REFERENCES

1. Leissa, A.W., and Clausen, W.E., "Deflection of a Circular Plate Having Mixed Boundary Conditions," AIAA Journal, Vol. 5, pp. 2287-2288, 1967.

2. Conway, H.D., and Farnham, K.A., "Deflections of Uniformly Loaded Circular Plates with Combinations of Clamped, Simply Supported and Free Boundary Conditions," International Journal of Mechanical Science, Vol 9, pp. 661-671, 1967.

3. Stahl, B., and Keer, L.M., "Bending of a Uniformly Loaded Circular Plate with Mixed Boundary Conditions," AIAA Journal, Vol. 10, pp. 739-743, 1972.

4. Sriswasdi, V., "Bending of a Uniformly Loaded Annular Plate with Mixed Boundary Conditions," Master's Thesis, Department of Mechanical Engineering, King Mongkut's Institute of Technology, Thonburi Campus, 1985.

5. Jaswon, M.A., and Maiti, M., "An Integral Equation Formulation of Plate Bending Problems," Journal of Engineering Mathematics, Vol 2, pp. 83-93, 1968.

6. Maiti, M., and Chakrabarty, S.K., "Integral Equation Solutions for Simply Supported Polygonal Plates," International Journal of Engineering Science, Vol. 12, pp. 793-806, 1974.

7. Altiero, N.J., and Sikarskie, D.L., "A Boundary Integral Method Applied to Plates of Arbitrary Plan Form," Computers and Structures, Vol. 9, pp. 163-168, 1978.

8. Wu, B.C., and Altiero, N.J., "A Boundary Integral Method Applied to Plates of Arbitrary Plan Form and Arbitrary Boundary Conditions," Computers and Structures, Vol. 10, pp. 703-707, 1979.

9. Stern, M., "A Gerneral Boundary Integral Formulation for the Numerical Solution of Plate Bending Problem," International Journal of Solids and Structures, Vol. 15, pp. 769-782, 1979.

10. Tottenham, H., "The Boundary Element Method for Plates and shells, " Developments in Boundary Element Methods," pp. 173-205, Applied Science Publishers, London, 1979.

11. Bezine, G., "A Boundary Integral Equation Method for Plate Flexure with Conditions inside the Domain," International Journal for Numerical Methods in Engineering, Vol. 17, pp. 1647-1657, 1981.

12. Paris, F., and Leon, S.D., "Boundary Element Method Applied to the Analysis of Thin Plates," Computers and Structures, Vol. 25, pp. 225-233, 1987.

13. Timoshenko, S., and Woinowsky-Krieger, S., Theory of Plates and Shells, McGraw-Hill, Kogakusha, 2nd ed., 1959.

14. Oden, J.T., Mechanics of Elastic Structures, McGraw-Hill, New York, 1967.

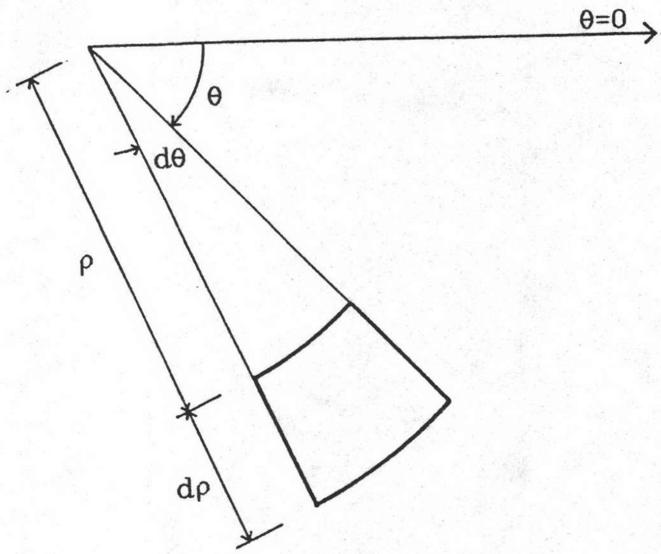
15. Brebbia, C.A., Telles, J.C.F., and Wrobel, L.C., Boundary Element Techniques, Springer-Verlag Berlin, Heidelberg, 1984.

16. Banerjee, P.K., and Butterfield, R., Boundary Element Methods in Engineering Science, McGraw-Hill, London, 1981.

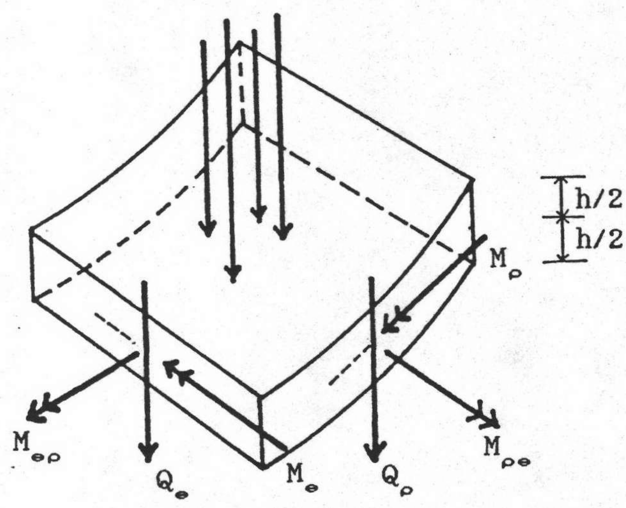
17. Kumar, K.S., "Boundary Integral Equations for Plate Bending," Master's Thesis, Division of Structural Engineering and Construction, Asian Institute of Technology, 1982.

18. Spiegel, M.R., Mathematical Handbook of Formulas and Tables, McGraw-Hill, New York, 1968.

19. Gradshteyn, I.S., and Ryzhik, I.M., Table of Integrals, Series, and Products, Translation edited by Alan Jeffrey, New York, 4th ed., 1965.

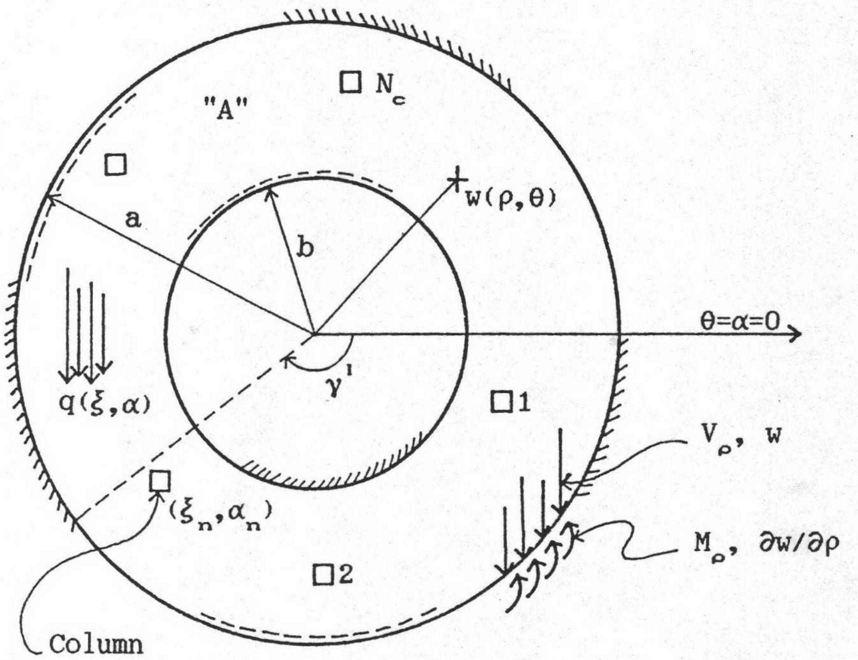


a) ELEMENT OF PLATE

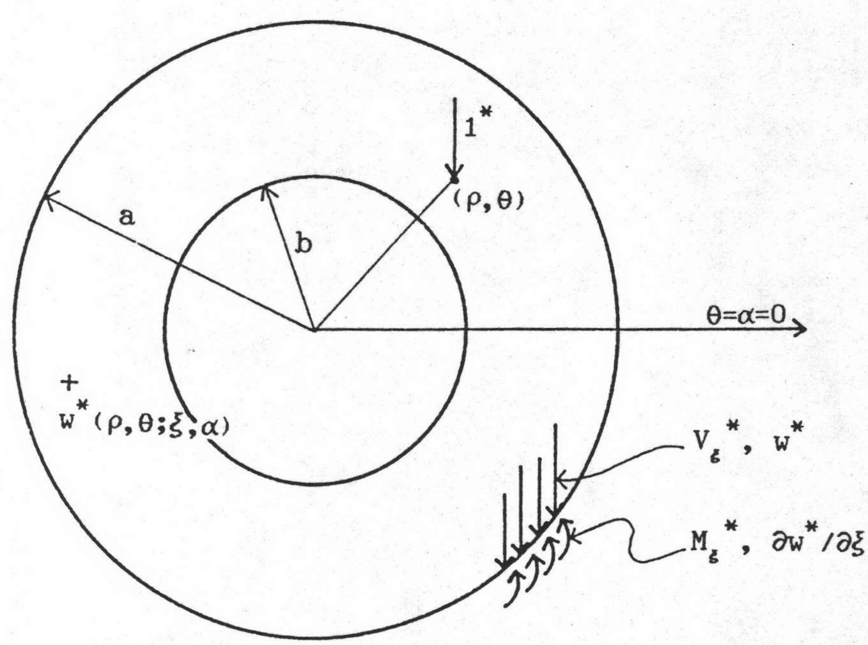


b) STRESS RESULTANTS

Fig. 1 Stress resultants in polar co-ordinate.

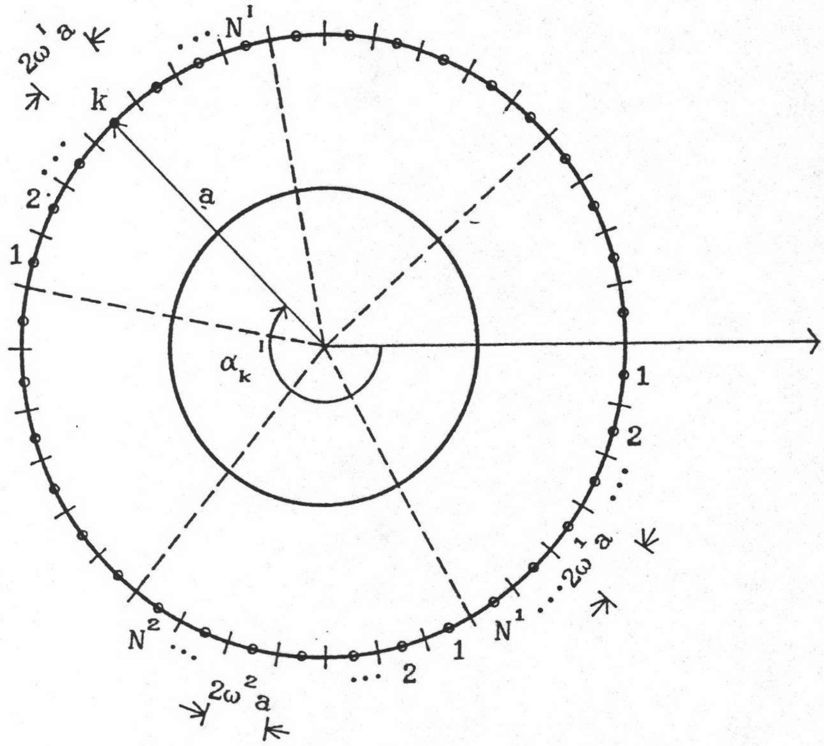


a) REAL SYSTEM

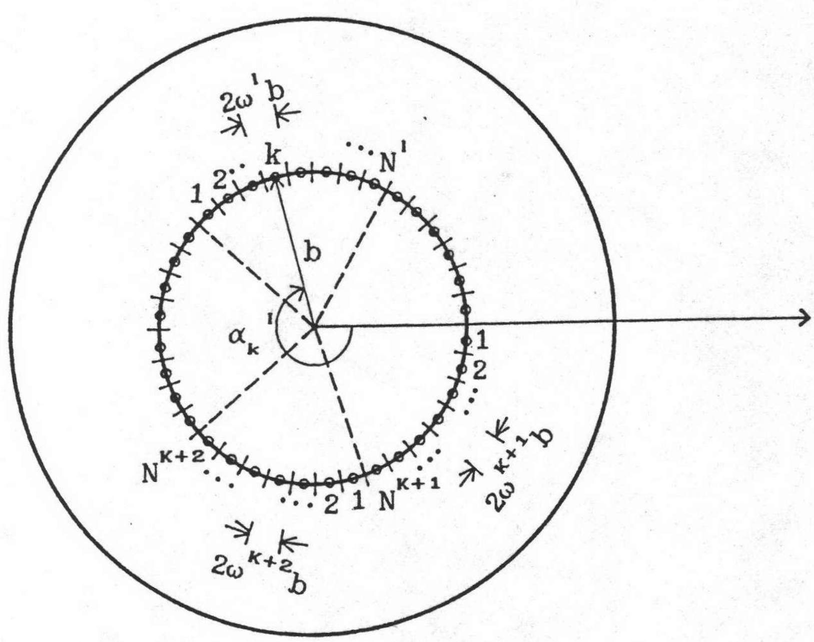


b) VIRTUAL SYSTEM

Fig. 2 Force and displacement systems in Betti's reciprocal theorem.



a) OUTER BOUNDARY



b) INNER BOUNDARY

Fig. 3 Subdivision of the boundary sections.

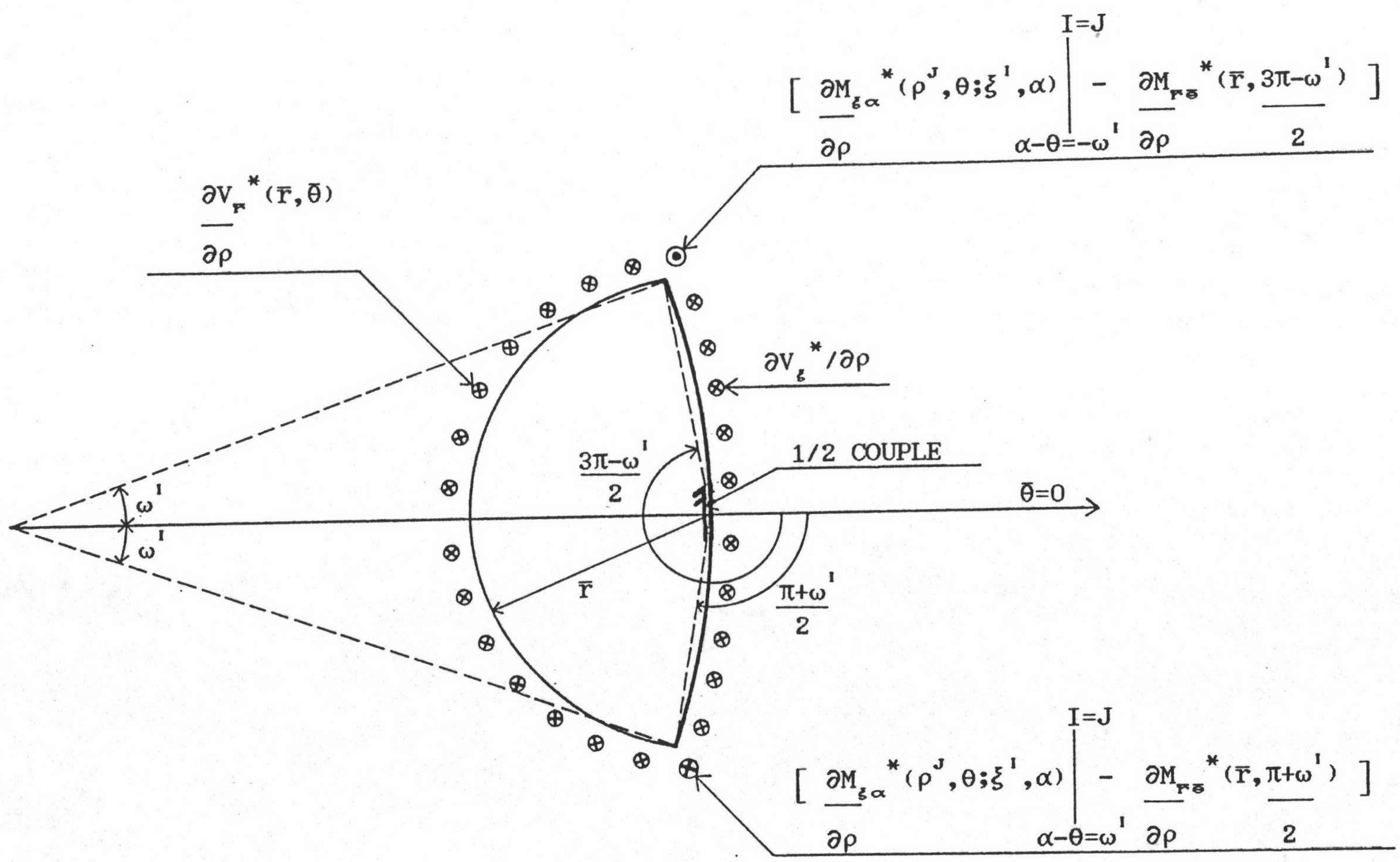
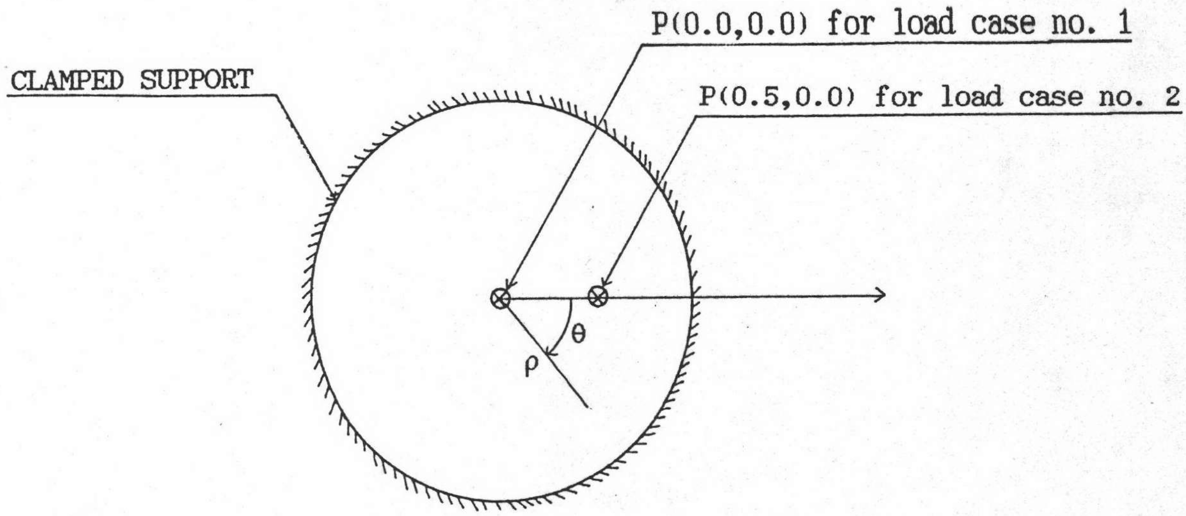
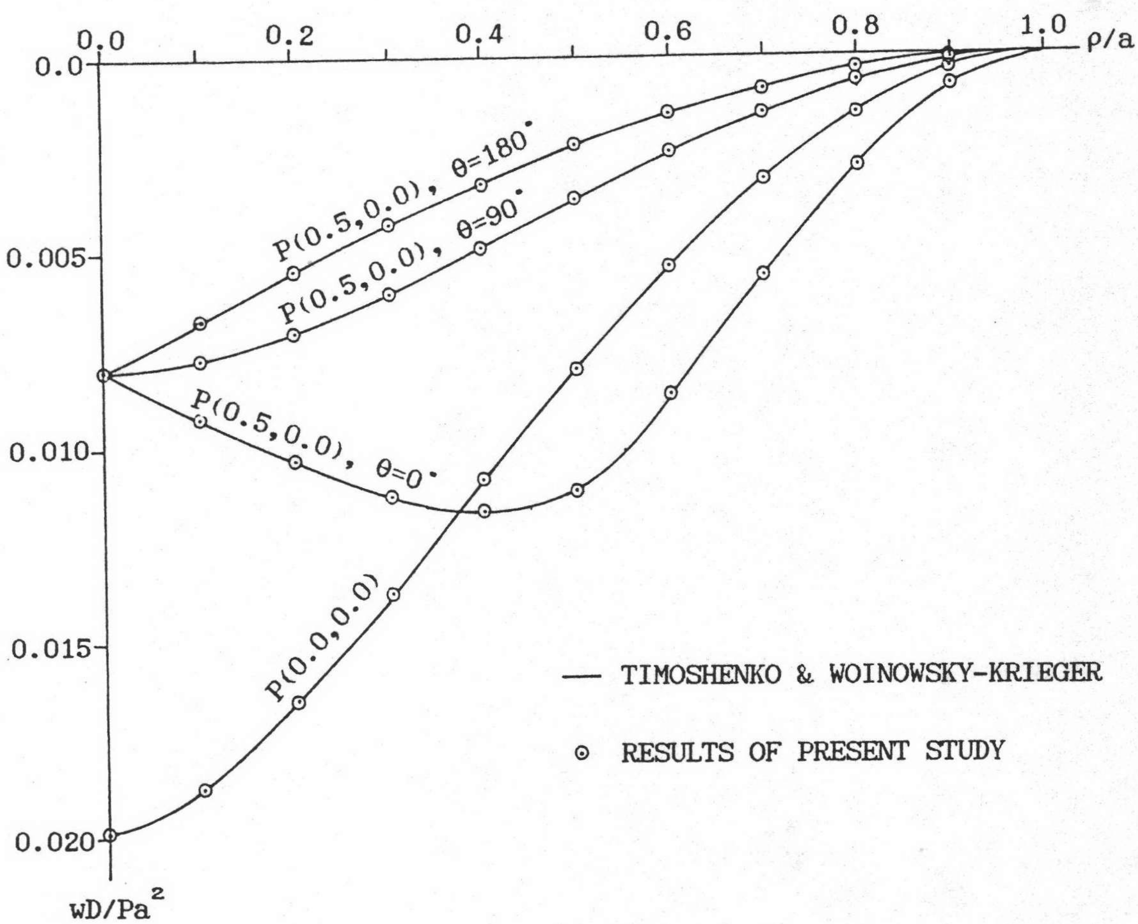


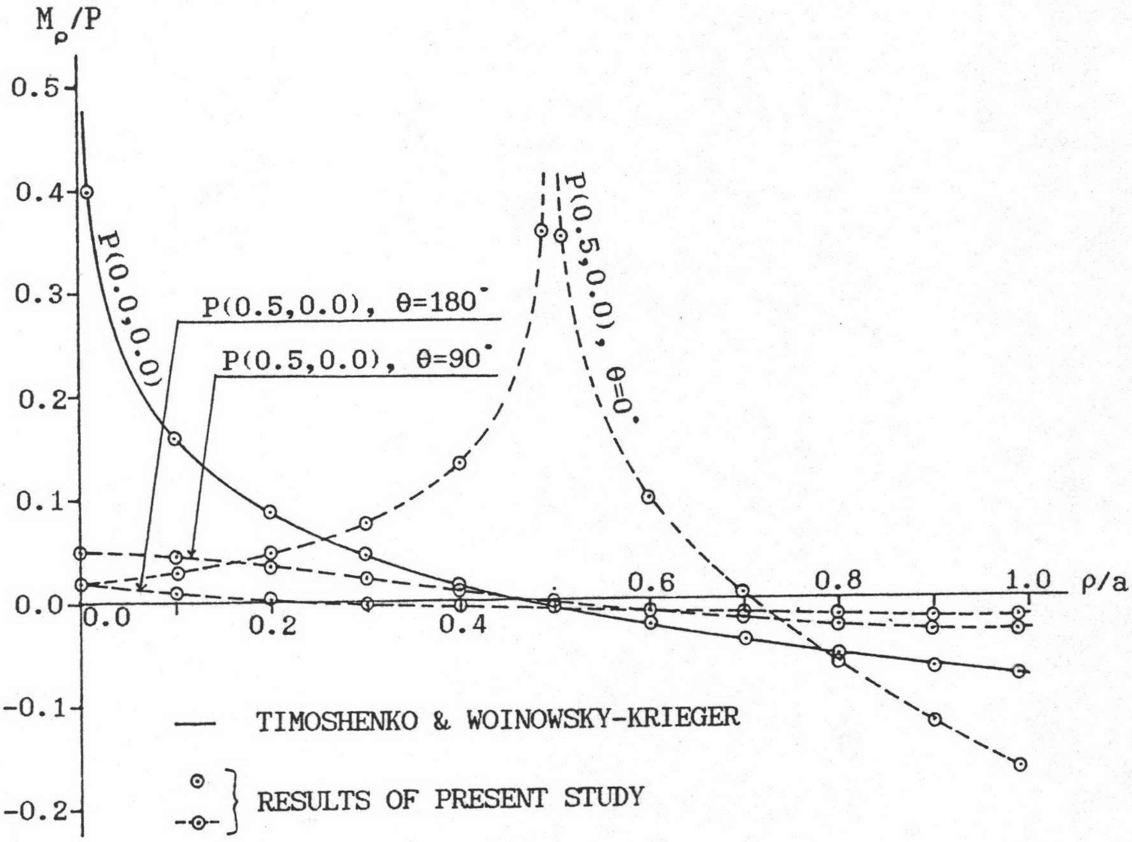
Fig. 4 Physical interpretation of the improper integral of $\partial V_\xi^*/\partial\rho$.



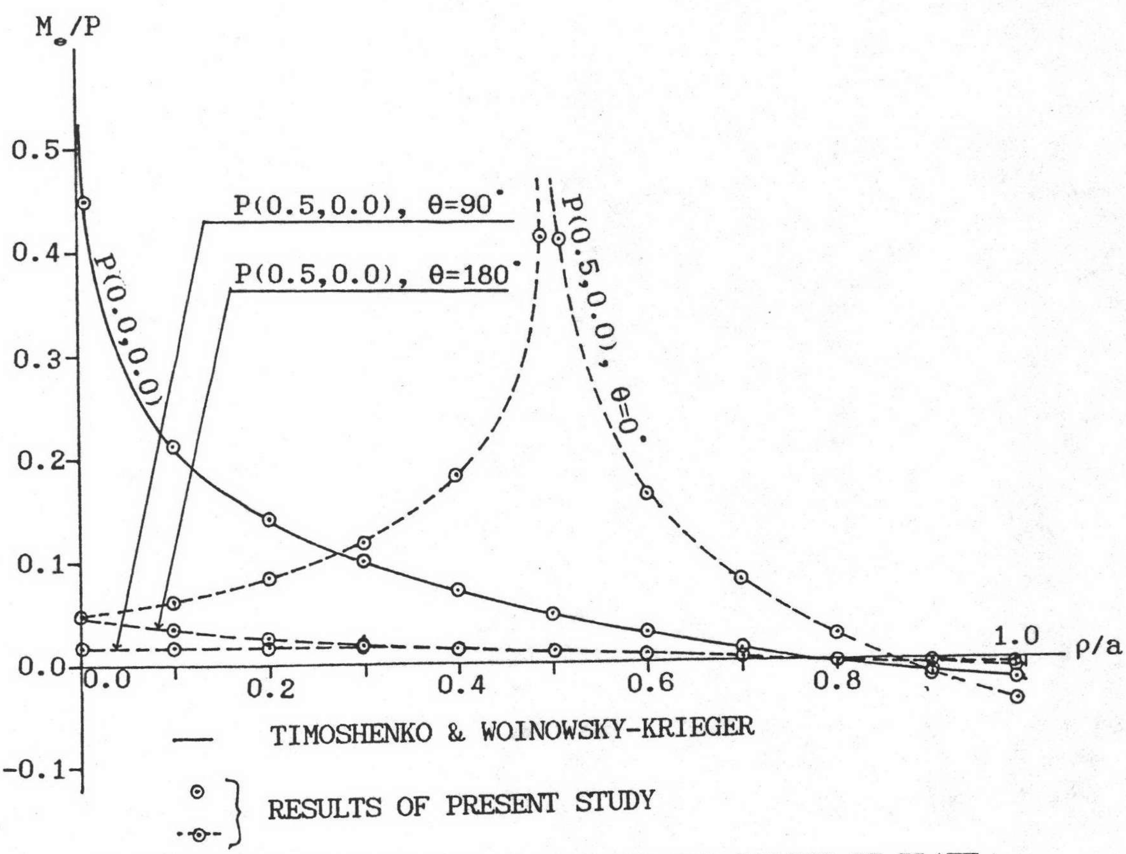
a) GEOMETRY (ν = 0.3)



b) DEFLECTIONS ALONG THE RADIUS OF PLATE



c) NORMAL BENDING MOMENTS ALONG THE RADIUS OF PLATE



d) TRANSVERSE BENDING MOMENTS ALONG THE RADIUS OF PLATE

Fig. 5 Clamped circular plate under a singular load.

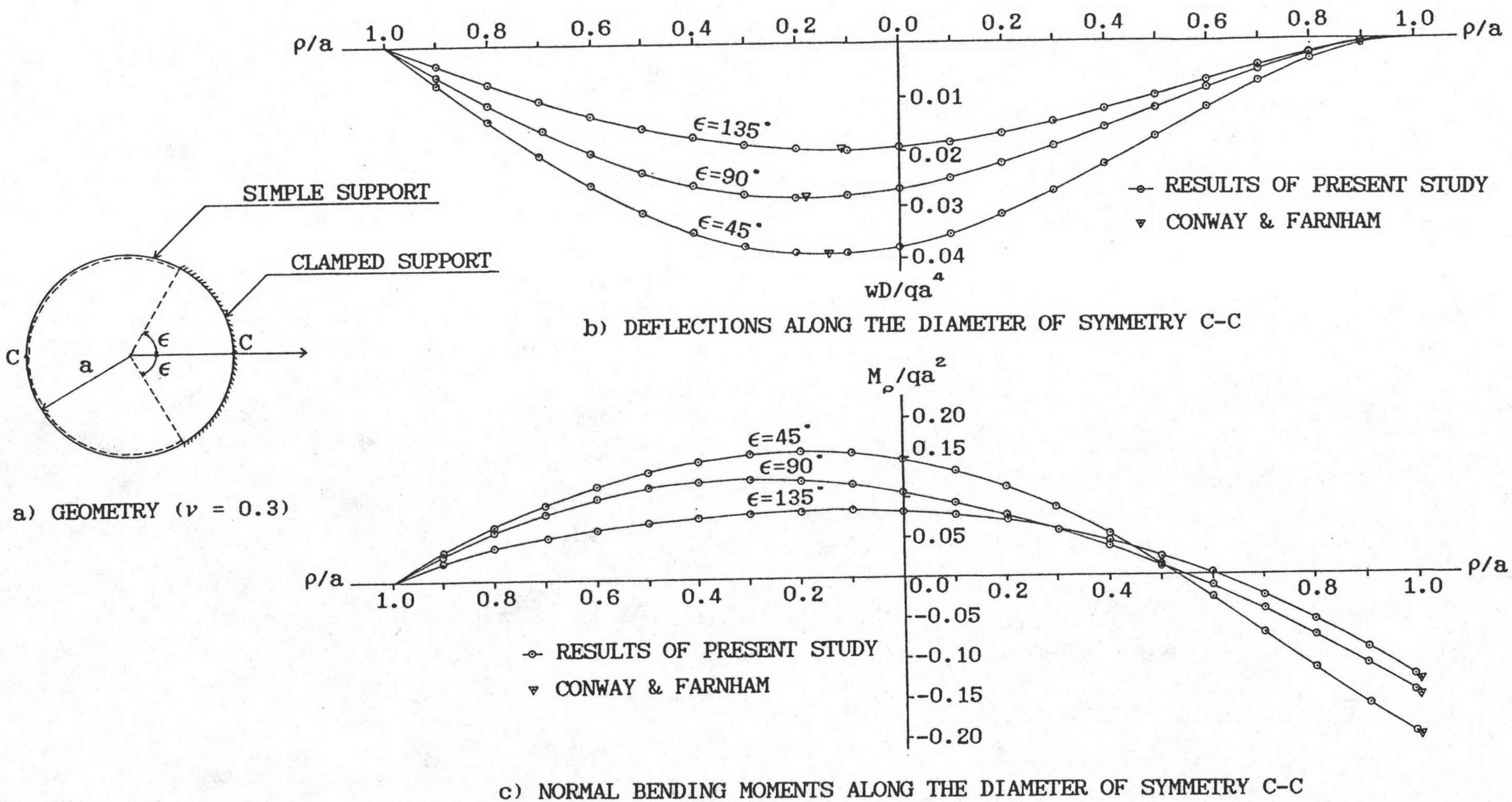
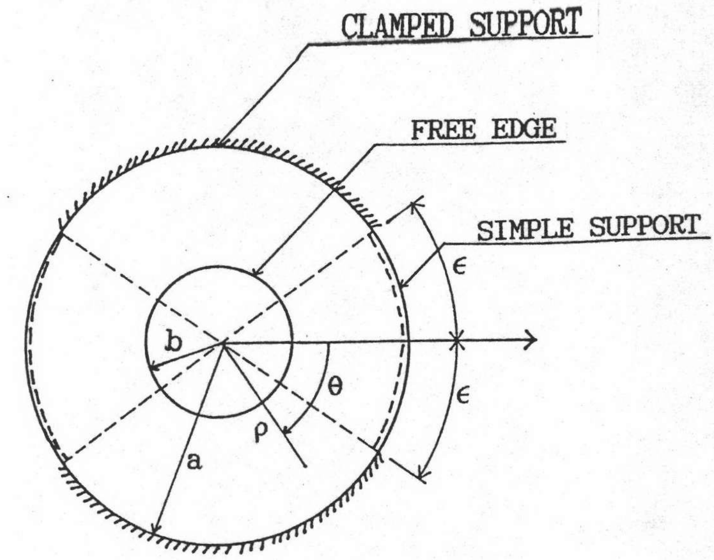
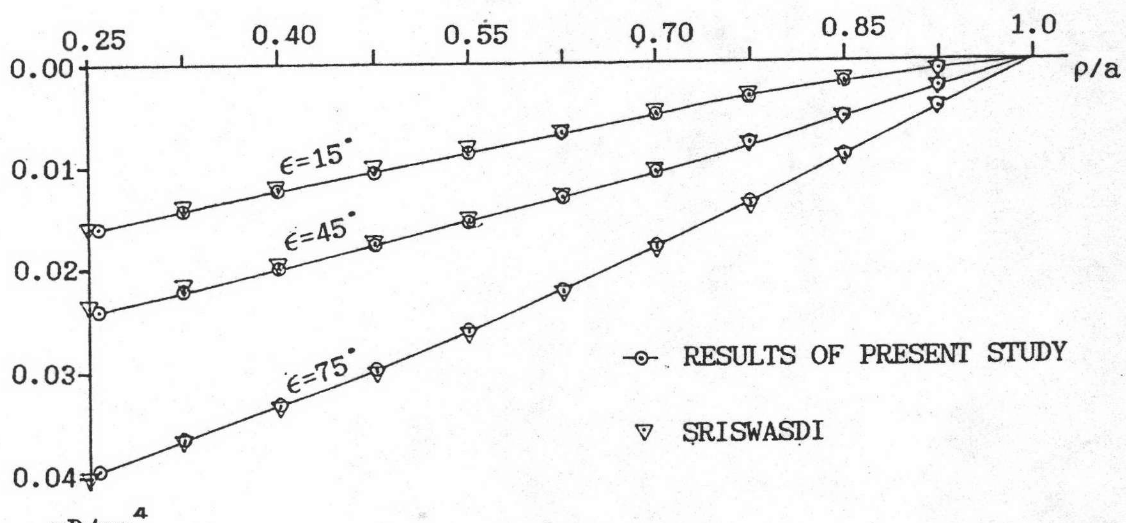


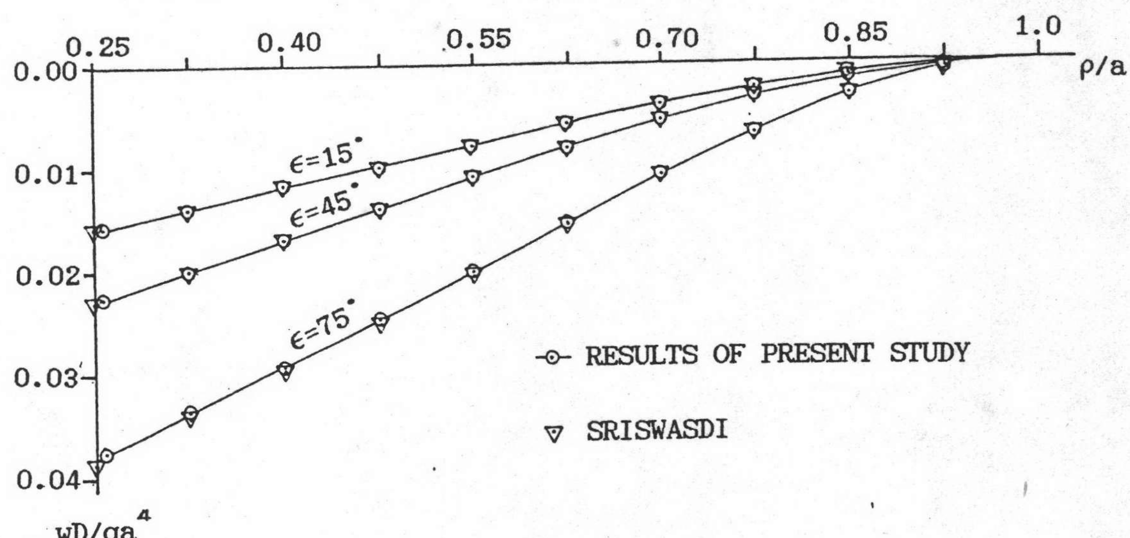
Fig. 6 Uniformly loaded circular plate with mixed simple and clamped supports.



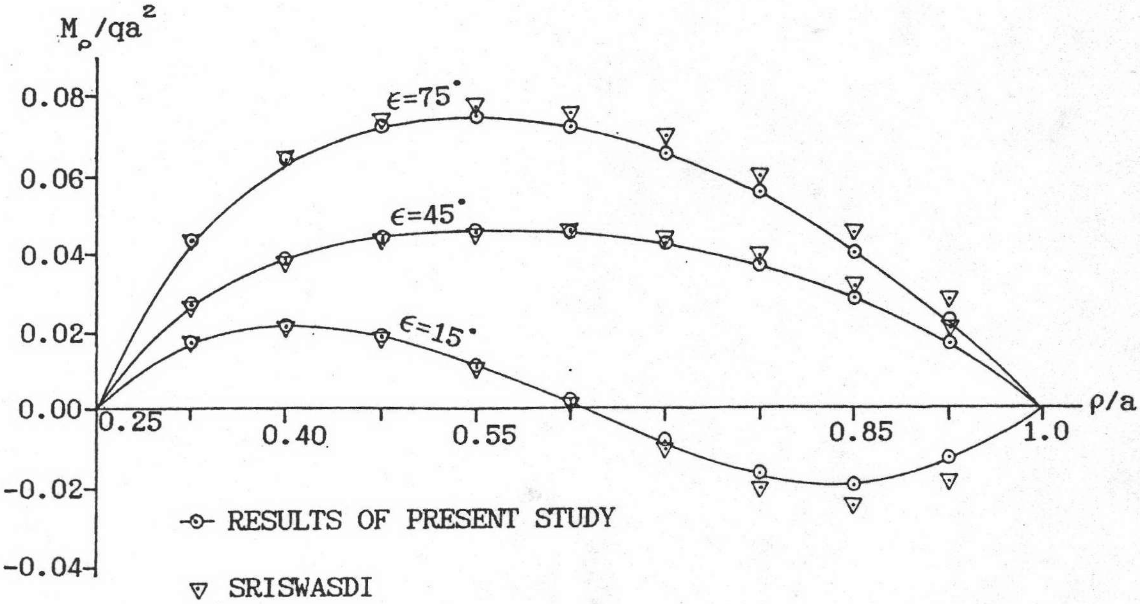
a) GEOMETRY (ν = 0.3, b/a=0.25)



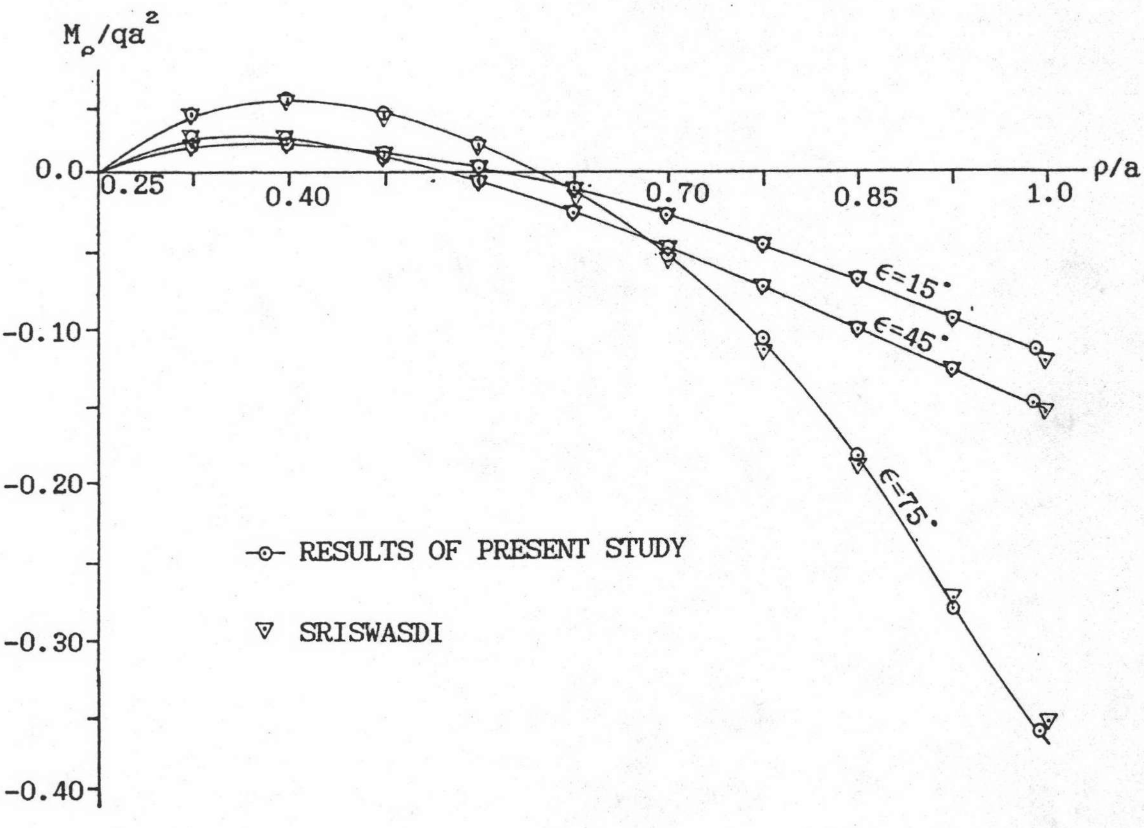
b) DEFLECTIONS ALONG THE RADIUS θ = 5°



c) DEFLECTIONS ALONG THE RADIUS θ = 85°

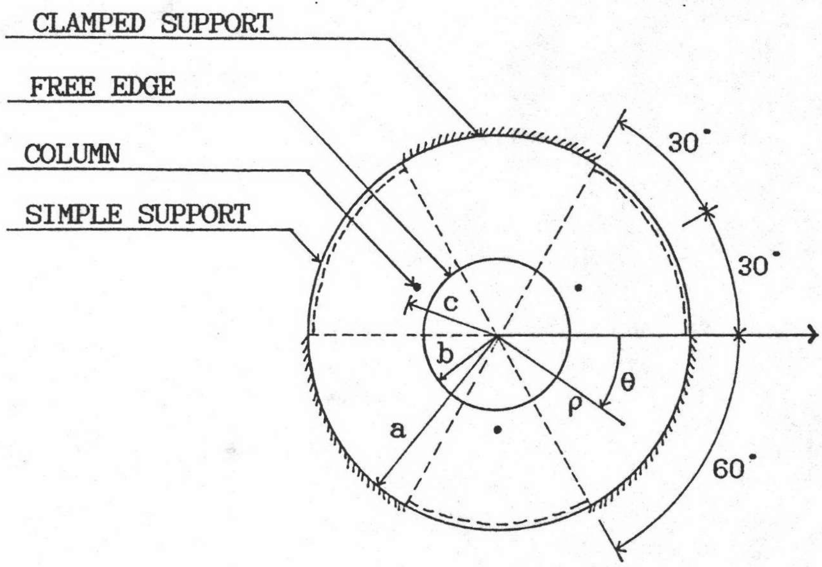


d) NORMAL BENDING MOMENTS ALONG THE RADIUS θ = 5°

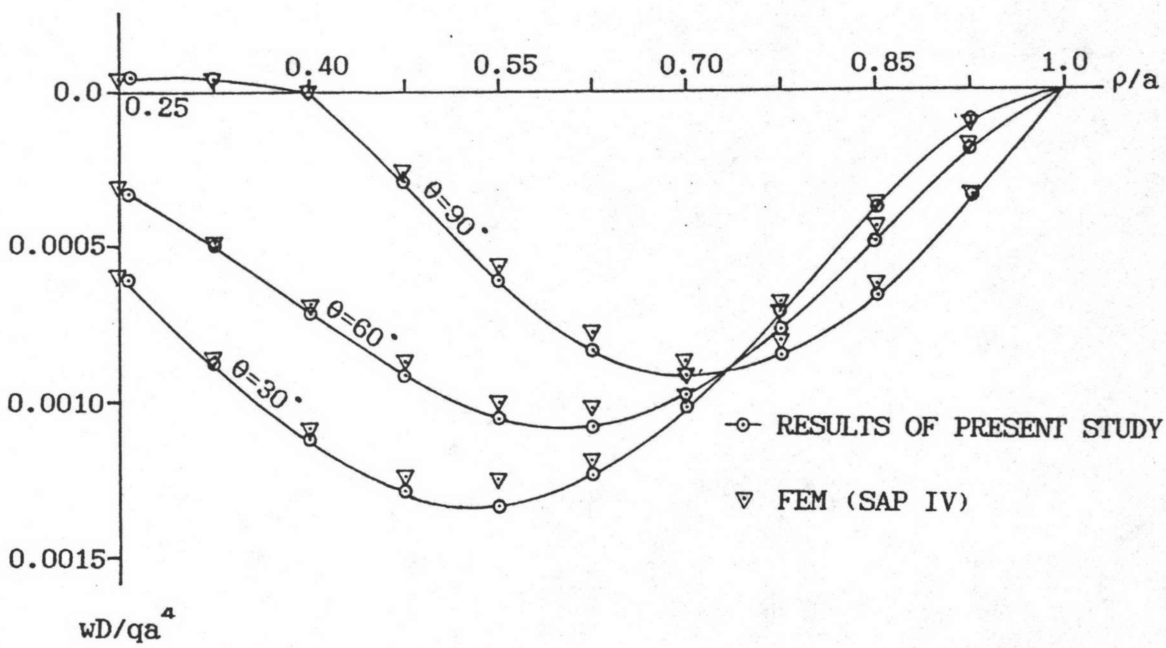


e) NORMAL BENDING MOMENTS ALONG THE RADIUS θ = 85°

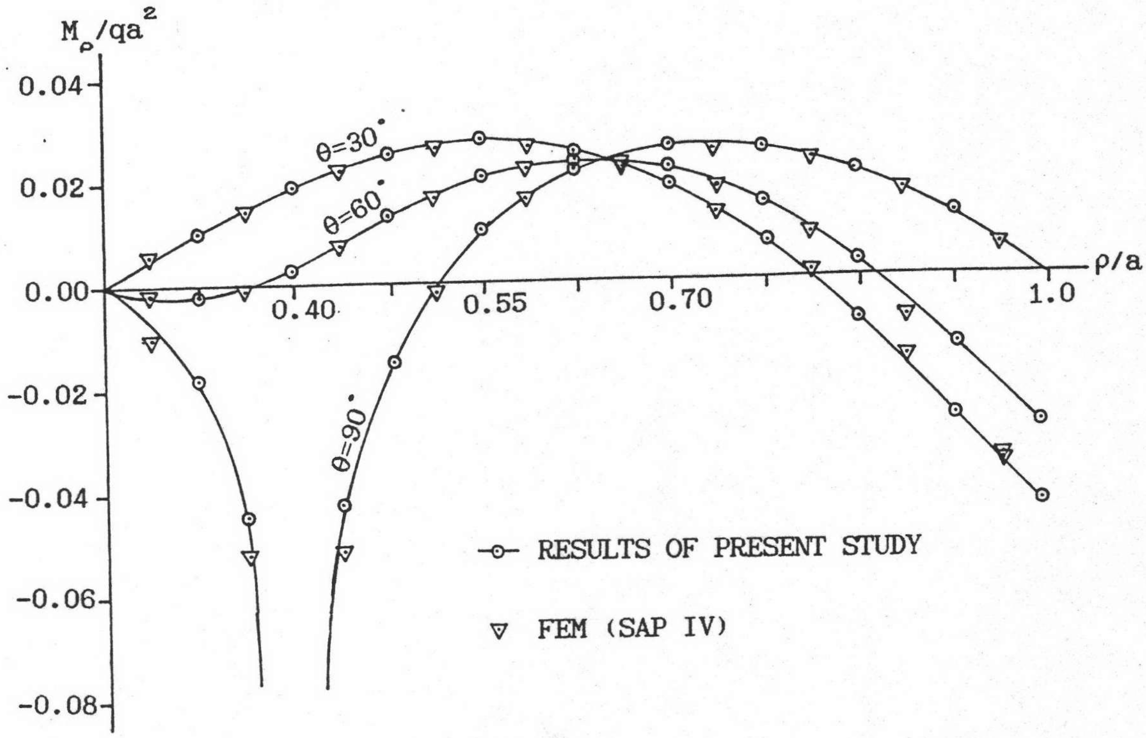
Fig. 7 Uniformly loaded annular plate with mixed free simple and clamped supports.



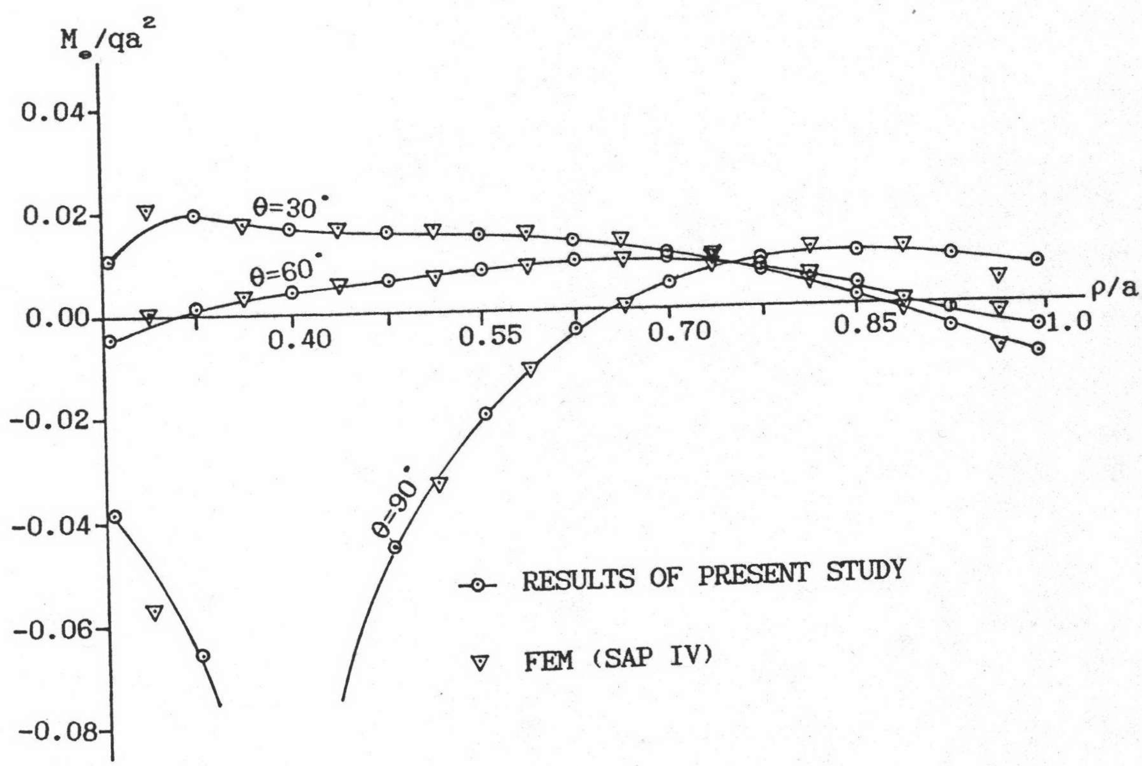
a) GEOMETRY (ν = 0.3, b/a=0.25, c/a=0.4)



b) DEFLECTIONS ALONG THE RADIUS θ = 30°, 60° AND 90°



c) NORMAL BENDING MOMENTS ALONG THE RADIUS θ = 30°, 60° AND 90°



d) TRANSVERSE BENDING MOMENTS ALONG THE RADIUS θ = 30°, 60° AND 90°

Fig. 8 Uniformly loaded annular plate with interior columns.

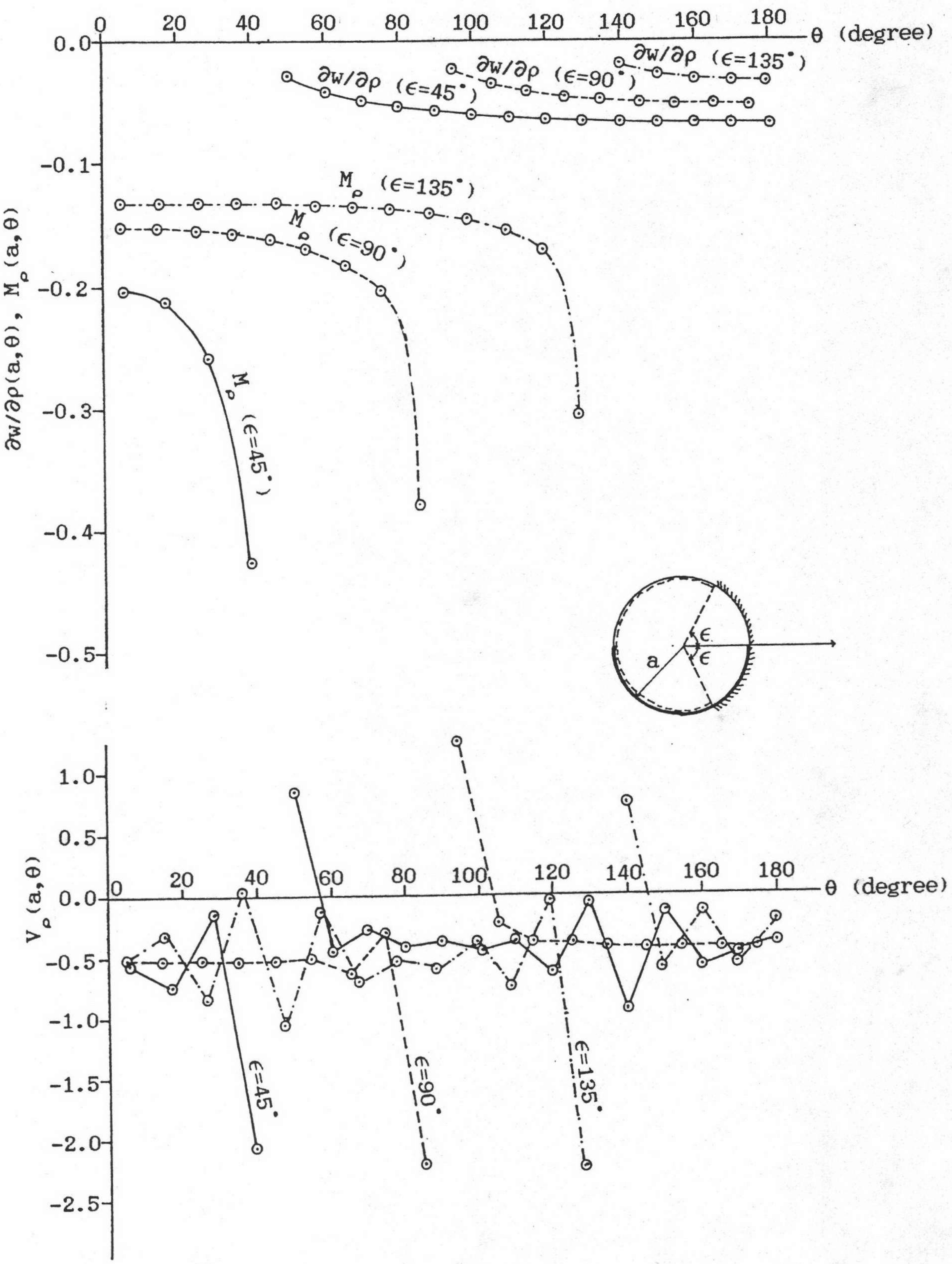


Fig. 9 Unknown values of ∂w/∂ρ, M_ρ and V_ρ along a half edge for circular plate in Example 2.

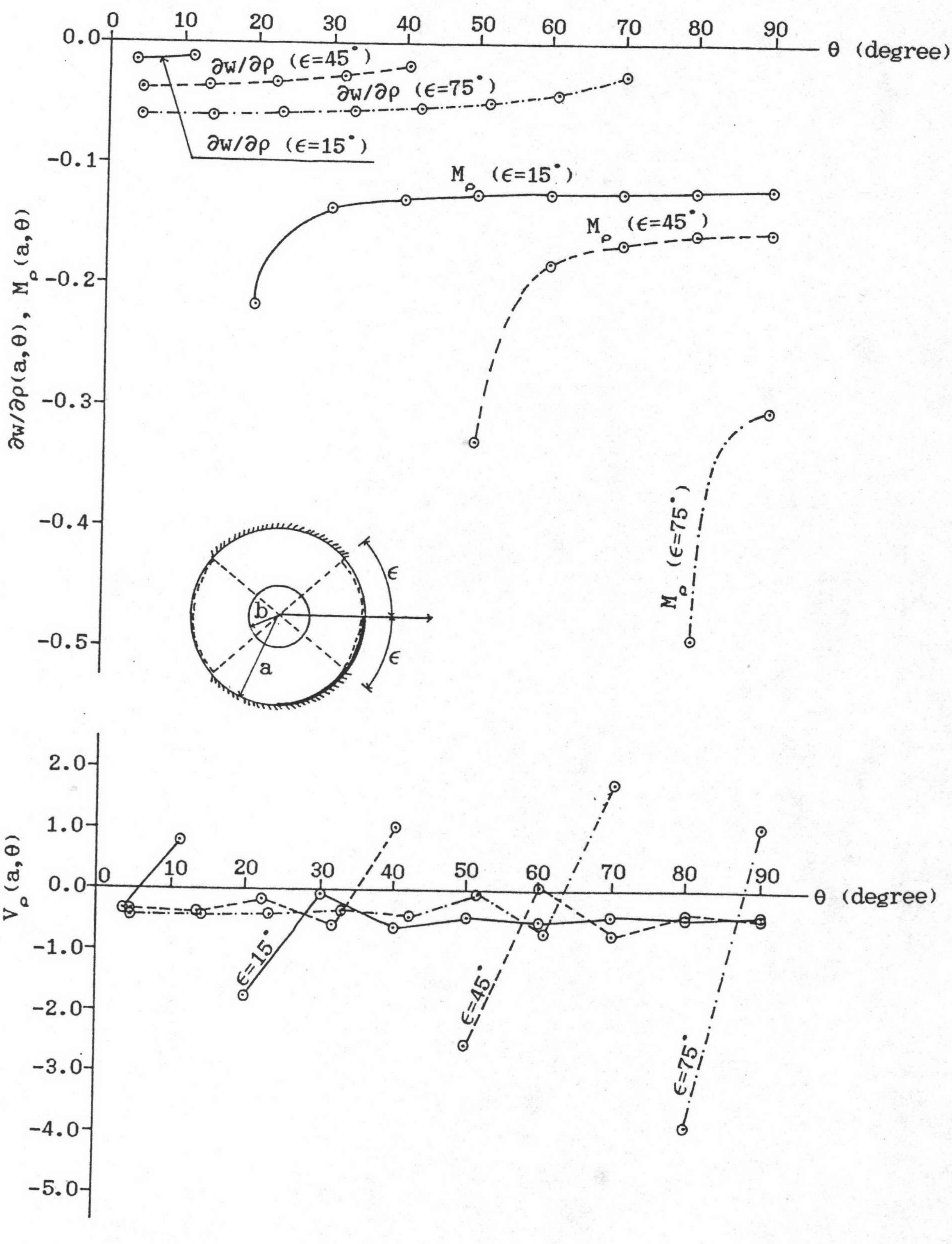


Fig. 10 Unknown values of $\partial w/\partial \rho$, M_ρ and V_ρ along a quarter of the outer edge for annular plate in Example 3.

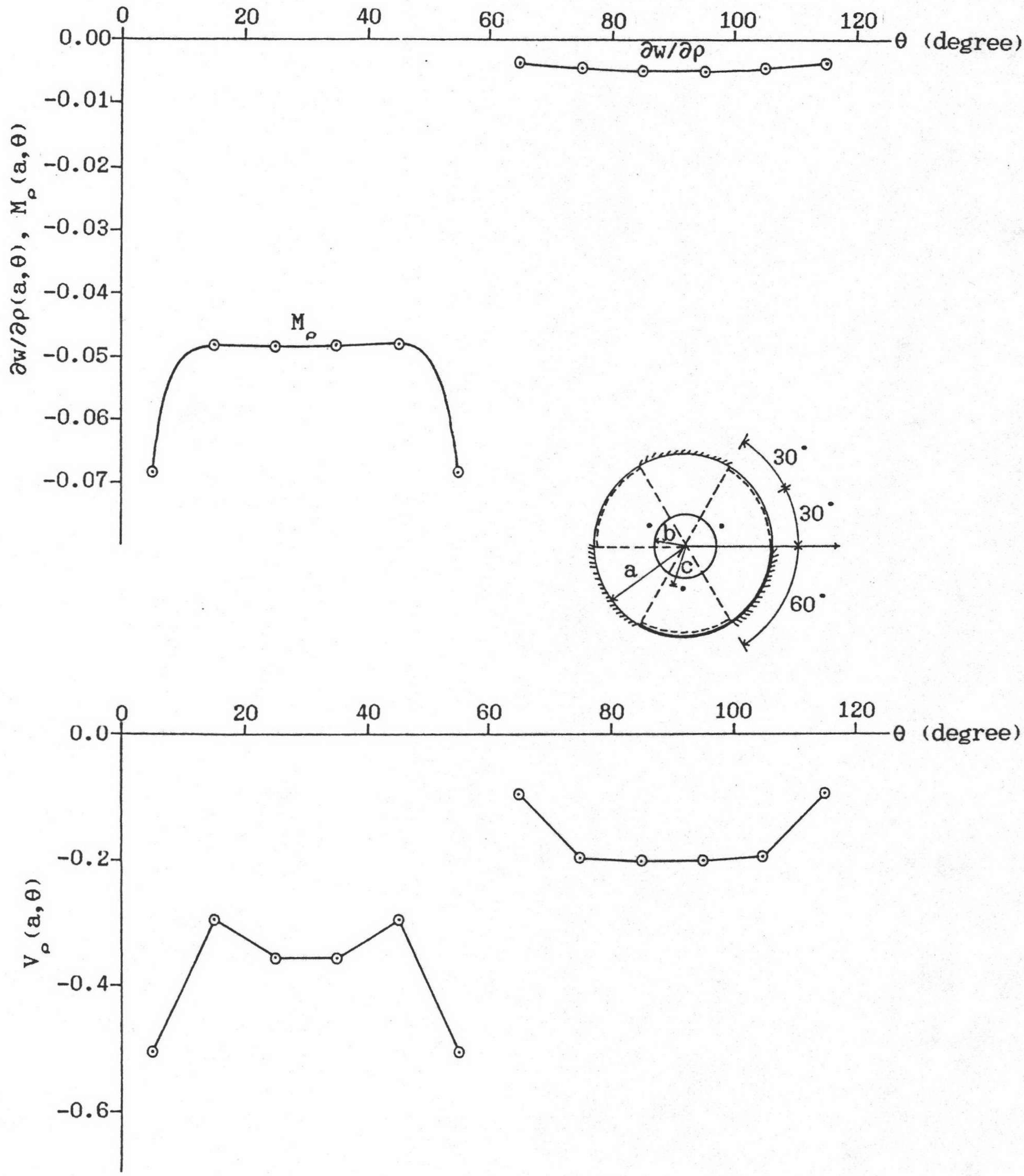


Fig. 11 Unknown values of $\partial w/\partial\rho$, M_ρ and V_ρ along a third of the outer edge for annular plate in Example 4.

APPENDIX

APPENDIX A

Influence Functions of the Virtual System

$$w^*(\rho,\theta;\xi,\alpha) = \frac{r^2 \ln r.}{8\pi D}$$

$$\frac{\partial w^*(\rho,\theta;\xi,\alpha)}{\partial \rho} = \frac{1}{8\pi D}(1+2\ln r)\{\rho-\xi\cos(\alpha-\theta)\}.$$

$$\frac{\partial w^*(\rho,\theta;\xi,\alpha)}{\partial \xi} = \frac{1}{8\pi D}(1+2\ln r)\{\xi-\rho\cos(\alpha-\theta)\}.$$

$$\frac{\partial^2 w^*(\rho,\theta;\xi,\alpha)}{\partial \xi \partial \rho} = \frac{1}{8\pi D}\Big[-(1+2\ln r)\cos(\alpha-\theta)$$

$$+ \frac{2}{r^2}\{\rho-\xi\cos(\alpha-\theta)\}\{\xi-\rho\cos(\alpha-\theta)\} \Big].$$

$$M_\xi^*(\rho,\theta;\xi,\alpha) = \frac{-1}{8\pi}\Big[(1+\nu)(1+2\ln r) + \frac{2}{r^2}\{\xi-\rho\cos(\alpha-\theta)\}^2$$

$$+ \frac{2\nu\rho^2\sin^2(\alpha-\theta)}{r^2} \Big].$$

$$\frac{\partial M_\xi^*(\rho,\theta;\xi,\alpha)}{\partial\rho} = \frac{-1}{2\pi r^4}\Big[\{\rho-\xi\cos(\alpha-\theta)\}\Big\{ \frac{r^2}{2}(1+\nu) - \{\xi-\rho\cos(\alpha-\theta)\}^2 \Big\}$$

$$+ \{\xi-\rho\cos(\alpha-\theta)\}\Big\{ \nu\rho\xi\sin^2(\alpha-\theta) - r^2\cos(\alpha-\theta) \Big\} \Big].$$

$$V_\xi^*(\rho,\theta;\xi,\alpha) = \frac{-1}{2\pi r^2}\Big[\{\xi-\rho\cos(\alpha-\theta)\}\Big\{ 1 + \frac{(1-\nu)\rho\cos(\alpha-\theta)}{2\xi} \Big\}$$

$$+ \frac{(1-\nu)\rho^2\{\rho^2-\xi^2\}\sin^2(\alpha-\theta)}{2\xi r^2} \Big].$$

$$\frac{\partial V_\xi^*(\rho,\theta;\xi,\alpha)}{\partial\rho} = \frac{1}{2\pi r^4}\Big[r^2\cos(\alpha-\theta) + 2\{\rho-\xi\cos(\alpha-\theta)\}\{\xi-\rho\cos(\alpha-\theta)\}$$

$$+ \rho(1-\nu)\{\xi-\rho\cos(\alpha-\theta)\}\Big\{ \cos^2(\alpha-\theta) - \frac{2\rho^2\sin^2(\alpha-\theta)}{r^2} \Big\}$$

$$+ \frac{(1-\nu)\{\rho^2-\xi^2\}}{2}\Big\{ \cos(\alpha-\theta) - \frac{2\rho\xi\sin^2(\alpha-\theta)}{r^2} \Big\} \Big].$$

APPENDIX B

Integrals of Influence Functions

For (ρ^J,θ_1^J) and (ξ^I,α_k^I) are coincident on the boundary:

$$\left[\int_{\alpha_k^I-\omega^I}^{\alpha_k^I+\omega^I} w^*(\rho^J,\theta_1^J;\xi^I,\alpha_k^I)\xi^I d\lambda\right]_{k=1}^{I=J} = \frac{(\xi^I)^3}{4\pi D}\Big[(\omega^I-\sin\omega^I)$$

$$\cdot\left\{\ln 2(\xi^I)^2(1-\cos\omega^I)-1\right\} + \left\{\frac{(\omega^I)^3}{18} + \frac{(\omega^I)^5}{1800} + \ldots + \frac{2B_n(\omega^I)^{2n+1}}{(2n+1)!} + \ldots\right\}\Big].$$

$$\left[\int_{\alpha_k^I-\omega^I}^{\alpha_k^I+\omega^I} \frac{\partial w^*}{\partial\rho}(\rho^J,\theta_1^J;\xi^I,\alpha_k^I)\xi^I d\lambda\right]_{k=1}^{I=J} = \frac{(\xi^I)^2}{4\pi D}\Big[(\omega^I-\sin\omega^I)$$

$$\cdot\left\{\ln 2(\xi^I)^2(1-\cos\omega^I)\right\} + \left\{\frac{(\omega^I)^3}{18} + \frac{(\omega^I)^5}{1800} + \ldots + \frac{2B_n(\omega^I)^{2n+1}}{(2n+1)!} + \ldots\right\}\Big].$$

$$\left[\int_{\alpha_k^I-\omega^I}^{\alpha_k^I+\omega^I} \frac{\partial^2 w^*}{\partial\xi\partial\rho}(\rho^J,\theta_1^J;\xi^I,\alpha_k^I)\xi^I d\lambda\right]_{k=1}^{I=J} = \frac{\xi^I}{4\pi D}\Big[2\omega^I - (\sin\omega^I)$$

$$\cdot\left\{1 + \ln 2(\xi^I)^2(1-\cos\omega^I)\right\}\Big].$$

$$\left[\int_{\alpha_k^I-\omega^I}^{\alpha_k^I+\omega^I} M_\xi^*(\rho^J,\theta_1^J;\xi^I,\alpha_k^I)\xi^I d\lambda\right]_{k=1}^{I=J} = \frac{-\xi^I}{4\pi}\left[(1+\nu)\left\{\omega^I \ln 2(\xi^I)^2(1-\cos\omega^I)\right.\right.$$

$$\left.\left. + \frac{(\omega^I)^3}{18} + \frac{(\omega^I)^5}{1800} + \dots + \frac{2B_n(\omega^I)^{2n+1}}{(2n+1)!} + \dots\right\} - (1-\nu)\sin\omega^I\right].$$

$$\left[\int_{\alpha_k^I-\omega^I}^{\alpha_k^I+\omega^I} \frac{\partial M_\xi^*}{\partial\rho}(\rho^J,\theta_1^J;\xi^I,\alpha_k^I)\xi^I d\lambda\right]_{k=1}^{I=J} = \frac{-1}{4\pi}\left[2\nu\omega^I - (1-\nu)\sin\omega^I\right].$$

$$\left[\int_{\alpha_k^I-\omega^I}^{\alpha_k^I+\omega^I} V_\xi^*(\rho^J,\theta_1^J;\xi^I,\alpha_k^I)\xi^I d\lambda\right]_{k=1}^{I=J} = \frac{-1}{2\pi}\left[\omega^I + \frac{(1-\nu)\sin\omega^I}{2}\right].$$

Where $B_n = \dfrac{(2n)!}{2^{2n-1}\pi^{2n}}\left\{1 + \dfrac{1}{2^{2n}} + \dfrac{1}{3^{2n}} + \dots\right\}.$

For $\rho^J = \xi^I$ and $\theta \neq \alpha$:

$$\left[\int w^*(\rho^J,\theta;\xi^I,\alpha)\xi^I d(\alpha-\theta)\right]^{I=J} = \frac{(\xi^I)^3}{8\pi D}\left[\left\{(\alpha-\theta)-\sin(\alpha-\theta)\right\}\right.$$

$$\cdot\left\{\ln 2(\xi^I)^2\{1-\cos(\alpha-\theta)\}-1\right\}$$

$$\left. + \left\{\frac{(\alpha-\theta)^3}{18} + \frac{(\alpha-\theta)^5}{1800} + \dots + \frac{2B_n(\alpha-\theta)^{2n+1}}{(2n+1)!} + \dots\right\}\right].$$

$$\left[\int^{I=J} \frac{\partial w^*(\rho^J,\theta;\xi^I,\alpha)\xi^I d(\alpha-\theta)}{\partial\rho}\right] = \frac{(\xi^I)^2}{8\pi D}\left[\left\{(\alpha-\theta)-\sin(\alpha-\theta)\right\}\right.$$

$$.\left\{\ln 2(\xi^I)^2\{1-\cos(\alpha-\theta)\}\right\}$$

$$\left.+\left\{\frac{(\alpha-\theta)^3}{18}+\frac{(\alpha-\theta)^5}{1800}+\ldots+\frac{2B_n(\alpha-\theta)^{2n+1}}{(2n+1)!}+\ldots\right\}\right].$$

$$\left[\int^{I=J} \frac{\partial^2 w^*(\rho^J,\theta;\xi^I,\alpha)\xi^I d(\alpha-\theta)}{\partial\xi\partial\rho}\right] = \frac{\xi^I}{8\pi D}\left[2(\alpha-\theta)-\{\sin(\alpha-\theta)\}\right.$$

$$\left..\left\{1+\ln 2(\xi^I)^2\{1-\cos(\alpha-\theta)\}\right\}\right].$$

$$\left[\int^{I=J} M_\xi^*(\rho^J,\theta;\xi^I,\alpha)\xi^I d(\alpha-\theta)\right] = \frac{-\xi^I}{8\pi}\left[(1+\nu)\left\{(\alpha-\theta)\right.\right.$$

$$.\ln 2(\xi^I)^2\{1-\cos(\alpha-\theta)\}$$

$$\left.\left.+\frac{(\alpha-\theta)^3}{18}+\frac{(\alpha-\theta)^5}{1800}+\ldots+\frac{2B_n(\alpha-\theta)^{2n+1}}{(2n+1)!}+\ldots\right\}-(1-\nu)\sin(\alpha-\theta)\right].$$

$$\left[\int^{I=J} \frac{\partial M_\xi^*(\rho^J,\theta;\xi^I,\alpha)\xi^I d(\alpha-\theta)}{\partial\rho}\right] = \frac{-1}{8\pi}\left[2\nu(\alpha-\theta)-(1-\nu)\sin(\alpha-\theta)\right].$$

$$\left[\int V_{\xi}^{*}(\rho^{J},\theta;\xi^{I},\alpha)\xi^{I}d(\alpha-\theta) \right]^{I=J} = \frac{-1}{4\pi}\left[(\alpha-\theta) + \frac{(1-\nu)}{2}\sin(\alpha-\theta) \right].$$

$$\left[\int \frac{\partial V_{\xi}^{*}}{\partial\rho}(\rho^{J},\theta;\xi^{I},\alpha)\xi^{I}d(\alpha-\theta) \right]^{I=J} = \frac{-1}{4\pi\xi^{I}}\left[\cot\frac{(\alpha-\theta)}{2} - \frac{(1-\nu)}{2}.\frac{\sin(\alpha-\theta)\cos(\alpha-\theta)}{1-\cos(\alpha-\theta)} \right].$$

For $\rho \neq \xi$:

$$\int w^{*}(\rho,\theta;\xi,\alpha)\xi d(\alpha-\theta) = \frac{\xi}{16\pi D}\left[\left\{(\rho^{2}+\xi^{2})(\alpha-\theta) - 2\rho\xi\sin(\alpha-\theta)\right\} \ln\{\rho^{2}+\xi^{2} - 2\rho\xi\cos(\alpha-\theta)\} - 2\rho\xi(\rho^{2}+\xi^{2}).FN3 + 4\rho^{2}\xi^{2}.FN4 \right].$$

$$\int \frac{\partial w^{*}}{\partial\xi}(\rho,\theta;\xi,\alpha)\xi d(\alpha-\theta) = \frac{\xi}{8\pi D}\left[(1+2\ln r)\left\{\xi(\alpha-\theta) - \rho\sin(\alpha-\theta)\right\} - 2\rho\xi^{2}.FN3 + 2\rho^{2}\xi.FN4 \right].$$

$$\int \frac{\partial w^{*}}{\partial\rho}(\rho,\theta;\xi,\alpha)\xi d(\alpha-\theta) = \frac{\xi}{8\pi D}\left[(1+2\ln r)\left\{\rho(\alpha-\theta) - \xi\sin(\alpha-\theta)\right\}\right.$$

$$- 2\rho^2\xi.FN3 + 2\rho\xi^2.FN4 \Big].$$

$$\int \frac{\partial^2 w^*(\rho,\theta;\xi,\alpha)\xi d(\alpha-\theta)}{\partial\rho\partial\xi} = \frac{\xi}{8\pi D}\Big[-(1+2\ln r)\sin(\alpha-\theta) + 4\rho\xi.FN1$$

$$- 2(\rho^2+\xi^2).FN2 \Big].$$

$$\int M^*(\rho,\theta;\xi,\alpha)\xi d(\alpha-\theta) = \frac{-\xi}{8\pi}\Big[(1+\nu)(\alpha-\theta)(1+2\ln r) + 2(\rho^2+\xi^2).FN1$$

$$- 4\rho\xi.FN2 - (1+\nu)2\rho\xi.FN3 - (1-\nu)2\rho^2.FN4 \Big].$$

$$\int \frac{\partial M^*(\rho,\theta;\xi,\alpha)\xi d(\alpha-\theta)}{\partial\rho} = \frac{-\xi}{2\pi}\Big[\frac{(3+\nu)}{2}\{\rho.FN1-\xi.FN2\} - \rho.FN4$$

$$- (\rho^3+3\rho\xi^2).FN5 + (\xi^3+3\xi\rho^2).FN6 + (\rho^3+2\rho\xi^2+\nu\rho\xi^2).FN7 - (1+\nu)\rho^2\xi.FN8 \Big].$$

$$\int V^*(\rho,\theta;\xi,\alpha)\xi d(\alpha-\theta) = \frac{-1}{4\pi}\Big[\{2\xi^2-(1-\nu)\rho^2\}.FN1 - (1+\nu)\rho\xi.FN2$$

$$+ (1-\nu)\rho^2.FN4 + (1-\nu)\rho^2(\rho^2-\xi^2).FN7 \Big].$$

$$\int \frac{\partial V^*(\rho,\theta;\xi,\alpha)}{\partial\rho}\xi d(\alpha-\theta) = \frac{-1}{2\pi}\Big[-(1-\nu)\rho.FN1 - \frac{(1+\nu)}{2}\xi.FN2 + (1-\nu)\rho.FN4$$

$$- \{4\rho\xi^2-(1-\nu)\rho(\rho^2+\xi^2)\}.FN5 + 2\xi(\nu\rho^2+\xi^2).FN6 + \{2\rho\xi^2+(1-\nu)\rho(\rho^2-2\xi^2)\}.FN7$$

$$+ (1-\nu)\xi\rho^2.FN8 - (1-\nu)2\rho^3(\rho^2-\xi^2).FN9 + (1-\nu)2\xi\rho^2(\rho^2-\xi^2).FN10 \Big].$$

Where

$$FN1 = \int \frac{d(\alpha-\theta)}{r^2} = \frac{2}{|\rho^2-\xi^2|}\tan^{-1}\Big\{\frac{(\rho+\xi)}{|\rho-\xi|}\tan\frac{(\alpha-\theta)}{2}\Big\}$$

$$FN2 = \int \frac{\cos(\alpha-\theta)d(\alpha-\theta)}{r^2} = -\frac{(\alpha-\theta)}{2\rho\xi} + \frac{(\rho^2+\xi^2)}{\rho\xi|\rho^2-\xi^2|}\tan^{-1}\Big\{\frac{(\rho+\xi)\tan}{|\rho-\xi|}\frac{(\alpha-\theta)}{2}\Big\}$$

$$FN3 = \int \frac{(\alpha-\theta)\sin(\alpha-\theta)d(\alpha-\theta)}{r^2} = \frac{1}{2\rho\xi}\Big[\frac{\pi}{2}\ln r^2 - \Big\{\frac{(\rho^2+\xi^2)\ln r^2}{2\rho\xi} - \frac{r^2}{2\rho\xi}\Big\}$$

$$- \frac{1}{2}.\frac{1}{3}\Big\{\frac{(\rho^2+\xi^2)^3\ln r^2}{(2\rho\xi)^3} - \frac{3(\rho^2+\xi^2)^2}{(2\rho\xi)^2}.\frac{r^2}{2\rho\xi} + \frac{3}{2}.\frac{(\rho^2+\xi^2)}{2\rho\xi}.\frac{(r^2)^2}{(2\rho\xi)^2} - \frac{1}{3}.\frac{(r^2)^3}{(2\rho\xi)^3}\Big\}$$

$$-\ldots- \frac{1.3.5\ldots(2n-1)}{2.4.6\ldots(2n).(2n+1)}\Big\{\frac{(\rho^2+\xi^2)^{2n+1}\ln r^2}{(2\rho\xi)^{2n+1}}$$

$$+ \sum_{m=1}^{2n+1} \frac{(-1)^m(2n+1)!(\rho^2+\xi^2)^{2n+1-m}}{m!(2n+1-m)!(2\rho\xi)^{2n+1-m}}.\frac{1}{m}\frac{r^{2m}}{(2\rho\xi)^m}\Big\} - \ldots \Big].$$

$$FN4 = \int \frac{\sin^2(\alpha-\theta)d(\alpha-\theta)}{r^2} = \frac{1}{2\rho\xi}\left[\sin(\alpha-\theta) - \frac{(\rho^2+\xi^2)}{2\rho\xi}\sin^{-1}\{\cos(\alpha-\theta)\}\right.$$

$$\left. - \frac{|\rho^2-\xi^2|}{2\rho\xi}\sin^{-1}\left\{\frac{(\rho^2+\xi^2)r^2-(\rho^2-\xi^2)^2}{2\rho\xi r^2}\right\}\right].$$

$$FN5 = \int \frac{d(\alpha-\theta)}{r^4} = \frac{1}{(\rho^2-\xi^2)^2}\left[\frac{2\rho\xi\sin(\alpha-\theta)}{r^2}\right.$$

$$\left. + \frac{2(\rho^2+\xi^2)}{|\rho^2-\xi^2|}\tan^{-1}\left\{\frac{(\rho+\xi)}{|\rho-\xi|}\tan\frac{(\alpha-\theta)}{2}\right\}\right].$$

$$FN6 = \int \frac{\cos(\alpha-\theta)d(\alpha-\theta)}{r^4} = \frac{1}{(\rho^2-\xi^2)^2}\left[\frac{(\rho^2+\xi^2)\sin(\alpha-\theta)}{r^2}\right.$$

$$\left. + \frac{4\rho\xi}{|\rho^2-\xi^2|}\tan^{-1}\left\{\frac{(\rho+\xi)}{|\rho-\xi|}\tan\frac{(\alpha-\theta)}{2}\right\}\right].$$

$$FN7 = \int \frac{\sin^2(\alpha-\theta)d(\alpha-\theta)}{r^4} = \frac{1}{2\rho\xi}\left[-\frac{\sin(\alpha-\theta)}{r^2} + \frac{1}{2\rho\xi}\sin^{-1}\{\cos(\alpha-\theta)\}\right.$$

$$\left. + \frac{(\rho^2+\xi^2)}{2\rho\xi|\rho^2-\xi^2|}\sin^{-1}\left\{\frac{(\rho^2+\xi^2)r^2-(\rho^2-\xi^2)^2}{2\rho\xi r^2}\right\}\right].$$

$$FN8 = \int \frac{\cos(\alpha-\theta)\sin^2(\alpha-\theta)d(\alpha-\theta)}{r^4} = \frac{1}{(2\rho\xi)^2}\left[-(1 + \frac{\rho^2+\xi^2}{r^2})\sin(\alpha-\theta)\right.$$

$$+ \frac{(\rho^2+\xi^2)}{\rho\xi} \sin^{-1}\left\{\cos(\alpha-\theta)\right\} - \frac{(\rho^4+\xi^4)}{\rho\xi\left|\rho^2-\xi^2\right|} \sin^{-1}\left\{\frac{(\rho^2+\xi^2)r^2-(\rho^2-\xi^2)^2}{2\rho\xi r^2}\right\} \Big].$$

$$FN9 = \int \frac{\sin^2(\alpha-\theta)d(\alpha-\theta)}{r^6} = \frac{1}{2\rho\xi} \Big[\left\{\frac{(\rho^2+\xi^2)r^2-(\rho^2-\xi^2)^2}{2(\rho^2-\xi^2)^2 r^4}\right\} \sin(\alpha-\theta)$$

$$+ \frac{\rho\xi}{\left|(\rho^2-\xi^2)^3\right|} \sin^{-1}\left\{\frac{(\rho^2+\xi^2)r^2-(\rho^2-\xi^2)^2}{2\rho\xi r^2}\right\} \Big].$$

$$FN10 = \int \frac{\cos(\alpha-\theta)\sin^2(\alpha-\theta)d(\alpha-\theta)}{r^6} = \frac{-1}{(2\rho\xi)^2} \Big[\left\{ \frac{\rho^2+\xi^2}{2r^4} + \frac{(\rho^2+\xi^2)^2}{2(\rho^2-\xi^2)^2 r^2} - \frac{1}{r^2} \right\}$$

$$.\sin(\alpha-\theta) + \frac{1}{2\rho\xi} \sin^{-1}\left\{\cos(\alpha-\theta)\right\}$$

$$+ \left\{ \frac{3(\rho^2+\xi^2)}{2\left|\rho^2-\xi^2\right|} - \frac{(\rho^2+\xi^2)^3}{2\left|(\rho^2-\xi^2)^3\right|} \right\} \frac{1}{2\rho\xi} \sin^{-1}\left\{\frac{(\rho^2+\xi^2)r^2-(\rho^2-\xi^2)^2}{2\rho\xi r^2}\right\} \Big].$$

APPENDIX C

List of the Computer Programs

Two computer programs are included in this appendix. The first one is the program for solving circular and annular plates with arbitrary combination of boundary conditions and interior column supports. The latter program is used for conveniently editing an input data file of the first program.

```
C***********************************************************************
C
C BOUNDARY ELEMENT METHOD FOR THE ANALYSIS OF CIRCULAR AND ANNULAR PLATE
C
C                                 BY
C                         Mr.Siriphan Amonsrisawat
C
C                      DEPARTMENT OF CIVIL ENGINEERING
C                         CHULALONGKORN UNIVERSITY
C                                 1988
C***********************************************************************
C
C  last modified 16-SEPT-88
C  Simple support, Clamped, Free Edges and Column supports Combination
C  with Arbitrary location of points load and Uniform load
C  MAIN PROGRAM
C
      COMMON /LIO/ LR,LW,LC,LM
      COMMON /SIZE/ MAX
      COMMON A(100000)
      CHARACTER*72 IFNAME, OFNAME, MFNAME
      INTEGER*2 STRING(28), NUM
      INTEGER*2 TIME1, TIME2, TIME3, TIME4
      EQUIVALENCE (STRING(7),TIME1)
      EQUIVALENCE (STRING(8),TIME2)
      EQUIVALENCE (STRING(9),TIME3)
      EQUIVALENCE (STRING(10),TIME4)
C  MAXIMUM STORAGE CAN BE USED
      MAX = 100000
C  SET INPUT-OUTPUT LOGICAL UNITS
      CALL TNOUA ('     INPUT DATA FILE NAME ===> ', INTS(30))
      READ (1,10) IFNAME
   10 FORMAT (A72)
      CALL TNOUA ('     INPUT THE FILE OUTPUT ===> ', INTS(31))
      READ (1,10) OFNAME
      OPEN (5,FILE=IFNAME)
      OPEN (6,FILE=OFNAME)
      CLOSE (6,STATUS='DELETE')
      OPEN (6,FILE=OFNAME)
      NUM = 28
      CALL TIMDAT (STRING,NUM)
      WRITE (6,1)
    1 FORMAT ('CPU TIME used before executing in SECONDS + TICKS')
      WRITE (6,3) TIME1, TIME2
      WRITE (6,2)
    2 FORMAT ('I/O TIME used before executing in SECONDS + TICKS')
      WRITE (6,3) TIME3, TIME4
    3 FORMAT (2I20)
      LR = 5
      LW = 1
      LC = 6
C
      CALL LABEL ('XXXX','XXXX','XXXX','XXXX','XXXX','XXXX','XXXX')
      CALL BEGIN
      CALL TIMDAT (STRING,NUM)
```

```
      WRITE (6,5)
    5 FORMAT ('CPU TIME used after executing in SECONDS + TICKS')
      WRITE (6,3) TIME1, TIME2
      WRITE (6,6)
    6 FORMAT ('I/O TIME used after executing in SECONDS + TICKS')
      WRITE (6,3) TIME3, TIME4
      STOP
      END
C*****************************************
C  DRIVEN PROGRAM OF OVERALL PROCESS
C*****************************************
      SUBROUTINE BEGIN
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON HEAD (20)
      COMMON A(1)
      COMMON /LIO/ LR,LW,LC,LM
      COMMON /SIZE/ MAX
      COMMON /CONST/ PR,PI,RIGID
      COMMON /ANNULAR/ KCIR,ROUT,RINN
      COMMON /BOUND/ NOUT,NINN,KOUT(10),KINN(10),GAMM(10),BETA(10),
     *               NELOUT(10),NELINN(10),NCOL,RCOL(10),ZECOL(10)
      COMMON /LOAD/ UNIF,NPT,PLOAD(10),PRHO(10),PANG(10)
      COMMON /DISP/ KW,WDIR(20),KP,KWPT,WRHO(30),WANG(30)
C
      PI = 3.141592653589793
C
C  READ TITLE HEADING
C
      READ (LR,1000) HEAD
      WRITE (LW,2000) HEAD
      WRITE (LC,2000) HEAD
      READ (LR,1000) HEAD
      WRITE (LW,2000) HEAD
      WRITE (LC,2000) HEAD
C
C  ANALYSIS FOR CIRCULAR OR ANNULAR PLATE
C
      READ (LR,1000) HEAD
      WRITE (LW,2000) HEAD
      WRITE (LC,2000) HEAD
      READ (LR,1010) KCIR
      WRITE (LW,2010) KCIR
      WRITE (LC,2010) KCIR
C
C  READ RADIUS OF PLATE
C
      READ (LR,1000) HEAD
      WRITE (LW,2000) HEAD
      WRITE (LC,2000) HEAD
      READ (LR,1020) ROUT
      WRITE (LW,2020) ROUT
      WRITE (LC,2020) ROUT
        IF (KCIR.EQ.0) GO TO 10
      READ (LR,1020) RINN
```

```
      WRITE (LW,2020) RINN
      WRITE (LC,2020) RINN
C
C  READ NUMBER OF COMBINATION EDGES FOR OUTER AND INNER EDGE
C
   10 READ (LR,1000) HEAD
      WRITE (LW,2000) HEAD
      WRITE (LC,2000) HEAD
      READ (LR,1010) NOUT
      WRITE (LW,2010) NOUT
      WRITE (LC,2010) NOUT
        IF (NOUT.GT.10) GO TO 991
        IF (KCIR.EQ.0) GO TO 20
      READ (LR,1010) NINN
      WRITE (LW,2010) NINN
      WRITE (LC,2010) NINN
        IF (NINN.GT.10) GO TO 992
C
C  READ TYPE OF COMBINATION EDGES FOR OUTER AND INNER EDGE
C
   20 READ (LR,1000) HEAD
      WRITE (LW,2000) HEAD
      WRITE (LC,2000) HEAD
      READ (LR,1010) (KOUT(I),I=1,NOUT)
      WRITE (LW,2030) (KOUT(I),I=1,NOUT)
      WRITE (LC,2030) (KOUT(I),I=1,NOUT)
        IF (KCIR.EQ.0) GO TO 30
      READ (LR,1010) (KINN(I),I=1,NINN)
      WRITE (LW,2030) (KINN(I),I=1,NINN)
      WRITE (LC,2030) (KINN(I),I=1,NINN)
C
C  READ THE BEGINNING ANGLE OF EACH BOUNDARY SECTION
C
   30 GAMM(1) = 0.0
        IF (NOUT.LE.1) GO TO 35
      READ (LR,1000) HEAD
      WRITE (LW,2000) HEAD
      WRITE (LC,2000) HEAD
      READ (LR,1020) (GAMM(I),I=1,NOUT)
      WRITE (LW,2040) (GAMM(I),I=1,NOUT)
      WRITE (LC,2040) (GAMM(I),I=1,NOUT)
        IF (GAMM(1).NE.0.0) GO TO 998
        DO 310 I = 1, NOUT
           GAMM(I) = GAMM(I) * PI / 180
  310   CONTINUE
   35   IF (KCIR.EQ.0) GO TO 40
      BETA(1) = 0.0
        IF (NINN.LE.1) GO TO 40
      READ (LR,1020) (BETA(I),I=1,NINN)
      WRITE (LW,2040) (BETA(I),I=1,NINN)
      WRITE (LC,2040) (BETA(I),I=1,NINN)
        IF (BETA(1).NE.0.0) GO TO 998
        DO 320 I = 1, NINN
           BETA(I) = BETA(I) * PI / 180
  320   CONTINUE
```

```
C
C  READ NUMBER OF ELEMENTS IN EACH BOUNDARY SECTION
C
   40 READ (LR,1000) HEAD
      WRITE (LW,2000) HEAD
      WRITE (LC,2000) HEAD
      READ (LR,1010) (NELOUT(I),I=1,NOUT)
      WRITE (LW,2030) (NELOUT(I),I=1,NOUT)
      WRITE (LC,2030) (NELOUT(I),I=1,NOUT)
        IF (KCIR.EQ.0) GO TO 50
      READ (LR,1010) (NELINN(I),I=1,NINN)
      WRITE (LW,2030) (NELINN(I),I=1,NINN)
      WRITE (LC,2030) (NELINN(I),I=1,NINN)
C
C  READ NUMBER OF COLUMN SUPPORTS AND LOCATIONS
C
   50 READ (LR,1000) HEAD
      WRITE (LW,2000) HEAD
      WRITE (LC,2000) HEAD
      READ (LR,1010) NCOL
      WRITE (LW,2010) NCOL
      WRITE (LC,2010) NCOL
        IF (NCOL.EQ.0) GO TO 60
        IF (NCOL.GT.10) GO TO 993
        DO 510 I = 1, NCOL
           READ (LR,1020) RCOL(I),ZECOL(I)
           WRITE (LW,2040) RCOL(I),ZECOL(I)
           WRITE (LC,2040) RCOL(I),ZECOL(I)
           ZECOL(I) = ZECOL(I) * PI / 180.
             IF (RCOL(I).GT.ROUT) GO TO 997
             IF (KCIR.EQ.0) GO TO 510
             IF (RCOL(I).LT.RINN) GO TO 997
  510   CONTINUE
C
C  READ THICKNESS OF PLATE, POISSON'S RATIO AND MODULUS OF ELASTICITY
C
   60 READ (LR,1000) HEAD
      WRITE (LW,2000) HEAD
      WRITE (LC,2000) HEAD
      READ (LR,1030) TH,PR,ELAS
      WRITE (LW,2050) TH,PR,ELAS
      WRITE (LC,2050) TH,PR,ELAS
        RIGID = ELAS*TH**3/(1-PR**2)/12
C
C  READ UNIFORM LOAD
C
      READ (LR,1000) HEAD
      WRITE (LW,2000) HEAD
      WRITE (LC,2000) HEAD
      READ (LR,1020) UNIF
      WRITE (LW,2020) UNIF
      WRITE (LC,2020) UNIF
C
C  READ NUMBER OF CONCENTRATED LOADS AND LOCATIONS  (PRHO,PANG)
C
```

```
   70 READ (LR,1000) HEAD
      WRITE (LW,2000) HEAD
      WRITE (LC,2000) HEAD
      READ (LR,1010) NPT
      WRITE (LW,2010) NPT
      WRITE (LC,2010) NPT
       IF (NPT.LE.0) GO TO 80
       DO 710 I = 1, NPT
          READ (LR,1020) PLOAD(I),PRHO(I),PANG(I)
          WRITE (LW,2040) PLOAD(I),PRHO(I),PANG(I)
          WRITE (LC,2040) PLOAD(I),PRHO(I),PANG(I)
            PANG(I) = PANG(I) * PI / 180.
  710  CONTINUE
C
C  READ ANGLE OF RADIUS ALONG WHICH RESULTS ARE REQUIRED
C
   80 READ (LR,1000) HEAD
      WRITE (LW,2000) HEAD
      WRITE (LC,2000) HEAD
      READ (LR,1010) KW
      WRITE (LW,2010) KW
      WRITE (LC,2010) KW
        IF (KW.LE.0) GO TO 90
      READ (LR,1020) (WDIR(I),I=1,KW)
      WRITE (LW,2040) (WDIR(I),I=1,KW)
      WRITE (LC,2040) (WDIR(I),I=1,KW)
        DO 810 I = 1, KW
           WDIR(I) = WDIR(I) * PI / 180.
  810   CONTINUE
      READ (LR,1000) HEAD
      WRITE (LW,2000) HEAD
      WRITE (LC,2000) HEAD
      READ (LR,1010) KP
      WRITE (LW,2010) KP
      WRITE (LC,2010) KP
C
C  READ LOCATIONS OF SPECIFIC POINT ON WHICH RESULTS ARE REQUIRED
C
   90 READ (LR,1000) HEAD
      WRITE (LW,2000) HEAD
      WRITE (LC,2000) HEAD
      READ (LR,1010) KWPT
      WRITE (LW,2010) KWPT
      WRITE (LC,2010) KWPT
       IF (KWPT.LE.0) GO TO 100
       DO 910 I = 1, KWPT
          READ (LR,1020) WRHO(I),WANG(I)
          WRITE (LW,2040) WRHO(I),WANG(I)
          WRITE (LC,2040) WRHO(I),WANG(I)
            WANG(I) = WANG(I) * PI / 180.
  910  CONTINUE
C
C  SET DIMENSION OF MATRIX
C
  100 NRS1 = 0
```

```
      DO 200 I = 1, NOUT
         NRS1 = NRS1 + NELOUT(I)
  200 CONTINUE
      IF (KCIR.EQ.0) GO TO 400
      DO 300 I = 1, NINN
         NRS1 = NRS1 + NELINN(I)
  300 CONTINUE
  400 NRS = NRS1 * 2
      IF (NCOL.LE.0) GO TO 450
      NRS = NRS + NCOL
C
  450 NT = KP
      IF (KWPT.LE.KP) GO TO 500
      NT = KWPT
  500 N1 = 1
      N2 = N1 + NRS*NRS
      N3 = N2 + NRS
      NE = N3 + NT*8
      IF (NE.GT.MAX) GO TO 999
      WRITE (LW,3000) NE
      WRITE (LC,3000) NE
C
C  GENERATE COEFFICIENT MATRIX
C
      CALL GENER (A(N1),NRS,NRS1)
      WRITE (LW,2100)
C      WRITE (LM,2100)
C      CALL DISP2 (A(N1),NRS,NRS)
C
C  GENERATE LOAD VECTOR
C
      CALL GENLD (A(N2),NRS1)
      WRITE (LW,2110)
C      WRITE (LM,2110)
C      CALL DISP1 (A(N2),NRS)
C
C  SOLVE FOR UNKNOWNS
C
      CALL SLNPD (A(N1),A(N2),DETER,NRS)
      WRITE (LW,2120) DETER
      WRITE (LC,2120) DETER
      WRITE (LW,2130)
      WRITE (LC,2130)
      CALL WTAPE2 (A(N2),NRS,1)
C
C  COMPUTE VALUES OF DEFLECTION AND STRESS RESULTANTS
C
      IF (KW.LE.0) GO TO 700
      DO 600 I = 1, KW
         WRITE (LW,*) 'DIRECTION NO. ',I
         CALL RESULT (A(N2),A(N3),I,KP,NRS1)
         WRITE (LW,2140) I
         WRITE (LC,2140) I
         CALL WTAPE2A (A(N3),KP)
  600 CONTINUE
```

```
  700 IF (KWPT.LE.0) GO TO 800
      CALL RESULT1 (A(N2),A(N3),KWPT,NRS1)
      WRITE (LW,2150)
      WRITE (LC,2150)
      CALL WTAPE2A (A(N3),KWPT)
C
  800 RETURN
C
  991 WRITE (LW,3040) 'NO. OF OUTER BOUNDARY '
      RETURN
  992 WRITE (LW,3040) 'NO. OF INNER BOUNDARY '
      RETURN
  993 WRITE (LW,3040) 'NO. OF COLUMNS '
      RETURN
  997 WRITE (LW,3020)
      RETURN
  998 WRITE (LW,3030)
      RETURN
  999 NMORE = NE - MAX
      WRITE (LW,3010) NE,NMORE
      RETURN
 1000 FORMAT (20A4)
 1010 FORMAT (12I5)
 1020 FORMAT (8F10.5)
 1030 FORMAT (2F10.5,D20.10)
 2000 FORMAT (/5X,20A4)
 2010 FORMAT (5X,'============>',I5)
 2020 FORMAT (5X,'============>',F10.5)
 2030 FORMAT (2X,12I5)
 2040 FORMAT (2X,8F10.5)
 2050 FORMAT (2X,2F10.5,D20.10)
 2100 FORMAT (/5X,'S (I,J)  ********************')
 2110 FORMAT (/5X,'LOAD VECTOR  *****************')
 2120 FORMAT (/5X,'DETERMINANT =',D20.10)
 2130 FORMAT (/5X,'UNKNOWNS SOLUTION  ************')
 2140 FORMAT (/1X,'*****RESULTS FOR RADIUS DIRECTION NO.',I3)
 2150 FORMAT (/1X,'*****RESULTS AT OTHER REQUIRED POSITION')
 3000 FORMAT (/10X,'********* MEMORY USED =',I10,' *********')
 3010 FORMAT (/5X,'REQUIRED STORAGES =',I10,' WHICH',I10,' EXCEEDED')
 3020 FORMAT (/5X,'COLUMN SUPPORTS IS OUT OF PLATE ')
 3030 FORMAT (/5X,'THE FIRST ANGLE MUST BE EQUAL ZERO ')
 3040 FORMAT (/5X,A25,'MUST NOT BE GREATER THAN 10 ')
      END
C*********************************
C  GENERATE COEFFICIENT MATRIX
C*********************************
      SUBROUTINE GENER (S,NRS,NRS1)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION S(NRS,1)
      COMMON /LIO/ LR,LW,LC,LM
      COMMON /ANNULAR/ KCIR,ROUT,RINN
      COMMON /BOUND/ NOUT,NINN,KOUT(10),KINN(10),GAMM(10),BETA(10),
     *               NELOUT(10),NELINN(10),NCOL,RCOL(10),ZECOL(10)
```

```
C
C
C  FOR NODAL POINTS ON THE OUTER BOUNDARY
C
C  ELEMENT LENGTH ==>EL
      NJ = 0
      NJ1 = NRS1
      DO 400 N = 1, NOUT
         EL = ELLE(0,N)
         OMEGA1 = EL/2./ROUT
         DO 390 J = 1, NELOUT(N)
            NJ = NJ + 1
            NJ1 = NJ1 + 1
           WRITE (LW,5000) NJ, NJ1
 5000      FORMAT (2I5)
C
C  INFLUENCE FUNCTIONS ON EACH INTERVAL OF OUTER BOUNDARY
            MI = 0
            MI1 = NRS1
            DO 100 M = 1, NOUT
               EL = ELLE(0,M)
               OMEGA2 = EL/2./ROUT
               DO 90 I = 1, NELOUT(M)
                  MI = MI + 1
                  MI1 = MI1 + 1
                  CALL CALR (ROUT,N,J,OMEGA1,ROUT,M,I,OMEGA2,R,GAM)
                  IF (R.LE.0.0) THEN
                     BN = BNSERY (OMEGA2)
                  ELSE
                     CALL ALIMIT (GAM,OMEGA2,GAM1,GAM2)
                     BN1 = BNSERY (GAM1)
                     BN2 = BNSERY (GAM2)
                  ENDIF
                  IF (KOUT(M) - 1) 10, 20, 30
C  FOR FREE EDGE : KOUT(-) = 0
  10              IF (R.LE.0.0) THEN
                   SM1 = -SPECV (OMEGA2) * 2. - 0.5
                   SM2 =  SPECM (ROUT,OMEGA2,BN) * 2.
                   SM3 = -SPECDV (ROUT,OMEGA2) * 2.
                   SM4 =  SPECDM (OMEGA2) * 2. - 0.5
                  ELSE
                   SM1 = -SPECV (GAM2) +SPECV (GAM1)
                   SM2 =  SPECM (ROUT,GAM2,BN2) -SPECM (ROUT,GAM1,BN1)
                   SM3 = -SPECDV (ROUT,GAM2) +SPECDV (ROUT,GAM1)
                   SM4 =  SPECDM (GAM2) -SPECDM (GAM1)
                  ENDIF
                   GO TO 40
C  FOR SIMPLE SUPPORT : KOUT(-) = 1
  20              IF (R.LE.0.0) THEN
                   SM1 =  SPECW (ROUT,OMEGA2,BN) * 2.
                   SM2 =  SPECM (ROUT,OMEGA2,BN) * 2.
                   SM3 =  SPECSL (ROUT,OMEGA2,BN) * 2.
                   SM4 =  SPECDM (OMEGA2) * 2. - 0.5
                  ELSE
                   SM1 =  SPECW (ROUT,GAM2,BN2) -SPECW (ROUT,GAM1,BN1)
```

```
                SM2 =  SPECM (ROUT,GAM2,BN2) -SPECM (ROUT,GAM1,BN1)
                SM3 =  SPECSL (ROUT,GAM2,BN2) -SPECSL (ROUT,GAM1,BN1)
                SM4 =  SPECDM (GAM2) -SPECDM (GAM1)
               ENDIF
                GO TO 40
C  FOR CLAMPED SUPPORT : KOUT(-) = 2
   30          IF (R.LE.0.0) THEN
                SM1 =  SPECW (ROUT,OMEGA2,BN) * 2.
                SM2 = -SPECSL (ROUT,OMEGA2,BN) * 2.
                SM3 =  SPECSL (ROUT,OMEGA2,BN) * 2.
                SM4 = -SPECDD (ROUT,OMEGA2) * 2.
               ELSE
                SM1 =  SPECW (ROUT,GAM2,BN2)  -SPECW (ROUT,GAM1,BN1)
                SM2 = -SPECSL (ROUT,GAM2,BN2) +SPECSL (ROUT,GAM1,BN1)
                SM3 =  SPECSL (ROUT,GAM2,BN2) -SPECSL (ROUT,GAM1,BN1)
                SM4 = -SPECDD (ROUT,GAM2,BN2) +SPECDD (ROUT,GAM1,BN1)
               ENDIF
C
   40          S(NJ,MI)   = SM1
               S(NJ,MI1)  = SM2
               S(NJ1,MI)  = SM3
               S(NJ1,MI1) = SM4
   90        CONTINUE
  100      CONTINUE
C
C  INFLUENCE FUNCTIONS ON EACH INTERVAL OF INNER BOUNDARY
           IF (KCIR.EQ.0) GO TO 210
           DO 200 M = 1, NINN
             EL = ELLE(1,M)
             OMEGA2 = EL/2./RINN
             DO 190 I = 1, NELINN(M)
               MI = MI + 1
               MI1 = MI1 + 1
               CALL CALR (ROUT,N,J,OMEGA1,RINN,M,I,OMEGA2,R,GAM)
               IF (KINN(M) - 1) 110, 120, 130
C  FOR FREE EDGE : KINN(-) = 0
  110          SM1 =  FUNCV (RINN,ROUT,R,GAM)
               SM2 = -FUNCM (RINN,ROUT,R,GAM)
               SM3 =  FUNCDV (RINN,ROUT,R,GAM)
               SM4 = -FUNCDM (RINN,ROUT,R,GAM)
                GO TO 140
C  FOR SIMPLE SUPPORT : KINN(-) = 1
  120          SM1 = -FUNCW (R)
               SM2 = -FUNCM (RINN,ROUT,R,GAM)
               SM3 = -FUNCSL (ROUT,RINN,R,GAM)
               SM4 = -FUNCDM (RINN,ROUT,R,GAM)
                GO TO 140
C  FOR CLAMPED SUPPORT : KINN(-) = 2
  130          SM1 = -FUNCW (R)
               SM2 =  FUNCSL (RINN,ROUT,R,GAM)
               SM3 = -FUNCSL (ROUT,RINN,R,GAM)
               SM4 =  FUNCDD (RINN,ROUT,R,GAM)
C
  140          S(NJ,MI)   = SM1 * EL
               S(NJ,MI1)  = SM2 * EL
```

```
                  S(NJ1,MI)  = SM3 * EL
                  S(NJ1,MI1) = SM4 * EL
  190           CONTINUE
  200         CONTINUE
C
C  INFLUENCE FUNCTIONS AT EACH COLUMN LOACTION
  210         IF (NCOL.EQ.0) GO TO 390
              DO 300 M = 1, NCOL
                 MI1 = MI1 + 1
                 CALL CALR1 (ROUT,N,J,OMEGA1,RCOL(M),ZECOL(M),R,GAM)
                 IF (R.LE.0.0) THEN
                  SM1 =  0.0
                  SM2 =  0.0
                 ELSE
                  SM1 =  FUNCW (R)
                  SM2 =  FUNCSL (ROUT,RCOL(M),R,GAM)
                 ENDIF
C
                  S(NJ,MI1)  = SM1
                  S(NJ1,MI1) = SM2
  300         CONTINUE
  390      CONTINUE
  400 CONTINUE
C
C
C  FOR NODAL POINTS ON THE INNER BOUNDARY
C
      IF (KCIR.EQ.0) GO TO 1000
      DO 900 N = 1, NINN
         EL = ELLE(1,N)
         OMEGA1 = EL/2./RINN
         DO 890 J = 1, NELINN(N)
            NJ = NJ + 1
            NJ1 = NJ1 +1
           WRITE (LW,5000) NJ, NJ1
C
C  INFLUENCE FUNCTIONS ON EACH INTERVAL OF OUTER BOUNDARY
            MI = 0
            MI1 = NRS1
            DO 600 M = 1, NOUT
               EL = ELLE(0,M)
               OMEGA2 = EL/2./ROUT
               DO 590 I = 1, NELOUT(M)
                  MI = MI + 1
                  MI1 = MI1 + 1
                  CALL CALR (RINN,N,J,OMEGA1,ROUT,M,I,OMEGA2,R,GAM)
                  IF (KOUT(M) - 1) 510, 520, 530
C  FOR FREE EDGE : KOUT(-) = 0
  510             SM1 = -FUNCV (ROUT,RINN,R,GAM)
                  SM2 =  FUNCM (ROUT,RINN,R,GAM)
                  SM3 = -FUNCDV (ROUT,RINN,R,GAM)
                  SM4 =  FUNCDM (ROUT,RINN,R,GAM)
                   GO TO 540
C  FOR SIMPLE SUPPORT : KOUT(-) = 1
  520             SM1 =  FUNCW (R)
```

```
                 SM2 =  FUNCM (ROUT,RINN,R,GAM)
                 SM3 =  FUNCSL (RINN,ROUT,R,GAM)
                 SM4 =  FUNCDM (ROUT,RINN,R,GAM)
                  GO TO 540
C  FOR CLAMPED SUPPORT : KOUT(-) = 2
  530            SM1 =  FUNCW (R)
                 SM2 = -FUNCSL (ROUT,RINN,R,GAM)
                 SM3 =  FUNCSL (RINN,ROUT,R,GAM)
                 SM4 = -FUNCDD (ROUT,RINN,R,GAM)
C
  540            S(NJ,MI)   = SM1 * EL
                 S(NJ,MI1)  = SM2 * EL
                 S(NJ1,MI)  = SM3 * EL
                 S(NJ1,MI1) = SM4 * EL
  590          CONTINUE
  600       CONTINUE
C
C  INFLUENCE FUNCTIONS ON EACH INTERVAL OF INNER BOUNDARY
            DO 700 M = 1, NINN
               EL = ELLE(1,M)
               OMEGA2 = EL/2./RINN
               DO 690 I = 1, NELINN(M)
                  MI = MI + 1
                  MI1 = MI1 + 1
                  CALL CALR (RINN,N,J,OMEGA1,RINN,M,I,OMEGA2,R,GAM)
                  IF (R.LE.0.0) THEN
                     BN = BNSERY (OMEGA2)
                  ELSE
                     CALL ALIMIT (GAM,OMEGA2,GAM1,GAM2)
                     BN1 = BNSERY (GAM1)
                     BN2 = BNSERY (GAM2)
                  ENDIF
                  IF (KINN(M) - 1) 610, 620, 630
C  FOR FREE EDGE : KINN(-) = 0
  610             IF (R.LE.0.0) THEN
                   SM1 =  SPECV (OMEGA2) * 2. - 0.5
                   SM2 = -SPECM (RINN,OMEGA2,BN) * 2.
                   SM3 =  SPECDV (RINN,OMEGA2) * 2.
                   SM4 = -SPECDM (OMEGA2) * 2. - 0.5
                  ELSE
                   SM1 =  SPECV (GAM2) -SPECV (GAM1)
                   SM2 = -SPECM (RINN,GAM2,BN2) +SPECM (RINN,GAM1,BN1)
                   SM3 =  SPECDV (RINN,GAM2) -SPECDV (RINN,GAM1)
                   SM4 = -SPECDM (GAM2) +SPECDM (GAM1)
                  ENDIF
                   GO TO 640
C  FOR SIMPLE SUPPORT : KINN(-) = 1
  620             IF (R.LE.0.0) THEN
                   SM1 = -SPECW (RINN,OMEGA2,BN) * 2.
                   SM2 = -SPECM (RINN,OMEGA2,BN) * 2.
                   SM3 = -SPECSL (RINN,OMEGA2,BN) * 2.
                   SM4 = -SPECDM (OMEGA2) * 2. - 0.5
                  ELSE
                   SM1 = -SPECW (RINN,GAM2,BN2) +SPECW (RINN,GAM1,BN1)
                   SM2 = -SPECM (RINN,GAM2,BN2) +SPECM (RINN,GAM1,BN1)
```

```
                 SM3 = -SPECSL (RINN,GAM2,BN2) +SPECSL (RINN,GAM1,BN1)
                 SM4 = -SPECDM (GAM2) +SPECDM (GAM1)
                ENDIF
                 GO TO 640
C  FOR CLAMPED SUPPORT : KINN(-) = 2
  630           IF (R.LE.0.0) THEN
                 SM1 = -SPECW (RINN,OMEGA2,BN) * 2.
                 SM2 =  SPECSL (RINN,OMEGA2,BN) * 2.
                 SM3 = -SPECSL (RINN,OMEGA2,BN) * 2.
                 SM4 =  SPECDD (RINN,OMEGA2) * 2.
                ELSE
                 SM1 = -SPECW (RINN,GAM2,BN2)  +SPECW (RINN,GAM1,BN1)
                 SM2 =  SPECSL (RINN,GAM2,BN2) -SPECSL (RINN,GAM1,BN1)
                 SM3 = -SPECSL (RINN,GAM2,BN2) +SPECSL (RINN,GAM1,BN1)
                 SM4 =  SPECDD (RINN,GAM2,BN2) -SPECDD (RINN,GAM1,BN1)
                ENDIF
C
  640           S(NJ,MI)   = SM1
                S(NJ,MI1)  = SM2
                S(NJ1,MI)  = SM3
                S(NJ1,MI1) = SM4
  690         CONTINUE
  700       CONTINUE
C
C  INFLUENCE FUNCTIONS AT EACH COLUMN LOACTION
            IF (NCOL.EQ.0) GO TO 890
            DO 800 M = 1, NCOL
              MI1 = MI1 + 1
              CALL CALR1 (RINN,N,J,OMEGA1,RCOL(M),ZECOL(M),R,GAM)
              IF (R.LE.0.0) THEN
               SM1 =  0.0
               SM2 =  0.0
              ELSE
               SM1 =  FUNCW (R)
               SM2 =  FUNCSL (RINN,RCOL(M),R,GAM)
              ENDIF
C
               S(NJ,MI1)  = SM1
               S(NJ1,MI1) = SM2
  800       CONTINUE
  890     CONTINUE
  900 CONTINUE
C
C
C  FOR POINTS ON THE COLUMN LOCATIONS
C
 1000 IF (NCOL.EQ.0) GO TO 1500
      DO 1400 N = 1, NCOL
        NJ1 = NJ1 +1
       WRITE (LW,5000) NJ1
C
C  INFLUENCE FUNCTIONS ON EACH INTERVAL OF OUTER BOUNDARY
            MI = 0
        MI1 = NRS1
        DO 1100 M = 1, NOUT
```

```
            EL = ELLE(0,M)
            OMEGA2 = EL/2./ROUT
            DO 1090 I = 1, NELOUT(M)
               MI = MI + 1
               MI1 = MI1 + 1
               CALL CALR1 (ROUT,M,I,OMEGA2,RCOL(N),ZECOL(N),R,GAM)
               IF (R.LE.0.0) THEN
                BN = BNSERY (OMEGA2)
               ELSE IF (RCOL(N).EQ.ROUT) THEN
                CALL ALIMIT (GAM,OMEGA2,GAM1,GAM2)
                BN1 = BNSERY (GAM1)
                BN2 = BNSERY (GAM2)
               ELSE IF (RCOL(N).GT.0.0) THEN
                ALP = ANGLE (0,M,I,OMEGA2)
                ALP1 = ALP - OMEGA2
                ALP2 = ALP + OMEGA2
                CALL FNINTEG (RCOL(N),ZECOL(N),ROUT,ALP1,ALP2,RS1,RS2)
               ENDIF
               IF (KOUT(M) - 1) 1010, 1020, 1030
C  FOR FREE EDGE : KOUT(-) = 0
 1010          IF (R.LE.0.0) THEN
                SM1 = -SPECV (OMEGA2) * 2. - 0.5
                SM2 =  SPECM (ROUT,OMEGA2,BN) * 2.
               ELSE IF (RCOL(N).EQ.ROUT) THEN
                SM1 = -SPECV (GAM2) +SPECV (GAM1)
                SM2 =  SPECM (ROUT,GAM2,BN2) -SPECM (ROUT,GAM1,BN1)
               ELSE IF (RCOL(N).EQ.0.0) THEN
                SM1 = -FUNCV (ROUT,RCOL(N),R,GAM) * EL
                SM2 =  FUNCM (ROUT,RCOL(N),R,GAM) * EL
               ELSE
                SM1 = -FINTV (RCOL(N),ZECOL(N),ROUT,RS1,RS2)
                SM2 =  FUNCM (ROUT,RCOL(N),R,GAM) * EL
               ENDIF
                GO TO 1040
C  FOR SIMPLE SUPPORT : KOUT(-) = 1
 1020          IF (R.LE.0.0) THEN
                SM1 =  SPECW (ROUT,OMEGA2,BN) * 2.
                SM2 =  SPECM (ROUT,OMEGA2,BN) * 2.
               ELSE IF (RCOL(N).EQ.ROUT) THEN
                SM1 =  SPECW (ROUT,GAM2,BN2) -SPECW (ROUT,GAM1,BN1)
                SM2 =  SPECM (ROUT,GAM2,BN2) -SPECM (ROUT,GAM1,BN1)
               ELSE IF (RCOL(N).EQ.0.0) THEN
                SM1 =  FUNCW (R) * EL
                SM2 =  FUNCM (ROUT,RCOL(N),R,GAM) * EL
               ELSE
                SM1 =  FUNCW (R) * EL
                SM2 =  FUNCM (ROUT,RCOL(N),R,GAM) * EL
               ENDIF
                GO TO 1040
C  FOR CLAMPED SUPPORT : KOUT(-) = 2
 1030          IF (R.LE.0.0) THEN
                SM1 =  SPECW (ROUT,OMEGA2,BN) * 2.
                SM2 = -SPECSL (ROUT,OMEGA2,BN) * 2.
               ELSE IF (RCOL(N).EQ.ROUT) THEN
                SM1 =  SPECW (ROUT,GAM2,BN2)  -SPECW (ROUT,GAM1,BN1)
```

```
                SM2 = -SPECSL (ROUT,GAM2,BN2) +SPECSL (ROUT,GAM1,BN1)
               ELSE IF (RCOL(N).EQ.0.0) THEN
                SM1 =  FUNCW (R) * EL
                SM2 = -FUNCSL (ROUT,RCOL(N),R,GAM) * EL
               ELSE
                SM1 =  FUNCW (R) * EL
                SM2 = -FUNCSL (ROUT,RCOL(N),R,GAM) * EL
               ENDIF
C
 1040           S(NJ1,MI)  = SM1
                S(NJ1,MI1) = SM2
 1090        CONTINUE
 1100     CONTINUE
C
C  INFLUENCE FUNCTIONS ON EACH INTERVAL OF INNER BOUNDARY
             IF (KCIR.EQ.0) GO TO 1210
          DO 1200 M = 1, NINN
             EL = ELLE(1,M)
             OMEGA2 = EL/2./RINN
             DO 1190 I = 1, NELINN(M)
               MI = MI + 1
               MI1 = MI1 + 1
               CALL CALR1 (RINN,M,I,OMEGA2,RCOL(N),ZECOL(N),R,GAM)
               IF (R.LE.0.0) THEN
                BN = BNSERY (OMEGA2)
               ELSE IF (RCOL(N).EQ.RINN) THEN
                CALL ALIMIT (GAM,OMEGA2,GAM1,GAM2)
                BN1 = BNSERY (GAM1)
                BN2 = BNSERY (GAM2)
               ELSE IF (RCOL(N).GT.0.0) THEN
                ALP = ANGLE (1,M,I,OMEGA2)
                ALP1 = ALP - OMEGA2
                ALP2 = ALP + OMEGA2
                CALL FNINTEG (RCOL(N),ZECOL(N),RINN,ALP1,ALP2,RS1,RS2)
               ENDIF
               IF (KINN(M) - 1) 1110, 1120, 1130
C  FOR FREE EDGE : KINN(-) = 0
 1110          IF (R.LE.0.0) THEN
                SM1 =  SPECV (OMEGA2) * 2. - 0.5
                SM2 = -SPECM (RINN,OMEGA2,BN) * 2.
               ELSE IF (RCOL(N).EQ.RINN) THEN
                SM1 =  SPECV (GAM2) -SPECV (GAM1)
                SM2 = -SPECM (RINN,GAM2,BN2) +SPECM (RINN,GAM1,BN1)
               ELSE IF (RCOL(N).EQ.0.0) THEN
                SM1 =  FUNCV (RINN,RCOL(N),R,GAM) * EL
                SM2 = -FUNCM (RINN,RCOL(N),R,GAM) * EL
               ELSE
                SM1 =  FINTV (RCOL(N),ZECOL(N),RINN,RS1,RS2)
                SM2 = -FUNCM (RINN,RCOL(N),R,GAM) * EL
               ENDIF
                GO TO 1140
C  FOR SIMPLE SUPPORT : KINN(-) = 1
 1120          IF (R.LE.0.0) THEN
                SM1 = -SPECW (RINN,OMEGA2,BN) * 2.
                SM2 = -SPECM (RINN,OMEGA2,BN) * 2.
```

```
                ELSE IF (RCOL(N).EQ.RINN) THEN
                 SM1 = -SPECW (RINN,GAM2,BN2) +SPECW (RINN,GAM1,BN1)
                 SM2 = -SPECM (RINN,GAM2,BN2) +SPECM (RINN,GAM1,BN1)
                ELSE IF (RCOL(N).EQ.0.0) THEN
                 SM1 = -FUNCW (R) * EL
                 SM2 = -FUNCM (RINN,RCOL(N),R,GAM) * EL
                ELSE
                 SM1 = -FUNCW (R) * EL
                 SM2 = -FUNCM (RINN,RCOL(N),R,GAM) * EL
                ENDIF
                 GO TO 1140
C  FOR CLAMPED SUPPORT : KINN(-) = 2
 1130           IF (R.LE.0.0) THEN
                 SM1 = -SPECW (RINN,OMEGA2,BN) * 2.
                 SM2 =  SPECSL (RINN,OMEGA2,BN) * 2.
                ELSE IF (RCOL(N).EQ.RINN) THEN
                 SM1 = -SPECW (RINN,GAM2,BN2)  +SPECW (RINN,GAM1,BN1)
                 SM2 =  SPECSL (RINN,GAM2,BN2) -SPECSL (RINN,GAM1,BN1)
                ELSE IF (RCOL(N).EQ.0.0) THEN
                 SM1 = -FUNCW (R) * EL
                 SM2 =  FUNCSL (RINN,RCOL(N),R,GAM) * EL
                ELSE
                 SM1 = -FUNCW (R) * EL
                 SM2 =  FUNCSL (RINN,RCOL(N),R,GAM) * EL
                ENDIF
C
 1140            S(NJ1,MI)  = SM1
                 S(NJ1,MI1) = SM2
 1190         CONTINUE
 1200      CONTINUE
C
C  INFLUENCE FUNCTIONS AT EACH COLUMN LOACTION
 1210      DO 1300 M = 1, NCOL
              MI1 = MI1 + 1
              CALL CALR2 (RCOL(N),ZECOL(N),RCOL(M),ZECOL(M),R,GAM)
              IF (R.LE.0.0) THEN
               SM1 = 0.0
              ELSE
               SM1 = FUNCW (R)
              ENDIF
C
            S(NJ1,MI1) = SM1
 1300      CONTINUE
 1400 CONTINUE
C
 1500 RETURN
      END
C*******************************************
C  GENERATE OF LOAD VECTOR
C*******************************************
      SUBROUTINE GENLD (Q,NRS1)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Q(1)
      COMMON /LIO/ LR,LW,LC,LM
```

```
      COMMON /ANNULAR/ KCIR,ROUT,RINN
      COMMON /BOUND/ NOUT,NINN,KOUT(10),KINN(10),GAMM(10),BETA(10),
     *               NELOUT(10),NELINN(10),NCOL,RCOL(10),ZECOL(10)
      COMMON /LOAD/ UNIF,NPT,PLOAD(10),PRHO(10),PANG(10)
C
      NJ = 0
      NJ1 = NRS1
C
C  FOR INTERVALS OF THE OUTER BOUNDARY
C
      DO 50 N = 1, NOUT
         EL = ELLE(0,N)
         OMEGA1 = EL/2./ROUT
         DO 40 J = 1, NELOUT(N)
            NJ = NJ + 1
            NJ1 = NJ1 + 1
            Q(NJ) = 0
            Q(NJ1) = 0
          WRITE (LW,1000) NJ, NJ1
 1000     FORMAT (2I5)
C  FOR CONCENTRATED LOADS
            IF (NPT.LE.0) GO TO 20
            DO 10 K = 1, NPT
               CALL CALR1 (ROUT,N,J,OMEGA1,PRHO(K),PANG(K),R,GAM)
               IF (R.LE.0.0) GO TO 10
               Q(NJ)  = Q(NJ) - FUNCW (R) * PLOAD(K)
               Q(NJ1) = Q(NJ1) - FUNCSL (ROUT,PRHO(K),R,GAM) * PLOAD(K)
   10       CONTINUE
C  FOR UNIFORMLY DISTRIBUTED LOAD
   20       IF (UNIF.EQ.0) GO TO 40
            Q(NJ)  = Q(NJ) - FUNCQ (ROUT)
            Q(NJ1) = Q(NJ1) - FUNCDQ (ROUT)
   40    CONTINUE
   50 CONTINUE
C
C  FOR INTERVALS OF THE INNER BOUNDARY
C
      IF (KCIR.EQ.0) GO TO 110
      DO 100 N = 1, NINN
         EL = ELLE (1,N)
         OMEGA1 = EL/2./RINN
         DO 90 J = 1, NELINN(N)
            NJ = NJ + 1
            NJ1 = NJ1 + 1
            Q(NJ) = 0.
            Q(NJ1) = 0.
          WRITE (LW,1000) NJ, NJ1
C  FOR CONCENTRATED LOADS
            IF (NPT.LE.0) GO TO 70
            DO 60 K = 1, NPT
               CALL CALR1 (RINN,N,J,OMEGA1,PRHO(K),PANG(K),R,GAM)
               IF (R.LE.0.0) GO TO 60
               Q(NJ)  = Q(NJ) - FUNCW (R) * PLOAD(K)
               Q(NJ1) = Q(NJ1) - FUNCSL (RINN,PRHO(K),R,GAM) * PLOAD(K)
   60       CONTINUE
```

```
C  FOR UNIFORMLY DISTRIBUTED LOAD
   70        IF (UNIF.EQ.0) GO TO 90
             Q(NJ)  = Q(NJ) - FUNCQ (RINN)
             Q(NJ1) = Q(NJ1) - FUNCDQ (RINN)
   90     CONTINUE
  100 CONTINUE
C
C  FOR COLUMN LOCATIONS
C
  110 IF (NCOL.EQ.0) GO TO 300
      DO 200 N = 1, NCOL
         NJ1 = NJ1 + 1
         Q(NJ1) = 0.
        WRITE (LW,1000) NJ1
C  FOR CONCENTRATED LOADS
            DO 190 K = 1, NPT
            CALL CALR2 (PRHO(K),PANG(K),RCOL(K),ZECOL(K),R,GAM)
            IF (R.LE.0.0) GO TO 190
            Q(NJ1)  = Q(NJ1) - FUNCW (R) * PLOAD(K)
  190    CONTINUE
C  FOR UNIFORMLY DISTRIBUTED LOAD
         IF (UNIF.EQ.0.0) GO TO 200
         Q(NJ1)  = Q(NJ1) - FUNCQ (RCOL(N))
  200 CONTINUE
C
  300 RETURN
      END
C*******************************************
C  RESULTS FOR EACH RADIUS DIRECTION
C*******************************************
      SUBROUTINE RESULT (Q,WD,IW,KD,NRS1)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Q(1),WD(KD,1),W(8)
      COMMON /ANNULAR/ KCIR,ROUT,RINN
      COMMON /DISP/ KW,WDIR(20),KP,KWPT,WRHO(30),WANG(30)
C
      IF (KCIR.EQ.0) THEN
         RIN = 0.0
      ELSE
         RIN = RINN
      ENDIF
      DO 20 L = 1, KP
         WRITE (1,100) L
  100    FORMAT (5X,I5)
         IF (L.EQ.1) THEN
            RDIR = RIN + (ROUT-RIN)/100.
         ELSE IF (L.EQ.KP) THEN
            RDIR = ROUT - (ROUT-RIN)/100.
         ELSE
            RDIR = RIN + (ROUT-RIN)*(L-1.)/(KP-1)
         ENDIF
         CALL CALW (Q,RDIR,WDIR(IW),NRS1,W)
         DO 10 N = 1, 8
            WD(L,N) = W(N)
```

```
   10    CONTINUE
   20 CONTINUE
C
      RETURN
      END
C*****************************************
C  RESULTS FOR EACH LOCATION INSIDE DOMAIN
C*****************************************
      SUBROUTINE RESULT1 (Q,WD,KD,NRS1)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Q(1),WD(KD,1),W(8)
      COMMON /DISP/ KW,WDIR(20),KP,KWPT,WRHO(30),WANG(30)
C
      DO 20 L = 1, KWPT
         WRITE (1,100) L
  100    FORMAT (5X,I5)
         CALL CALW (Q,WRHO(L),WANG(L),NRS1,W)
         DO 10 N = 1, 8
            WD(L,N) = W(N)
   10    CONTINUE
   20 CONTINUE
C
      RETURN
      END
C************************************************
C  SUBPROGRAM FOR SOLVING RESULTS
C  OF EACH RADIUS DIRECTIONS OR ARBITRARY LOCATIONS
C************************************************
      SUBROUTINE CALW (Q,RHO,ZETA,NRS1,W)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Q(1),W(1),SM(16)
      COMMON /LIO/ LR,LW,LC,LM
      COMMON /CONST/ PR,PI,RIGID
      COMMON /ANNULAR/ KCIR,ROUT,RINN
      COMMON /BOUND/ NOUT,NINN,KOUT(10),KINN(10),GAMM(10),BETA(10),
     *               NELOUT(10),NELINN(10),NCOL,RCOL(10),ZECOL(10)
      COMMON /LOAD/ UNIF,NPT,PLOAD(10),PRHO(10),PANG(10)
C
      DO 1 I = 1, 8
         W(I) = 0.0
    1 CONTINUE
C
C  RESULT DUE TO EACH INTERVAL ON THE OUTER BOUNDARY
      MI1 = 0
      MI2 = NRS1
      DO 20 M = 1, NOUT
         EL = ELLE (0,M)
         DO 10 I = 1, NELOUT(M)
            MI1 = MI1 + 1
            MI2 = MI2 + 1
            CALL PBOUND (0,M,I,RHO,ZETA,SM)
            DO 5 J = 1, 8
               I1 = 2*J - 1
```

```
                I2 = 2*J
                W(J) = W(J) + Q(MI1)*SM(I1) + Q(MI2)*SM(I2)
    5       CONTINUE
   10    CONTINUE
   20 CONTINUE
C
C  RESULT DUE TO EACH INTERVAL ON THE INNER BOUNDARY
      IF (KCIR.EQ.0) GO TO 50
      DO 40 M = 1, NINN
         EL = ELLE (1,M)
         DO 30 I = 1, NELINN(M)
            MI1 = MI1 + 1
            MI2 = MI2 + 1
            CALL PBOUND (1,M,I,RHO,ZETA,SM)
            DO 25 J = 1, 8
               I1 = 2*J - 1
               I2 = 2*J
               W(J) = W(J) + Q(MI1)*SM(I1) + Q(MI2)*SM(I2)
   25       CONTINUE
   30    CONTINUE
   40 CONTINUE
C
C  RESULT DUE TO COLUMN REACTIONS
   50 IF (NCOL.EQ.0) GO TO 70
      DO 60 M = 1, NCOL
         MI2 = MI2 + 1
         CALL CALR2 (RCOL(M),ZECOL(M),RHO,ZETA,R,GAM)
         IF (R.LE.0.0) GO TO 60
         W(1) = W(1) + Q(MI2)*FUNCW (R)
         W(2) = W(2) + Q(MI2)*FUNCM1A (RCOL(M),RHO,R,GAM)
         W(3) = W(3) + Q(MI2)*FUNCM2A (RCOL(M),RHO,R,GAM)
         W(4) = W(4) + Q(MI2)*FUNCM3A (RCOL(M),RHO,R,GAM)
         W(5) = W(5) + Q(MI2)*FUNCQ1A (RCOL(M),RHO,R,GAM)
         W(6) = W(6) + Q(MI2)*FUNCV1A (RCOL(M),RHO,R,GAM)
         W(7) = W(7) + Q(MI2)*FUNCQ2A (RCOL(M),RHO,R,GAM)
         W(8) = W(8) + Q(MI2)*FUNCV2A (RCOL(M),RHO,R,GAM)
   60 CONTINUE
C
C  RESULT DUE TO CONCENTRATED LOADS
   70 IF (NPT.EQ.0) GO TO 90
      DO 80 K = 1, NPT
         CALL CALR2 (PRHO(K),PANG(K),RHO,ZETA,R,GAM)
         IF (R.LE.0.0) GO TO 80
         W(1) = W(1) + PLOAD(K)*FUNCW (R)
         W(2) = W(2) + PLOAD(K)*FUNCM1A (PRHO(K),RHO,R,GAM)
         W(3) = W(3) + PLOAD(K)*FUNCM2A (PRHO(K),RHO,R,GAM)
         W(4) = W(4) + PLOAD(K)*FUNCM3A (PRHO(K),RHO,R,GAM)
         W(5) = W(5) + PLOAD(K)*FUNCQ1A (PRHO(K),RHO,R,GAM)
         W(6) = W(6) + PLOAD(K)*FUNCV1A (PRHO(K),RHO,R,GAM)
         W(7) = W(7) + PLOAD(K)*FUNCQ2A (PRHO(K),RHO,R,GAM)
         W(8) = W(8) + PLOAD(K)*FUNCV2A (PRHO(K),RHO,R,GAM)
   80 CONTINUE
C
C  RESULT DUE TO UNIFORM LOAD
   90 IF (UNIF.EQ.0.0) GO TO 91
```

```
      W(1) = W(1) + FUNCQ (RHO)
      W(2) = W(2) + FUNCQM1 (RHO)
      W(3) = W(3) + FUNCQM2 (RHO)
      W(5) = W(5) + FUNCQQ1 (RHO)
      W(6) = W(6) + FUNCQV1 (RHO)
C
C  RESULTS AT THE BOUNDARY OF PLATES
   91 IF (RHO.EQ.ROUT) THEN
         DO 95 I = 1, NOUT
            IF (KOUT(I).NE.0) GO TO 95
            IF (I.EQ.NOUT) THEN
               P2 = DABS(2.*PI-ZETA)
               IF (P2.LE.0.00001) IP = 0
               IF (IP.EQ.0.AND.KOUT(1).EQ.0) GO TO 92
               IF (ZETA.EQ.GAMM(I).AND.KOUT(I-1).EQ.0) GO TO 92
               IF (ZETA.LT.GAMM(I)) GO TO 95
            ELSE
               IF (ZETA.EQ.0.0.AND.KOUT(NOUT).EQ.0) GO TO 92
               IF (ZETA.EQ.GAMM(I).AND.KOUT(I-1).EQ.0) GO TO 92
               IF (ZETA.EQ.GAMM(I+1).AND.KOUT(I+1).EQ.0) GO TO 92
               IF (ZETA.LT.GAMM(I).OR.ZETA.GT.GAMM(I+1)) GO TO 95
            ENDIF
   92       DO 94 N = 1, 8
               W(N) = 2.*W(N)
   94       CONTINUE
            GO TO 100
   95    CONTINUE
      ELSE
         IF (KCIR.EQ.0) GO TO 100
         IF (RHO.NE.RINN) GO TO 100
         DO 99 I = 1, NINN
            IF (KINN(I).NE.0) GO TO 99
            IF (I.EQ.NINN) THEN
               P2 = DABS(2.*PI-ZETA)
               IF (P2.LE.0.00001) IP = 0
               IF (IP.EQ.0.AND.KINN(1).EQ.0) GO TO 96
               IF (ZETA.EQ.BETA(I).AND.KINN(I-1).EQ.0) GO TO 96
               IF (ZETA.LT.BETA(I)) GO TO 99
            ELSE
               IF (ZETA.EQ.0.0.AND.KINN(NINN).EQ.0) GO TO 96
               IF (ZETA.EQ.BETA(I).AND.KINN(I-1).EQ.0) GO TO 96
               IF (ZETA.EQ.BETA(I+1).AND.KINN(I+1).EQ.0) GO TO 96
               IF (ZETA.LT.BETA(I).OR.ZETA.GT.BETA(I+1)) GO TO 99
            ENDIF
   96       DO 98 N = 1, 8
               W(N) = 2.*W(N)
   98       CONTINUE
            GO TO 100
   99    CONTINUE
      ENDIF
C
  100 RETURN
      END
```

```
C*****************************************************
C  FIND DEFLECTION AND STRESS RESULTANTS
C  AT EACH LOCATION DUE TO EACH BOUNDARY INTERVAL
C*****************************************************
      SUBROUTINE PBOUND (ID,M,I,RHO,ZET,SM)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION SM(1)
      COMMON /ANNULAR/ KCIR,ROUT,RINN
      COMMON /BOUND/ NOUT,NINN,KOUT(10),KINN(10),GAMM(10),BETA(10),
     *               NELOUT(10),NELINN(10),NCOL,RCOL(10),ZECOL(10)
C
      EL = ELLE(ID,M)
      IF (ID.EQ.0) THEN
         R1 = ROUT
         KTYPE = KOUT(M)
      ELSE
         R1 = RINN
         KTYPE = KINN(M)
      ENDIF
      OMEGA = EL/2./R1
      CALL CALR1 (R1,M,I,OMEGA,RHO,ZET,R,GAM)
      IF (R.LE.0.0) THEN
         BN = BNSERY (OMEGA)
      ELSE IF (RHO.EQ.R1) THEN
         CALL ALIMIT (GAM,OMEGA,GAM1,GAM2)
         BN1 = BNSERY (GAM1)
         BN2 = BNSERY (GAM2)
      ELSE IF (RHO.GT.0.0) THEN
         ALP = ANGLE (ID,M,I,OMEGA)
         ALP1 = ALP - OMEGA
         ALP2 = ALP + OMEGA
         CALL FNINTEG1 (RHO,ZET,R1,ALP1,ALP2,RS1,RS2)
         CALL FNINTEG2 (RHO,ZET,R1,ALP1,ALP2,RS1,RS2)
      ENDIF
      IF (KTYPE - 1) 10, 20, 30
C
C  FOR FREE EDGE : KTYPE = 0
   10 IF (RHO.LE.0.0) THEN
          SM(1) = -FUNCV (R1,RHO,R,GAM) * EL
          SM(2) =  FUNCM (R1,RHO,R,GAM) * EL
          SM(3) = -FUNCM1D (R1,RHO,R,GAM) * EL
          SM(4) =  FUNCM1C (R1,RHO,R,GAM) * EL
          SM(5) = -FUNCM2D (R1,RHO,R,GAM) * EL
          SM(6) =  FUNCM2C (R1,RHO,R,GAM) * EL
          SM(7) = -FUNCM3D (R1,RHO,R,GAM) * EL
          SM(8) =  FUNCM3C (R1,RHO,R,GAM) * EL
          SM(9) = -FUNCQ1D (R1,RHO,R,GAM) * EL
          SM(10)=  FUNCQ1C (R1,RHO,R,GAM) * EL
          SM(11)= -FUNCV1D (R1,RHO,R,GAM) * EL
          SM(12)=  FUNCV1C (R1,RHO,R,GAM) * EL
          SM(13)= -FUNCQ2D (R1,RHO,R,GAM) * EL
          SM(14)=  FUNCQ2C (R1,RHO,R,GAM) * EL
          SM(15)= -FUNCV2D (R1,RHO,R,GAM) * EL
          SM(16)=  FUNCV2C (R1,RHO,R,GAM) * EL
```

```
      ELSE
         SM(1) = -FINTV (RHO,ZET,R1,RS1,RS2)
         SM(2) =  FUNCM (R1,RHO,R,GAM) * EL
         SM(3) = -FINTM1D (RHO,ZET,R1,RS1,RS2)
         SM(4) =  FINTM1C (RHO,ZET,R1,RS1,RS2)
         SM(5) = -FINTM2D (RHO,ZET,R1,RS1,RS2)
         SM(6) =  FINTM2C (RHO,ZET,R1,RS1,RS2)
         SM(7) = -FINTM3D (RHO,ZET,R1,RS1,RS2)
         SM(8) =  FINTM3C (RHO,ZET,R1,RS1,RS2)
         SM(9) = -FINTQ1D (RHO,ZET,R1,RS1,RS2)
         SM(10)=  FINTQ1C (RHO,ZET,R1,RS1,RS2)
         SM(11)= -FINTV1D (RHO,ZET,R1,RS1,RS2)
         SM(12)=  FINTV1C (RHO,ZET,R1,RS1,RS2)
         SM(13)= -FINTQ2D (RHO,ZET,R1,RS1,RS2)
         SM(14)=  FINTQ2C (RHO,ZET,R1,RS1,RS2)
         SM(15)= -FINTV2D (RHO,ZET,R1,RS1,RS2)
         SM(16)=  FINTV2C (RHO,ZET,R1,RS1,RS2)
      ENDIF
       GO TO 40
C   FOR SIMPLE SUPPORT : KTYPE = 1
   20 IF (RHO.LE.0.0) THEN
         SM(1) =  FUNCW (R) * EL
         SM(2) =  FUNCM (R1,RHO,R,GAM) * EL
         SM(3) =  FUNCM1A (R1,RHO,R,GAM) * EL
         SM(4) =  FUNCM1C (R1,RHO,R,GAM) * EL
         SM(5) =  FUNCM2A (R1,RHO,R,GAM) * EL
         SM(6) =  FUNCM2C (R1,RHO,R,GAM) * EL
         SM(7) =  FUNCM3A (R1,RHO,R,GAM) * EL
         SM(8) =  FUNCM3C (R1,RHO,R,GAM) * EL
         SM(9) =  FUNCQ1A (R1,RHO,R,GAM) * EL
         SM(10)=  FUNCQ1C (R1,RHO,R,GAM) * EL
         SM(11)=  FUNCV1A (R1,RHO,R,GAM) * EL
         SM(12)=  FUNCV1C (R1,RHO,R,GAM) * EL
         SM(13)=  FUNCQ2A (R1,RHO,R,GAM) * EL
         SM(14)=  FUNCQ2C (R1,RHO,R,GAM) * EL
         SM(15)=  FUNCV2A (R1,RHO,R,GAM) * EL
         SM(16)=  FUNCV2C (R1,RHO,R,GAM) * EL
      ELSE
         SM(1) =  FUNCW (R) * EL
         SM(2) =  FUNCM (R1,RHO,R,GAM) * EL
         SM(3) =  FUNCM1A (R1,RHO,R,GAM) * EL
         SM(4) =  FINTM1C (RHO,ZET,R1,RS1,RS2)
         SM(5) =  FUNCM2A (R1,RHO,R,GAM) * EL
         SM(6) =  FINTM2C (RHO,ZET,R1,RS1,RS2)
         SM(7) =  FINTM3A (RHO,ZET,R1,RS1,RS2)
         SM(8) =  FINTM3C (RHO,ZET,R1,RS1,RS2)
         SM(9) =  FINTQ1A (RHO,ZET,R1,RS1,RS2)
         SM(10)=  FINTQ1C (RHO,ZET,R1,RS1,RS2)
         SM(11)=  FINTV1A (RHO,ZET,R1,RS1,RS2)
         SM(12)=  FINTV1C (RHO,ZET,R1,RS1,RS2)
         SM(13)=  FINTQ2A (RHO,ZET,R1,RS1,RS2)
         SM(14)=  FINTQ2C (RHO,ZET,R1,RS1,RS2)
         SM(15)=  FINTV2A (RHO,ZET,R1,RS1,RS2)
         SM(16)=  FINTV2C (RHO,ZET,R1,RS1,RS2)
      ENDIF
```

```
      GO TO 40
C  FOR CLAMPED SUPPORT : KTYPE = 2
   30 IF (RHO.LE.0.0) THEN
         SM(1) =  FUNCW (R) * EL
         SM(2) = -FUNCSL (R1,RHO,R,GAM) * EL
         SM(3) =  FUNCM1A (R1,RHO,R,GAM) * EL
         SM(4) = -FUNCM1B (R1,RHO,R,GAM) * EL
         SM(5) =  FUNCM2A (R1,RHO,R,GAM) * EL
         SM(6) = -FUNCM2B (R1,RHO,R,GAM) * EL
         SM(7) =  FUNCM3A (R1,RHO,R,GAM) * EL
         SM(8) = -FUNCM3B (R1,RHO,R,GAM) * EL
         SM(9) =  FUNCQ1A (R1,RHO,R,GAM) * EL
         SM(10)= -FUNCQ1B (R1,RHO,R,GAM) * EL
         SM(11)=  FUNCV1A (R1,RHO,R,GAM) * EL
         SM(12)= -FUNCV1B (R1,RHO,R,GAM) * EL
         SM(13)=  FUNCQ2A (R1,RHO,R,GAM) * EL
         SM(14)= -FUNCQ2B (R1,RHO,R,GAM) * EL
         SM(15)=  FUNCV2A (R1,RHO,R,GAM) * EL
         SM(16)= -FUNCV2B (R1,RHO,R,GAM) * EL
      ELSE
         SM(1) =  FUNCW (R) * EL
         SM(2) = -FUNCSL (R1,RHO,R,GAM) * EL
         SM(3) =  FUNCM1A (R1,RHO,R,GAM) * EL
         SM(4) = -FINTM1B (RHO,ZET,R1,RS1,RS2)
         SM(5) =  FUNCM2A (R1,RHO,R,GAM) * EL
         SM(6) = -FINTM2B (RHO,ZET,R1,RS1,RS2)
         SM(7) =  FINTM3A (RHO,ZET,R1,RS1,RS2)
         SM(8) = -FINTM3B (RHO,ZET,R1,RS1,RS2)
         SM(9) =  FINTQ1A (RHO,ZET,R1,RS1,RS2)
         SM(10)= -FINTQ1B (RHO,ZET,R1,RS1,RS2)
         SM(11)=  FINTV1A (RHO,ZET,R1,RS1,RS2)
         SM(12)= -FINTV1B (RHO,ZET,R1,RS1,RS2)
         SM(13)=  FINTQ2A (RHO,ZET,R1,RS1,RS2)
         SM(14)= -FINTQ2B (RHO,ZET,R1,RS1,RS2)
         SM(15)=  FINTV2A (RHO,ZET,R1,RS1,RS2)
         SM(16)= -FINTV2B (RHO,ZET,R1,RS1,RS2)
      ENDIF
C
   40 IF (ID.EQ.0) GO TO 50
      DO 45 J = 1, 16
         SM(J) = -SM(J)
   45 CONTINUE
C
   50 RETURN
      END
C**********************************************
C  FIND ELEMENT LENGTH
C**********************************************
      FUNCTION ELLE (ID,M)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /ANNULAR/ KCIR,ROUT,RINN
      COMMON /BOUND/ NOUT,NINN,KOUT(10),KINN(10),GAMM(10),BETA(10),
     *               NELOUT(10),NELINN(10),NCOL,RCOL(10),ZECOL(10)
```

```
C
      IF (ID.NE.0) GO TO 10
      IF (M.EQ.NOUT) THEN
         ELLE = (2.*PI - GAMM(M)) / NELOUT(M) * ROUT
      ELSE
         ELLE = (GAMM(M+1) - GAMM(M)) / NELOUT(M) * ROUT
      ENDIF
      RETURN
C
   10 IF (M.EQ.NINN) THEN
         ELLE = (2.*PI - BETA(M)) / NELINN(M) * RINN
      ELSE
         ELLE = (BETA(M+1) - BETA(M)) / NELINN(M) * RINN
      ENDIF
      RETURN
C
      END
C*****************************************************
C  FIND DISTANCE "R" BETWEEN 2 POINTS AT BOUNDARY
C*****************************************************
      SUBROUTINE CALR (R1,N,J,OMEGA1,R2,M,I,OMEGA2,R,GAM)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /ANNULAR/ KCIR,ROUT,RINN
C
      IF (R1.EQ.R2.AND.N.EQ.M.AND.J.EQ.I) GO TO 10
C
      ID1 = 0
      IF (R1.EQ.RINN) ID1 = 1
      ZETA = ANGLE (ID1,N,J,OMEGA1)
      ID2 = 0
      IF (R2.EQ.RINN) ID2 = 1
      ALPHA = ANGLE (ID2,M,I,OMEGA2)
      GAM  = ALPHA - ZETA
      TEMP  = R1*R1 + R2*R2 - 2.*R1*R2*DCOS(GAM)
      IF (TEMP.LE.0.000001) TEMP = 0.0
      R = DSQRT (TEMP)
      RETURN
C
   10 R = 0.0
      RETURN
      END
C*****************************************************
C  FIND DISTANCE "R" BETWEEN ANY POINT TO NODAL POINT
C*****************************************************
      SUBROUTINE CALR1 (R1,N,J,OMEGA1,R2,ZETA,R,GAM)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /ANNULAR/ KCIR,ROUT,RINN
C
      ID1 = 0
      IF (R1.EQ.RINN) ID1 = 1
      ALPHA = ANGLE (ID1,N,J,OMEGA1)
      GAM  = ALPHA - ZETA
      TEMP  = R1*R1 + R2*R2 - 2.*R1*R2*DCOS(GAM)
```

```
      IF (TEMP.LE.0.000001) TEMP = 0.0
      R = DSQRT (TEMP)
      RETURN
C
      END
C*******************************************************
C  FIND DISTANCE "R" BETWEEN TWO POINT
C*******************************************************
      SUBROUTINE CALR2 (R1,ALPHA,R2,ZETA,R,GAM)
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      GAM  = ALPHA - ZETA
      TEMP  = R1*R1 + R2*R2 - 2.*R1*R2*DCOS(GAM)
      IF (TEMP.LE.0.000001) TEMP = 0.0
      R = DSQRT (TEMP)
      RETURN
C
      END
C************************************************
C  FIND LOCATION OF ELEMENT
C************************************************
      FUNCTION ANGLE (ID,M,I,OME)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /BOUND/ NOUT,NINN,KOUT(10),KINN(10),GAMM(10),BETA(10),
     *               NELOUT(10),NELINN(10),NCOL,RCOL(10),ZECOL(10)
C
      IF (ID.NE.0) GO TO 10
C
      ANGLE = GAMM(M) + 2.*OME*(I-0.5)
      RETURN
C
   10 ANGLE = BETA(M) + 2.*OME*(I-0.5)
      RETURN
C
      END
C************************************************
C  FIND INFLUENCED FUNCTIONS OF VIRTUAL SYSTEM
C  **W,M,V,dW/dn,dSL/dn,dM/dn,dV/dn**
C************************************************
C  FUNCTION OF DEFLECTION
      FUNCTION FUNCW (R)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      FUNCW = R*R*DLOG(R) /8./PI/RIGID
      RETURN
      END
C+++++++++++++++++++++++++
C  FUNCTION OF SLOPE
      FUNCTION FUNCSL (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
```

```
      COMMON /CONST/ PR,PI,RIGID
C
      FUNCSL = (1.+2.*DLOG(R)) * (XI-RHO*DCOS(GR)) /8./PI/RIGID
      RETURN
      END
C++++++++++++++++++++++++++++++
C  FUNCTION OF DERIVATIVE OF SLOPE
      FUNCTION FUNCDD (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      FUNCDD = (-1.*(1.+2.*DLOG(R))*DCOS(GR) + 2.*(RHO-XI*DCOS(GR))
     *         *(XI-RHO*DCOS(GR))/R**2) /8./PI/RIGID
      RETURN
      END
C++++++++++++++++++++++++++
C  FUNCTION OF NORMAL MOMENT
      FUNCTION FUNCM (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      FUNCM = -1.*((1.+PR)*(1.+2.*DLOG(R)) + 2.*(XI-RHO*DCOS(GR))**2
     *        /R**2 + 2.*PR*RHO**2*DSIN(GR)**2/R**2) /8./PI
      RETURN
      END
C++++++++++++++++++++++++++++++
C  FUNCTION OF DERIVATIVE OF NORMAL MOMENT
      FUNCTION FUNCDM (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      CCR = RHO - XI*DCOS(GR)
      CCX = XI - RHO*DCOS(GR)
      FUNCDM = -1. * (CCR*(R**2*(1+PR)/2. - CCX**2) + CCX*(PR*RHO*XI
     *         *DSIN(GR)**2 - R**2*DCOS(GR))) /2./PI/R**4
      RETURN
      END
C++++++++++++++++++++++++++++++
C  FUNCTION OF NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FUNCV (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      FUNCV = -1.*((XI-RHO*DCOS(GR))*(1.+(1.-PR)*RHO*DCOS(GR)/2./XI)
     *        + (1.-PR)*RHO**2*DSIN(GR)**2*(RHO**2-XI**2)/2./XI/R**2)
     *        /2./PI/R**2
      RETURN
      END
C++++++++++++++++++++++++++++++
C  FUNCTION OF DERIVATIVE OF NORMAL CIRCHHOFF'S SHEAR
      FUNCTION FUNCDV (XI,RHO,R,GR)
```

```
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      CCR = RHO - XI*DCOS(GR)
      CCX = XI - RHO*DCOS(GR)
      FUNCDV = (R**2*DCOS(GR) + 2.*CCR*CCX - RHO*(1-PR)*CCX*(2.*RHO**2
     *         *DSIN(GR)**2/R**2 - DCOS(GR)**2) + (1.-PR)/2.*(DCOS(GR)
     *         - 2.*RHO*XI*DSIN(GR)**2/R**2)*(RHO*RHO - XI*XI))
     *         /2./PI/R**4
      RETURN
      END
C*****************************************************************
C  FIND INTEGRATION OF INFLUENCED FUNCTIONS
C  WHEN THE VIRTUAL LOAD IS ON THE SAME BOUNDARY
C  FINC INFLUENCED  **W,M,V,dW/dn,dSL/dn,dM/dn,dV/dn**
C*****************************************************************
C  INTEGRAL FUNCTION OF DEFLECTION
      FUNCTION SPECW (A,OME,BN)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
C     DEL = DABS(OME-PI)
      DEL1 = DABS(OME)
      IF (DEL1.LE.0.000001) THEN
       SPECW = 0.0
C     ELSE IF (DEL.LE.0.000001) THEN
C      SPECW = A**3*(-2.*PI/A/A - PI + BN) /8./PI/RIGID
      ELSE
       SPECW = A**3*((OME-DSIN(OME)) * (DLOG(2.*A**2*(1.-DCOS(OME)))
     *         - 1.) + BN) /8./PI/RIGID
      ENDIF
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++
C  INTEGRAL FUNCTION OF SLOPE
      FUNCTION SPECSL (A,OME,BN)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
C     DEL = DABS(OME-PI)
      DEL1 = DABS(OME)
      IF (DEL1.LE.0.000001) THEN
       SPECSL = 0.0
C     ELSE IF (DEL.LE.0.000001) THEN
C      SPECSL = A**2*(-2.*PI/A/A + BN) /8./PI/RIGID
      ELSE
       SPECSL = A**2*((OME-DSIN(OME)) * (DLOG(2.*A**2*(1.-DCOS(OME))))
     *          + BN) /8./PI/RIGID
      ENDIF
      RETURN
      END
```

```
C++++++++++++++++++++++++++++++++++++++++++++++
C  INTEGRAL FUNCTION OF DERIVATIVE OF SLOPE
      FUNCTION SPECDD (A,OME)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
C     DEL = DABS(OME-PI)
      DEL1 = DABS(OME)
      IF (DEL1.LE.0.000001) THEN
       SPECDD = 0.0
C     ELSE IF (DEL.LE.0.000001) THEN
C      SPECDD = A/4./RIGID
      ELSE
       SPECDD = A*(2.*OME - DSIN(OME)*(1.+DLOG(2.*A**2*(1.-DCOS(OME)))))
     *          /8./PI/RIGID
      ENDIF
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++
C  INTEGRAL FUNCTION OF NORMAL MOMENT
      FUNCTION SPECM (A,OME,BN)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
C     DEL = DABS(OME-PI)
      DEL1 = DABS(OME)
      IF (DEL1.LE.0.000001) THEN
       SPECM = 0.0
C     ELSE IF (DEL.LE.0.000001) THEN
C      SPECM = A*(1.+PR)*(2.*PI/A/A - BN) /8./PI
      ELSE
       SPECM = -1.*A*((1.+PR) * (OME*DLOG(2.*A**2*(1.-DCOS(OME))) + BN)
     *          - (1.-PR)*DSIN(OME)) /8./PI
      ENDIF
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++
C  INTEGRAL FUNCTION OF DERIVATIVE OF NORMAL MOMENT
      FUNCTION SPECDM (OME)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      SPECDM  =  -1.*(2.*PR*OME - (1.-PR)*DSIN(OME))
     *           /8./PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++
C  INTEGRAL FUNCTION OF NORMAL KIRCHHOFF'S SHEAR
      FUNCTION SPECV (OME)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
```

```
C
      SPECV = -1.*(OME + (1.-PR)/2*DSIN(OME)) /4./PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++
C  INTEGRAL FUNCTION OF DERIVATIVE OF NORMAL SHEAR
      FUNCTION SPECDV (A,OME)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      DEL1 = DABS(OME)
      IF (DEL1.LE.0.000001) THEN
       SPECDV = 0.0
      ELSE
       CC = 1. - DCOS(OME)
       SPECDV = -1.*(DCOS(OME/2.)/DSIN(OME/2.) - (1.-PR)*DSIN(OME)
     *          *DCOS(OME)/2./CC) /4./PI/A
      ENDIF
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++
C  SUBPROGRAM FOR CHANGING LIMIT OF INTEGRATIONS
C++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE ALIMIT (GR,OME,GR1,GR2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      GR1 = GR - OME
      GR2 = GR + OME
C
      IF (GR1.LT.-PI.AND.GR2.LE.-PI) THEN
         GR1 = 2.*PI + GR1
         GR2 = 2.*PI + GR2
      ELSE IF (GR1.GE.PI.AND.GR2.GT.PI) THEN
         GRT = GR1
         GR1 = 2.*PI - GR2
         GR2 = 2.*PI - GRT
      ENDIF
C
      RETURN
      END
C*********************************************
C  FIND VALUE OF BERNOULLI NUMBERS
C*********************************************
      FUNCTION BNSERY (OME)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      BNSERY = 0.0
      DO 50 I = 1, 100
         II = 2*I
         NN = 1
```

```
      DO 10 K = 2, II
         NN = NN*K
 10   CONTINUE
C
      IF (I.EQ.1) THEN
         BN = 1./6.
      ELSE IF (I.EQ.2) THEN
         BN = 1./30.
      ELSE IF (I.EQ.3) THEN
         BN = 1./42.
      ENDIF
      IF (I.LE.3) GO TO 35
C
      SERY = 1.
      DO 20 J =1, 100
         TERM = 1./(1.+J)**(2.*I)
         SERY = SERY + TERM
         IF (TERM.LE.0.0000000001) GO TO 30
 20   CONTINUE
C
 30   BN = NN*SERY/(2.**(II-1))/(PI**II)
 35   I2 = 2*I + 1
      N2 = 1
      DO 40 K = 2, I2
         N2 = N2*K
 40   CONTINUE
C
 45   TERM = 2.*BN*OME**I2/N2
      BNSERY = BNSERY + TERM
      ERR = DABS (TERM)
      IF (ERR.LE.0.0000001) GO TO 60
 50 CONTINUE
C
 60 RETURN
    END
C*************************************************************
C  FIND INTEGRATION OF INFLUENCED FUNCTION OF EACH INTERVAL
C  WHEN THE VIRTUAL LOAD IS INSIDE DOMAIN
C    **W,M,V,,dW/dn,dSL/dn,dM/dn,dV/dn**
C*************************************************************
C  INTEGRAL FUNCTION OF DEFLECTION
      FUNCTION FINTW (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX1 = 2.*RHO*XI
      RXS = RHO*RHO + XI*XI
      FINTW = XI*(FINT(34) - RX1*RXS*FINT(9) + RX1*RX1*FINT(10))
     *        /16./PI/RIGID
      RETURN
      END
C++++++++++++++++++++++++++
C  INTEGRAL FUNCTION OF SLOPE
```

```
      FUNCTION FINTSL (ID,RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      IF (ID.EQ.0) THEN
         FINTSL = XI * (FINT(35) - 2.*RHO*XI*XI*FINT(9)
     *             + 2.*RHO*RHO*XI*FINT(10)) /8./PI/RIGID
      ELSE
         FINTSL = RHO * (FINT(36) - 2.*RHO*XI*XI*FINT(9)
     *             + 2.*RHO*RHO*XI*FINT(10)) /8./PI/RIGID
      ENDIF
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  INTEGRAL FUNCTION OF DERIVATIVE OF SLOPE
      FUNCTION FINTDD (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      FINTDD = XI*(- 1.*FINT(38) + 4.*RHO*XI*FINT(1)
     *          - 2.*(RHO*RHO+XI*XI)*FINT(5)) /8./PI/RIGID
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  INTEGRAL FUNCTION OF NORMAL MOMENT
      FUNCTION FINTM (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      FINTM = -1.*XI*((1.+PR)*FINT(37) - (1.+PR)*2.*RHO*XI*FINT(9)
     *        - 4.*RHO*XI*FINT(5) + 2.*(RHO*RHO+XI*XI)*FINT(1)
     *        - 2.*RHO*RHO*(1-PR)*FINT(10)) /8./PI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  INTEGRAL FUNCTION OF DERIVATIVE OF NORMAL MOMENT
      FUNCTION FINTDM (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RH = RHO*RHO
      XX = XI*XI
      FINTDM = -1.*XI*((3.+PR)/2.*(RHO*FINT(1)-XI*FINT(5))-RHO*FINT(10)
     *         - (RH*RHO+3.*RHO*XX)*FINT(2) + (3.*RH*XI+XX*XI)*FINT(6)
     *         + (RH*RHO+(2.+PR)*RHO*XX)*FINT(11)
     *         - (1.+PR)*RH*XI*FINT(14)) /2./PI
      RETURN
```

```
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++
C  INTEGRAL FUNCTION OF NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FINTV (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RH = RHO*RHO
      XX = XI*XI
      FINTV = -1.*XI*((2.*XX-(1.-PR)*RH)/2./XI*FINT(1) -(1.+PR)/2.*RHO
     *        *FINT(5) + (1.-PR)*RH/2./XI*FINT(10) + (1.-PR)*RH*(RH-XX)
     *        /2./XI*FINT(11)) /2./PI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++
C  INTEGRAL FUNCTION OF DERIVATIVE OF NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FINTDV (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO/XI
      RH = RHO*RHO
      XX = XI*XI
      RX1 = 2.*RHO*XI
      RXS = RH + XX
      RXD = RH - XX
      PRD = 1. - PR
      FINTDV = -1.*XI*(-1.*(1.+PR)/2.*FINT(5) - PRD*RX*FINT(1)
     *         + PRD*RX*FINT(10) - (2.*RX1-PRD*RX*RXS)*FINT(2)
     *         +2.*(PR*RH+XX)*FINT(6) +(RX1+PRD*RX*(RH-2.*XX))*FINT(11)
     *         + PRD*RH*FINT(14) - 2.*RX*RH*PRD*RXD*FINT(12)
     *         + 2.*RH*PRD*RXD*FINT(15)) /2./PI
      RETURN
      END
C************************************************************
C  SUBPROGRAM TO COMPUTE INTEGRATED TERMS USED IN EACH FUNCTION
      SUBROUTINE FNINTEG (RHO,ZET,XI,AL1,AL2,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION GR(4),RS(4)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RX1 = 2.*RX
      RXS = RHO + XI
      RXD = RHO - XI
      RXA = DABS(RXD)
      RXS1 = RHO*RHO + XI*XI
      RXD1 = RHO*RHO - XI*XI
      RXA1 = DABS(RXD1)
```

```
      RXD2 = RXD1*RXD1
      CALL CALR2 (XI,AL1,RHO,ZET,R1,GR(1))
      RS1 = R1*R1
      CALL CALR2 (XI,AL2,RHO,ZET,R2,GR(2))
      RS2 = R2*R2
C
C
      IF (GR(1).LT.-PI.AND.GR(2).LE.-PI) THEN
         GR(1) = 2.*PI + GR(1)
         GR(2) = 2.*PI + GR(2)
      ELSE IF (GR(1).LT.0.0.AND.GR(2).LE.0.0) THEN
         TEM   =  GR(1)
         GR(1) = -GR(2)
         GR(2) = -TEM
         TEM = RS1
         RS1 = RS2
         RS2 = TEM
      ELSE IF (GR(1).GE.PI.AND.GR(2).GT.PI) THEN
         TEM = GR(1)
         GR(1) = 2.*PI - GR(2)
         GR(2) = 2.*PI - TEM
         TEM = RS1
         RS1 = RS2
         RS2 = TEM
      ENDIF
C
      RS(1) = RS1
      RS(2) = RS2
      II = 1
      IF (GR(1).LT.0.0) THEN
         II = 2
         GR(4) = -GR(1)
         GR(3) = 0.0
         GR(1) = 0.0
         RS(4) = RS(1)
         RS(3) = RXD*RXD
         RS(1) = RXD*RXD
      ELSE IF (GR(2).GT.PI) THEN
         II = 2
         GR(4) = PI
         GR(3) = 2*PI - GR(2)
         GR(2) = PI
         RS(4) = RXS*RXS
         RS(3) = RS(2)
         RS(2) = RXS*RXS
      ENDIF
C
      DO 10 I = 1, 47
         FINT(I) = 0.0
   10 CONTINUE
C
      I1 = -1
      I2 = 0
      DO 100 I = 1, II
         I1 = I1 + 2
```

```
         I2 = I2 + 2
C
         FN1 = 2./RXA1*DATAN(RXS/RXA*DTAN(GR(I1)/2))
         FN2 = 2./RXA1*DATAN(RXS/RXA*DTAN(GR(I2)/2))
      FINT1 = FN2 - FN1
      FINT(1) = FINT(1) + FINT1
C
C  5-INTEGRATION OF FUNCTION "COS(GR)/R**2"
      FINT(5) = FINT(5) - 1.*((GR(I2)-GR(I1))-RXS1*FINT1)/RX1
C
C  9-INTEGRATION OF FUNCTION "AL*SIN(GR)/R**2"
      GRM = (GR(I1) + GR(I2)) /2.
      RSM = RXS1 - RX1*DCOS(GRM)
      ELLE = (GR(I2) - GR(I1)) *XI
      FINT(9) = FINT(9) + GRM*DSIN(GRM)/RSM *ELLE
C
C 10-INTEGRATION OF FUNCTION "SIN(GR)**2/R**2"
      FS = DSIN(GR(I2)) - DSIN(GR(I1))
      FN1 = (RXS1*RS(I1)-RXD2)/RX1/RS(I1)
       IF (FN1.GT.1.0)  FN1 = 1.0
       IF (FN1.LT.-1.0) FN1 = -1.0
      FN2 = (RXS1*RS(I2)-RXD2)/RX1/RS(I2)
       IF (FN2.GT.1.0)  FN2 = 1.0
       IF (FN2.LT.-1.0) FN2 = -1.0
      FS1 = DASIN(FN2) - DASIN(FN1)
      FS2 = DASIN(DCOS(GR(I2))) - DASIN(DCOS(GR(I1)))
      FINT(10) = FINT(10) +(FS - RXA1/RX1*FS1 - RXS1/RX1*FS2) /RX1
C
C  2-INTEGRATION OF FUNCTION "1/R**4"
      FSR = DSIN(GR(I2))/RS(I2) - DSIN(GR(I1))/RS(I1)
      FINT(2) = FINT(2) + (RX1*FSR + RXS1*FINT1) /RXD2
C
C  6-INTEGRATION OF FUNCTION "COS(GR)/R**4"
      FINT(6) = FINT(6) + (RXS1*FSR + RX1*FINT1) /RXD2
C
C 11-INTEGRATION OF FUNCTION "SIN(GR)**2/R**4"
      FINT(11) = FINT(11) +(-1.*FSR+RXS1/RX1/RXA1*FS1+FS2/RX1) /RX1
C
C 14-INTEGRATION OF FUNCTION "COS(GR)*SIN(GR)**2/R**4"
      FINT(14) = FINT(14) +(-1.*FS -RXS1*FSR + (RHO**4+XI**4)/RX/RXA1
     *          *FS1 + RXS1/RX*FS2) /RX1/RX1
C
C 12-INTEGRATION OF FUNCTION "SIN(GR)**2/R**6"
      FN1 = ((RXS1*RS(I1)-RXD2)/2./RXD2/RS(I1)**2)*DSIN(GR(I1))
      FN2 = ((RXS1*RS(I2)-RXD2)/2./RXD2/RS(I2)**2)*DSIN(GR(I2))
      FINT(12) = FINT(12) + (FN2 - FN1 + RX/RXD2/RXA1*FS1) /RX1
C
C 15-INTEGRATION OF FUNCTION "COS(GR)*SIN(GR)**2/R**6"
      FN1 = (RXS1/2./RS(I1)**2-RXS1*RXS1/2./RS(I1)/RXD2-1./RS(I1))
     *      *DSIN(GR(I1))
      FN2 = (RXS1/2./RS(I2)**2-RXS1*RXS1/2./RS(I2)/RXD2-1./RS(I2))
     *      *DSIN(GR(I2))
      FSR  = (3.*RXS1/2./RXA1 - RXS1**3/2./RXD2/RXA1) /RX1
      FINT(15) = FINT(15) - (FN2 - FN1 + FSR*FS1 + FS2/RX1) /RX1/RX1
C
```

```
C 34-INTEGRATE BY PARTS OF FUNCTION "W"
      FN1 = DLOG(RS(I1))
      FN2 = DLOG(RS(I2))
      FINT(34) = FINT(34) + (RXS1*GR(I2)-RX1*DSIN(GR(I2)))*FN2
     *           - (RXS1*GR(I1)-RX1*DSIN(GR(I1)))*FN1
C
C 35,36-INTEGRATE BY PARTS OF FUNCTION "dW/dn"
      FN1 = 1. + FN1
      FN2 = 1. + FN2
      FINT(35) = FINT(35) + (XI*GR(I2)-RHO*DSIN(GR(I2)))*FN2
     *           - (XI*GR(I1)-RHO*DSIN(GR(I1)))*FN1
      FINT(36) = FINT(36) + (RHO*GR(I2)-XI*DSIN(GR(I2)))*FN2
     *           - (RHO*GR(I1)-XI*DSIN(GR(I1)))*FN1
C
C 37-INTEGRATE BY PARTS OF FUNCTION "M"
      FINT(37) = FINT(37) + GR(I2)*FN2 - GR(I1)*FN1
C
C 38-INTEGRATE BY PARTS OF FUNCTION "dSL/dn"
      FINT(38) = FINT(38) + DSIN(GR(I2))*FN2 - SIN(GR(I1))*FN1
C
  100 CONTINUE
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++
C  SUBPROGRAM TO COMPUTE A SERY USED IN INTEGRATED FUNCTION
      FUNCTION FNSERY (RHO,XI,RS1,RS2)
C
      IMPLICIT REAL*16(A-E,G,H,O-Z)
      IMPLICIT REAL*8(F)
C      COMMON /CONST/ PR,PI,RIGID
C
      RX = 2.*RHO*XI
      RXS = RHO*RHO + XI*XI
         TFNSERY = (RXS*(QLOG(RS2)-QLOG(RS1)) - RS2 + RS1) / RX
         DO 100 I = 1, 300
          AN1 = -1.0
          AN2 = 0.0
          ANN = 1.0
          DO 10 K = 1, I
           AN1 = AN1 + 2.0
           AN2 = AN2 + 2.0
           ANN = ANN * AN1 / AN2
   10     CONTINUE
          II = 2*I
          I1 = II + 1
          SUM = (RXS/RX)**I1*(QLOG(RS2)-QLOG(RS1)) *ANN/I1
          DO 50 J = 1, I1
             MM = I1 - J
             IF (J.GE.MM) THEN
                M1 = J + 1
                M2 = MM
                IER = 1
             ELSE
                M1 = MM + 1
                M2 = J
```

```
                IER = 0
             ENDIF
             AM1 = 1.0
             DO 20 K = 1, M2
                AM1 = AM1*FLOAT(K)
   20        CONTINUE
             AM2 = 1.0
             DO 30 K = M1, I1
                AM2 = AM2*FLOAT(K)
   30        CONTINUE
             SUM1 = (-1)**J*AM2/AM1*(RXS/RX)**MM
     *              *((RS2/RX)**J-(RS1/RX)**J)/J
             SUM1 = SUM1 *ANN/I1
             SUM = SUM + SUM1
   50     CONTINUE
C
   60     TFNSERY = TFNSERY + SUM
          ERR = QABS (SUM)
          IF (ERR.LE.0.000001) GO TO 200
          IF (I.EQ.1) GO TO 90
          IF (ERR.GT.ERR1) GO TO 200
   90     ERR1 = ERR
  100    CONTINUE
  200 FNSERY = DBLEQ(TFNSERY)
      RETURN
      END
C******************************************************
C  INFLUENCED FUNCTION FOR UNIFORMLY DISTRIBUTED LOAD
C******************************************************
C  INTEGRAL OF DEFLECTION
      FUNCTION FUNCQ (RHO)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /ANNULAR/ KCIR,ROUT,RINN
      COMMON /LOAD/ UNIF,NPT,PLOAD(10),PRHO(10),PANG(10)
C
      RAT = RHO/ROUT
      FUNCQ = UNIF*ROUT**4*(RAT**4 + 4.*RAT**2*(2.*DLOG(ROUT) + 1.)
     *        + 4.*DLOG(ROUT) - 1.) /64./RIGID
      IF (KCIR.EQ.0) GO TO 10
      RAT = RHO/RINN
      FUNCQ = FUNCQ - UNIF*RINN**4*(DLOG(RHO)*(2.*RAT**2 + 1.) + 1.)
     *        /16./RIGID
   10 RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++
C  INTEGRAL OF DERIVATIVE OF DEFLECTION
      FUNCTION FUNCDQ (RHO)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /ANNULAR/ KCIR,ROUT,RINN
      COMMON /LOAD/ UNIF,NPT,PLOAD(10),PRHO(10),PANG(10)
C
```

```
      RAT = RHO/ROUT
      FUNCDQ = UNIF*ROUT**3*(RAT**3 + 2.*RAT*(2.*DLOG(ROUT) + 1.))
     *         /16./RIGID
      IF (KCIR.EQ.0) GO TO 10
      RAT = RHO/RINN
      FUNCDQ = FUNCDQ - UNIF*RINN**3*(2.*RAT*(2.*DLOG(RHO) + 1.)
     *         + 1./RAT) /16./RIGID
   10 RETURN
      END
C***********************************************
C  FIND MOMENT & SHEAR OF INFLUENCED FUNCTION
C   *** W,dW/dn,M,V ***
C***********************************************
C  NORMAL BENDING MOMENT DUE TO DEFLECTION
      FUNCTION FUNCM1A (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      FUNCM1A = -1.*((1.+PR)*(1.+2.*DLOG(R)) + 2.*(RHO-XI*DCOS(GR))**2
     *          /R**2 + 2.*PR*XI**2*DSIN(GR)**2/R**2) /8./PI
      RETURN
      END
C++++++++++++++++++++++++++++++
C  TRANSVERSE BENDING MOMENT DUE TO DEFLECTION
      FUNCTION FUNCM2A (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      FUNCM2A = -1.*((1.+PR)*(1.+2.*DLOG(R)) + 2.*(RHO-XI*DCOS(GR))**2
     *          *PR/R**2 + 2.*XI**2*DSIN(GR)**2/R**2) /8./PI
      RETURN
      END
C++++++++++++++++++++++++++++++
C  TWISTING MOMENT DUE TO DEFLECTION
      FUNCTION FUNCM3A (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      FUNCM3A = -1.*(1.-PR)*XI*DSIN(GR)*(RHO-XI*DCOS(GR))/4./PI/R**2
      RETURN
      END
C++++++++++++++++++++++++++++++
C  NORMAL SHEAR DUE TO DEFLECTION
      FUNCTION FUNCQ1A (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      FUNCQ1A = -1.*(RHO-XI*DCOS(GR))/2./PI/R**2
      RETURN
      END
```

```
C+++++++++++++++++++++++++++++++
C  NORMAL KIRCHHOFF'S SHEAR DUE TO DEFLECTION
      FUNCTION FUNCV1A (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      FUNCV1A = -1.*(RHO-XI*DCOS(GR))/2./PI/R**2
      IF (RHO.LE.0.0) GO TO 10
      FUNCV1A = FUNCV1A - (1.-PR)*(XI*DCOS(GR)*(RHO-XI*DCOS(GR))
     *          + XI**2*DSIN(GR)**2*(XI**2-RHO**2)/R**2)
     *          /RHO/4./PI/R**2
   10 RETURN
      END
C+++++++++++++++++++++++++++++++
C  TRANSVERSE SHEAR DUE TO DEFLECTION
      FUNCTION FUNCQ2A (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      FUNCQ2A = XI*DSIN(GR) /2./PI/R**2
      RETURN
      END
C+++++++++++++++++++++++++++++++
C  TRANSVERSE KIRCHHOFF'S SHEAR DUE TO DEFLECTION
      FUNCTION FUNCV2A (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      FUNCV2A = XI*DSIN(GR)*((3-PR)/2.-(1-PR)*(RHO-XI*DCOS(GR))**2/R/R)
     *          /2./PI/R**2
      RETURN
      END
C*******************************************
C  NORMAL BENDING MOMENT DUE TO SLOPE
      FUNCTION FUNCM1B (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      CCR = RHO - XI*DCOS(GR)
      CCX = XI - RHO*DCOS(GR)
      FUNCM1B = -1.*((1.+PR)*CCX -2.*DCOS(GR)*CCR +2.*PR*XI*DSIN(GR)**2
     *          -2.*CCR**2*CCX/R**2 -2.*PR*XI**2*DSIN(GR)**2*CCX/R**2)
     *          /4./PI/R**2
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE BENDING MOMENT DUE TO SLOPE
      FUNCTION FUNCM2B (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
```

```
C
      CCR = RHO - XI*DCOS(GR)
      CCX = XI - RHO*DCOS(GR)
      FUNCM2B = -1.*((1.+PR)*CCX -2.*PR*DCOS(GR)*CCR +2.*XI*DSIN(GR)**2
     *          -2.*PR*CCR**2*CCX/R**2 -2.*XI**2*DSIN(GR)**2*CCX/R**2)
     *          /4./PI/R**2
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++
C  TWISTING MOMENT DUE TO SLOPE
      FUNCTION FUNCM3B (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      FUNCM3B = -(1.-PR)*(RHO*DSIN(GR) -2.*XI*DSIN(GR)*DCOS(GR)
     *          -2.*XI*DSIN(GR)*(RHO-XI*DCOS(GR))*(XI-RHO*DCOS(GR))
     *          /R**2) /4./PI/R**2
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++
C  NORMAL SHEAR DUE TO SLOPE
      FUNCTION FUNCQ1B (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      CCR = RHO - XI*DCOS(GR)
      CCX = XI - RHO*DCOS(GR)
      FUNCQ1B = (DCOS(GR) + 2.*CCR*CCX/R**2) /2./PI/R**2
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++
C  NORMAL KIRCHHOFF'S SHEAR DUE TO SLOPE
      FUNCTION FUNCV1B (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      CCR = RHO - XI*DCOS(GR)
      CCR1= RHO - 2.*XI*DCOS(GR)
      CCX = XI - RHO*DCOS(GR)
      FUNCV1B = (DCOS(GR) + 2.*CCR*CCX/R**2) /2./PI/R**2
      IF (RHO.LE.0.0) GO TO 10
      FUNC = (1.-PR)*(DCOS(GR)*CCR1/R/R -2.*XI*RHO**2*DSIN(GR)**2/R**4
     *       +4.*RHO*XI**2*DSIN(GR)**2*DCOS(GR)/R**4 -2.*XI*DCOS(GR)
     *       *CCR*CCX/R**4 +8*RHO*XI**2*DSIN(GR)**2*CCR*CCX/R**6)/4./PI
      FUNCV1B = FUNCV1B - FUNC/RHO
   10 RETURN
      END
C++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE SHEAR DUE TO SLOPE
      FUNCTION FUNCQ2B (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
```

```
      COMMON /CONST/ PR,PI,RIGID
C
      CCR = RHO - XI*DCOS(GR)
      CCX = XI - RHO*DCOS(GR)
      FUNCQ2B = DSIN(GR)*(1.-2.*XI*CCX/R**2) /2./PI/R**2
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE KIRCHHOFF'S SHEAR DUE TO SLOPE
      FUNCTION FUNCV2B (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      CCR = RHO - XI*DCOS(GR)
      CCX = XI - RHO*DCOS(GR)
      FUNC1 = DSIN(GR)*(1.-2.*XI*CCX/R**2) /2./PI/R**2
      FUNC2 = (1.-PR)*(DSIN(GR)/R/R -2.*DSIN(GR)*(RHO**2-XI**2)/R**4
     *        +6.*XI*DSIN(GR)*DCOS(GR)*CCR/R**4 +8.*XI*DSIN(GR)
     *        *CCR**2*CCX/R**6) /4./PI
      FUNCV2B = FUNC1 + FUNC2
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++
C  NORMAL BENDING MOMENT DUE TO NORMAL BENDING MOMENT
      FUNCTION FUNCM1C (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      PR1 = 1.+PR
      PR2 = PR1*PR1
      PR3 = 1.-PR
      PR4 = 1.+PR*PR
      CCR = RHO - XI*DCOS(GR)
      CCX = XI - RHO*DCOS(GR)
      FUNCM1C = RIGID*(PR2/R/R +2.*PR4*DCOS(GR)**2/R/R +2.*PR*PR3
     *         *DSIN(GR)**2/R**2 -2.*PR1*CCR*CCR/R**4 -2.*CCX*CCX/R**4
     *        +8.*DCOS(GR)*CCR*CCX/R**4 +2.*PR*XI*XI*DSIN(GR)**2/R**4
     *        -4.*PR*RHO*XI*DSIN(GR)**2*DCOS(GR)/R**4 -2.*PR*CCX*CCX
     *        /R**4 +2.*PR*PR*XI*DSIN(GR)**2*CCX/R**4 -2.*PR*PR1*XI*XI
     *        *DSIN(GR)**2/R**4 -8.*PR*XI*DSIN(GR)**2*CCX/R**4
     *        -10.*PR*PR*RHO*XI*DSIN(GR)**2*DCOS(GR)/R**4 +8.*CCR*CCR
     *        *CCX*CCX/R**6 -8.*PR*RHO*XI*DSIN(GR)**2*CCR*CCX/R**6
     *        +8.*PR*PR*RHO*RHO*XI*XI*DSIN(GR)**4/R**6 +8.*PR*XI*XI
     *        *DSIN(GR)**2*CCX*CCX/R**6) /4./PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE BENDING MOMENT DUE TO NORMAL BENDING MOMENT
      FUNCTION FUNCM2C (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
```

```
      PR1 = 1.+PR
      PR2 = PR1*PR1
      PR3 = 1.-PR
      PR4 = 1.-PR*PR
      S1 = DSIN(GR)
      S2 = S1*S1
      C1 = DCOS(GR)
      C2 = C1*C1
      CCR = RHO - XI*DCOS(GR)
      CCX = XI - RHO*DCOS(GR)
      FUNCM2C = RIGID*(PR2/R/R +4.*PR*C2/R/R +2.*PR3*S2/R/R -2.*PR
     *          *PR1*CCR*CCR/R**4 -2.*PR1*CCX*CCX/R**4 +8.*PR*C1*CCR
     *          *CCX/R**4 -2.*PR4*XI*XI*S2/R**4 -4.*PR*(3.+PR)*RHO*XI
     *          *S2*C1/R**4 -8.*XI*S2*CCX/R**4 +8.*XI*XI*S2*CCX*CCX
     *          /R**6 +8.*PR*RHO*RHO*XI*XI*S2*S2/R**6 +8.*PR*CCR*CCR
     *          *CCX*CCX/R**6 -8.*PR*PR*RHO*XI*S2*CCR*CCX/R**6)/4./PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++
C  TWISTING MOMENT DUE TO NORMAL BENDING MOMENT
      FUNCTION FUNCM3C (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      CCR = RHO - XI*DCOS(GR)
      CCX = XI - RHO*DCOS(GR)
      FUNCM3C = -1.*RIGID*(1.-PR)*((1.-PR)*DSIN(GR)*DCOS(GR)/R/R
     *          +2.*PR*RHO*XI*DSIN(GR)**3/R**4 +(3.+PR)*XI*DSIN(GR)*CCR
     *          /R**4 -2.*(1.-PR)*RHO*DSIN(GR)*DCOS(GR)*CCR/R**4
     *          -2.*XI*DSIN(GR)*DCOS(GR)*CCX/R**4 -4.*PR*RHO*RHO*XI*CCR
     *          *DSIN(GR)**3/R**6 -4*XI*DSIN(GR)*CCR*CCX*CCX/R**6)/2/PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++
C  NORMAL SHEAR DUE TO NORMAL BENDING MOMENT
      FUNCTION FUNCQ1C (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      XX = XI*XI
      RH = RHO*RHO
      S1 = DSIN(GR)
      S2 = S1*S1
      C1 = DCOS(GR)
      C2 = C1*C1
      CCR = RHO - XI*C1
      CCX = XI - RHO*C1
      R2 = R*R
      R4 = R2*R2
      R6 = R2*R4
      R8 = R4*R4
      FUNC1 = -1.*(-4.*(1.+PR)*CCR/R4 -2*(3+PR)*C2*CCR/R4 -2.*(1-PR)
     *        *S2*CCR/R4 +8.*C1*CCX/R4 +4.*(1.-2.*PR)*XI*S2*C1/R4
```

```
     *       +4.*(1.+PR)*CCR**3/R6 +16.*CCR*CCX*CCX/R6 -24.*C1*CCR*CCR
     *       *CCX/R6 -4*PR*XI*S2*CCR*CCX/R6 +4.*(5-PR)*XX*S2*CCR/R6
     *       -16.*(1.-2.*PR)*RHO*XI*S2*C1*CCR/R6
     *       +4.*PR*RHO*XI*S2*C1*CCR/R6 -4*PR*RHO*XI*S2*CCX/R6
     *       +8.*PR*RHO*XX*S2*S2/R6 -8.*XX*S2*C1*CCX/R6
     *       -24.*CCR**3*CCX*CCX/R8 +24.*PR*RHO*XI*S2*CCR*CCR*CCX/R8
     *       -24*PR*RH*XX*S2*S2*CCR/R8 -24*XX*S2*CCR*CCX*CCX/R8) /2./PI
      FUNCQ1C = -1.*RIGID*FUNC1
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++
C  NORMAL KIRCHHOFF'S SHEAR DUE TO NORMAL BENDING MOMENT
      FUNCTION FUNCV1C (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      XX = XI*XI
      RH = RHO*RHO
      S1 = DSIN(GR)
      S2 = S1*S1
      C1 = DCOS(GR)
      C2 = C1*C1
      CCR = RHO - XI*C1
      CCX = XI - RHO*C1
      R2 = R*R
      R4 = R2*R2
      R6 = R2*R4
      R8 = R4*R4
      FUNC1 = -1.*(-4.*(1.+PR)*CCR/R4 -2*(3+PR)*C2*CCR/R4 -2.*(1-PR)
     *       *S2*CCR/R4 +8.*C1*CCX/R4 +4.*(1.-2.*PR)*XI*S2*C1/R4
     *       +4.*(1.+PR)*CCR**3/R6 +16.*CCR*CCX*CCX/R6 -24.*C1*CCR*CCR
     *       *CCX/R6 -4*PR*XI*S2*CCR*CCX/R6 +4.*(5-PR)*XX*S2*CCR/R6
     *       -16.*(1.-2.*PR)*RHO*XI*S2*C1*CCR/R6
     *       +4.*PR*RHO*XI*S2*C1*CCR/R6 -4*PR*RHO*XI*S2*CCX/R6
     *       +8.*PR*RHO*XX*S2*S2/R6 -8.*XX*S2*C1*CCX/R6
     *       -24.*CCR**3*CCX*CCX/R8 +24.*PR*RHO*XI*S2*CCR*CCR*CCX/R8
     *       -24*PR*RH*XX*S2*S2*CCR/R8 -24*XX*S2*CCR*CCX*CCX/R8) /2./PI
      FUNCV1C = -1.*RIGID*FUNC1
      IF (RHO.LE.0.0) GO TO 10
      FUNC2 = (1.-PR)*((1.-PR)*S2/R2 -(1.-PR)*C2/R2 +2.*(1.-4.*PR)*RHO
     *       *XI*S2*C1/R4 -(3.+PR)*XX*S2/R4 -(3.+PR)*XI*C1*CCR/R4
     *       -2.*(1.-PR)*RHO*S2*CCR/R4 +2.*(1.-PR)*RHO*C2*CCR/R4
     *       +2.*(2.-PR)*RHO*XI*S2*C1/R4 -2.*XI*S2*CCX/R4 +2.*XI*C2
     *       *CCX/R4 +12.*PR*RH*XX*S2*S2/R6 +4.*(3.+PR)*RHO*XX*S2*CCR
     *       /R6 -4.*(2.-5.*PR)*RH*XI*S2*C1*CCR/R6 -8.*RHO*XX*S2*C1
     *       *CCX/R6 +4.*XI*C1*CCR*CCX*CCX/R6 +4.*XX*S2*CCX*CCX/R6
     *       +8.*RHO*XI*S2*CCR*CCX/R6 -24.*PR*RHO*RH*XX*S2*S2*CCR/R8
     *       -24.*RHO*XX*S2*CCR*CCX*CCX/R8) /2./PI
      FUNCV1C = FUNCV1C + RIGID*FUNC2/RHO
   10 RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE SHEAR DUE TO NORMAL BENDING MOMENT
      FUNCTION FUNCQ2C (XI,RHO,R,GR)
```

```
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      XX = XI*XI
      RH = RHO*RHO
      S1 = DSIN(GR)
      S2 = S1*S1
      C1 = DCOS(GR)
      C2 = C1*C1
      CCR = RHO - XI*C1
      CCX = XI - RHO*C1
      R2 = R*R
      R4 = R2*R2
      R6 = R2*R4
      R8 = R4*R4
      PR1 = 1.+PR
      PR2 = 1.-PR
      FUNC1 = -1.*(4.*PR1*XI*S1/R4 -4.*PR2*RHO*S1*C1/R4 -2.*(1.-5.*PR)
     *        *XI*S1*C2/R4 +6.*PR2*XI*S1*S2/R4 +8.*RHO*XX*S1*C1/R6
     *        -4.*PR1*XI*S1*CCR*CCR/R6 -8.*XI*S1*CCX*CCX/R6
     *        +8.*(2.+PR)*XI*S1*C1*CCR*CCX/R6 +16.*(1.-3.*PR)*RHO*XX*S1
     *        *S2*C1/R6 -8.*RHO*XX*S1*C1*C2/R6 -8.*XX*S1*S2*CCX/R6
     *        -4.*(5.-PR)*XI*XX*S1*S2/R6 +24.*XI*S1*CCR*CCR*CCX*CCX/R8
     *        -24*PR*RHO*XX*S1*S2*CCR*CCX/R8 +24*PR*RH*XI*XX*S1*S2*S2/R8
     *        +24.*XI*XX*S1*S2*CCX*CCX/R8) /2/PI
      FUNCQ2C = -1.*RIGID*FUNC1
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE KIRCHHOFF'S SHEAR DUE TO NORMAL BENDING MOMENT
      FUNCTION FUNCV2C (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      XX = XI*XI
      RH = RHO*RHO
      S1 = DSIN(GR)
      S2 = S1*S1
      C1 = DCOS(GR)
      C2 = C1*C1
      CCR = RHO - XI*C1
      CCX = XI - RHO*C1
      R2 = R*R
      R4 = R2*R2
      R6 = R2*R4
      R8 = R4*R4
      PR1 = 1.+PR
      PR2 = 1.-PR
      FUNC1 = -1.*(4.*PR1*XI*S1/R4 -4.*PR2*RHO*S1*C1/R4 -2.*(1.-5.*PR)
     *        *XI*S1*C2/R4 +6.*PR2*XI*S1*S2/R4 +8.*RHO*XX*S1*C1/R6
     *        -4.*PR1*XI*S1*CCR*CCR/R6 -8.*XI*S1*CCX*CCX/R6
     *        +8.*(2.+PR)*XI*S1*C1*CCR*CCX/R6 +16.*(1.-3.*PR)*RHO
     *        *XX*S1*S2*C1/R6 -8*RHO*XX*S1*C1*C2/R6 -8*XX*S1*S2*CCX/R6
```

```
     *        -4.*(5.-PR)*XI*XX*S1*S2/R6 +24.*XI*S1*CCR*CCR*CCX*CCX/R8
     *        -24*PR*RHO*XX*S1*S2*CCR*CCX/R8 +24*PR*RH*XI*XX*S1*S2*S2/R8
     *        +24.*XI*XX*S1*S2*CCX*CCX/R8) /2/PI
      FUNC2 = (1.-PR)*(-4.*(1-PR)*S1*C1*CCR/R4 +2.*PR*XI*S1*S2/R4
     *        +(3.+PR)*XI*S1/R4 -2.*(1-PR)*RHO*S1*C1/R4 +2.*XI*S1*C2/R4
     *        -8.*PR*RHO*XI*S1*S2*CCR/R6 -4.*(3+PR)*XI*S1*CCR*CCR/R6
     *        +8.*(1-PR)*RHO*S1*C1*CCR*CCR/R6 +16*XI*S1*C1*CCR*CCX/R6
     *        -8.*PR*RHO*XI*S1*S2*CCR/R6 -4.*PR*RH*XI*S1*S2/R6 -4.*XI
     *        *S1*CCX*CCX/R6 +24.*PR*RH*XI*S1*S2*CCR*CCR/R8 +24.*XI
     *        *S1*CCR*CCR*CCX*CCX/R8) /2./PI
      FUNCV2C = -1.*RIGID*(FUNC1 - FUNC2)
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++
C  NORMAL BENDING MOMENT DUE TO NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FUNCM1D (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      XX = XI*XI
      RH = RHO*RHO
      S1 = DSIN(GR)
      S2 = S1*S1
      C1 = DCOS(GR)
      C2 = C1*C1
      CCR = RHO - XI*C1
      CCX = XI - RHO*C1
      R2 = R*R
      R4 = R2*R2
      R6 = R2*R4
      R8 = R4*R4
      PR1 = 1.+PR
      PR2 = 1.-PR
      FUNCM1D = RIGID*(-1.*PR2/XI/R2 +PR2*PR2*S2/XI/R2 -(4.+5.*PR
     *          -PR*PR)*XI/R4 +(5.-3.*PR-2*PR*PR)*RH/XI/R4
     *          -(1.-8.*PR-PR*PR)
     *          *RHO*C1/R4 +(1.+2.*PR-3.*PR*PR)*XI*S2/R4 +(1.-2.*PR
     *          +PR*PR)*RH*S2/XI/R4 +4.*PR2*PR2*RHO*S2*C1/R4
     *          +12.*PR1*RH*XI/R6 +8.*XX*XI/R6 -4.*PR2*RH*RH/XI/R6
     *          -4.*(5.+PR)*RHO*XX*C1/R6 +4.*(1.-3.*PR)*RH*RHO*C1/R6
     *          -8.*PR2*XX*XI*S2/R6 +(6.-20.*PR-2.*PR*PR)*RH*XI*S2/R6
     *          -(14.-12*PR-2*PR*PR)*RH*RH*S2/XI/R6 +8.*PR2*PR2*RH*RHO
     *          *S2*C1/R6 -4.*(1.-4.*PR+3.*PR*PR)*RHO*XX*S2*C1/R6
     *          +4.*PR2*PR2*RH*XI*S2*C2/R6 +12.*PR2*RH*RH*S2*CCR*CCR
     *          /XI/R8 -12.*PR2*RH*XI*S2*CCR*CCR/R8 +12.*PR*PR2*RH
     *          *XI*S2*S2*(RH-XX)/R8) /2./PI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE BENDING MOMENT DUE TO NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FUNCM2D (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
```

```
C
      XX = XI*XI
      RH = RHO*RHO
      S1 = DSIN(GR)
      S2 = S1*S1
      C1 = DCOS(GR)
      C2 = C1*C1
      CCR = RHO - XI*C1
      CCX = XI - RHO*C1
      R2 = R*R
      R4 = R2*R2
      R6 = R2*R4
      R8 = R4*R4
      PRR = PR*PR
      PR1 = 1.+PR
      PR2 = 1.-PR
      FUNCM2D = RIGID*(-1.*PR*PR2/XI/R2 -PR2*PR2*S2/XI/R2 -(3.+3.*PR
     *          +2.*PRR)*XI/R4 +(2.+3*PR-5*PRR)*RH/XI/R4 +(1.+7.*PRR)
     *          *RHO*C1/R4 -(1.+2.*PR-3.*PRR)*XI*S2/R4 -PR2*PR2
     *          *RH*S2/XI/R4 -4.*PR2*PR2*RHO*S2*C1/R4 +12.*PR1*PR
     *          *RH*XI/R6 +8.*PR*XX*XI/R6 -4.*PR*PR2*RH*RH/XI/R6
     *          -4*PR*(5.+PR)*RHO*XX*C1/R6 +4*PR*(1.-3*PR)*RH*RHO*C1/R6
     *          +8.*PR2*XX*XI*S2/R6 +(2.+4.*PR-22.*PRR)*RH*XI*S2/R6
     *          -(2.+12.*PR-14.*PRR)*RH*RH*S2/XI/R6 -8.*PR2*PR2*RH*RHO
     *          *S2*C1/R6 +4.*(1.-4.*PR+3.*PRR)*RHO*XX*S2*C1/R6
     *          -4.*PR2*PR2*RH*XI*S2*C2/R6 +12.*PR*PR2*RH*RH*S2*CCR*CCR
     *          /XI/R8 -12.*PR*PR2*RH*XI*S2*CCR*CCR/R8 +12.*PR2*RH
     *          *XI*S2*S2*(RH-XX)/R8) /2./PI
       RETURN
       END
C++++++++++++++++++++++++++++++++++++++++
C  TWISTING MOMENT DUE TO NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FUNCM3D (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      XX = XI*XI
      RH = RHO*RHO
      S1 = DSIN(GR)
      S2 = S1*S1
      C1 = DCOS(GR)
      C2 = C1*C1
      CCR = RHO - XI*C1
      CCX = XI - RHO*C1
      R2 = R*R
      R4 = R2*R2
      R6 = R2*R4
      R8 = R4*R4
      PR1 = 1.+PR
      PR2 = 1.-PR
      FUNCM3D = -1.*RIGID*PR2*(-1.*PR2*S1*C1/XI/R2 +PR1*S1*CCR/R4
     *          +2.*PR2*RHO*S1*C1*CCR/XI/R4 -2.*XI*S1*C1/R4 +2.*PR2
     *          *XI*S1*C1/R4 -2.*PR2*RHO*S1*C2/R4 -3.*PR2*RH*S1*C1
     *          /XI/R4 -8.*XI*S1*CCR*CCX/R6 +8.*PR2*RHO*RH*S2*S1/R6
```

```
     *         -4.*PR2*RHO*XX*S2*S1/R6 -4.*PR2*RHO*S1*C1*CCR*CCX/R6
     *         -4.*PR2*RHO*XI*S1*C1*CCR/R6 +4.*PR2*RH*RHO*S1*C1*CCR
     *         /XI/R6 -12.*PR2*RH*RH*S1*S2*CCR/R8 +12.*PR2*RH*XX
     *         *S1*S2*CCR/R8) /2./PI
      RETURN
      END
C++++++++++++++++++++++++++++++
C  NORMAL SHEAR DUE TO NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FUNCQ1D (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      XX = XI*XI
      RH = RHO*RHO
      S1 = DSIN(GR)
      S2 = S1*S1
      C1 = DCOS(GR)
      C2 = C1*C1
      CCR = RHO - XI*C1
      CCX = XI - RHO*C1
      R2 = R*R
      R4 = R2*R2
      R6 = R2*R4
      R8 = R4*R4
      R10 = R8*R2
      PR1 = 1.+PR
      PR2 = 1.-PR
      FUNCQ1D = -1.*RIGID*(16.*PR2*RHO/XI/R4 -2.*(1.-5.*PR)*C1/R4
     *          +(52+60*PR)*RHO*XI/R6 -44.*PR2*RH*RHO/XI/R6
     *          +(40.-96*PR)*RH*C1/R6 -(48+8.*PR)*XX*C1/R6
     *          +(16-80*PR)*RHO*XI*S2/R6 -64.*PR2*RH*RHO*S2/XI/R6
     *          -24.*(7.+PR)*RHO*XX*XI/R8 -(48.+144.*PR)*RH*RHO*XI/R8
     *          +24.*PR2*RH*RH*RHO/XI/R8 +48.*XX*XI*C1/R8
     *          +(192+96.*PR)*RH*XX*C1/R8 -48*(1-2*PR)*RH*RH*C1/R8
     *          +24.*(4.+2.*PR)*RHO*XX*XI*S2/R8 -72.*(1.-3.*PR)*RH*RHO
     *          *XI*S2/R8 +168*PR2*RH*RH*RHO*S2/XI/R8
     *          -216.*PR2*RH*RH*S2*C1/R8 +(120-216.*PR)*RH*XX*S2*C1/R8
     *          +288*PR2*RH*RHO*XX*XI*S2/R10 -192.*PR2*RH*RH*RHO*XI*S2
     *          /R10 -96.*PR2*RH*RH*RH*RHO*S2/XI/R10 -96.*PR2*RH*XX*XX
     *          *S2*C1/R10 -192.*PR2*RH*RH*XX*S2*C1/R10 +288*PR2*RH*RH
     *          *RH*S2*C1/R10 -192.*PR2*RH*RHO*XX*XI*S2*S2/R10
     *          +192.*PR2*RH*RH*RHO*XI*S2*S2/R10) /2./PI
      RETURN
      END
C++++++++++++++++++++++++++++++
C  NORMAL KIRCHHOFF'S SHEAR DUE TO NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FUNCV1D (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      XX = XI*XI
      RH = RHO*RHO
      S1 = DSIN(GR)
```

```
      S2 = S1*S1
      C1 = DCOS(GR)
      C2 = C1*C1
      CCR = RHO - XI*C1
      CCX = XI - RHO*C1
      R2 = R*R
      R4 = R2*R2
      R6 = R2*R4
      R8 = R4*R4
      R10 = R8*R2
      PR1 = 1.+PR
      PR2 = 1.-PR
      FUNCV1D = -1.*RIGID*(16.*PR2*RHO/XI/R4 -2.*(1.-5.*PR)*C1/R4
     *          +(52+60*PR)*RHO*XI/R6 -44.*PR2*RH*RHO/XI/R6
     *          +(40.-96*PR)*RH*C1/R6 -(48+8.*PR)*XX*C1/R6
     *          +(16-80*PR)*RHO*XI*S2/R6 -64.*PR2*RH*RHO*S2/XI/R6
     *          -24.*(7.+PR)*RHO*XX*XI/R8 -(48.+144.*PR)*RH*RHO*XI/R8
     *          +24.*PR2*RH*RH*RHO/XI/R8 +48.*XX*XI*C1/R8
     *          +(192+96.*PR)*RH*XX*C1/R8 -48*(1-2*PR)*RH*RH*C1/R8
     *          +24.*(4.+2.*PR)*RHO*XX*XI*S2/R8 -72.*(1.-3.*PR)*RH*RHO
     *          *XI*S2/R8 +168*PR2*RH*RH*RHO*S2/XI/R8
     *          -216.*PR2*RH*RH*S2*C1/R8 +(120-216.*PR)*RH*XX*S2*C1/R8
     *          +288*PR2*RH*RHO*XX*XI*S2/R10 -192.*PR2*RH*RH*RHO*XI*S2
     *          /R10 -96.*PR2*RH*RH*RH*RHO*S2/XI/R10 -96.*PR2*RH*XX*XX
     *          *S2*C1/R10 -192.*PR2*RH*RH*XX*S2*C1/R10 +288*PR2*RH*RH
     *          *RH*S2*C1/R10 -192.*PR2*RH*RHO*XX*XI*S2*S2/R10
     *          +192.*PR2*RH*RH*RHO*XI*S2*S2/R10) /2./PI
      IF (RHO.LE.0.0) GO TO 10
      FUNCV1D = FUNCV1D + RIGID*PR2*(PR2/XI/R2 -2.*PR2*S2/XI/R2
     *          +(1.+3.*PR)*XI/R4 +PR2*RH/XI/R4 +(3.-5.*PR)*RHO*C1/R4
     *          -2.*(1.+3.*PR)*XI*S2/R4 -2.*PR2*RH*S2/XI/R4
     *          -14*PR2*RHO*S2*C1/R4 -8*XX*XI/R6 +4*(1-3*PR)*RH*XI/R6
     *          -4*PR2*RH*RH/XI/R6 +8*PR1*RHO*XX*C1/R6 +16*XX*XI*S2/R6
     *          -8*(4-7*PR)*RH*XI*S2/R6 +8*PR2*RH*RH*S2/XI/R6
     *          +8*(1-6*PR)*RHO*XX*S2*C1/R6 -28*PR2*RH*RHO*S2*C1/R6
     *          +32*PR2*RH*XI*S2*S2/R6 -12*(1+7*PR)*RH*XX*XI*S2/R8
     *          -36*PR2*RH*RH*XI*S2/R8 -12*(5-9*PR)*RH*RHO*XX*S2*C1/R8
     *          +48*RHO*XX*XX*S2*C1/R8 +60*PR2*RHO**5*S2*C1/R8
     *          -48*(2-3*PR)*RH*XX*XI*S2*S2/R8+96*PR2*RH*RH*XI*S2*S2/R8
     *          +24*PR2*RH*RHO*XX*S2*S2*C1/R8-96*PR2*RHO**6*XI*S2*S2/R10
     *          +96*PR2*RH*RH*XX*XI*S2*S2/R10 -96*PR2*RH*RHO*XX*XX*S2
     *          *S2*C1/R10 +96*PR2*RHO**5*XX*S2*S2*C1/R10) /2/PI/RHO
   10 RETURN
      END
C+++++++++++++++++++++++++++++++
C  TRANSVERSE SHEAR DUE TO NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FUNCQ2D (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      XX = XI*XI
      RH = RHO*RHO
      S1 = DSIN(GR)
      S2 = S1*S1
```

```
      C1 = DCOS(GR)
      C2 = C1*C1
      CCR = RHO - XI*C1
      CCX = XI - RHO*C1
      R2 = R*R
      R4 = R2*R2
      R6 = R2*R4
      R8 = R4*R4
      R10 = R8*R2
      PR1 = 1.+PR
      PR2 = 1.-PR
      FUNCQ2D = RIGID*(-2*(1-5*PR)*S1/R4 -8*(6+PR)*XX*S1/R6
     *          +8*(4-5*PR)*RH*S1/R6 -16*(1-5*PR)*RHO*XI*S1*C1/R6
     *          +32*PR2*RH*RHO*S1*C1/XI/R6 +48*XX*XX*S1/R8
     *          +(144-96*PR)*RH*XX*S1/R8 -144*PR2*RHO**4*S1/R8
     *          -48*(2+PR)*RHO*XX*XI*S1*C1/R8+24*(1-3*PR)*RHO*RH*XI*S1
     *          *C1/R8 -24*PR2*RHO**5*S1*C1/XI/R8 -24*(5-9*PR)*RH
     *          *XX*S1*C2/R8 +168*PR2*RHO**4*S1*C2/R8 +96*PR2*RHO**6
     *          *S1/R10 -96*PR2*RH*XX*XX*S1/R10 +192*PR2*RH*RHO*XX
     *          *XI*S1*C1/R10 -192*PR2*RHO**5*XI*S1*C1/R10
     *          +96*PR2*RH*XX*XX*S1*C2/R10-96*PR2*RHO**6*S1*C2/R10
     *          -192*PR2*RH*RHO*XX*XI*S1*C2*C1/R10 +192*PR2*RHO**5*XI
     *          *S1*C2*C1/R10) /2./PI
      RETURN
      END
C++++++++++++++++++++++++++++++
C  TRANSVERSE KIRCHHOFF'S SHEAR DUE TO NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FUNCV2D (XI,RHO,R,GR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
C
      XX = XI*XI
      RH = RHO*RHO
      S1 = DSIN(GR)
      S2 = S1*S1
      C1 = DCOS(GR)
      C2 = C1*C1
      CCR = RHO - XI*C1
      CCX = XI - RHO*C1
      R2 = R*R
      R4 = R2*R2
      R6 = R2*R4
      R8 = R4*R4
      R10 = R8*R2
      PR1 = 1.+PR
      PR2 = 1.-PR
      FUNCV2D = RIGID*(-2*(1-5*PR)*S1/R4 -8*(6+PR)*XX*S1/R6
     *          +8*(4-5*PR)*RH*S1/R6 -16*(1-5*PR)*RHO*XI*S1*C1/R6
     *          +32*PR2*RH*RHO*S1*C1/XI/R6 +48*XX*XX*S1/R8
     *          +(144-96*PR)*RH*XX*S1/R8 -144*PR2*RHO**4*S1/R8
     *          -48*(2+PR)*RHO*XX*XI*S1*C1/R8+24*(1-3*PR)*RHO*RH*XI*S1
     *          *C1/R8 -24*PR2*RHO**5*S1*C1/XI/R8 -24*(5-9*PR)*RH
     *          *XX*S1*C2/R8 +168*PR2*RHO**4*S1*C2/R8 +96*PR2*RHO**6
     *          *S1/R10 -96*PR2*RH*XX*XX*S1/R10 +192*PR2*RH*RHO*XX
```

```
     *          *XI*S1*C1/R10 -192*PR2*RHO**5*XI*S1*C1/R10
     *          +96*PR2*RH*XX*XX*S1*C2/R10-96*PR2*RHO**6*S1*C2/R10
     *          -192*PR2*RH*RHO*XX*XI*S1*C2*C1/R10 +192*PR2*RHO**5*XI
     *          *S1*C2*C1/R10) /2./PI
      FUNCV2D = FUNCV2D + RIGID*PR2*(PR1*S1/R4 -6*PR2*S1*C2/R4
     *          +4*(5-7*PR)*RH*S1/R6 -4*(3-PR)*XX*S1/R6
     *          +8*(1+4*PR)*RHO*XI*S1*C1/R6 +20*PR2*RH*RHO*S1*C1/XI/R6
     *          -24*PR*XX*S1*C2/R6 -12*PR2*RH*S1*C2/R6
     *          -24*PR2*RHO*XI*S1*C2*C1/R6 +12*(9-5*PR)*RH*XX*S1/R8
     *          -108*PR2*RH*RH*S1/R8 -48*(3-PR)*RHO*XX*XI*S1*C1/R8
     *          +48*(2-3*PR)*RH*RHO*XI*S1*C1/R8 -24*PR2*RHO**5*S1*C1
     *          /XI/R8 -12*(5-13*PR)*RH*XX*S1*C2/R8 +48*XX*XX*S1*C2/R8
     *          +132*PR2*RH*RH*S1*C2/R8 +48*(1-2*PR)*RHO*XX*XI*S1*C2
     *          *C1/R8 -72*PR2*RH*RHO*XI*S1*C2*C1/R8 -24*PR2*RH*XX*S1
     *          *C2*C2/R8 -96*PR2*RH*RH*XX*S1/R10+96*PR2*RHO**6*S1/R10
     *          +192*PR2*RH*RHO*XX*XI*S1*C1/R10 -192*PR2*RHO**5*XI*S1
     *          *C1/R10 +192*PR2*RH*RH*XX*S1*C2/R10 -96*PR2*RH*XX*XX
     *          *S1*C2/R10 -96*PR2*RHO**6*S1*C2/R10 -192*PR2*RH*RHO*XX
     *          *XI*S1*C2*C1/R10 +192*PR2*RHO**5*XI*S1*C2*C1/R10
     *          +96*PR2*RH*XX*XX*S1*C2*C2/R10 -96*PR2*RH*RH*XX*S1*C2
     *          *C2/R10) /2./PI
      RETURN
      END
C**************************************************
C  FIND M&V INTEGRATION OF INFLUENCED FUNCTION
C    **W,M,V,,dW/dn**
C**************************************************
C  NORMAL BENDING MOMENT DUE TO DEFLECION
      FUNCTION FINTM1A (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      FINTM1A = -1.*XI*((1.+PR)*FINT(37) - (1.+PR)*2.*RHO*XI*FINT(9)
     *          - 4.*RHO*XI*FINT(5) + 2.*(RHO*RHO+XI*XI)*FINT(1)
     *          - 2.*XI*XI*(1-PR)*FINT(10)) /8./PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE BENDING MOMENT DUE TO DEFLECION
      FUNCTION FINTM2A (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      FINTM2A = -1.*XI*((1.+PR)*FINT(37) - (1.+PR)*2.*RHO*XI*FINT(9)
     *          - 4.*PR*RHO*XI*FINT(5) + 2.*PR*(RHO*RHO+XI*XI)*FINT(1)
     *          + 2.*XI*XI*(1-PR)*FINT(10)) /8./PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TWISTING MOMENT DUE TO DEFLECION
      FUNCTION FINTM3A (RHO,ZET,XI,RS1,RS2)
```

```
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      FINTM3A = -1.*XI*XI*(1.-PR)*(RHO*FINT(20) - XI*FINT(24))
     *          /4./PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  NORMAL SHEAR DUE TO DEFLECION
      FUNCTION FINTQ1A (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RH = RHO*RHO
      XX = XI*XI
      FINTQ1A = -1.*XI*(RHO*FINT(1) -XI*FINT(5)) /2./PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  NORMAL KIRCHHOFF'S SHEAR DUE TO DEFLECION
      FUNCTION FINTV1A (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RH = RHO*RHO
      XX = XI*XI
      FINTV1A = -1.*XI*((2.*RH-(1.-PR)*XX)/2./RHO*FINT(1)
     *          -(1.+PR)/2.*XI*FINT(5) +(1.-PR)*XX/2./RHO*FINT(10)
     *          +(1.-PR)*XX*(XX-RH)/2./RHO*FINT(11)) /2./PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE SHEAR DUE TO DEFLECION
      FUNCTION FINTQ2A (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RH = RHO*RHO
      XX = XI*XI
      FINTQ2A = XX*FINT(20) /2./PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE KIRCHHOFF'S SHEAR DUE TO DEFLECION
      FUNCTION FINTV2A (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
```

```
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RH = RHO*RHO
      XX = XI*XI
      FINTV2A = XX*((3.+PR)/2.*FINT(20) -(1.-PR)*RH*FINT(21)
     *          +2.*(1.-PR)*RHO*XI*FINT(25) -(1.-PR)*XX*FINT(28))
     *          /2./PI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  NORMAL BENDING MOMENT DUE TO SLOPE
      FUNCTION FINTM1B (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRD = 1. - PR
      PRS3 = 3. + PR
      FINTM1B = -1.*XI*(PRS3*XI*FINT(1) - PRS3*RHO*FINT(5)
     *          -2.*PRD*XI*FINT(10) - 2.*XI*(XX+3.*RH)*FINT(2)
     *          +2.*RHO*(3.*XX+RH)*FINT(6) +2.*XI*(2.*RH+PRD*XX)
     *          *FINT(11) -2.*PRD*RX*XI*FINT(14)) /4./PI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE BENDING MOMENT DUE TO SLOPE
      FUNCTION FINTM2B (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRD = 1. - PR
      PRS3 = 1. + 3.*PR
      FINTM2B = -1.*XI*(PRS3*XI*FINT(1) - PRS3*RHO*FINT(5)
     *          +2.*PRD*XI*FINT(10) - 2.*PR*XI*(XX+3.*RH)*FINT(2)
     *          +2.*PR*RHO*(3.*XX+RH)*FINT(6) +2.*XI*(2.*PR*RH
     *          -PRD*XX)*FINT(11) +2.*PRD*RX*XI*FINT(14)) /4./PI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TWISTING MOMENT DUE TO SLOPE
      FUNCTION FINTM3B (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
```

```
C
      RX = RHO*XI
      RX1 = 2.*RX
      RH = RHO*RHO
      XX = XI*XI
      PRD = 1. - PR
      PRS3 = 1. + 3.*PR
      FINTM3B = -1.*PRD*XI*(RHO*FINT(20) - RX1*XI*FINT(21)
     *          +XI*RX1*FINT(28)) /4./PI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  NORMAL SHEAR DUE TO SLOPE
      FUNCTION FINTQ1B (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RX1 = 2.*RX
      RH = RHO*RHO
      XX = XI*XI
      FINTQ1B = XI*(FINT(5) + 2.*RX1*FINT(2) -2.*(RH+XX)
     *          *FINT(6) -RX1*FINT(11)) /2./PI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  NORMAL KIRCHHOFF'S SHEAR DUE TO SLOPE
      FUNCTION FINTV1B (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RX1 = 2.*RX
      RH = RHO*RHO
      XX = XI*XI
      PRD = 1. - PR
      FINTV1B = XI*((1.+PR)/2.*FINT(5) + 2.*RX1*FINT(2) -2.*(RH+XX)
     *          *FINT(6) -(1.+PR)*RX*FINT(11) -PRD*XX*FINT(14)
     *          -8.*PRD*RX*XX*FINT(12) +4.*PRD*XX*(RH+XX)*FINT(15)
     *          +4.*PRD*RX*XX*FINT(17)) /2./PI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE SHEAR DUE TO SLOPE
      FUNCTION FINTQ2B (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
```

```
      RX1 = 2.*RX
      RH = RHO*RHO
      XX = XI*XI
      FINTQ2B = XI*(FINT(20) - 2.*XX*FINT(21) +RX1*FINT(25)) /2./PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE KIRCHHOFF'S SHEAR DUE TO SLOPE
      FUNCTION FINTV2B (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RX1 = 2.*RX
      RH = RHO*RHO
      XX = XI*XI
      PRD = 1. - PR
      FINTV2B = XI*((3.-PR)/2.*FINT(20) -(1.+PR)*XX*FINT(21)
     *          -PRD*RH*FINT(21) +(5.-3.*PR)*RX*FINT(25) -3.*PRD*XX
     *          *FINT(28) +4.*PRD*RH*XX*FINT(22) -8.*PRD*RX*XX
     *          *FINT(26) +4.*PRD*XX*XX*FINT(29) -4.*PRD*RH*RX
     *          *FINT(26) +8.*PRD*RH*XX*FINT(29) -4.*PRD*RX*XX
     *          *FINT(31)) /2./PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  NORMAL BENDING MOMENT DUE TO NORMAL BENDING MOMENT
      FUNCTION FINTM1C (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRR = PR*PR
      FINTM1C = RIGID*XI*((3.+2.*PR+3*PRR)*FINT(1) -(2.-2*PR+4*PRR)
     *          *FINT(10) -4.*(3+PR)*(RH+XX)*FINT(2) +8.*(3+PR)*RX
     *          *FINT(6) +((10+2.*PR)*RH+(10-6*PR)*XX)*FINT(11)
     *          -(8.-4*PR+12*PRR)*RX*FINT(14) +8.*(RH*RH+6*RH*XX
     *          +XX*XX)*FINT(3) -32*RX*(RH+XX)*FINT(7) -(8*RH*RH
     *          +(48+8*PR)*RH*XX+8*(1-PR)*XX*XX)*FINT(12) +8.*(2+PR)
     *          *RX*RH*FINT(15) +8*(2-PR)*RX*XX*FINT(15) +8.*(1+PRR)
     *          *RX*RX*FINT(17)) /4./PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE BENDING MOMENT DUE TO NORMAL BENDING MOMENT
      FUNCTION FINTM2C (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
```

```
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRR = PR*PR
      FINTM2C = RIGID*XI*((1.+6.*PR+PRR)*FINT(1) +2*(1-3*PR)*FINT(10)
     *          -(2+12*PR+2*PRR)*(RH+XX)*FINT(2) +4*(1+6*PR+PRR)*RX
     *          *FINT(6) +((2+10*PR)*RH-(10-10*PR-4*PRR)*XX)*FINT(11)
     *          +4*(2-5*PR-PRR)*RX*FINT(14) +8.*PR*(RH*RH+6*RH*XX
     *          +XX*XX)*FINT(3) -32*PR*RX*(RH+XX)*FINT(7) -(8*PR*RH*RH
     *          -(8-48*PR-16*PRR)*RH*XX-8*(1-PR)*XX*XX)*FINT(12)
     *          -((16-16*PR-8*PRR)*RX*XX-8*PR*(2+PR)*RH*RX)*FINT(15)
     *          -8.*(1-2*PR-PRR)*RX*RX*FINT(17)) /4./PI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TWISTING MOMENT DUE TO NORMAL BENDING MOMENT
      FUNCTION FINTM3C (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRR = PR*PR
      PRD = 1.-PR
      FINTM3C = -1*RIGID*XI*PRD*(PRD*FINT(24) +3.*(1+PR)*RX*FINT(21)
     *          -2*PRD*RH*FINT(25) -(5+PR)*XX*FINT(26) +4.*PRD*RX
     *          *FINT(28) -4*RX*XX*FINT(22) -4*PR*RH*RX*FINT(22)
     *          +4*(2+PR)*RH*XX*FINT(26) +4*XX*XX*FINT(26) -4*PRD
     *          *RH*RX*FINT(29) -8*RX*XX*FINT(29) +4*PRD*RH*XX
     *          *FINT(31)) /2/PI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  NORMAL SHEAR DUE TO NORMAL BENDING MOMENT
      FUNCTION FINTQ1C (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRR = PR*PR
      PRD = 1.-PR
      FINTQ1C = RIGID*XI*(-6*(3+PR)*RHO*FINT(2) +6*(3+PR)*XI*FINT(6)
     *          +4*(3+PR)*RHO*FINT(11) -12*PR*XI*FINT(14)
     *          +4*(11+PR)*RHO*RH*FINT(3) +12*(11+PR)*RX*XI*FINT(3)
     *          -12*(11+PR)*RH*XI*FINT(7) -4*(11+PR)*XI*XX*FINT(7)
     *          -40*RH*RHO**FINT(12) -32*(3+2*PR)*RX*XI*FINT(12)
```

```
     *           +4*(12+11*PR)*RH*XI*FINT(15) +12*PR*XI*XX*FINT(15)
     *           +48*PR*RX*XI*FINT(17) -24*RH*RH*RHO*FINT(4)
     *           -240*RH*RHO*XX*FINT(4) -120*RX*XX*XI*FINT(4)
     *           +120*RH*RH*XI*FINT(8) +240*RH*XX*XI*FINT(8)
     *           +24*XI**5*FINT(8) +24*RHO**5*FINT(13) +8*(33+9*PR)
     *           *RHO**3*XX*FINT(13) +24*(4+PR)*RX*XX*XI*FINT(13)
     *           -24*(3+PR)*RH*RH*XI*FINT(16) -24*(5+3*PR)*RH*XX*XI
     *           *FINT(16) -24*(2+3*PR)*RH*RHO*XX*FINT(18) -24*PR
     *           *RHO*XX*XX*FINT(18) +48*PR*RH*XX*XI*FINT(19)) /2/PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  NORMAL KIRCHHOFF'S SHEAR DUE TO NORMAL BENDING MOMENT
      FUNCTION FINTV1C (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRR = PR*PR
      PRD = 1.-PR
      FINTV1C = RIGID*XI*(-6*(3+PR)*RHO*FINT(2) +6*(3+PR)*XI*FINT(6)
     *           +4*(3+PR)*RHO*FINT(11) -12*PR*XI*FINT(14)
     *           +4*(11+PR)*RHO*RH*FINT(3) +12*(11+PR)*RX*XI*FINT(3)
     *           -12*(11+PR)*RH*XI*FINT(7) -4*(11+PR)*XI*XX*FINT(7)
     *           -40*RH*RHO**FINT(12) -32*(3+2*PR)*RX*XI*FINT(12)
     *           +4*(12+11*PR)*RH*XI*FINT(15) +12*PR*XI*XX*FINT(15)
     *           +48*PR*RX*XI*FINT(17) -24*RH*RH*RHO*FINT(4)
     *           -240*RH*RHO*XX*FINT(4) -120*RX*XX*XI*FINT(4)
     *           +120*RH*RH*XI*FINT(8) +240*RH*XX*XI*FINT(8)
     *           +24*XI**5*FINT(8) +24*RHO**5*FINT(13) +8*(33+9*PR)
     *           *RHO**3*XX*FINT(13) +24*(4+PR)*RX*XX*XI*FINT(13)
     *           -24*(3+PR)*RH*RH*XI*FINT(16) -24*(5+3*PR)*RH*XX*XI
     *           *FINT(16) -24*(2+3*PR)*RH*RHO*XX*FINT(18) -24*PR
     *           *RHO*XX*XX*FINT(18) +48*PR*RH*XX*XI*FINT(19)) /2/PI
      FINTV1C = FINTV1C +RIGID*PRD*XI/RHO*(2*PRD*FINT(10) -PRD*FINT(1)
     *           +2*PRD*RH*FINT(2) +(5+PR)*XX*FINT(2) -(7-PR)*RX*FINT(6)
     *           -4*PRD*RH*FINT(11) -2*(5+PR)*XX*FINT(11) +14*PRD*RX
     *           *FINT(14) -12*RH*XX*FINT(3) -4*XX*XX*FINT(3)
     *           +4*RH*RX*FINT(7) +12*RX*XX*FINT(7) +16*(4-PR)*RH*XX
     *           *FINT(12) +8*XX*XX*FINT(12) -20*PRD*RH*RX*FINT(15)
     *           -4*(11+PR)*RX*XX*FINT(15) -32*PRD*RH*XX*FINT(17)
     *           -24*RH*RH*XX*FINT(13) -72*RH*XX*XX*FINT(13)
     *           +72*RX**3*FINT(16) +24*RHO*XI**5*FINT(16)
     *           +24*PRD*RH*RH*XX*FINT(18) +48*RH*XX*XX*FINT(18)
     *           -24*PRD*RX**3*FINT(19)) /2/PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE SHEAR DUE TO NORMAL BENDING MOMENT
      FUNCTION FINTQ2C (RHO,ZET,XI,RS1,RS2)
C
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRR = PR*PR
      PRD = 1.-PR
      FINTQ2C = RIGID*XI*(2*(5-PR)*XI*FINT(21) -4*PRD*RHO*FINT(25)
     *          -8*(1-2*PR)*XI*FINT(28) -4*(1+PR)*RH*XI*FINT(22)
     *          -4*(9-PR)*XX*XI*FINT(22) +8*(9-4*PR)*RX*XI*FINT(26)
     *          -8*(3+PR)*RH*XI*FINT(29) +8*(1-2*PR)*XX*XI*FINT(29)
     *          -8*(2-7*PR)*RX*XI*FINT(31) +24*RH*XI*XX*FINT(23)
     *          +24*XI**5*FINT(23) -24*(2-PR)*RH*RHO*XX*FINT(27)
     *          -24*(4-PR)*RX*XX*XI*FINT(27) +24*RH*RH*XI*FINT(30)
     *          +24*(5-2*PR)*RH*XI*XX*FINT(30) -24*(2+PR)*RH*RX*XI
     *          *FINT(32) -24*PR*RHO*XX*XX*FINT(32)
     *          +48*PR*RH*XX*XI*FINT(33)) /2/PI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE KIRCHHOFF'S SHEAR DUE TO NORMAL BENDING MOMENT
      FUNCTION FINTV2C (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRR = PR*PR
      PRD = 1.-PR
      FINTV2C = RIGID*XI*(2*(5-PR)*XI*FINT(21) -4*PRD*RHO*FINT(25)
     *          -8*(1-2*PR)*XI*FINT(28) -4*(1+PR)*RH*XI*FINT(22)
     *          -4*(9-PR)*XX*XI*FINT(22) +8*(9-4*PR)*RX*XI*FINT(26)
     *          -8*(3+PR)*RH*XI*FINT(29) +8*(1-2*PR)*XX*XI*FINT(29)
     *          -8*(2-7*PR)*RX*XI*FINT(31) +24*RH*XI*XX*FINT(23)
     *          +24*XI**5*FINT(23) -24*(2-PR)*RH*RHO*XX*FINT(27)
     *          -24*(4-PR)*RX*XX*XI*FINT(27) +24*RH*RH*XI*FINT(30)
     *          +24*(5-2*PR)*RH*XI*XX*FINT(30) -24*(2+PR)*RH*RX*XI
     *          *FINT(32) -24*PR*RHO*XX*XX*FINT(32)
     *          +48*PR*RH*XX*XI*FINT(33)) /2/PI
      FINTV2C = FINTV2C + RIGID*PRD*XI*(3*(1+PR)*XI*FINT(21) -6*PRD
     *          *RHO*FINT(25) +6*PRD*XI*FINT(28) -12*(1+2*PR)*RH*XI
     *          *FINT(22) -4*XI*XX*FINT(22) +8*PRD*RH*RHO*FINT(26)
     *          +24*(2+PR)*RX*XI*FINT(26) -36*PRD*RH*XI*FINT(29)
     *          -4*(7+PR)*XX*XI*FINT(29) +24*PRD*RX*XI*FINT(31)
     *          +24*PR*RH*RH*XI*FINT(23) +24*RH*XX*XI*FINT(23)
     *          -48*(1+PR)*RH*RX*XI*FINT(27) -48*RX*XX*XI*FINT(27)
     *          +24*PRD*RH*RH*XI*FINT(30) +24*(4+PR)*RH*XX*XI*FINT(30)
     *          +24*XI**5*FINT(30) -48*PRD*RH*RX*XI*FINT(32)
     *          -48*RX*XX*XI*FINT(32) +24*PRD*RH*XI*XX*FINT(33)) /2/PI
      RETURN
```

```
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++
C  NORMAL BENDING MOMENT DUE TO NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FINTM1D (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRR = PR*PR
      PRD = 1.-PR
      FINTM1D = RIGID*XI*(-1*PRD/XI*FINT(1) +PRD*PRD/XI*FINT(10)
     *          -(4+5*PR-PRR)*XI*FINT(2) +(5-3*PR-2*PRR)*RH/XI*FINT(2)
     *          -(1-8*PR-PRR)*RHO*FINT(6) +(1+2*PR-3*PRR)*XI*FINT(11)
     *          +(1-2*PR+PRR)*RH/XI*FINT(11) +4*PRD*PRD*RHO*FINT(14)
     *          -4*PRD*RH*RH/XI*FINT(3) +12*(1+PR)*RH*XI*FINT(3)
     *          +8*XI*XX*FINT(3) +4*(1-3*PR)*RH*RHO*FINT(7)
     *          -4*(5+PR)*RX*XI*FINT(7) -(14-12*PR-2*PRR)*RH*RH/XI
     *          *FINT(12) +(10-28*PR+2*PRR)*RH*XI*FINT(12)
     *          -8*PRD*XI*XX*FINT(12) +8*PRD*PRD*RH*RHO*FINT(15)
     *          -4*(1-4*PR+3*PRR)*RX*XI*FINT(15) -4*PRD*PRD*RH*XI
     *          *FINT(17) +12*PRD*RH**3/XI*FINT(13) -12*PRD*RH*XX*XI
     *          *FINT(13) -24*PRD*RHO**5*FINT(16) +24*PRD*RH*RX*XI
     *          *FINT(16) -12*PRD*PRD*RH*RH*XI*FINT(18)
     *          +12*PRD*PRD*RH*XX*XI*FINT(18)) /2/PI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE BENDING MOMENT DUE TO NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FINTM2D (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRR = PR*PR
      PRD = 1.-PR
      FINTM2D = RIGID*XI*(-1*PR*PRD/XI*FINT(1) -PRD*PRD/XI*FINT(10)
     *          -(3+3*PR+2*PRR)*XI*FINT(2) +(2+3*PR-5*PRR)*RH/XI*FINT(2)
     *          +(1+7*PRR)*RHO*FINT(6) -(1+2*PR-3*PRR)*XI*FINT(11)
     *          -(1-2*PR+PRR)*RH/XI*FINT(11) -4*PRD*PRD*RHO*FINT(14)
     *          +12*PR*(1+PR)*RH*XI*FINT(3) -4*PR*PRD*RH*RH/XI*FINT(3)
     *          +8*PR*XI*XX*FINT(3) +4*PR*(1-3*PR)*RH*RHO*FINT(7)
     *          -4*PR*(5+PR)*RX*XI*FINT(7) -(2+12*PR-14*PRR)*RH*RH/XI
     *          *FINT(12) -(2-12*PR+26*PRR)*RH*XI*FINT(12)
     *          +8*PRD*XI*XX*FINT(12) -8*PRD*PRD*RH*RHO*FINT(15)
     *          +4*(1-4*PR+3*PRR)*RX*XI*FINT(15) +4*PRD*PRD*RH*XI
     *          *FINT(17) +12*PR*PRD*RH**3/XI*FINT(13) -12*PR*PRD*RH*XX
     *          *XI*FINT(13) -24*PR*PRD*RHO**5*FINT(16) +24*PR*PRD*RH
```

```
     *          *RX*XI*FINT(16) +12*PRD*PRD*RH*RH*XI*FINT(18)
     *          -12*PRD*PRD*RH*XX*XI*FINT(18)) /2/PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TWISTING MOMENT DUE TO NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FINTM3D (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRR = PR*PR
      PRD = 1.-PR
      FINTM3D = -1*RIGID*PRD*XI*(-1*PRD/XI*FINT(24) +(1+PR)*RHO*FINT(21)
     *          -PRD*RH/XI*FINT(25) -(1+3*PR)*XI*FINT(25) -4*PRD*RHO
     *          *FINT(28) +8*PRD*RH*RHO*FINT(22) -4*(3-PR)*RX*XI
     *          *FINT(22) +4*PRD*RH*RH/XI*FINT(26) +8*PR*RH*XI*FINT(26)
     *          +8*XX*XI*FINT(26) -8*PRD*RH*RHO*FINT(29) +4*(1-3*PR)
     *          *RX*XI*FINT(29) -4*PRD*RH*XI*FINT(31) -12*PRD*RHO**5
     *          *FINT(23) +12*PRD*RH*RX*XI*FINT(23) +12*PRD*RH*RH*XI
     *          *FINT(27) -12*PRD*RH*XX*XI*FINT(27) +12*PRD*RHO**5
     *          *FINT(30) -12*PRD*RH*RX*XI*FINT(30) -12*PRD*RH*RH*XI
     *          *FINT(32) +12*PRD*RH*XX*XI*FINT(32)) /2/PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  NORMAL SHEAR DUE TO NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FINTQ1D (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRR = PR*PR
      PRD = 1.-PR
      FINTQ1D = RIGID*XI*(16*PRD*RHO/XI*FINT(2) -2*(1-5*PR)*FINT(6)
     *          +(52+60*PR)*RHO*XI*FINT(3) -44*PRD*RH*RHO/XI*FINT(3)
     *          +(40-96*PR)*RH*FINT(7) -(48+8*PR)*XX*FINT(7)
     *          +(16-80*PR)*RHO*XI*FINT(12) -64*PRD*RH*RHO/XI*FINT(12)
     *          -24*(7+PR)*RX*XX*FINT(4) -(48+144*PR)*RX*RH*FINT(4)
     *          +24*PRD*RHO**5/XI*FINT(4) +48*XX*XX*FINT(8)
     *          +(192+96*PR)*RH*XX*FINT(8) -48*(1-2*PR)*RH*RH*FINT(8)
     *          +24*(4+2*PR)*RX*XX*FINT(13) -72*(1-3*PR)*RX*RH*FINT(13)
     *          +168*PRD*RH*RH*RHO/XI*FINT(13) -216*PRD*RH*RH*FINT(16)
     *          +(120-216*PR)*RH*XX*FINT(16) +288*PRD*RX**3*FINT(39)
     *          -192*PRD*RH*RH*RX*FINT(39) -96*PRD*RHO**7/XI*FINT(39)
     *          -96*PRD*RH*XX*XX*FINT(40) -192*PRD*RH*RH*XX*FINT(40)
     *          +288*PRD*RH**3*FINT(40) -192*PRD*RX**3*FINT(41)
```

```
     *         +192*PRD*RH*RH*RX*FINT(41)) /2./PI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  NORMAL KIRCHHOFF'S SHEAR DUE TO NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FINTV1D (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRR = PR*PR
      PRD = 1.-PR
      FINTV1D = RIGID*XI*(16*PRD*RHO/XI*FINT(2) -2*(1-5*PR)*FINT(6)
     *         +(52+60*PR)*RHO*XI*FINT(3) -44*PRD*RH*RHO/XI*FINT(3)
     *         +(40-96*PR)*RH*FINT(7) -(48+8*PR)*XX*FINT(7)
     *         +(16-80*PR)*RHO*XI*FINT(12) -64*PRD*RH*RHO/XI*FINT(12)
     *         -24*(7+PR)*RX*XX*FINT(4) -(48+144*PR)*RX*RH*FINT(4)
     *         +24*PRD*RHO**5/XI*FINT(4) +48*XX*XX*FINT(8)
     *         +(192+96*PR)*RH*XX*FINT(8) -48*(1-2*PR)*RH*RH*FINT(8)
     *         +24*(4+2*PR)*RX*XX*FINT(13) -72*(1-3*PR)*RX*RH*FINT(13)
     *         +168*PRD*RH*RH*RHO/XI*FINT(13) -216*PRD*RH*RH*FINT(16)
     *         +(120-216*PR)*RH*XX*FINT(16) +288*PRD*RX**3*FINT(39)
     *         -192*PRD*RH*RH*RX*FINT(39) -96*PRD*RHO**7/XI*FINT(39)
     *         -96*PRD*RH*XX*XX*FINT(40) -192*PRD*RH*RH*XX*FINT(40)
     *         +288*PRD*RH**3*FINT(40) -192*PRD*RX**3*FINT(41)
     *         +192*PRD*RH*RH*RX*FINT(41)) /2./PI
      FINTV1D = FINTV1D + RIGID*PRD*XI/RHO*(PRD/XI*FINT(1)
     *         -2*PRD/XI*FINT(10) +(1+3*PR)*XI*FINT(2)
     *         +PRD*RH/XI*FINT(2) +(3-5*PR)*RHO*FINT(6)
     *         -2*(1+3*PR)*XI*FINT(11) -2*PRD*RH/XI*FINT(11)
     *         -14*PRD*RHO*FINT(14) -8*XX*XI*FINT(3)
     *         +4*(1-3*PR)*RH*XI*FINT(3) -4*PRD*RH*RH/XI*FINT(3)
     *         +8*(1+PR)*RHO*XX*FINT(7) +16*XX*XI*FINT(12)
     *         -8*(4-7*PR)*RH*XI*FINT(12) +8*PRD*RH*RH/XI*FINT(12)
     *         +8*(1-6*PR)*RHO*XX*FINT(15) -28*PRD*RH*RHO*FINT(15)
     *         +32*PRD*RH*XI*FINT(17) -12*(1+7*PR)*RH*XX*XI*FINT(13)
     *         -36*PRD*RH*RH*XI*FINT(13)-12*(5-9*PR)*RX*RH*XI*FINT(16)
     *         +48*RX*XX*XI*FINT(16) +60*PRD*RHO**5*FINT(16)
     *         -48*(2-3*PR)*RH*XX*XI*FINT(18) +96*PRD*RH*RH*XI
     *         *FINT(18) +24*PRD*RH*RX*XI*FINT(19) -96*PRD*RH**3
     *         *XI*FINT(41) +96*PRD*RH*RH*XX*XI*FINT(41) -96*PRD*RH
     *         *RHO*XX*XX*FINT(42) +96*PRD*RHO**5*XX*FINT(42)) /2/PI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE SHEAR DUE TO NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FINTQ2D (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
```

```
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRR = PR*PR
      PRD = 1.-PR
      FINTQ2D = RIGID*XI*(-2.*(1-5*PR)*FINT(21) -8*(6+PR)*XX*FINT(22)
     *          +8*(4-5*PR)*RH*FINT(22) -16*(1-5*PR)*RX*FINT(26)
     *          +32*PRD*RH*RHO/XI*FINT(26) +48*XX*XX*FINT(23)
     *          +(144-96*PR)*RH*XX*FINT(23) -144*PRD*RH*RH*FINT(23)
     *          -48*(2+PR)*RX*XX*FINT(27) +24*(1-3*PR)*RX*RH*FINT(27)
     *          -24*PRD*RH*RH*RHO/XI*FINT(27) -24*(5-9*PR)*RH*XX
     *          *FINT(30) +168*PRD*RH*RH*FINT(30) +96*PRD*RH**3
     *          *FINT(43) -96*PRD*RH*XX*XX*FINT(43) +192*PRD*RH*RHO
     *          *XX*XI*FINT(44) -192*PRD*RH*RH*RX*FINT(44)
     *          +96*PRD*RH*XX*XX*FINT(45) -96*PRD*RH**3*FINT(45)
     *          -192*PRD*RH*RX*XX*FINT(46) +192*PRD*RH*RH*RX*FINT(46))
     *          /2./PI
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  TRANSVERSE KIRCHHOFF'S SHEAR DUE TO NORMAL KIRCHHOFF'S SHEAR
      FUNCTION FINTV2D (RHO,ZET,XI,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RH = RHO*RHO
      XX = XI*XI
      PRR = PR*PR
      PRD = 1.-PR
      FINTV2D = RIGID*XI*(-2.*(1-5*PR)*FINT(21) -8*(6+PR)*XX*FINT(22)
     *          +8*(4-5*PR)*RH*FINT(22) -16*(1-5*PR)*RX*FINT(26)
     *          +32*PRD*RH*RHO/XI*FINT(26) +48*XX*XX*FINT(23)
     *          +(144-96*PR)*RH*XX*FINT(23) -144*PRD*RH*RH*FINT(23)
     *          -48*(2+PR)*RX*XX*FINT(27) +24*(1-3*PR)*RX*RH*FINT(27)
     *          -24*PRD*RH*RH*RHO/XI*FINT(27) -24*(5-9*PR)*RH*XX
     *          *FINT(30) +168*PRD*RH*RH*FINT(30) +96*PRD*RH**3
     *          *FINT(43) -96*PRD*RH*XX*XX*FINT(43) +192*PRD*RH*RHO
     *          *XX*XI*FINT(44) -192*PRD*RH*RH*RX*FINT(44)
     *          +96*PRD*RH*XX*XX*FINT(45) -96*PRD*RH**3*FINT(45)
     *          -192*PRD*RH*RX*XX*FINT(46) +192*PRD*RH*RH*RX*FINT(46))
     *          /2./PI
      FINTV2D = FINTV2D + RIGID*PRD*XI*((1+PR)*FINT(21)
     *          -6*PRD*FINT(28) +4*(5-7*PR)*RH*FINT(22)
     *          -4*(3-PR)*XX*FINT(22) +8*(1+4*PR)*RX*FINT(26)
     *          +20*PRD*RH*RHO/XI*FINT(26) -24*PR*XX*FINT(29)
     *          -12*PRD*RH*FINT(29) -24*PRD*RX*FINT(31)
     *          +12*(9-5*PR)*RX*RX*FINT(23) -108*PRD*RH*RH*FINT(23)
     *          -48*(3-PR)*RX*XX*FINT(27) +48*(2-3*PR)*RX*RH*FINT(27)
     *          -24*PRD*RH*RH*RHO/XI*FINT(27) -12*(5-13*PR)*RH*XX
     *          *FINT(30) +48*XX*XX*FINT(30) +132*PRD*RH*RH*FINT(30)
     *          +48*(1-2*PR)*RX*XX*FINT(32) -72*PRD*RX*RH*FINT(32)
```

```
     *          -24*PRD*RH*XX*FINT(33) -96*PRD*RH*RH*XX*FINT(43)
     *          +96*PRD*RH**3*FINT(43) +192*PRD*RX**3*FINT(44)
     *          -192*PRD*RX*RH*RH*FINT(44) +192*PRD*RH*RH*XX*FINT(45)
     *          -96*PRD*RH*XX*XX*FINT(45) -96*PRD*RH**3*FINT(45)
     *          -192*PRD*RX**3*FINT(46) +192*PRD*RH*RH*RX*FINT(46)
     *          +96*PRD*RH*XX*XX*FINT(47) -96*PRD*RH*RH*XX*FINT(47))
     *          /2/PI
      RETURN
      END
C*****************************************************************
C  SUBPROGRAM TO COMPUTE INTEGRATED TERMS USED IN EACH FUNCTION
      SUBROUTINE FNINTEG1 (RHO,ZET,XI,AL1,AL2,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION GR(4),RS(4)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RX1 = 2.*RX
      RXS = RHO + XI
      RXD = RHO - XI
      RXA = DABS(RXD)
      RXS1 = RHO*RHO + XI*XI
      RXS2 = RXS1*RXS1
      RXD1 = RHO*RHO - XI*XI
      RXD2 = RXD1*RXD1
      RXA1 = DABS(RXD1)
      CALL CALR2 (XI,AL1,RHO,ZET,R1,GR(1))
      RS1 = R1*R1
      CALL CALR2 (XI,AL2,RHO,ZET,R2,GR(2))
      RS2 = R2*R2
C
C
      IF (GR(1).LT.-PI.AND.GR(2).LE.-PI) THEN
         GR(1) = 2.*PI + GR(1)
         GR(2) = 2.*PI + GR(2)
      ELSE IF (GR(1).LT.0.0.AND.GR(2).LE.0.0) THEN
         TEM  =  GR(1)
         GR(1) = -GR(2)
         GR(2) = -TEM
         TEM = RS1
         RS1 = RS2
         RS2 = TEM
      ELSE IF (GR(1).GE.PI.AND.GR(2).GT.PI) THEN
         TEM = GR(1)
         GR(1) = 2.*PI - GR(2)
         GR(2) = 2.*PI - TEM
         TEM = RS1
         RS1 = RS2
         RS2 = TEM
      ENDIF
C
      RS(1) = RS1
      RS(2) = RS2
```

```
      II = 1
      IF (GR(1).LT.0.0) THEN
         II = 2
         GR(4) = -GR(1)
         GR(3) = 0.0
         GR(1) = 0.0
         RS(4) = RS(1)
         RS(3) = RXD*RXD
         RS(1) = RXD*RXD
      ELSE IF (GR(2).GT.PI) THEN
         II = 2
         GR(4) = PI
         GR(3) = 2*PI - GR(2)
         GR(2) = PI
         RS(4) = RXS*RXS
         RS(3) = RS(2)
         RS(2) = RXS*RXS
      ENDIF
C
      DO 10 I = 1, 47
         FINT(I) = 0.0
   10 CONTINUE
C
      I1 = -1
      I2 = 0
      DO 100 I = 1, II
         I1 = I1 + 2
         I2 = I2 + 2
C
C  1-INTEGRATION OF FUNCTION "1/R**2"
         FN1 = 2./RXA1*DATAN(RXS/RXA*DTAN(GR(I1)/2))
         FN2 = 2./RXA1*DATAN(RXS/RXA*DTAN(GR(I2)/2))
      FINT1 = FN2 - FN1
      FINT(1) = FINT(1) + FINT1
C
C  2-INTEGRATION OF FUNCTION "1/R**4"
      FN = DSIN(GR(I2))/RS(I2) - DSIN(GR(I1))/RS(I1)
      FINT(2) = FINT(2) + (RX1*FN + RXS1*FINT1) /RXD2
C
C  3-INTEGRATION OF FUNCTION "1/R**6"
      FN1 = DSIN(GR(I2))/RS(I2)**2 - DSIN(GR(I1))/RS(I1)**2
      CC1 = 3.*RX*RXS1/RXD2
      CC2 = (2.*RXS2 + RX1*RX1) /2./RXD2
      FINT(3) = FINT(3) + (CC1*FN + RX*FN1 + CC2*FINT1) /RXD2
C
C  4-INTEGRATION OF FUNCTION "1/R**8"
      FN2 = DSIN(GR(I2))/RS(I2)**3 - DSIN(GR(I1))/RS(I1)**3
      CC1 = (11.*RX*RXS2 + 2.*RX1**3) /RXD2/RXD2
      CC2 = 5.*RX*RXS1/RXD2
      CC3 = 3.*RXS1*(2.*RXS2 + 3.*RX1*RX1) /RXD2/RXD2/2.
      FINT(4) = FINT(4) + (CC1*FN + CC2*FN1 + RX1*FN2
     *            + CC3*FINT1) /RXD2/3.
C
C  5-INTEGRATION OF FUNCTION "COS(GR)/R**2"
      FINT(5) = FINT(5) - 1.*((GR(I2)-GR(I1))-RXS1*FINT1)/RX1
```

```
C
C  6-INTEGRATION OF FUNCTION "COS(GR)/R**4"
      FINT(6) = FINT(6) + (RXS1*FN + RX1*FINT1) /RXD2
C
C  7-INTEGRATION OF FUNCTION "COS(GR)/R**6"
      CC1 = (RXS2 + 2.*RX1*RX1)/RXD2
      CC2 = 3.*RXS1*RX1/RXD2
      FINT(7) = FINT(7) + (CC1*FN + RXS1*FN1 + CC2*FINT1) /RXD2/2.
C
C  8-INTEGRATION OF FUNCTION "COS(GR)/R**8"
      CC1 = (2.*RXS2*RXS1 + 13.*RXS1*RX1*RX1)/RXD2/RXD2/2.
      CC2 = (2.*RXS2 + 3.*RX1*RX1)/RXD2/2.
      CC3 = (12.*RX1*RXS2 + 3.*RX1**3)/RXD2/RXD2/2.
      FINT(8) = FINT(8) + (CC1*FN + CC2*FN1 + RXS1*FN2
     *          + CC3*FINT1) /RXD2/3.
C
C  9-INTEGRATION OF FUNCTION "AL*SIN(GR)/R**2"
      GRM = (GR(I2) + GR(I1)) /2.
      RSM = RXS1 - RX1*DCOS(GRM)
      ELLE = (GR(I2) - GR(I1)) *XI
      FINT(9) = FINT(9) + GRM*DSIN(GRM)/RSM *ELLE
C
C 10-INTEGRATION OF FUNCTION "SIN(GR)**2/R**2"
      FN1 = DSIN(GR(I2)) - DSIN(GR(I1))
      CC1 = (RXS1*RS(I1)-RXD2)/RX1/RS(I1)
       IF (CC1.GT.1.0)  CC1 = 1.0
       IF (CC1.LT.-1.0) CC1 = -1.0
      CC2 = (RXS1*RS(I2)-RXD2)/RX1/RS(I2)
       IF (CC2.GT.1.0)  CC2 = 1.0
       IF (CC2.LT.-1.0) CC2 = -1.0
      FN2 = DASIN(CC2) - DASIN(CC1)
      FN3 = DASIN(DCOS(GR(I2))) - DASIN(DCOS(GR(I1)))
      FINT(10) = FINT(10) + (FN1 - RXA1/RX1*FN2 - RXS1/RX1*FN3) /RX1
C
C 11-INTEGRATION OF FUNCTION "SIN(GR)**2/R**4"
      FINT(11) = FINT(11) + (-1.*FN+RXS1/RX1/RXA1*FN2+FN3/RX1) /RX1
C
C 12-INTEGRATION OF FUNCTION "SIN(GR)**2/R**6"
      FNN1 = ((RXS1*RS(I1)-RXD2)/2./RXD2/RS(I1)**2)*DSIN(GR(I1))
      FNN2 = ((RXS1*RS(I2)-RXD2)/2./RXD2/RS(I2)**2)*DSIN(GR(I2))
      FINTT1 = (FNN2 - FNN1 + RX/RXD2/RXA1*FN2) /RX1
      FINT(12) = FINT(12) + FINTT1
C
C 13-INTEGRATION OF FUNCTION "SIN(GR)**2/R**8"
      FNN1 = ((RXS2*RS(I1)-RXS1*RXD2)/2./RXD2**2/RS(I1)**2)
     *       *DSIN(GR(I1))
      FNN2 = ((RXS2*RS(I2)-RXS1*RXD2)/2./RXD2**2/RS(I2)**2)
     *       *DSIN(GR(I2))
      FN4 = (DSIN(GR(I2))/RS(I2))**3 - (DSIN(GR(I1))/RS(I1))**3
      CC1 = RX1*RX1/3./RXD2
      CC2 = (RXS2*RXS1 - RXS1*RXD2)/2./RXD2/RXD2/RX1/RXA1
      FINTT2 = (FNN2 - FNN1 + CC1*FN4 + CC2*FN2) /RX1
      FINT(13) = FINT(13) + FINTT2
C
C 39-INTEGRATION OF FUNCTION "SIN(GR)**2/R**10"
```

```
      FNN1 = DSIN(GR(I2))**3/RS(I2)**4 - DSIN(GR(I1))**3/RS(I1)**4
      CC1 = RX1/4./RXD2
      CC2 = 5*RXS1/4./RXD2
      FINTT3 = CC1*FNN1 + CC2*FINTT2 - FINTT1/4./RXD2
      FINT(39) = FINT(39) + FINTT3
C
C 14-INTEGRATION OF FUNCTION "COS(GR)*SIN(GR)**2/R**4"
      FINT(14) = FINT(14) + (-1.*FN1 - RXS1*FN + (RHO**4+XI**4)/RX/RXA1
     *          *FN2 + RXS1/RX*FN3) /RX1/RX1
C
C 15-INTEGRATION OF FUNCTION "COS(GR)*SIN(GR)**2/R**6"
      FNN1 = (RXS1/2./RS(I1)**2-RXS1*RXS1/2./RS(I1)/RXD2-1./RS(I1))
     *       *DSIN(GR(I1))
      FNN2 = (RXS1/2./RS(I2)**2-RXS1*RXS1/2./RS(I2)/RXD2-1./RS(I2))
     *       *DSIN(GR(I2))
      FNN3 = (3.*RXS1/2./RXA1 - RXS1**3/2./RXD2/RXA1) /RX1
      FINT(15) = FINT(15) - (FNN2 - FNN1 + FNN3*FN2 + FN3/RX1)/RX1/RX1
C
C 16-INTEGRATION OF FUNCTION "COS(GR)*SIN(GR)**2/R**8"
      FNN1 = (RXD2 - RXS1*RS(I1)) /RXD2**2/RS(I1)**2/2.*DSIN(GR(I1))
      FNN2 = (RXD2 - RXS1*RS(I2)) /RXD2**2/RS(I2)**2/2.*DSIN(GR(I2))
      CC1 = RXS1/3./RXD2
      CC2 = RX1/2./RXD2/RXD2/RXA1
      FINT(16) = FINT(16) + (CC1*FN4 - FNN2 + FNN1 + CC2*FN2)
C
C 40-INTEGRATION OF FUNCTION "COS(GR)*SIN(GR)**2/R**10"
      FINT(40) = FINT(40) + FINTT3*RXS1/RX1 - FINTT2/RX1
C
C 17-INTEGRATION OF FUNCTION "SIN(GR)**4/R**6"
      FNN1 = (RXD2-RXS1*RS(I1)) /(RXD2*RS(I1))**2*RX*DSIN(GR(I1))**5
      FNN2 = (RXD2-RXS1*RS(I2)) /(RXD2*RS(I2))**2*RX*DSIN(GR(I2))**5
      FNN3 = (RXD2-RXS1*RS(I1)+2.*RXS2) /2/RX1/RXD2**2*DSIN(GR(I1))**3
      FNN4 = (RXD2-RXS1*RS(I2)+2.*RXS2) /2/RX1/RXD2**2*DSIN(GR(I2))**3
      FNN5 = 3.*(RXD2-RXS1*RS(I1)+2.*RXS2) /2/RX1**3/RXD2*DSIN(GR(I1))
      FNN6 = 3.*(RXD2-RXS1*RS(I2)+2.*RXS2) /2/RX1**3/RXD2*DSIN(GR(I2))
      CC1 = 3.*(RHO**4+XI**4)/RX1**4/RXA1
      CC2 = 3.*RXS1/RX1**4
      FINT1 = FNN2-FNN1 +FNN4-FNN3 -FNN6+FNN5 +CC1*FN2 + CC2*FN3
      FINT(17) = FINT(17) + FINT1
C
C 18-INTEGRATION OF FUNCTION "SIN(GR)**4/R**8"
      FNN1 = (2.*RXD2**2+RXD2*RXS1*RS(I1)+(4.*RXD2-RXS2)*RS(I1)**2)
     *       /6./(RXD2*RS(I1))**3*RX1*DSIN(GR(I1))**5
      FNN2 = (2.*RXD2**2+RXD2*RXS1*RS(I2)+(4.*RXD2-RXS2)*RS(I2)**2)
     *       /6./(RXD2*RS(I2))**3*RX1*DSIN(GR(I2))**5
      FNN3 = (7.*RXD2*RXS1-2*RXS1**3+(RXS2-4.*RXD2)*RS(I1))
     *       /6./RX1/RXD2**3*DSIN(GR(I1))**3
      FNN4 = (7.*RXD2*RXS1-2*RXS1**3+(RXS2-4.*RXD2)*RS(I2))
     *       /6./RX1/RXD2**3*DSIN(GR(I2))**3
      FNN5 = (5.*RXD2*RXS1-2*RXS1**3+(RXS2-2.*RXD2)*RS(I1))
     *       /2./RX1**3/RXD2**2*DSIN(GR(I1))
      FNN6 = (5.*RXD2*RXS1-2*RXS1**3+(RXS2-2.*RXD2)*RS(I2))
     *       /2./RX1**3/RXD2**2*DSIN(GR(I2))
      CC1 = (RXS1**3-3.*RXD2*RXS1)/2./RX1**4/RXD2/RXA1
      FINT2 = FNN2-FNN1 -FNN4+FNN3 +FNN6-FNN5 +CC1*FN2 - FN3/RX1**4
```

```
      FINT(18) = FINT(18) + FINT2
C
C 41-INTEGRATION OF FUNCTION "SIN(GR)**4/R**10"
      FNN1 = DSIN(GR(I2))**5/RS(I2)**4 - DSIN(GR(I1))**5/RS(I1)**4
      CC1 = RX1/4./RXD2
      CC2 = 3*RXS1/4./RXD2
      FINT3 = CC1*FNN1 + CC2*FINT2 + FINT1/4./RXD2
      FINT(41) = FINT(41) + FINT3
C
C 19-INTEGRATION OF FUNCTION "COS(GR)*SIN(GR)**4/R**8"
      FINT(19) = FINT(19) + (RXS1/RX1*FINT2 - FINT1/RX1)
C
C 42-INTEGRATION OF FUNCTION "COS(GR)*SIN(GR)**4/R**10"
      FINT(42) = FINT(42) + FINT3*RXS1/RX1 - FINT2/RX1
C
C 34-INTEGRATE BY PARTS OF FUNCTION "W"
      FN1 = DLOG(RS(I1))
      FN2 = DLOG(RS(I2))
      FINT(34) = FINT(34) + (RXS1*GR(I2)-RX1*DSIN(GR(I2)))*FN2
     *           - (RXS1*GR(I1)-RX1*DSIN(GR(I1)))*FN1
C
C 35,36-INTEGRATE BY PARTS OF FUNCTION "dW/dn"
      FN1 = 1. + FN1
      FN2 = 1. + FN2
      FINT(35) = FINT(35) + (XI*GR(I2)-RHO*DSIN(GR(I2)))*FN2
     *           - (XI*GR(I1)-RHO*DSIN(GR(I1)))*FN1
      FINT(36) = FINT(36) + (RHO*GR(I2)-XI*DSIN(GR(I2)))*FN2
     *           - (RHO*GR(I1)-XI*DSIN(GR(I1)))*FN1
C
C 37-INTEGRATE BY PARTS OF FUNCTION "M"
      FINT(37) = FINT(37) + GR(I2)*FN2 - GR(I1)*FN1
C
C 38-INTEGRATE BY PARTS OF FUNCTION "dSL/dn"
      FINT(38) = FINT(38) + DSIN(GR(I2))*FN2 - DSIN(GR(I1))*FN1
C
  100 CONTINUE
      RETURN
      END
C***************************************************************
C  SUBPROGRAM TO COMPUTE INTEGRATED TERMS USED IN EACH FUNCTION
      SUBROUTINE FNINTEG2 (RHO,ZET,XI,AL1,AL2,RS1,RS2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION GR(4),RS(2)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /FINTEG/ FINT(47)
C
      RX = RHO*XI
      RX1 = 2.*RX
      RXS = RHO + XI
      RXD = RHO - XI
      RXA = DABS(RXD)
      RXS1 = RHO*RHO + XI*XI
      RXS2 = RXS1*RXS1
      RXD1 = RHO*RHO - XI*XI
```

```
      RXD2 = RXD1*RXD1
      RXA1 = DABS(RXD1)
      I1 = 1
      CALL CALR2 (XI,AL1,RHO,ZET,R1,GR(I1))
      RS(I1) = R1*R1
      I2 = 2
      CALL CALR2 (XI,AL2,RHO,ZET,R2,GR(I2))
      RS(I2) = R2*R2
C
C 20-INTEGRATION OF FUNCTION "SIN(GR)/R**2"
      FN = DLOG(RS(I2)) - DLOG(RS(I1))
      FINT(20) = FN/RX1
C
C 21-INTEGRATION OF FUNCTION "SIN(GR)/R**4"
      FN1 = 1/RS(I2) - 1./RS(I1)
      FINT(21) = -1.*FN1/RX1
C
C 22-INTEGRATION OF FUNCTION "SIN(GR)/R**6"
      FN2 = 1/RS(I2)**2 - 1./RS(I1)**2
      FINT(22) = -1.*FN2/RX1/2.
C
C 23-INTEGRATION OF FUNCTION "SIN(GR)/R**8"
      FN3 = 1/RS(I2)**3 - 1./RS(I1)**3
      FINT(23) = -1.*FN3/RX1/3.
C
C 43-INTEGRATION OF FUNCTION "SIN(GR)/R**10"
      FN4 = 1/RS(I2)**4 - 1./RS(I1)**4
      FINT(43) = -1.*FN4/RX1/4.
C
C 24-INTEGRATION OF FUNCTION "SIN(GR)*COS(GR)/R**2"
      FNN1 = DCOS(GR(I2)) - DCOS(GR(I1))
      FINT(24) = FNN1/RX1 + RXS1/RX1/RX1*FN
C
C 25-INTEGRATION OF FUNCTION "SIN(GR)*COS(GR)/R**4"
      FINT(25) = -1.*RXS1/RX1/RX1*FN1 - FN/RX1/RX1
C
C 26-INTEGRATION OF FUNCTION "SIN(GR)*COS(GR)/R**6"
      FINT(26) = FN1/RX1/RX1 - RXS1/2/RX1/RX1*FN2
C
C 27-INTEGRATION OF FUNCTION "SIN(GR)*COS(GR)/R**8"
      FINT(27) = FN2/2./RX1/RX1 - RXS1/3./RX1/RX1*FN3
C
C 44-INTEGRATION OF FUNCTION "SIN(GR)*COS(GR)/R**10"
      FINT(44) = FN3/3./RX1/RX1 - RXS1/4./RX1/RX1*FN4
C
C 28-INTEGRATION OF FUNCTION "SIN(GR)*COS(GR)**2/R**4"
      FINT(28) = -1.*FNN1/RX1/RX1 - RXS2/RX1**3*FN1
     *           - 2.*RXS1/RX1**3*FN
C
C 29-INTEGRATION OF FUNCTION "SIN(GR)*COS(GR)**2/R**6"
      FNN2 = DCOS(GR(I2))/RS(I2)**2 - DCOS(GR(I1))/RS(I1)**2
      CC1 = 3.*RXS2/2./RX1**3
      CC2 = 2.*RXS1/RX1**2
      FINT(29) = FN2*CC1 - CC2*FNN2 + FN/RX1**3
C
```

```
C 30-INTEGRATION OF FUNCTION "SIN(GR)*COS(GR)**2/R**8"
      FNN3 = DCOS(GR(I2))/RS(I2)**3 - DCOS(GR(I1))/RS(I1)**3
      FNN4 = DCOS(GR(I2))**2/RS(I2)**3 - DCOS(GR(I1))**2/RS(I1)**3
      CC1 = RXS1/RX1**2
      CC2 = RXS2/RX1**3/3
      FINT(30) = -1.*FNN4/RX1 + CC1*FNN3 - FN3*CC2
C
C 45-INTEGRATION OF FUNCTION "SIN(GR)*COS(GR)**2/R**10"
      FINT1 = DCOS(GR(I2))/RS(I2)**4 - DCOS(GR(I1))/RS(I1)**4
      FINT2 = DCOS(GR(I2))**2/RS(I2)**4 - DCOS(GR(I1))**2/RS(I1)**4
      FINT(45) = FINT1*RXD1/RX1/RX1/3. - FINT2/RX1/2.
     *           - FN4*RXS2/RX1**3/12.
C
C 31-INTEGRATION OF FUNCTION "SIN(GR)*COS(GR)**3/R**6"
      FNN5 = (DCOS(GR(I2))/RS(I2))**2 - (DCOS(GR(I1))/RS(I1))**2
      FNN6 = DCOS(GR(I2))**3/RS(I2)**2 - DCOS(GR(I1))**3/RS(I1)**2
      CC1 = 2.*RXS1/RX1**2
      CC2 = 5.*RXS2*RXS1/RX1**4/2.
      CC3 = 3.*RXS1/RX1**4
      FINT(31) = FNN6/RX1 - CC1*FNN5 - FNN2*CC1/RX1
     *           + CC2*FN2 - CC3*FN
C
C 32-INTEGRATION OF FUNCTION "SIN(GR)*COS(GR)**3/R**8"
      CC1 = 3.*RXS1/RX1**2
      CC2 = 9.*RXS2/RX1**3/2.
      CC3 = 11.*RXS2*RXS1/RX1**4/6.
      FINTT1 = -1.*FNN4*CC1 + CC2*FNN3 - FN3*CC3 - FN/RX1**4
      FINT(32) = FINTT1
C
C 46-INTEGRATION OF FUNCTION "SIN(GR)*COS(GR)**3/R**10"
      FINT3 = DCOS(GR(I2))**3/RS(I2)**4 - DCOS(GR(I1))**3/RS(I1)**4
      CC1 = 3.*RXS1/RX1**2/2.
      CC2 = RXS2/RX1**3
      CC3 = RXS2*RXS1/RX1**4/4.
      FINT(46) = -1.*FINT3/RX1 + CC1*FINT2 - CC2*FINT1 + CC3*FN4
C
C 33-INTEGRATION OF FUNCTION "SIN(GR)*COS(GR)**4/R**8"
      FNN5 = DCOS(GR(I2))**4/RS(I2)**3 - DCOS(GR(I1))**4/RS(I1)**3
      CC1 = 12.*RXS2/RX1**3
      CC2 = 18.*RXS2*RXS1/RX1**4.
      CC3 = 22.*RXS2*RXS2/RX1**5/3.
      FINT(33) = FNN5/RX1 - CC1*FNN4 + CC2*FNN3
     *           - FN3*CC3 - FN*4.*RXS1/RX1**5
C
C 47-INTEGRATION OF FUNCTION "SIN(GR)*COS(GR)**4/R**10"
      FINT1 = DCOS(GR(I2))**4/RS(I2)**4 - DCOS(GR(I1))**4/RS(I1)**4
      FINT(47) = -1.*FINT1/RX1/4. - FINTT1/RX1
C
      RETURN
      END
C****************************************************
C  INFLUENCED FUNCTION FOR UNIFORMLY DISTRIBUTED LOAD
C****************************************************
C  INTEGRAL OF NORMAL BENDING MOMENT
      FUNCTION FUNCQM1 (RHO)
```

```
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /ANNULAR/ KCIR,ROUT,RINN
      COMMON /LOAD/ UNIF,NPT,PLOAD(10),PRHO(10),PANG(10)
C
      RAT = RHO/ROUT
      PR1 = 1.+PR
      PR2 = 1.-PR
      PR3 = 3.+PR
      FUNCQM1 = -1.*UNIF*ROUT**2*(PR3*RAT**2 + 2.*PR1*(2.*DLOG(ROUT)
     *        + 1.)) /16.
      IF (KCIR.EQ.0) GO TO 10
      RAT = RHO/RINN
      FUNCQM1 = FUNCQM1 - UNIF*RINN**2*(PR2/RAT**2 -4.*PR1*DLOG(RHO)
     *        -2.*PR3) /16.
   10 RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++
C  INTEGRAL OF TRANSVERSE BENDING MOMENT
      FUNCTION FUNCQM2 (RHO)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /ANNULAR/ KCIR,ROUT,RINN
      COMMON /LOAD/ UNIF,NPT,PLOAD(10),PRHO(10),PANG(10)
C
      RAT = RHO/ROUT
      PR1 = 1.+PR
      PR2 = 1.-PR
      PR3 = 1.+3.*PR
      FUNCQM2 = -1.*UNIF*ROUT**2*(PR3*RAT**2 + 2.*PR1*(2.*DLOG(ROUT)
     *          + 1.)) /16.
      IF (KCIR.EQ.0) GO TO 10
      RAT = RHO/RINN
      FUNCQM2 = FUNCQM2 + UNIF*RINN**2*(PR2/RAT**2 +4.*PR1*DLOG(RHO)
     *          + 2.*PR3) /16.
   10 RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++
C  INTEGRAL OF NORMAL SHEAR
      FUNCTION FUNCQQ1 (RHO)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /ANNULAR/ KCIR,ROUT,RINN
      COMMON /LOAD/ UNIF,NPT,PLOAD(10),PRHO(10),PANG(10)
C
      FUNCQQ1 = -1.*UNIF*RHO/2.
      IF (KCIR.EQ.0) GO TO 10
      FUNCQQ1 = FUNCQV1 + UNIF*RINN**2/RHO/2.
   10 RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++
C  INTEGRAL OF NORMAL KIRCHHOFF'S SHEAR
```

```
      FUNCTION FUNCQV1 (RHO)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CONST/ PR,PI,RIGID
      COMMON /ANNULAR/ KCIR,ROUT,RINN
      COMMON /LOAD/ UNIF,NPT,PLOAD(10),PRHO(10),PANG(10)
C
      FUNCQV1 = -1.*UNIF*RHO/2.
      IF (KCIR.EQ.0) GO TO 10
      FUNCQV1 = FUNCQV1 + UNIF*RINN**2/RHO/2.
   10 RETURN
      END
C==================================
C  LABEL PRINTING SUBROUTINE
C==================================
      SUBROUTINE LABEL (A1,A2,A3,A4,A5,A6,A7)
C
      COMMON /LIO/ LR,LW,LC,LM
C
      WRITE (LW,2000) A1,A2,A3,A4,A5,A6,A7
      RETURN
 2000 FORMAT (/5X,7A4)
      END
C================================
C  PRINTING OF 1-D ARRAY
C================================
      SUBROUTINE DISP1 (A,NC)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(1)
      COMMON /LIO/ LR,LW,LC,LM
C
      J1 = 0
      J2 = 0
      ICJ = 9
   10 J1 = J2 + 1
      J2 = J1 + ICJ
      IF (J2.GE.NC) J2 = NC
      WRITE (LM,2000) (J,J=J1, J2)
      WRITE (LM,2001)
      WRITE (LM,2010) (A(J),J=J1, J2)
      WRITE (LM,2001)
      IF (J2.NE.NC) GO TO 10
      RETURN
 2000 FORMAT (12X,10I10)
 2001 FORMAT (/5X,' ')
 2010 FORMAT (2X,10X,10F10.5)
      END
C================================
C  WRITE 1-D ARRAY ON TAPE # 6
C================================
      SUBROUTINE WTAPE1 (A,NC)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(1)
```

```
      COMMON /LIO/ LR,LW,LC,LM
C
      J1 = 0
      J2 = 0
      ICJ = 9
   10 J1 = J2 + 1
      J2 = J1 + ICJ
      IF (J2.GE.NC) J2 = NC
      WRITE (LC,2000) (J,J=J1, J2)
      WRITE (LC,2001)
      WRITE (LC,2010) (A(J),J=J1, J2)
      WRITE (LC,2001)
      IF (J2.NE.NC) GO TO 10
      RETURN
 2000 FORMAT (6X,10I14)
 2001 FORMAT (/5X,' ')
 2010 FORMAT (2X,10X,10D14.5)
      END
C==========================================
C  PRINTING OF 2-D ARRAY
C==========================================
      SUBROUTINE DISP2 (A,NR,NC)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NR,1)
      COMMON /LIO/ LR,LW,LC,LM
C
      J1 = 0
      J2 = 0
      ICJ = 9
   10 J1 = J2 + 1
      J2 = J1 + ICJ
      IF (J2.GE.NC) J2 = NC
      WRITE (LM,2000) (J,J=J1,J2)
      WRITE (LM,2001)
      DO 100 I = 1, NR
        IP = I
        WRITE (LM,2010) IP,(A(I,J),J=J1,J2)
  100 CONTINUE
      WRITE (LM,2001)
      WRITE (LM,2001)
      IF (J2.NE.NC) GO TO 10
      RETURN
 2000 FORMAT  (12X,10I10)
 2001 FORMAT  (/5X,' ')
 2010 FORMAT  (2X,I5,5X,10F10.5)
      END
C==========================================
C  WRITE 2-D ARRAY ON TAPE
C==========================================
      SUBROUTINE WTAPE2 (A,NR,NC)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NR,1)
      COMMON /LIO/ LR,LW,LC,LM
```

```
C
      J1 = 0
      J2 = 0
      ICJ = 9
   10 J1 = J2 + 1
      J2 = J1 + ICJ
      IF (J2.GE.NC) J2 = NC
      WRITE (LC,2000) (J,J=J1,J2)
      WRITE (LC,2001)
      DO 100 I = 1, NR
        IP = I
        WRITE (LC,2010) IP,(A(I,J),J=J1,J2)
  100 CONTINUE
      WRITE (LC,2001)
      WRITE (LC,2001)
      IF (J2.NE.NC) GO TO 10
      RETURN
 2000 FORMAT  (6X,10I14)
 2001 FORMAT  (/5X,' ')
 2010 FORMAT  (2X,I5,5X,10D14.5)
      END
C++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE WTAPE2A (A,NR)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NR,1)
      COMMON /LIO/ LR,LW,LC,LM
C
      WRITE (LC,2001)
      WRITE (LC,2002)
      WRITE (LC,2001)
      DO 100 I = 1, NR
        WRITE (LC,2010) I,(A(I,J),J=1,8)
  100 CONTINUE
      WRITE (LC,2001)
      WRITE (LC,2001)
      RETURN
 2001 FORMAT  (/5X,' ')
 2002 FORMAT  (1X,'PT.NO.',7X,'w',13X,'Mr',12X,'Mt',12X,'Mrt',11X,'Qr'
     *,12X,'Vr',12X,'Qt',12X,'Vt')
 2010 FORMAT  (1X,I3,10D14.5)
      END
C****************************************
C  SOLUTION FOR LINEAR SYSTEM OF EQUATIONS
C****************************************
      SUBROUTINE SLNPD (A,B,D,M)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(M,1),B(1),IVEC(200)
      COMMON /LIO/ LR,LW,LC,LM
C
      DO 1 I = 1, M
         IVEC(I) = I
    1 CONTINUE
      MM = M - 1
```

```
      DO 8 J = 1, MM
          WRITE (LW,5000) J
 5000     FORMAT (I5)
        J1 = J + 1
        IP = J
        JP = J
        DO 2 IROW = J, M
        DO 2 ICOL = J, M
           IF (DABS(A(IROW,ICOL)).LE.DABS(A(IP,JP))) GO TO 2
           IP = IROW
           JP = ICOL
    2   CONTINUE
        IF (IP.EQ.J.AND.JP.EQ.J) GO TO 6
        IF (IP.EQ.J) GO TO 4
        DO 3 I = J, M
           ATEMP = A(J,I)
           A(J,I) = A(IP,I)
           A(IP,I) = ATEMP
    3   CONTINUE
        ATEMP = B(J)
        B(J) = B(IP)
        B(IP) = ATEMP
    4   DO 5 I = 1, M
           ATEMP = A(I,J)
           A(I,J) = A(I,JP)
           A(I,JP) = ATEMP
    5   CONTINUE
        I = IVEC(J)
        IVEC(J) = IVEC(JP)
        IVEC(JP) = I
    6   DO 7 IROW = J1, M
           C = -A(IROW,J) / A(J,J)
           B(IROW) = B(IROW) + B(J)*C
        DO 7 ICOL = J1, M
           A(IROW,ICOL) = A(IROW,ICOL) + C*A(J,ICOL)
    7   CONTINUE
    8 CONTINUE
C
C     COMPUTE LAST UNKNOWN
C
      B(M) = B(M)/A(M,M)
C
C     APPLY BACKSUBSTITUTION PROCESS TO COMPUTE REMAINING UNKNOWNS
C
      DO 10 L = 1, MM
         K = M - L
         K1 = K + 1
          WRITE (LW,5000) K
         DO 9 J = K1, M
            B(K) = B(K) - A(K,J)*B(J)
    9    CONTINUE
         B(K) = B(K)/A(K,K)
   10 CONTINUE
C
C  REARRANGE X IN THE ORDER (X1,X2,X3,....)
```

```
C
      DO 13 I = 1, M
      DO 11 K = I, M
         IF (IVEC(K).NE.I) GO TO 11
         BTEMP = B(I)
         B(I) = B(K)
         B(K) = BTEMP
         GO TO 12
   11 CONTINUE
   12 IVEC(K) = IVEC(I)
   13 CONTINUE
C
C  COMPUTE VALUE OF DETERMINANT
C
      D = 1.
      DO 250 I = 1, M
        D = D*A(I,I)
  250 CONTINUE
  300 RETURN
 2002 FORMAT (/10X,'**** SINGULAR IN ROW',I5)
      END
C***********************************************************************
```

```
C*********************************************************************
C
C PROGRAM FOR EDITING DATA FILE OF BEM. FOR CIRCULAR AND ANNULAR PLATE
C
C*********************************************************************
C
      DIMENSION KOUT(10),KINN(10),GAMM(10),BETA(10),NELOUT(10)
     *          ,NELINN(10),RCOL(10),ZECOL(10),PLOAD(10),PRHO(10)
     *          ,PANG(10),WDIR(8),WRHO(20),WANG(20)
      CHARACTER*72 IFNAME
      LW = 1
      LR = 5
      CALL TNOUA ('     INPUT DATA FILE NAME ===> ', INTS(30))
      READ (LW,1000) IFNAME
      OPEN (5,FILE=IFNAME)
      CLOSE (5,STATUS='DELETE')
      OPEN (5,FILE=IFNAME)
C
      WRITE (LW,2000)
      WRITE (LR,2000)
      WRITE (LW,2010)
      WRITE (LR,2010)
C
      WRITE (LW,2020)
      WRITE (LR,2020)
      WRITE (LW,2030)
      READ (LW,*) KCIR
      WRITE (LR,1010) KCIR
C
      IF (KCIR.NE.0) GO TO 10
      WRITE (LW,2040)
      WRITE (LR,2040)
      READ (LW,*) ROUT
      WRITE (LR,1020) ROUT
      GOTO 20
   10 WRITE (LW,2050)
      WRITE (LR,2050)
      WRITE (LW,2060)
      READ (LW,*) ROUT
      WRITE (LR,1020) ROUT
      WRITE (LW,2070)
      READ (LW,*) RINN
      WRITE (LR,1020) RINN
C
   20 WRITE (LW,2080)
      WRITE (LR,2080)
      IF (KCIR.EQ.0) GO TO 30
      WRITE (LW,2090)
   30 READ (LW,*) NOUT
      WRITE (LR,1010) NOUT
      IF (KCIR.EQ.0) GO TO 40
      WRITE (LW,2100)
      READ (LW,*) NINN
      WRITE (LR,1010) NINN
C
```

```
   40 WRITE (LW,2110)
      WRITE (LR,2110)
      WRITE (LW,2120)
      IF (KCIR.EQ.0) GO TO 50
      WRITE (LW,2090)
   50 DO 55 I = 1, NOUT
         WRITE (LW,1010) I
         READ (LW,*) KOUT(I)
   55 CONTINUE
      WRITE (LR,1010) (KOUT(I),I=1,NOUT)
      IF (KCIR.EQ.0) GO TO 60
      WRITE (LW,2100)
      DO 56 I = 1, NINN
         WRITE (LW,1010) I
         READ (LW,*) KINN(I)
   56 CONTINUE
      WRITE (LR,1010) (KINN(I),I=1,NINN)
C
   60 IF (NOUT.LE.1) GO TO 76
      WRITE (LW,2130)
      WRITE (LR,2130)
      IF (KCIR.EQ.0) GO TO 70
      WRITE (LW,2090)
   70 DO 75 I = 1, NOUT
         WRITE (LW,1010) I
         IF (I.EQ.1) THEN
            GAMM(I) = 0.
            WRITE (LW,1020) GAMM(I)
         ELSE
            READ (LW,*) GAMM(I)
         ENDIF
   75 CONTINUE
      WRITE (LR,1020) (GAMM(I),I=1,NOUT)
   76 IF (KCIR.EQ.0) GO TO 80
      IF (NINN.LE.1) GO TO 80
      WRITE (LW,2100)
      DO 78 I = 1, NINN
         WRITE (LW,1010) I
         IF (I.EQ.1) THEN
            BETA(I) = 0.
            WRITE (LW,1020) BETA(I)
         ELSE
            READ (LW,*) BETA(I)
         ENDIF
   78 CONTINUE
      WRITE (LR,1020) (BETA(I),I=1,NINN)
C
   80 WRITE (LW,2140)
      WRITE (LR,2140)
      IF (KCIR.EQ.0) GO TO 90
      WRITE (LW,2090)
   90 DO 95 I = 1, NOUT
         WRITE (LW,1010) I
         READ (LW,*) NELOUT(I)
   95 CONTINUE
```

```
      WRITE (LR,1010) (NELOUT(I),I=1,NOUT)
      IF (KCIR.EQ.0) GO TO 100
      WRITE (LW,2100)
      DO 96 I = 1, NINN
         WRITE (LW,1010) I
         READ (LW,*) NELINN(I)
   96 CONTINUE
      WRITE (LR,1010) (NELINN(I),I=1,NINN)
C
  100 WRITE (LW,2150)
      WRITE (LR,2150)
      READ (LW,*) NCOL
      WRITE (LR,1010) NCOL
      IF (NCOL.LE.0) GO TO 120
      WRITE (LW,2160)
      DO 110 I = 1, NCOL
         WRITE (LW,1010) I
         WRITE (LW,2162)
         READ (LW,*) RCOL(I)
         WRITE (LW,2164)
         READ (LW,*) ZECOL(I)
         WRITE (LR,1020) RCOL(I),ZECOL(I)
  110 CONTINUE
C
  120 WRITE (LR,2170)
      WRITE (LW,2180)
      READ (LW,*) TH
      WRITE (LW,2190)
      READ (LW,*) PR
      WRITE (LW,2200)
      READ (LW,*) ELAS
      WRITE (LR,1040) TH,PR,ELAS
C
      WRITE (LW,2210)
      WRITE (LR,2210)
      READ (LW,*) UNIF
      WRITE (LR,1020) UNIF
C
      WRITE (LW,2220)
      WRITE (LR,2220)
      READ (LW,*) NPT
      WRITE (LR,1010) NPT
      IF (NPT.LE.0) GO TO 140
      WRITE (LW,2230)
      DO 130 I = 1, NPT
         WRITE (LW,1010) I
         WRITE (LW,2235)
         READ (LW,*) PLOAD(I)
         WRITE (LW,2162)
         READ (LW,*) PRHO(I)
         WRITE (LW,2164)
         READ (LW,*) PANG(I)
         WRITE (LR,1020) PLOAD(I), PRHO(I), PANG(I)
  130 CONTINUE
C
```

```
  140 WRITE (LR,2240)
      WRITE (LW,2250)
      READ (LW,*) KW
      WRITE (LR,1010) KW
      IF (KW.LE.0) GO TO 150
      WRITE (LW,2240)
      DO 145 I = 1, KW
         WRITE (LW,1010) I
         READ (LW,*) WDIR(I)
  145 CONTINUE
      WRITE (LR,1020) (WDIR(I),I=1,KW)
C
      WRITE (LR,2255)
      WRITE (LW,2256)
      READ (LW,*) KP
      WRITE (LR,1010) KP
C
  150 WRITE (LW,2260)
      WRITE (LR,2260)
      WRITE (LW,2270)
      READ (LW,*) KWPT
      WRITE (LR,1010) KWPT
      IF (KWPT.LE.0) GO TO 170
      WRITE (LW,2260)
      DO 160 I = 1, KWPT
         WRITE (LW,1010) I
         WRITE (LW,2162)
         READ (LW,*) WRHO(I)
         WRITE (LW,2164)
         READ (LW,*) WANG(I)
         WRITE (LR,1020) WRHO(I), WANG(I)
  160 CONTINUE
  170 STOP
C
 1000 FORMAT (A72)
 1010 FORMAT (12I5)
 1020 FORMAT (8F10.5)
 1030 FORMAT (D20.10)
 1040 FORMAT (2F10.5,D20.10)
 2000 FORMAT ('BOUNDARY ELEMENT METHOD FOR THE ANALYSIS OF CIRCULAR AND
     *ANNULAR PLATE')
 2010 FORMAT (' ')
 2020 FORMAT ('ANALYSIS FOR CIRCULAR OR ANNULAR PLATE')
 2030 FORMAT ('   INPUT -0- FOR CIRCULAR  -1- FOR ANNULAR PLATE')
 2040 FORMAT ('RADIUS OF CIRCULAR PLATE')
 2050 FORMAT ('RADIUS OF ANNULAR PLATE')
 2060 FORMAT ('   INPUT OUTER RADIUS')
 2070 FORMAT ('   INPUT INNER RADIUS')
 2080 FORMAT ('NUMBER OF COMBINATION BOUNDARY')
 2090 FORMAT ('   FOR OUTER EDGE')
 2100 FORMAT ('   FOR INNER EDGE')
 2110 FORMAT ('TYPE OF COMBINATION BOUNDARY')
 2120 FORMAT ('   INPUT -0- FOR FREE  -1- FOR SIMPLE  -2- FOR CLAMP')
 2130 FORMAT ('THE BEGINING ANGLE OF EACH BOUNDARY (DEGREE)')
 2140 FORMAT ('NUMBER OF ELEMENT IN EACH BOUNDARY')
```

```
 2150 FORMAT ('NUMBER OF COLUMN SUPPORTS')
 2160 FORMAT ('   INPUT RADIUS AND ANGLE FOR EACH COLUMN')
 2162 FORMAT ('   R = ')
 2164 FORMAT ('   ANGLE = ')
 2170 FORMAT ('THICKNESS OF PLATE, POISSON RATIO AND MODULUS OF ELASTICI
     *TY')
 2180 FORMAT ('   INPUT THICKNESS OF PLATE')
 2190 FORMAT ('   INPUT POISSON RATIO')
 2200 FORMAT ('   INPUT MODULUS OF ELASTICITY')
 2210 FORMAT ('UNIFORM LOAD')
 2220 FORMAT ('NUMBER OF CONCENTRATED LOADS AND LOCATIONS')
 2230 FORMAT ('   INPUT POINT LOADS, RADIUS AND ANGLE FOR EACH LOAD')
 2235 FORMAT ('   POINT LOADS = ')
 2240 FORMAT ('RADIUS DIRECTION OF RESULT')
 2250 FORMAT ('   INPUT NO. OF RADIUS DIRECTION FOR RESULT')
 2255 FORMAT ('   INPUT NO. OF POINT PER LINE')
 2256 FORMAT ('NUMBER OF POINT PER LINE')
 2260 FORMAT ('POSITION OF OTHER REQUIRED RESULT')
 2270 FORMAT ('   INPUT NO. OF OTHER REQUIRED RESULT POSITION')
      END
C***********************************************************************
```

APPENDIX D

Use of the Computer Programs

Data Input

I. HEADING

	variable	entry
(1)	HEAD	Title of problem (80 alphabets)

II. MASTER CONTROL PARAMETERS

	variable	entry
(1)	KCIR	Type of plate: EQ.0; circular plate NE.0; annular plate
(2)	ROUT	Radius of circular plate or outer radius of annular plate
(3)	RINN	Inner radius of annular plate
(4)	NOUT	Number of combination boundary conditions of outer edge
(5)	NINN	Number of combination boundary conditions of inner edge

III. BOUNDARY CONDITIONS

	variable	entry
(1)	I	Section number

(2)	KOUT(I)	Type of boundary conditions of outer edge: EQ.0; free edge EQ.1; simple support EQ.2; clamped support
(3)	KINN(I)	Type of boundary conditions of inner edge: EQ.0; free edge EQ.1; simple support EQ.2; clamped support
(4)	GAMM(I)	Beginning angle of section of outer edge
(5)	BETA(I)	Beginning angle of section of inner edge
(6)	NELOUT(I)	Number of intervals in section of outer edge
(6)	NELINN(I)	Number of intervals in section of inner edge

IV. INTERIOR COLUMN SUPPORT

	variable	entry
(1)	NCOL	Number of support
(2)	I	Support number
(3)	RCOL(I)	R -ordinate
	ZECOL(I)	θ -ordinate

V. GEOMETRIC AND MATERIAL PROPERTIES

	variable	entry
(1)	TH	Plate thickness
(2)	PR	Poisson's ratio

(3)	ELAS	Modulus of Elasticity

VI. LOADING DATA

	variable	entry
(1)	UNIF	Uniformly distributed load
(2)	NPT	Number of concentrated loads
(3)	I	Concentrated load number
(4)	PLOAD(I)	Magnitude of z-direction force
	PRHO(I)	R -ordinate
	PANG(I)	θ -ordinate

VII. SOLUTION OUTPUT

	variable	entry
(1)	KW	Number of solution lines in radius direction
(2)	I	Solution line number
(3)	WDIR(I)	Angle of solution line
(4)	KP	Number of points in solution line
(5)	KWPT	Number of solution locations
(6)	I	Location number
(7)	WRHO(I)	R -ordinate
	WANG(I)	θ -ordinate

Example

Data input file and results of Example 4 in Chapter IV can be obtained in the following:

a) DATA FILE

```
BOUNDARY ELEMENT METHOD FOR THE ANALYSIS OF CIRCULAR AND ANNULAR PLATE

ANALYSIS FOR CIRCULAR OR ANNULAR PLATE
    1
RADIUS OF ANNULAR PLATE
   1.00000
   0.25000
NUMBER OF COMBINATION BOUNDARY
    6
    1
TYPE OF COMBINATION BOUNDARY
    2    1    2    1    2    1
    0
THE BEGINING ANGLE OF EACH BOUNDARY (DEGREE)
   0.00000  60.00000 120.00000 180.00000 240.00000 300.00000
NUMBER OF ELEMENT IN EACH BOUNDARY
    6    6    6    6    6    6
   18
NUMBER OF COLUMN SUPPORTS
    3
   0.40000  90.00000
   0.40000 210.00000
   0.40000 330.00000
THICKNESS OF PLATE, POISSON RATIO AND MODULUS OF ELASTICITY
   0.01750   0.30000    0.2037551000D+07
UNIFORM LOAD
   1.00000
NUMBER OF CONCENTRATED LOADS AND LOCATIONS
    0
RADIUS DIRECTION OF RESULT
    3
  30.00000  60.00000  90.00000
   INPUT NO. OF POINT PER LINE
   11
POSITION OF OTHER REQUIRED RESULT
    2
   0.39800  90.00000
   0.40200  90.00000
```

b) RESULTS

```
CPU TIME used before executing in SECONDS + TICKS
               121                  286
I/O TIME used before executing in SECONDS + TICKS
               12                   182

    BOUNDARY ELEMENT METHOD FOR THE ANALYSIS OF CIRCULAR AND ANNULAR PLATE


    ANALYSIS FOR CIRCULAR OR ANNULAR PLATE
    ============>   1

    RADIUS OF ANNULAR PLATE
    ============>   1.00000
    ============>   0.25000

    NUMBER OF COMBINATION BOUNDARY
    ============>   6
    ============>   1

    TYPE OF COMBINATION BOUNDARY
     2    1    2    1    2    1
     0

    THE BEGINING ANGLE OF EACH BOUNDARY (DEGREE)
    0.00000  60.00000 120.00000 180.00000 240.00000 300.00000

    NUMBER OF ELEMENT IN EACH BOUNDARY
     6    6    6    6    6    6
    18

    NUMBER OF COLUMN SUPPORTS
    ============>   3
    0.40000  90.00000 .
    0.40000 210.00000
    0.40000 330.00000

    THICKNESS OF PLATE, POISSON RATIO AND MODULUS OF ELASTICITY
    0.01750   0.30000    0.2037551000D+07

    UNIFORM LOAD
    ============>   1.00000

    NUMBER OF CONCENTRATED LOADS AND LOCATIONS
    ============>   0

    RADIUS DIRECTION OF RESULT
    ============>   3
   30.00000  60.00000  90.00000
```

```
   INPUT NO. OF POINT PER LINE
::::::::::::>  11

POSITION OF OTHER REQUIRED RESULT
::::::::::::>  2
0.39800  90.00000
0.40200  90.00000

    ********* MEMORY USED =      12521 *********

DETERMINANT =    0.1043177194-161
```

*****RESULTS FOR RADIUS DIRECTION NO. 1

PT.NO.	W	Mr	Mt	Mrt	Vr	Vt
1	0.61140D-03	0.35183D-02	0.11194D-01	0.16561D-09	-0.49272D+03	-0.49827D-07
2	0.87531D-03	0.97427D-02	0.20284D-01	-0.52985D-12	-0.23032D+00	-0.32217D-08
3	0.11245D-02	0.18873D-01	0.17425D-01	-0.64171D-12	-0.17980D-01	-0.63524D-10
4	0.12914D-02	0.25195D-01	0.16506D-01	-0.15254D-12	-0.39816D-02	-0.30020D-12
5	0.13353D-02	0.27495D-01	0.15561D-01	-0.56177D-13	-0.45545D-01	0.12186D-11
6	0.12406D-02	0.25135D-01	0.14030D-01	-0.37526D-13	-0.92210D-01	0.17319D-12
7	0.10191D-02	0.18083D-01	0.11494D-01	-0.33196D-13	-0.14106D+00	-0.30833D-11
8	0.70950D-03	0.66843D-02	0.75944D-02	-0.36748D-13	-0.19127D+00	-0.20505D-10
9	0.37488D-03	-0.85815D-02	0.20823D-02	-0.52847D-13	-0.24289D+00	-0.16629D-09
10	0.10112D-03	-0.27375D-01	-0.45297D-02	-0.10436D-12	-0.29461D+00	-0.34841D-08
11	-0.13337D-04	-0.44110D-01	-0.10067D-01	-0.26582D-11	0.65603D+02	-0.38343D-04

*****RESULTS FOR RADIUS DIRECTION NO. 2

PT.NO.	W	Mr	Mt	Mrt	Vr	Vt
1	0.32942D-03	-0.12353D-02	-0.41796D-02	-0.13014D+05	-0.24452D+03	0.71709D+10
2	0.50013D-03	-0.25429D-02	0.15151D-02	-0.18396D+01	-0.40458D-01	0.13069D+04
3	0.71311D-03	0.26601D-02	0.45327D-02	-0.15939D+00	0.95313D-01	0.21722D+02
4	0.91485D-03	0.12538D-01	0.61946D-02	-0.37481D-01	0.89034D-01	0.22105D+01
5	0.10517D-02	0.20470D-01	0.79763D-02	-0.25230D-01	0.53683D-01	0.31329D+00
6	0.10793D-02	0.23570D-01	0.93766D-02	-0.48524D-01	-0.96442D-03	-0.14663D+00
7	0.98097D-03	0.21539D-01	0.94993D-02	-0.12853D+00	-0.64125D-01	-0.68617D+00
8	0.76937D-03	0.14650D-01	0.78105D-02	-0.42061D+00	-0.12979D+00	-0.27764D+01
9	0.48401D-03	0.30018D-02	0.41790D-02	-0.21108D+01	-0.19505D+00	-0.17666D+02
10	0.19187D-03	-0.13572D-01	-0.67163D-03	-0.33569D+02	-0.25701D+00	-0.33643D+03
11	0.13584D-04	-0.29209D-01	-0.47065D-02	-0.34581D+06	0.22758D+02	-0.36499D+07

*****RESULTS FOR RADIUS DIRECTION NO. 3

PT.NO.	W	Mr	Mt	Mrt	Vr	Vt
1	-0.45342D-04	-0.52675D-02	-0.37884D-01	-0.30903D-09	0.37988D+02	0.12474D-06
2	-0.34134D-04	-0.18594D-01	-0.65358D-01	-0.19460D-10	-0.96595D+00	0.82498D-08
3	-0.32363D-13	0.38241D-01	0.21257D-01	-0.23761D-11	0.94393D-01	0.32314D-09
4	0.29324D-03	-0.15270D-01	-0.49568D-01	-0.68678D-12	0.10991D+01	0.46715D-10
5	0.60996D-03	0.10129D-01	-0.19798D-01	-0.37770D-12	0.48594D+00	0.12340D-10
6	0.83199D-03	0.21352D-01	-0.40183D-02	-0.45614D-12	0.25199D+00	0.52003D-11
7	0.91676D-03	0.25558D-01	0.49130D-02	-0.93264D-12	0.11340D+00	0.50739D-11
8	0.85391D-03	0.24993D-01	0.93049D-02	-0.24057D-11	0.12986D-01	0.12971D-10
9	0.65445D-03	0.20452D-01	0.10316D-01	-0.85098D-11	-0.68074D-01	-0.15404D-10
10	0.34795D-03	0.12054D-01	0.90584D-02	-0.94015D-10	-0.13766D+00	-0.61176D-08
11	0.69750D-04	0.31427D-02	0.73114D-02	-0.88975D-06	-0.21101D+01	0.10395D-01

*****RESULTS AT OTHER REQUIRED POSITION

PT.NO.	W	Mr	Mt	Mrt	Vr	Vt
1	-0.40369D-05	-0.16255D+00	-0.20153D+00	-0.24754D-11	-0.42111D+02	0.34569D-09
2	0.46733D-05	-0.16246D+00	-0.20116D+00	-0.22782D-11	0.42191D+02	0.30331D-09

CPU TIME used after executing in SECONDS + TICKS
190 88
I/O TIME used after executing in SECONDS + TICKS
13 123

VITA

Mr. Siriphan Amonsrisawat was born on December 28, 1959 in Trad and graduated B.Eng. in Civil Engineering from King Mongkut's Institute of Technology, Thonburi Campus in 1982.