c
      m=k
      n=-l
      o=-h
      call stor(m,n,o,nbcc,bcc)
c
      m=l
      n=h
      o=k
      call stor(m,n,o,nbcc,bcc)
c
      m=-l
      n=h
      o=-k
      call stor(m,n,o,nbcc,bcc)
c
      m=-l
      n=-h
      o=k
      call stor(m,n,o,nbcc,bcc)
c
      m=l
      n=-h
      o=-k
      call stor(m,n,o,nbcc,bcc)
c
      m=-l
      n=-k
      o=-h
      call stor(m,n,o,nbcc,bcc)
c
      m=l
      n=-k
      o=h
      call stor(m,n,o,nbcc,bcc)
c
      m=l
      n=k
      o=-h
      call stor(m,n,o,nbcc,bcc)
c
      m=-l
      n=k
      o=h
      call stor(m,n,o,nbcc,bcc)
c
      m=-h
      n=-l
      o=-k
      call stor(m,n,o,nbcc,bcc)
c
      m=h
      n=-l
      o=k
      call stor(m,n,o,nbcc,bcc)
c
      m=h
      n=l
```

```
        o=-k
        call stor(m,n,o,nbcc,bcc)
c
        m=-h
        n=l
        o=k
        call stor(m,n,o,nbcc,bcc)
c
        m=-k
        n=-h
        o=-l
        call stor(m,n,o,nbcc,bcc)
c
        m=k
        n=-h
        o=l
        call stor(m,n,o,nbcc,bcc)
c
        m=k
        n=h
        o=-l
        call stor(m,n,o,nbcc,bcc)
c
        m=-k
        n=h
        o=l
        call stor(m,n,o,nbcc,bcc)
c
20    continue

c
c     Delete repeated indices for fcc
c
        jj=0
        do 30 ii=1,nfcc
          h=fcc(ii,1)
          k=fcc(ii,2)
          l=fcc(ii,3)
          if((h.eq.0).and.(k.eq.0).and.(l.eq.0))goto 30
          do 40 j=ii+1,nfcc
            m=fcc(j,1)
            n=fcc(j,2)
            o=fcc(j,3)
            if((m.eq.0).and.(n.eq.0).and.(o.eq.0))goto 40
            if((h.eq.m).and.(k.eq.n).and.(l.eq.o))then
              fcc(j,1)=0
              fcc(j,2)=0
              fcc(j,3)=0
            end if
40        continue
            jj=jj+1
            dum(jj,1)=h
            dum(jj,2)=k
            dum(jj,3)=l
30      continue
c
        do 50 i=1,jj
          fcc(i,1)=dum(i,1)
          fcc(i,2)=dum(i,2)
          fcc(i,3)=dum(i,2)
50      continue
c
```

```
c     Number of (hkl)-indices in fcc-array
c
        nfcc=jj
c
c
c     Delete repeated indices for bcc
c
        jj=0
        do 70 ii=1,nbcc
          h=bcc(ii,1)
          k=bcc(ii,2)
          l=bcc(ii,3)
          if((h.eq.0).and.(k.eq.0).and.(l.eq.0))goto 70
          do 80 j=ii+1,nbcc
            m=bcc(j,1)
            n=bcc(j,2)
            o=bcc(j,3)
            if((m.eq.0).and.(n.eq.0).and.(o.eq.0))goto 80
            if((h.eq.m).and.(k.eq.n).and.(l.eq.o))then
              bcc(j,1)=0
              bcc(j,2)=0
              bcc(j,3)=0
            end if
80        continue
            jj=jj+1
            dum(jj,1)=h
            dum(jj,2)=k
            dum(jj,3)=l
70      continue
c
        do 90 i=1,jj
          bcc(i,1)=dum(i,1)
          bcc(i,2)=dum(i,2)
          bcc(i,3)=dum(i,3)
90      continue
c
c     Number of (hkl)-indices in bcc-array
c
        nbcc=jj
c
        stop
c
c
c     Subroutine HEXSYM creates all symmetrically possible  plane
c     indices(hkl) for hexagonal crystal system.
c     ihcp -  number  of  (hkl)-indices  in  hcpind-array  to  be
c     permutated
c     dum - dummy array
c
      subroutine hexsym(hcp,hcpind,ihcp,nhcp)
        integer ii,j,jj,nhcp,ihcp,h,k,i,l,m,n,o
        integer*1 hcp(500,3),dum(500,3),hcpind(3,25)
c
c     Symmetry operations in hexagonal system
c
      do 10 j=1,ihcp
        h=hcpind(1,j)
        k=hcpind(2,j)
        l=hcpind(3,j)
        i=-(h+k)
        call stor(h,k,l,nhcp,hcp)
c
```

```
          m=-i
          n=-h
          o=l
          call stor(m,n,o,nhcp,hcp)
c
          m=k
          n=i
          call stor(m,n,o,nhcp,hcp)
c
          m=-h
          n=-k
          call stor(m,n,o,nhcp,hcp)
c
          m=i
          n=h
          call stor(m,n,o,nhcp,hcp)
c
          m=-k
          n=-i
          call stor(m,n,o,nhcp,hcp)
c
          o=-l
          m=h
          n=i
          call stor(m,n,o,nhcp,hcp)
c
          m=k
          n=h
          call stor(m,n,o,nhcp,hcp)
c
          m=i
          n=k
          call stor(m,n,o,nhcp,hcp)
c
          m=-i
          n=-k
          call stor(m,n,o,nhcp,hcp)
c
          m=-h
          n=-i
          call stor(m,n,o,nhcp,hcp)
c
          m=-k
          n=-h
          call stor(m,n,o,nhcp,hcp)
c
10      continue
c
c       Delete repeated (hkl)-indices
c
          jj=0
          do 30 ii=1,nhcp
            h=hcp(ii,1)
            k=hcp(ii,2)
            l=hcp(ii,3)
            if((h.eq.0).and.(k.eq.0).and.(l.eq.0))goto 30
            do 40 j=ii+1,nhcp
              m=hcp(j,1)
              n=hcp(j,2)
              o=hcp(j,3)
              if((m.eq.0).and.(n.eq.0).and.(o.eq.0))goto 40
              if((h.eq.m).and.(k.eq.n).and.(l.eq.o))then
```

```
              hcp(j,1)=0
              hcp(j,2)=0
              hcp(j,3)=0
            end if
40        continue
            jj=jj+1
            dum(jj,1)=h
            dum(jj,2)=k
            dum(jj,3)=l
30        continue
c
          do 50 i=1,jj
            hcp(i,1)=dum(i,1)
            hcp(i,2)=dum(i,2)
            hcp(i,3)=dum(i,3)
50        continue
c
c       Number of (hkl)-indices in hcp-array
c
          nhcp=jj
c
          stop
c
c
c
c       Subroutine ORTSYM creates all symmetrically possible  plane
c       indices (hkl) for orthorhombic crystal system.
c       ihcp -  number  of  (hkl)-indices  in  ortind-array  to  be
c       permutated
c       dum - dummy array
c
        subroutine ortsym(ort,ortind,iort,nort,aort,bort,cort)
          integer ii,j,jj,nort,iort,h,k,l,m,n,o
          integer*1 ort(700,3),dum(700,3),ortind(3,100)
          real d1,d2,aort,bort,cort,a,b,c
c
c       Symmetry operations in orthorhombic system
c
        do 10 j=1,iort
          h=ortind(1,j)
          k=ortind(2,j)
          l=ortind(3,j)
          call stor(h,k,l,nort,ort)
c
          m=h
          n=-k
          o=-l
          call stor(m,n,o,nort,ort)
c
          m=-h
          n=k
          o=-l
          call stor(m,n,o,nort,ort)
c
          m=-h
          n=-k
          o=l
          call stor(m,n,o,nort,ort)
c
10      continue
c
c       Sorting  of (hkl) according to the value of the spacing  of
```

```
c     the planes d
c
        a=aort*aort
        b=bort*bort
        c=cort*cort
c
        do 25 ii=1,nort
       d1=(ort(ii,1)**2)/a+(ort(ii,2)**2)/b+(ort(ii,3)**2)/c
          if(d1.lt.0.001)goto 25
          do 26 jj=ii,nort
       d2=(ort(jj,1)**2)/a+(ort(jj,2)**2)/b+(ort(jj,3)**2)/c
            if(d2.lt.0.0001)goto 26
            if(d1.gt.d2)then
              h=ort(ii,1)
              k=ort(ii,2)
              l=ort(ii,3)
              ort(ii,1)=ort(jj,1)
              ort(ii,2)=ort(jj,2)
              ort(ii,3)=ort(jj,3)
              ort(jj,1)=h
              ort(jj,2)=k
              ort(jj,3)=l
              d1=d2
            end if
26        continue
25      continue
c
c     Delete repeated (hkl)-indices from ort-array
c
        jj=0
        do 30 ii=1,nort
          h=ort(ii,1)
          k=ort(ii,2)
          l=ort(ii,3)
          if((h.eq.0).and.(k.eq.0).and.(l.eq.0))goto 30
          do 40 j=ii+1,nort
            m=ort(j,1)
            n=ort(j,2)
            o=ort(j,3)
            if((m.eq.0).and.(n.eq.0).and.(o.eq.0))goto 40
            if((h.eq.m).and.(k.eq.n).and.(l.eq.o))then
              ort(j,1)=0
              ort(j,2)=0
              ort(j,3)=0
            end if
40        continue
            jj=jj+1
            dum(jj,1)=h
            dum(jj,2)=k
            dum(jj,3)=l
30      continue
c
        do 50 i=1,jj
          ort(i,1)=dum(i,1)
          ort(i,2)=dum(i,2)
          ort(i,3)=dum(i,2)
50      continue
c
c     Number of (hkl)-indices in ort-array
c
        nort=jj
c
```

```
        stop
c
c
c
c     Subroutine   STOR   writes   the   (hkl)-indices   into   an
c     appropriate array (fcc,bcc,hcp or ort).
c     STOR is called from CUBSYM, HEXSYM and ORTSYM
c     npl - number of the planes
c     array - name of the array
c
        subroutine stor(h,k,l,npl,array)
        integer h,k,l,npl
        integer*1 array(700,3)
c
        npl=npl+1
        array(npl,1)=h
        array(npl,2)=k
        array(npl,3)=l
c
        return
        end
c
c
c     Subroutine  FCCPLN finds if any of the (hkl)-indices  match
c     the input data and then calculates all planes belonging  to
c     the same zone.  If a match is found,  the (hkl)-indices  of
c     the planes are written to the output file DRAWDAT.
c
c     The format of the data in DRAWDAT is as follows:
c     Integer number 1 indicates the fcc system
c     nr - identification number of the diffraction pattern
c     l,lambda,abcc - camera length, wave length and unit vector
c     nout - number of following integer triplets
c       first triplet u,v,w - zone axis
c       next triplets h,k,l - plane indices
c
      subroutine fccpln(fccind)
      integer i,ii,j,jj,h1,k1,l1,h2,k2,l2,h3,k3,l3,nplna,nplnb
      integer zeiga(19),zeigb(19),nfcc,nbcc,nhcp,nort,nr
      integer*1 fccind(3,19)
      integer*1 fcc(700,3),bcc(700,3),hcp(700,3),ort(700,3)
      integer out(200,3),nout
      real d,da,db,cosab,u,v,w,u1,v1,w1
      real afcc,abcc,ahcp,chcp,aort,bort,cort,l,lambda
      real alpha,alpha1,alpha2
      real da1,da2,db1,db2
      common /dat1/ fcc,bcc,hcp,ort,afcc,abcc,ahcp,chcp,alpha1,
     *alpha2,da1,da2,db1,db2,alpha,nfcc,nbcc,nhcp,l,lambda,nr,
     *nort,aort,bort,cort
c
c     Zeroise pointer arrays
c
      do 5 i=1,19
        zeiga(i)=0
        zeigb(i)=0
5     continue
c
c     Find matching planes
c
      nplna=0
      nplnb=0
      do 10 i=1,19
```

```
        h1=fccind(1,i)
        k1=fccind(2,i)
        l1=fccind(3,i)
        d=(afcc**2)/(h1*h1+k1*k1+l1*l1)
        d=sqrt(d)
        if((d.ge.da1).and.(d.le.da2))then
          nplna=nplna+1
          zeiga(nplna)=i
          print*,'OK for A !     d=',d
          write(3,*)'OK for A !     d=',d
        end if
        if((d.ge.db1).and.(d.le.db2))then
          nplnb=nplnb+1
          zeigb(nplnb)=i
          print*,'OK for B !     d=',d
          write(3,*)'OK for B !     d=',d
        end if
10    continue
c
c     Write report file
c
      if((nplna.eq.0).or.(nplnb.eq.0))then
        print*,'*************************   '
        print*,'  No planes found for fcc '
        print*,'*************************'
        write(3,*)'*************************   '
        write(3,*)'  No planes found for fcc '
        write(3,*)'*************************'
        return
      end if
c
c     Compute plane angles
c
      do 20 i=1,nplna
        h1=fccind(1,zeiga(i))
        k1=fccind(2,zeiga(i))
        l1=fccind(3,zeiga(i))
        da=h1**2+k1**2+l1**2
        da=sqrt(da)
        do 30 ii=1,nplnb
          h2=fccind(1,zeigb(ii))
          k2=fccind(2,zeigb(ii))
          l2=fccind(3,zeigb(ii))
          db=h2**2+k2**2+l2**2
          db=sqrt(db)
          u1=0
          v1=0
          w1=0
          nout=0
          do 40 j=1,nfcc
            h2=fcc(j,1)
            k2=fcc(j,2)
            l2=fcc(j,3)
            if((h2.eq.0).and.(k2.eq.0).and.(l2.eq.0))goto 40
            d=h2**2+k2**2+l2**2
            d=sqrt(d)
            if((db.ge.(d-.01)).and.(db.le.(d+.01)))then
              cosab=(h1*h2+k1*k2+l1*l2)/(da*db)
              if(cosab.ge.1)then
                cosab=.999999
              end if
              if(cosab.le.-1)then
```

```
              cosab=-.999999
            end if
            cosab=acos(cosab)
            if((cosab.ge.alpha1).and.(cosab.le.alpha2))then
c
c     Find the zone axis
c
              u=k1*l2-l1*k2
              v=l1*h2-h1*l2
              w=h1*k2-k1*h2
              if((u*u+v*v+w*w).le.0.01)goto 40
         if((abs((u1**2+v1**2+w1**2)-(u*u+v*v+w*w))).ge.0.01)then
c
c     Write report file
c
              print*,'****'
              print*,'Found fcc:'
              print*,'Zone axis:   [',u,v,w,']'
              print*,'1st vector:  (',h1,k1,l1,')'
              print*,'2nd vector:  (',h2,k2,l2,')'
              write(3,*)'****'
              write(3,*)'Found fcc:'
              write(3,*)'Zone axis:   [',u,v,w,']'
              write(3,*)'1st vector:  (',h1,k1,l1,')'
              write(3,*)'2nd vector:  (',h2,k2,l2,')'
c
              nout=nout+1
              out(nout,1)=ifix(u)
              out(nout,2)=ifix(v)
              out(nout,3)=ifix(w)
              nout=nout+1
              out(nout,1)=h1
              out(nout,2)=k1
              out(nout,3)=l1
              nout=nout+1
              out(nout,1)=h2
              out(nout,2)=k2
              out(nout,3)=l2
c
c     Find planes belonging to the zone [u,v,w]
c
              do 50 jj=1,nfcc
                h3=fcc(jj,1)
                k3=fcc(jj,2)
                l3=fcc(jj,3)
                if((h3*h3+k3*k3+l3*l3).gt..01)then
                  if((abs(h3*u+k3*v+l3*w)).le.0.01)then
                    print*,'Next vector: (',h3,k3,l3,')'
                    write(3,*)'Next vector: (',h3,k3,l3,')'
c
              nout=nout+1
              out(nout,1)=h3
              out(nout,2)=k3
              out(nout,3)=l3
c
                  end if
                end if
50            continue
c
c     Write output to the DRAWDAT file
c
              write(4,*)1
```

```
                write(4,*)nr
                write(4,*)1,lambda,afcc
                write(4,*)nout
                do 55 jj=1,nout
                  write(4,*)(out(jj,jjj),jjj=1,3)
55              continue
                nout=0
c
               end if
                u1=u
                v1=v
                w1=w
              end if
            end if
40        continue
30      continue
20    continue
c
      return
      end
c
c
c     Subroutine  BCCPLN finds if any of the (hkl)-indices  match
c     the input data and then calculates all planes belonging  to
c     the same zone.  If a match is found,  the (hkl)-indices  of
c     the planes are written to the output file DRAWDAT.
c
c     The format of the data in DRAWDAT is as follows:
c     Integer number 2 indicates the bcc system
c     nr - identification number of the diffraction pattern
c     1,lambda,abcc - camera length, wave length and unit vector
c     nout - number of following integer triplets
c       first triplet u,v,w - zone axis
c       next triplets h,k,l - plane indices
c
      subroutine bccpln(bccind)
      integer i,ii,j,jj,h1,k1,l1,h2,k2,l2,h3,k3,l3,nplna,nplnb
      integer zeiga(28),zeigb(28),nfcc,nbcc,nhcp,nort
      integer*1 bccind(3,28)
      integer out(200,3),nout,nr
      integer*1 fcc(700,3),bcc(700,3),hcp(700,3),ort(700,3)
      real d,da,db,cosab,u,v,w,u1,v1,w1
      real afcc,abcc,ahcp,chcp,aort,bort,cort,l,lambda
      real alpha,alpha1,alpha2
      real da1,da2,db1,db2
      common /dat1/ fcc,bcc,hcp,ort,afcc,abcc,ahcp,chcp,alpha1,
     *alpha2,da1,da2,db1,db2,alpha,nfcc,nbcc,nhcp,l,lambda,nr
     *nort,aort,bort,cort
c
c     Zeroise pointer arrays
c
      do 5 i=1,28
        zeiga(i)=0
        zeigb(i)=0
5     continue
c
c     Find matching planes
c
      nplna=0
      nplnb=0
      do 10 i=1,28
        h1=bccind(1,i)
```

```
        k1=bccind(2,i)
        l1=bccind(3,i)
        d=(abcc**2)/(h1*h1+k1*k1+l1*l1)
        d=sqrt(d)
        if((d.ge.da1).and.(d.le.da2))then
          nplna=nplna+1
          zeiga(nplna)=i
          print*,'OK for A !     d=',d
          write(3,*)'OK for A !     d=',d
        end if
        if((d.ge.db1).and.(d.le.db2))then
          nplnb=nplnb+1
          zeigb(nplnb)=i
          print*,'OK for B !     d=',d
          write(3,*)'OK for B !     d=',d
        end if
10    continue
c
c     Write report file
c
      if((nplna.eq.0).or.(nplnb.eq.0))then
        print*,'*************************   '
        print*,'  No planes found for bcc '
        print*,'*************************'
        write(3,*)'*************************   '
        write(3,*)'  No planes found for bcc '
        write(3,*)'*************************'
        return
      end if
c
c     Compute plane angles
c
      do 20 i=1,nplna
        h1=bccind(1,zeiga(i))
        k1=bccind(2,zeiga(i))
        l1=bccind(3,zeiga(i))
        da=h1**2+k1**2+l1**2
        da=sqrt(da)
        do 30 ii=1,nplnb
          h2=bccind(1,zeigb(ii))
          k2=bccind(2,zeigb(ii))
          l2=bccind(3,zeigb(ii))
          db=h2**2+k2**2+l2**2
          db=sqrt(db)
          u1=0
          v1=0
          w1=0
          nout=0
          do 40 j=1,nbcc
            h2=bcc(j,1)
            k2=bcc(j,2)
            l2=bcc(j,3)
            if((h2.eq.0).and.(k2.eq.0).and.(l2.eq.0))goto 40
            d=h2**2+k2**2+l2**2
            d=sqrt(d)
            if((db.ge.(d-.01)).and.(db.le.(d+.01)))then
              cosab=(h1*h2+k1*k2+l1*l2)/(da*db)
              cosab=acos(cosab)
              if((cosab.ge.alpha1).and.(cosab.le.alpha2))then
c
c     Find the zone axis
c
```

```
                u=k1*l2-l1*k2
                v=l1*h2-h1*l2
                w=h1*k2-k1*h2
c
c
                if((u*u+v*v+w*w).le.0.01)goto 40
          if((abs((u1**2+v1**2+w1**2)-(u*u+v*v+w*w))).gt.0.01)then
c
c       Write report file
c
                print*,'****'
                print*,'Found bcc:'
                print*,'Zone axis:   [',u,v,w,']'
                print*,'1st vector:  (',h1,k1,l1,')'
                print*,'2nd vector:  (',h2,k2,l2,')'
                write(3,*)'****'
                write(3,*)'Found bcc:'
                write(3,*)'Zone axis:   [',u,v,w,']'
                write(3,*)'1st vector:  (',h1,k1,l1,')'
                write(3,*)'2nd vector:  (',h2,k2,l2,')'
c
c
                nout=nout+1
                out(nout,1)=ifix(u)
                out(nout,2)=ifix(v)
                out(nout,3)=ifix(w)
                nout=nout+1
                out(nout,1)=h1
                out(nout,2)=k1
                out(nout,3)=l1
                nout=nout+1
                out(nout,1)=h2
                out(nout,2)=k2
                out(nout,3)=l2
c
c       Find planes belonging to the zone [u,v,w]
c
                do 50 jj=1,nbcc
                  h3=bcc(jj,1)
                  k3=bcc(jj,2)
                  l3=bcc(jj,3)
                  if((h3*h3+k3*k3+l3*l3).gt..01)then
                    if((abs(h3*u+k3*v+l3*w)).le.0.01)then
                      print*,'Next vector: (',h3,k3,l3,')'
                      write(3,*)'Next vector: (',h3,k3,l3,')'
c
                nout=nout+1
                out(nout,1)=h3
                out(nout,2)=k3
                out(nout,3)=l3
                     end if
                   end if
c
50              continue
c
c       Write DRAWDAT
c
                write(4,*)2
                write(4,*)nr
                write(4,*)1,lambda,abcc
                write(4,*)nout
                do 55 jj=1,nout
```

```
                write(4,*)(out(jj,jjj),jjj=1,3)
55              continue
                nout=0
c
               end if
                u1=u
                v1=v
                w1=w
              end if
            end if
40        continue
30      continue
20    continue
c
      return
      end
c
c
c     Subroutine  HCPPLN finds if any of the (hkl)-indices  match
c     the input data and then calculates all planes belonging  to
c     the same zone.  If a match is found,  the (hkl)-indices  of
c     the planes are written to the output file DRAWDAT.
c
c     The format of the data in DRAWDAT is as follows:
c     Integer number 3 indicates the hcp system
c     nr - identification number of the diffraction pattern
c     l,lambda,abcc,chcp -  camera length,  wave length and  unit
c     vectors
c     nout - number of following integer triplets
c       first triplet u,v,w - zone axis
c       next triplets h,k,l - plane indices
c
      subroutine hcppln(hcpind)
      integer i,ii,j,jj,h1,k1,l1,h2,k2,l2,h3,k3,l3,nplna,nplnb
      integer i1,i2,i3
      integer zeiga(25),zeigb(25),nhcp,nfcc,nbcc,nort
      integer*1 hcpind(4,25)
      integer*1 fcc(700,3),bcc(700,3),hcp(700,3),ort(700,3)
      integer out(200,3),nout,nr
      real d,da,db,cosab,qphi,u,v,t,w,u1,v1,t1,w1
      real afcc,abcc,ahcp,chcp,aort,bort,cort,l,lambda
      real alpha,alpha1,alpha2
      real da1,da2,db1,db2
      common /dat1/ fcc,bcc,hcp,ort,afcc,abcc,ahcp,chcp,alpha1,
     *alpha2,da1,da2,db1,db2,alpha,nfcc,nbcc,nhcp,l,lambda,nr,
     *nort,aort,bort,cort
c
      qphi=2*((chcp/ahcp)**2)/3
c
c     Zeroise pointer arrays
c
      do 5 i=1,25
        zeiga(i)=0
        zeigb(i)=0
5     continue
c
c     Find matching planes
c
      nplna=0
      nplnb=0
      do 10 i=1,25
        h1=hcpind(1,i)
```

```
          k1=hcpind(2,i)
          i1=hcpind(3,i)
          l1=hcpind(4,i)
          d=(3*ahcp**2/2)/(h1*h1+k1*k1+i1*i1+l1*l1*(1/qphi))
          d=sqrt(d)
          if((d.ge.da1).and.(d.le.da2))then
            nplna=nplna+1
            zeiga(nplna)=i
            print*,'OK for A !     d=',d
            write(3,*)'OK for A !     d=',d
          end if
          if((d.ge.db1).and.(d.le.db2))then
            nplnb=nplnb+1
            zeigb(nplnb)=i
            print*,'OK for B !     d=',d
            write(3,*)'OK for B !     d=',d
          end if
10      continue
c
c       Write report file
c
        if((nplna.eq.0).or.(nplnb.eq.0))then
          print*,'************************   '
          print*,'  No planes found for hcp '
          print*,'************************'
          write(3,*)'************************   '
          write(3,*)'  No planes found for hcp '
          write(3,*)'************************'
          return
        end if
c
c       Compute plane angles
c
        do 20 i=1,nplna
          h1=hcpind(1,zeiga(i))
          k1=hcpind(2,zeiga(i))
          i1=hcpind(3,zeiga(i))
          l1=hcpind(4,zeiga(i))
          da=h1**2+k1**2+i1**2+(1/qphi)*l1**2
          da=sqrt(da)
          do 30 ii=1,nplnb
            h2=hcpind(1,zeigb(ii))
            k2=hcpind(2,zeigb(ii))
            i2=hcpind(3,zeigb(ii))
            l2=hcpind(4,zeigb(ii))
            db=h2**2+k2**2+i2**2+(1/qphi)*l2**2
            db=sqrt(db)
            u1=0
            v1=0
            t1=0
            w1=0
            nout=0
            do 40 j=1,nhcp
              h2=hcp(j,1)
              k2=hcp(j,2)
              i2=-(h2+k2)
              l2=hcp(j,3)
              if((h2.eq.0).and.(k2.eq.0).and.(l2.eq.0))goto 40
              d=h2**2+k2**2+i2**2+(1/qphi)*l2**2
              d=sqrt(d)
              if((db.ge.(d-.01)).and.(db.le.(d+.01)))then
                cosab=(h1*h2+k1*k2+i1*i2+(1/qphi)*l1*l2)/(da*db)
```

```
            if(cosab.ge.1)then
              cosab=.999999
            end if
            if(cosab.le.-1)then
             cosab=-.999999
            end if
            cosab=acos(cosab)
            if((cosab.ge.alpha1).and.(cosab.le.alpha2))then
c
c     Find the zone axis
c
              u=l2*(2*k1+h1)-l1*(2*k2+h2)
              v=l1*(2*h2+k2)-l2*(2*h1+k1)
              w=3*(h1*k2-h2*k1)
              t=-(u+v)
              if((u*u+v*v+w*w).le.0.01)goto 40
     if((abs((u1**2+v1**2+t1**2+w1**2)-(u*u+v*v+t*t+w*w))).gt.
    *.01)then
c
c     Write report file
c
              print*,'****'
              print*,'Found hcp:'
              print*,'Zone axis:   [',u,v,t,w,']'
              print*,'1st vector:  (',h1,k1,i1,l1,')'
              print*,'2nd vector:  (',h2,k2,i2,l2,')'
              write(3,*)'****'
              write(3,*)'Found hcp:'
              write(3,*)'Zone axis:   [',u,v,t,w,']'
              write(3,*)'1st vector:  (',h1,k1,i1,l1,')'
              write(3,*)'2nd vector:  (',h2,k2,i2,l2,')'
c
c
              nout=nout+1
              out(nout,1)=ifix(u)
              out(nout,2)=ifix(v)
              out(nout,3)=ifix(w)
              nout=nout+1
              out(nout,1)=h1
              out(nout,2)=k1
              out(nout,3)=l1
              nout=nout+1
              out(nout,1)=h2
              out(nout,2)=k2
              out(nout,3)=l2
c
c     Find planes belonging to the zone [u,v,t,w]
c
              do 50 jj=1,nhcp
                h3=hcp(jj,1)
                k3=hcp(jj,2)
                i3=-(h3+k3)
                l3=hcp(jj,3)
                if((h3*h3+k3*k3+l3*l3).gt..01)then
                  if((abs(h3*u+k3*v+i3*t+l3*w)).le.0.01)then
c
c     Write report file
c
                    write(3,*)'Next vector: (',h3,k3,i3,l3,')'
                    print*,'Next vector: (',h3,k3,i3,l3,')'
c
              nout=nout+1
```

```
                out(nout,1)=h3
                out(nout,2)=k3
                out(nout,3)=l3
c
                      end if
                    end if
50              continue
c
c     Write DRAWDAT
c
                write(4,*)3
                write(4,*)nr
                write(4,*)l,lambda,ahcp,chcp
                write(4,*)nout
                do 55 jj=1,nout
                  write(4,*)(out(jj,jjj),jjj=1,3)
55              continue
                nout=0
c
               end if
               u1=u
               v1=v
               t1=t
               w1=w
              end if
            end if
40        continue
30      continue
20    continue
c
      return
      end
c
c
c     Subroutine  ORTPLN finds if any of the (hkl)-indices  match
c     the input data and then calculates all planes belonging  to
c     the same zone.  If a match is found,  the (hkl)-indices  of
c     the planes are written to the output file DRAWDAT.
c
c     The format of the data in DRAWDAT is as follows:
c     Integer number 4 indicates the ort system
c     nr - identification number of the diffraction pattern
c     l,lambda,aort,bort,cort -  camera length,  wave length  and
c     unit vectors
c     nout - number of following integer triplets
c       first triplet u,v,w - zone axis
c       next triplets h,k,l - plane indices
c
      subroutine ortpln(ortind)
      integer i,ii,j,jj,h1,k1,l1,h2,k2,l2,h3,k3,l3,nplna,nplnb
      integer zeiga(100),zeigb(100),nfcc,nbcc,nhcp,nort,nr
      integer*1 ortind(3,100)
      integer*1 fcc(700,3),bcc(700,3),hcp(700,3),ort(700,3)
      integer out(200,3),nout
      real d,da,db,cosab,u,v,w,u1,v1,w1
      real afcc,abcc,ahcp,chcp,aort,bort,cort,l,lambda
      real alpha,alpha1,alpha2
      real da1,da2,db1,db2
      common /dat1/ fcc,bcc,hcp,ort,afcc,abcc,ahcp,chcp,alpha1,
     *alpha2,da1,da2,db1,db2,alpha,nfcc,nbcc,nhcp,l,lambda,nr,
     *nort,aort,bort,cort
c
```

```
c     Zeroise pointer arrays
c
      do 5 i=1,100
        zeiga(i)=0
        zeigb(i)=0
5     continue
c
c     Find matching planes
c
      nplna=0
      nplnb=0
      do 10 i=1,100
        h1=ortind(1,i)
        k1=ortind(2,i)
        l1=ortind(3,i)
c        print*,'Checking ort hkl',h1,k1,l1
c        write(3,*)'Checking ort hkl',h1,k1,l1
        d=(aort*aort*bort*bort*cort*cort)/(h1*h1*bort*bort*cort*
     *cort+k1*k1*cort*cort*aort*aort+l1*l1*aort*aort*bort*bort)
        d=sqrt(d)
        if((d.ge.da1).and.(d.le.da2))then
          nplna=nplna+1
          zeiga(nplna)=i
          print*,'OK for A !     d=',d
          write(3,*)'OK for A !     d=',d
        end if
        if((d.ge.db1).and.(d.le.db2))then
          nplnb=nplnb+1
          zeigb(nplnb)=i
          print*,'OK for B !     d=',d
          write(3,*)'OK for B !     d=',d
        end if
10    continue
c
c     Write report file
c
      if((nplna.eq.0).or.(nplnb.eq.0))then
        print*,'************************   '
        print*,'  No planes found for ort '
        print*,'************************'
        write(3,*)'************************   '
        write(3,*)'  No planes found for ort '
        write(3,*)'************************'
        return
      end if
c
c     Compute plane angles
c
      do 20 i=1,nplna
        h1=ortind(1,zeiga(i))
        k1=ortind(2,zeiga(i))
        l1=ortind(3,zeiga(i))
       da=(aort*aort*bort*bort*cort*cort)/(h1*h1*bort*bort*cort*
     *cort+k1*k1*cort*cort*aort*aort+l1*l1*aort*aort*bort*bort)
        da=sqrt(da)
        do 30 ii=1,nplnb
          h2=ortind(1,zeigb(ii))
          k2=ortind(2,zeigb(ii))
          l2=ortind(3,zeigb(ii))
        db=(aort*aort*bort*bort*cort*cort)/(h2*h2*bort*bort*cort*
     *cort+k2*k2*cort*cort*aort*aort+l2*l2*aort*aort*bort*bort)
          db=sqrt(db)
```

```
          u1=0
          v1=0
          w1=0
          nout=0
          do 40 j=1,nort
            h2=ort(j,1)
            k2=ort(j,2)
            l2=ort(j,3)
            if((h2.eq.0).and.(k2.eq.0).and.(l2.eq.0))goto 40
       d=(aort*aort*bort*bort*cort*cort)/(h2*h2*bort*bort*cort*
     *cort+k2*k2*cort*cort*aort*aort+l2*l2*aort*aort*bort*bort)
            d=sqrt(d)
            if((db.ge.(d-.001)).and.(db.le.(d+.001)))then
      cosab=(h1*h2/aort/aort+k1*k2/bort/bort+l1*l2/cort/cort)*
     *(da*db)
             if(cosab.ge.1)then
               cosab=.999999
             end if
             if(cosab.le.-1)then
               cosab=-.999999
             end if
             cosab=acos(cosab)
             if((cosab.ge.alpha1).and.(cosab.le.alpha2))then
c
c      Find the zone axis
c
               u=k1*l2-l1*k2
               v=l1*h2-h1*l2
               w=h1*k2-k1*h2
               if((u*u+v*v+w*w).le.0.01)goto 40
        if((abs((u1**2+v1**2+w1**2)-(u*u+v*v+w*w))).ge.0.01)then
c
c      Write report file
c
               print*,'****'
               print*,'Found ort:'
               print*,'Zone axis:   [',u,v,w,']'
               print*,'1st vector:  (',h1,k1,l1,')'
               print*,'2nd vector:  (',h2,k2,l2,')'
               write(3,*)'****'
               write(3,*)'Found ort:'
               write(3,*)'Zone axis:   [',u,v,w,']'
               write(3,*)'1st vector:  (',h1,k1,l1,')'
               write(3,*)'2nd vector:  (',h2,k2,l2,')'
c
               nout=nout+1
               out(nout,1)=ifix(u)
               out(nout,2)=ifix(v)
               out(nout,3)=ifix(w)
               nout=nout+1
               out(nout,1)=h1
               out(nout,2)=k1
               out(nout,3)=l1
               nout=nout+1
               out(nout,1)=h2
               out(nout,2)=k2
               out(nout,3)=l2
c
c      Find planes belonging to the zone [u,v,w]
c
               do 50 jj=1,nort
                 h3=ort(jj,1)
```

```
                  k3=ort(jj,2)
                  l3=ort(jj,3)
                  if((h3*h3+k3*k3+l3*l3).gt..01)then
                    if((abs(h3*u+k3*v+l3*w)).le.0.01)then
                      print*,'Next vector: (',h3,k3,l3,')'
                      write(3,*)'Next vector: (',h3,k3,l3,')'
c
                nout=nout+1
                out(nout,1)=h3
                out(nout,2)=k3
                out(nout,3)=l3
c
                     end if
                   end if
50              continue
c
c     Write DRAWDAT
c
                write(4,*)4
                write(4,*)nr
                write(4,*)l,lambda,aort,bort,cort
                write(4,*)nout
                do 55 jj=1,nout
                  write(4,*)(out(jj,jjj),jjj=1,3)
55              continue
                nout=0
c
               end if
                u1=u
                v1=v
                w1=w
              end if
            end if
40        continue
30      continue
20    continue
c
      return
      end
c
```

Program DRAW

```
C     The program DRAW uses the file DRAWDAT created by the program
C     DIFFRA and draws a diffraction pattern on a computer screen. This
C     program is software and hardware dependent as it will run only on
C     ATARI 1040ST computer under FORTRAN77.
C
      PROGRAM DRAW
C
      INTEGER AESRET,HANDLE,WKXYWH,DESKX,DESKY,DESKW,DESKH
      INTEGER WX,WY,WW,WH,DESKHA,MYWIND,WFNAME,NAMEP
C
C     Clip rectangle:
      INTEGER*2 CLIPR(0:3)
C
C     Variables for calculating pattern points:
      REAL A, B, XSCALE, YSCALE
C
C     VDI parameter arrays:
      INTEGER*2 POINTS(0:256),WORKIN(0:10),WORKOU(0:56)
C
C     Variables for computation
      INTEGER OUT(200,3),NOUT,NR,TYPE,I,HKL1(3),HKL2(3),UVW(3)
      REAL L,LAMBDA,ADIR,BDIR,CDIR,D11,D22,RPHI(200,2)
C
C     Scale factors
      XSCALE=.321875
      YSCALE=.3075
C
C     Read data from DROWDAT
      OPEN(4,FILE='B:DRAWDAT',STATUS='OLD')
13    READ(4,*,END=11)TYPE
      READ(4,*,END=11)NR
C
C     Type of the crystal system:
C     1=fcc, 2=bcc, 3=hcp, 4=ort
C
      IF(TYPE.EQ.3)THEN
        READ(4,*,END=11)L,LAMBDA,ADIR,CDIR
      END IF
      IF(TYPE.EQ.4)THEN
        READ(4,*,END=11)L,LAMBDA,ADIR,BDIR,CDIR
      END IF
      IF((TYPE.EQ.1).OR.(TYPE.EQ.2))THEN
        READ(4,*,END=11)L,LAMBDA,ADIR
      END IF
C
C     Read zone axis [u,v,w]
C
      READ(4,*,END=11)NOUT
      READ(4,*,END=11)(UVW(I),I=1,3)
C
C     Read plane indices (h,k,l)
c
      DO 10 I=1,NOUT-1
        READ(4,*,END=11)(OUT(I,J),J=1,3)
10    CONTINUE
C
      GOTO 14
11    PRINT*,'END OF FILE REACHED!'
C
```

```
C     Computation of the R and PHI coordinates
C
14    DO 20 I=1,3
        HKL1(I)=OUT(1,I)
20    CONTINUE
C
C     Hexagonal system
c
      IF(TYPE.EQ.3)THEN
C
        CDIR=(CDIR/ADIR)**2
        CDIR=(CDIR*2)/3
        D1=HKL1(1)**2+HKL1(2)**2+(HKL1(1)+HKL1(2))**2+
     +(1/CDIR)*HKL1(3)**2
        D11=D1*2/3/ADIR**2
        D11=SQRT(D11)
        D1=SQRT(D1)
        RPHI(1,1)=L*LAMBDA*D11
        RPHI(1,2)=0
C
        DO 30 I=2,NOUT-1
          DO 40 J=1,3
            HKL2(J)=OUT(I,J)
40        CONTINUE
C
          D2=HKL2(1)**2+HKL2(2)**2+(HKL2(1)+HKL2(2))**2+
     +(1/CDIR)*HKL2(3)**2
          D22=D2*2/3/ADIR**2
          D22=SQRT(D22)
          D2=SQRT(D2)
C
          RPHI(I,1)=L*LAMBDA*D22
          RPHI(I,2)=(HKL1(1)*HKL2(1)+HKL1(2)*HKL2(2)+HKL1(3)*
     +HKL2(3)*(1/CDIR)+(HKL1(1)+HKL1(2))*(HKL2(1)+HKL2(2)))/D1/D2
          IF(RPHI(I,2).GE.1.0)THEN
            RPHI(I,2)=.999999
          END IF
          IF(RPHI(I,2).LE.-1.0)THEN
            RPHI(I,2)=-.999999
          END IF
          RPHI(I,2)=ACOS(RPHI(I,2))
30      CONTINUE
      END IF
C
C     Cubic system
C
      IF((TYPE.EQ.1).OR.(TYPE.EQ.2))THEN
C
        D1=HKL1(1)**2+HKL1(2)**2+HKL1(3)**2
        D11=D1/ADIR**2
        D11=SQRT(D11)
        D1=SQRT(D1)
        RPHI(1,1)=L*LAMBDA*D11
        RPHI(1,2)=0
C
        DO 50 I=2,NOUT-1
          DO 60 J=1,3
            HKL2(J)=OUT(I,J)
60        CONTINUE
C
          D2=HKL2(1)**2+HKL2(2)**2+HKL2(3)**2
          D22=D2/ADIR**2
```

```
          D22=SQRT(D22)
          D2=SQRT(D2)
C
          RPHI(I,1)=L*LAMBDA*D22
          RPHI(I,2)=(HKL1(1)*HKL2(1)+HKL1(2)*HKL2(2)+HKL1(3)*
     +HKL2(3))/D1/D2
          IF(RPHI(I,2).GE.1.0)THEN
           RPHI(I,2)=0.999999
          END IF
          IF(RPHI(I,2).LE.-1.0)THEN
            RPHI(I,2)=-.999999
          END IF
          RPHI(I,2)=ACOS(RPHI(I,2))
50      CONTINUE
       END IF
C
C      Orthorhombic system
C
       IF(TYPE.EQ.4)THEN
C
         D1=(HKL1(1)**2)/ADIR/ADIR+(HKL1(2)**2)/BDIR/BDIR+
     *(HKL1(3)**2)/CDIR/CDIR
         D11=D1
         D11=SQRT(D11)
         D1=SQRT(D1)
         RPHI(1,1)=L*LAMBDA*D11
         RPHI(1,2)=0
C
         DO 90 I=2,NOUT-1
           DO 100 J=1,3
             HKL2(J)=OUT(I,J)
100        CONTINUE
C
           D2=(HKL2(1)**2)/ADIR/ADIR+(HKL2(3)**2)/CDIR/CDIR+
     *(HKL2(2)**2)/BDIR/BDIR
           D22=D2
           D22=SQRT(D22)
           D2=SQRT(D2)
C
           RPHI(I,1)=L*LAMBDA*D22
           RPHI(I,2)=(HKL1(1)*HKL2(1)/ADIR/ADIR+HKL1(3)*
     +HKL2(3)/CDIR/CDIR+HKL1(2)*HKL2(2)/BDIR/BDIR)/(D1*D2)
           IF(RPHI(I,2).GE.1.0)THEN
             RPHI(I,2)=.999999
           END IF
           IF(RPHI(I,2).LE.-1.0)THEN
             RPHI(I,2)=-.999999
           END IF
           RPHI(I,2)=ACOS(RPHI(I,2))
90       CONTINUE
       END IF
C
C      Calculate X,Y-coordinates
C
       DO 70 I=0,256
         POINTS(I)=0
70     CONTINUE
       IF(NOUT.GT.62)THEN
         NOUT=62
       END IF
       J=1
       DO 80 I=1,NOUT-1
```

```
        POINTS(I+J)=NINT((RPHI(I,1)/XSCALE)*COS(RPHI(I,2)))
        POINTS(I+1+J)=NINT((RPHI(I,1)*SIN(RPHI(I,2))/YSCALE))
C
C     Second set of coordinates because of centre of symmetry
C
        POINTS(I+2+J)=-POINTS(I+J)
        POINTS(I+3+J)=-POINTS(I+1+J)
        J=J+3
80    CONTINUE
C
C     Write data on screen
C
      PRINT*,'TYPE=',TYPE
      PRINT*,'PHOTO NR.=',NR
      PRINT*,'ZONE :'
      WRITE(*,21)(UVW(I),I=1,3)
21    FORMAT(3I3)
      PRINT*,'1ST HKL :'
      WRITE(*,22)(HKL1(I),I=1,3)
22    FORMAT(3I3)
C
C     Initiate graphics application
      CALL APPLIN
C
C     Open virtual workstation
      HANDLE=0
      DO 1 I=0,9
        WORKIN(I)=1
1     CONTINUE
      WORKIN(7) = 0
      WORKIN(10) = 2
      WORKIN(3)=3
      CALL VOPNVW(WORKIN, HANDLE, WORKOU)
C
C     Determine desktop window work area size
      WKXYWH = 4
      DESKHA = 0
      CALL WINDGE(DESKHA, WKXYWH, DESKX, DESKY, DESKW, DESKH)
C
C     Determine my window work area size
      WW = DESKW/2
      WH = DESKH
      WX = DESKX+WW
      WY = DESKY
C
C     Create window half size
      CALL WINDCR(1, WX, WY, WW, WH)
      MYWIND = AESRET()
C
C     Open the window
      CALL WINDOP(MYWIND, WX, WY, WW, WH)
C
C     Set clip rectangle to work area of window
      CALL WINDGE(MYWIND, WKXYWH, WX, WY, WW, WH)
      CLIPR(0) = WX
      CLIPR(1) = WY
      CLIPR(2) = WX+WW
      CLIPR(3) = WY+WH
      CALL VSCLIP(HANDLE, 1, CLIPR)
C
C     Hide mouse while drawing
      CALL GRAFMO(256, 0)
```

```
C
C     Fill rectangle with background (white)
      CALL VRRECF(HANDLE, CLIPR)
C
C     Calculate coordinates
      A=WX+WW/2
      B=WY+WH/2
      DO 5 I=0,254,2
      POINTS(I)=POINTS(I)+A
5     POINTS(I+1)=POINTS(I+1)+B
C
C     Draw the pattern
      CALL VPMARK(HANDLE, 127, POINTS)
C
C     Wait until keyboard
      CALL EVNTKE
C
C     Show mouse after update
      CALL GRAFMO(0,0)
      CALL GRAFMO(257,0)
C
C     Close and delete window
      CALL WINDCL(MYWIND)
      CALL WINDDE(MYWIND)
      CALL VCLRWK(HANDLE)
      CALL VCLSVW(HANDLE)
C
C     Application exit
      CALL APPLEX
C
      PRINT*,'NEW PATTERN ?       (Y=1,N=0)'
      READ(*,*)I
      IF(I.EQ.1)GOTO 13
C
      STOP
      END
```