

AD-A039 394

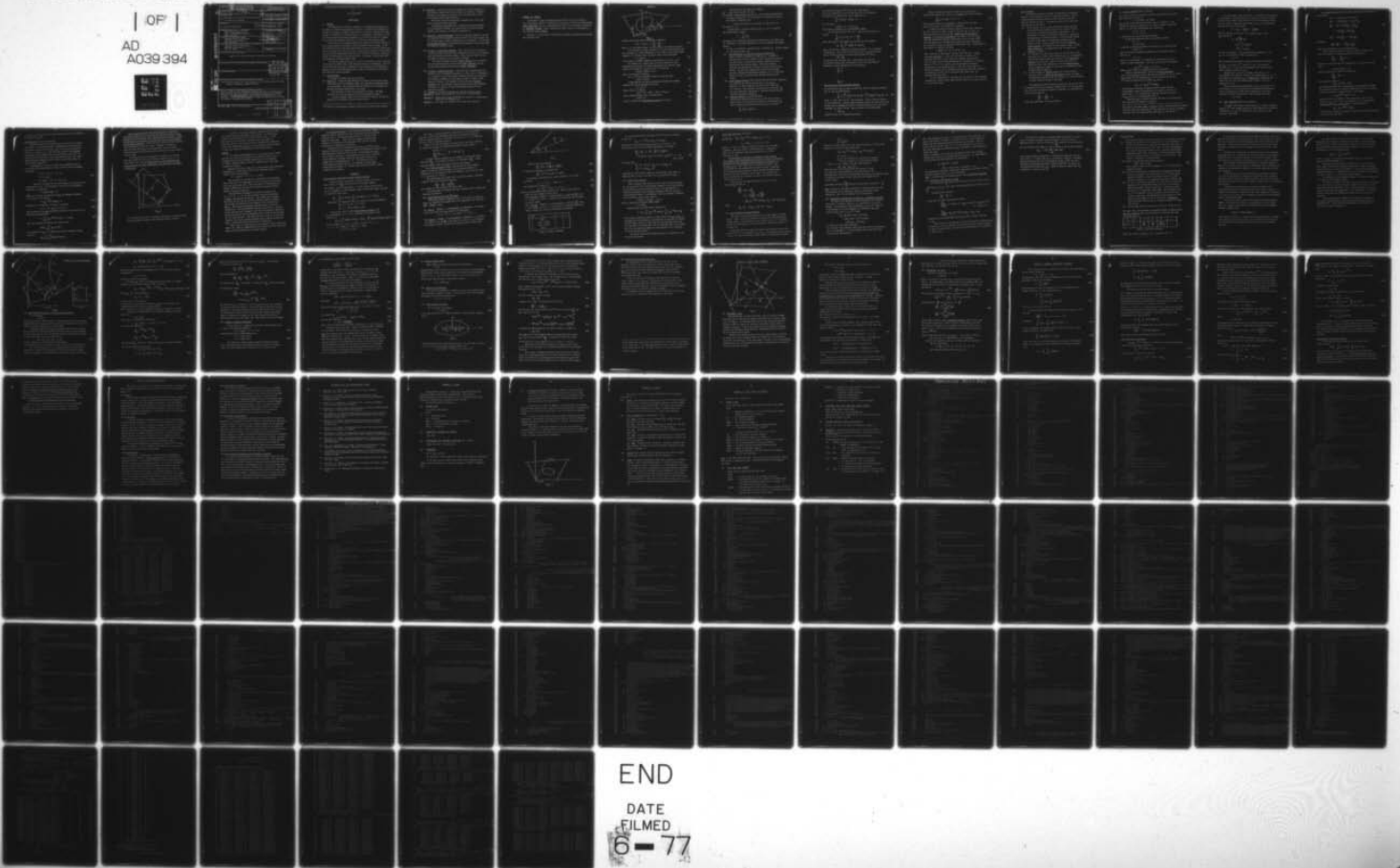
UNIVERSITY COLL CORK (IRELAND) DEPT OF MATHEMATICAL--ETC F/G 12/1
FUNDAMENTALS OF THE EDGE-FUNCTION METHOD VIA LAPLACE'S EQUATION--ETC(U)
FEB 77 P M QUINLAN AF-AFOSR-2786-75

UNCLASSIFIED

EOARD-TR-77-01

NL

| OF |
AD
A039 394



END

DATE
FILMED
6-77

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER 14 EDARD-TR-77-01	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 Fundamentals of The Edge-Function Method via Laplace's Equation.	5. TYPE OF REPORT & PERIOD COVERED Scientific Interim rept. 1 July 1975 - 30 June 1976	
7. AUTHOR(s) 10 Patrick M. Quinlan	8. CONTRACT OR GRANT NUMBER(s) 15 AF-AFOSR-75-2786-75	
9. PERFORMING ORGANIZATION NAME AND ADDRESS University College of Cork Dept. of Mathematical Physics Cork, Ireland.	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 9767-99-56 12 99	
11. CONTROLLING OFFICE NAME AND ADDRESS European Office of Aerospace, Research & Development, FPO N.Y. 09510	12. REPORT DATE 11 1 February 1977	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) European Office of Aerospace Research & Development FPO N.Y. 09510	13. NUMBER OF PAGES 50 + 2 progs	
	15. SECURITY CLASS. (of this report)	
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE	

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
DDC
RECEIVED
MAY 13 1977
RECEIVED
A

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Edge-Function Method, Structural Analysis, Linear Boundary Value Problems, Laplace's Equation; Harmonic Fitting
The Edge-Function Method

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
The main algebraic and programming features of E.F.M. are illustrated by application to Laplace's equation for polygonal regions with elliptical cavities. The Basic Functions - Vertex, Edge and Mapped Polars - are obtained. Vertex Functions, Harmonic Fitting and submission into elements are covered. Programs are attached.

ACCESSION IN	White Section
DDC	Self Section
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	<input type="checkbox"/>
BY	
DIST	
APPROVAL	
ANAL. SEC. OR SPECIAL	
BY	
APPROVAL	
ANAL. SEC. OR SPECIAL	

402.948

42

ADA 039394
 DDC FILE COPY

by

P. M. QUINLAN⁽¹⁾

Introduction

1. OUTLINE

The Edge-Function Method as applied to two-dimensional problems is a procedure for obtaining approximate solutions to boundary value problems in regular or irregular polygonal regions with, or without, cavities and cracks. It may be described as a piecing together of "asymptotic" solutions to a set of linear partial differential equations for the several parts of a domain D to satisfy the boundary conditions in a discrete least squares sense.

The method was originally developed by Quinlan [1] for the solution of the torsion problem for prismatic bars of polygonal cross-section. In the succeeding 10 years it has been successfully applied to problems in the bending of isotropic thin plates [2,3,4]; coupled linear systems in elastostatics [4,5,6,7] and in moderately thick plates and shallow shells [8]; cracks and stress concentrations in elastostatics [9]; vibrations of Thin Plates [10] and vibrations of Shallow Shells [11].

The present paper arose out of a course of lectures on E.F.M. given by the author in December 1975 at The Georgia Institute of Technology. The aim is to illustrate as simply as possible the main algebraic and programming features of the method by applying it to Laplace's equation as it arises in heat flow and torsion problems.

2. PLAN OF PAPER

The paper is divided into four sections:

- (a) Section 1: Introduces Edge-Functions and Vertex Functions for steady state heat flow in a polygonal region the temperatures being specified on the boundary. The resulting boundary identities are satisfied by harmonic matching using Fourier sine series.

The importance of the vertex equations is shown. The Edge-Functions are obtained using a half-strip mathematical model.

A simple program, LAPEX, and an illustrative example, is given in Appendix A for problems covered by this section.

(1) Professor of Mathematical Physics, University College, Cork, Ireland.

- (b) Section 2: Extension is made to regions with mixed conditions on the boundary segments, and circular holes are included. Harmonic matching is now based on using the (full) Fourier series, with corresponding (full) Edge-Functions.

Provision is made for Logarithmic Singularities, where they arise, using Log-Vertex Functions.

Programming requirements for a comprehensive program LAPGEN, as a development of LAPEX, are discussed and a suitable program LAPGEN, with examples, is attached in Appendix B.

- (c) Section 3 - Curved Boundaries: This uses conformal mapping to extend the treatment of the previous section to holes of elliptic shape, and to circular and/or elliptic indentations on the boundary. The ensuing Curved Edge-Functions are developed, and programmed in subroutine PMAP in LAPGEN for elliptic curves.

- (d) Section 4 - Other Basic Problems: This section treats the case of a reentrant angle, and also where singularities occur on the boundary, using appropriate half-strip models. These features have not been included in LAPGEN, but are available in QUINP.

The models used may be termed "BASIC MODELS" for the problem, the corresponding regions being termed "BASIC REGIONS" and the corresponding functions being termed "BASIC FUNCTIONS".

- (e) Section 5 - Harmonic Fitting: A discrete least squares method for fitting a set of orthogonal functions to a given curve is developed. It is applied to solve the boundary identity problem by minimising the boundary residuals, on each segment, using a discrete least squares criterion. This process is given the name "Harmonic Fitting", and follows from Harmonic Matching when the integrals concerned are replaced by the corresponding trapezoidal rule quadrature formula.

- (f) Section 6 - Distinctive Features of the Edge Function Method.
Appendix A: LAPEX, with example of heat flow in a quadrilateral.

Appendix B: LAPGEN with example involving the torsion of a quadrilateral region with an elliptic hole

Appendix C: Data Input for LAPGEN on the "black box" principle.

SUMMARY OF METHOD

Accordingly to apply The Edge-Function Method to any new situation e.g. thin plates, shells, 3-D Elasticity, we require to find the corresponding BASIC FUNCTIONS, and fit a superposition of these to the boundary conditions by "HARMONIC FITTING" using a computation scheme similar to that given in the attached program LAPGEN.

Sections 3, 4 and 5 may be omitted when studying the Edge-Function Method for the first time.

SECTION 1

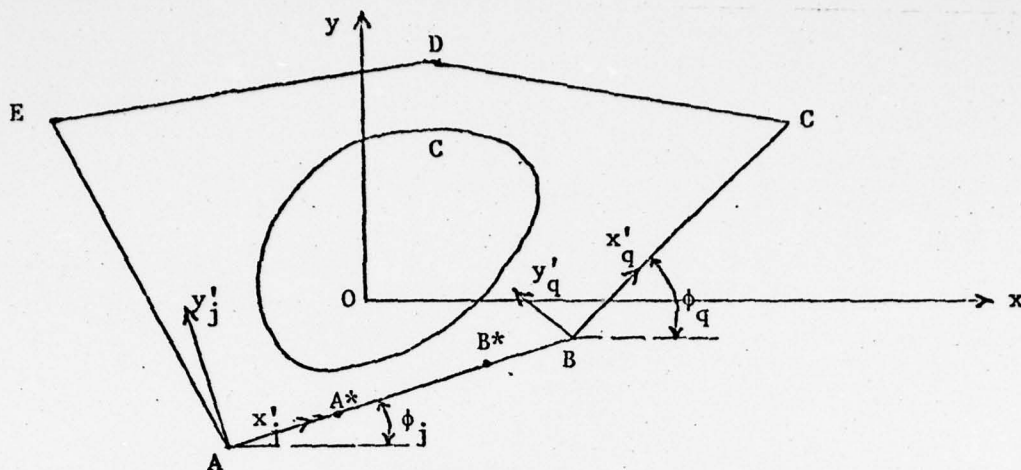


Fig. 1

Consider the linear partial differential equation:

$$L(u) = f(x,y); \quad L \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}, \quad (1)$$

where L , in the present paper, is restricted to the Laplace operator.

It is required to find a function $u = u(x,y)$ which satisfies the above equation in a domain Ω bounded by any number q' piecewise smooth segments, or straight lines, and which takes prescribed values

$$u = g_q(x,y) \quad (2)$$

on the various boundary segments corresponding to $q = 1, 2, \dots, q'$.

For example in a steady state heat flow problem u is the temperature and $f(x,y)$ is the heat generation term, and for a heat source of strength Q at a point (u,v) , f is given by

$$f(x,y) = Q\delta(x-u, y-v), \quad (3)$$

where δ is the delta function.

If $u = u^p$ is a particular solution of eqn.(1), then

$$L(u^p) = f(x,y) \quad (4)$$

A more general solution, involving the addition of another function u^c , may be written in the form

$$u = u^p + u^c \quad (5)$$

where, since L is linear,

$$L(u^p + u^c) = L(u^p) + L(u^c) = f(x,y), \quad (6)$$

and on subtracting eqn.(4), it follows that

$$L(u^c) = 0, \quad (7)$$

which is called the complementary equation of eqn.(1).

The problem is now split into finding

- (a) a particular solution u^P , and
- (b) suitable complementary functions u^C , with sufficient flexibility to enable solution $u^P + u^C$ to satisfy, in an approximate manner, the boundary conditions (2).

The present paper is restricted to the case $f = 0$, for which $u^P = 0$.

On obtaining I suitable solutions u_i^C , $i = 1, I$; of eqn.(7) and superposing, we obtain

$$u = \sum_{i=1}^I A_i u_i^C \quad (8)$$

involving the I arbitrary superposition constants A_i , and these must be evaluated to satisfy conditions (2) in an acceptable approximate manner.

I eqns. are required to determine the I constants A_i . Various schemes for producing I eqns. are:

- (a) Point-Matching, or collocation [Conway, Leissa etc.]

Select I points on boundary in same manner - equidistant distribution, or have a somewhat closer spacing near corners - and write eqns. to ensure that boundary conditions (2) are satisfied at the selected points. Check the boundary residuals (or deviations) at points in between the selected points. The system of simultaneous equations may become unstable, giving ill-conditioned equations, as I is increased. There is an art in selecting the optimum spacing for the points.

- (b) Least squares fitting This requires that the boundary conditions be satisfied, in a least squares sense, at M points, where $M > I$, there being more points selected than there are unknowns. This improves the process up to about $M = 2I$, but much additional computation is involved in producing the resulting normalised equations.

- (c) Harmonic Matching (Quinlan 1962-70)

Consider the boundary identity on the q^{th} side, in fig.1, where points on the side are specified by the parameter x'_q , which requires, on substituting from eqn.(8) in (2), that

$$\sum_{i=1}^I A_i u_i^C \equiv g_q(x, y), \quad (9)$$

for all points $P(x'_q)$ on q^{th} side in the range $(0, a_q)$.
 Since the coordinates (x, y) for any point P can be expressed
 in terms of x'_q , identity (9) can be expressed as

$$\sum_{i=1}^I A_i u_i^c(x'_q) - g_q(x'_q) \equiv 0, \quad (10)$$

or, for short,

$$\psi_q(x'_q) \equiv 0; \quad 0 \leq x'_q \leq a_q, \quad (11)$$

with similar identities for each boundary segment.

On expanding identity (11) in a Fourier sine series, we obtain

$$\sum_{N=1}^{\infty} C_N \sin nx'_q \equiv 0; \quad n = \frac{N\pi}{a_q}, \quad (12)$$

where the Fourier coefficients C_N are given by the integral

$$C_N = \frac{2}{a_q} \int_0^{a_q} \psi_q(x'_q) \sin nx'_q dx'_q \quad (13)$$

Since we have but I degrees of freedom in eqn.(8) - or I arbitrary constants A_i - and if these were to be shared out equally between the q' boundary segments, we would then allocate N^* , where

$$N^* = I/q' \quad (14)$$

to each boundary identity (11). Accordingly, the best way to satisfy identity (12) would appear to be to set the first N^* harmonics to zero, giving rise to the equations

$$\begin{aligned} C_1 &= 0 \\ C_2 &= 0 \\ &\cdot \\ &\cdot \\ &\cdot \\ C_{N^*} &= 0 \end{aligned} \quad (15)$$

This process is called Harmonic Matching

On substituting from eqs.(11) and (10) into (13), we obtain, on setting $C_N = 0$, for the N th harmonic:

$$\sum_{i=1}^I A_i \int_0^{a_q} u_i^c(x'_q) \sin nx'_q dx'_q - \int_0^{a_q} g_q(x'_q) \sin nx'_q dx'_q = 0, \quad (16)$$

where N goes from 1 to N^* . On evaluating, a linear equation results in the unknowns A_i . Similar linear equations follow for each side of the polygon and accordingly a total of I linear equations result for the I unknowns, A_i , which system may be written in the matrix form

$$\vec{P}\vec{x} = \vec{Q}, \quad (17)$$

where \vec{P} is the $I \times I$ coefficients matrix.

However setting the first N^* harmonics to zero does not satisfy (12) exactly, but leaves as a residual the higher harmonics

$$\sum_{N=N}^{\infty} C_N \sin nx'_q; \quad \bar{N} = N^* + 1 \quad (18)$$

and we require this to be negligible - which will be the case only if the series is sufficiently convergent.

As shown in texts on Fourier series the sine series expansion of (11) will have a Gibbs effect at the ends $x'_q = 0; x'_q = a_q$ - with convergence only of order $\frac{1}{M}$ - unless the function being expanded, ψ_q , is zero at both ends of the range. In this latter case the convergence improves considerably, to be of order $1/M^3$, and consequently the residual series (18) can be made negligible, in a practical sense, by taking N^* sufficiently large. Accordingly we must ensure in identity (11) that

$$\psi_q(x'_q) = 0; \quad x'_q = 0 \text{ and } x'_q = a_q \quad (19)$$

This involves two additional equations for each side - called the Vertex Equations - and if we still wish to restrict to N^* equations per side, we must reduce the harmonic equations set (15) for each side by two, corresponding to truncation at L harmonics where $L = N^* - 2$. It is obvious, due to the greatly increased convergence of the expansion (12) to $1/M^3$, that this scheme is a better utilisation of the available resources per side (N^* constants) in seeking to satisfy, in an acceptable approximate way, the boundary identities (11).

If the temperature is continuous at all points on the boundary, including all corner points, then eqs.(19) reduce to a single equation $\psi = 0$ at each vertex.

If solution form (8) is capable of providing an acceptable solution to the problem, then it is especially important that it should represent the main features of the solution in all the critical regions of Fig. 1. On examining some critical parts of the boundary we note that:

- (a) in any corner, say at A, the solution (1-8) must approach the solution in an infinite sector B'AE', where B' and E' are points at infinity on AB and AE respectively. In particular, if there is any singular behaviour - physically corresponding to infinite stresses - in the infinite sector solution, there must be corresponding singularities in solution (1-8). Hence we must first solve the infinite sector problem, and include its more singular terms - called Vertex Functions - in superposition (8) for each of the vertices in Figure 1.
- (b) at a distance from any corner where the vertex effects have moderated, say on A*B*, solution (8) should, in the neighbourhood of A*B*, behave like the solution for the half-plane problem bounded by A'B', where A' and B' are points at infinity on AB. Such solutions are termed Edge-Functions, and an appropriate number of these must be included for each straight side of Figure 1.
- (c) in the vicinity of interior hole C the solution should approach that of a hole C in an infinite region, and again an appropriate number of the more singular (or characteristic) of these solutions should be included.

The appropriate "mix" of functions from (a), (b) and (c) will be determined later from analytic considerations. We now proceed to solve the basic problems for Laplace's equation, as exemplified by the heat flow problem:-

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad (21)$$

where the superscript c has been omitted.

[a] VERTEX FUNCTIONS (Angle Problem)

Polar type solutions of Laplace's equation are provided by the well known harmonics of order λ :

$$u = Ar^\lambda \cos\lambda\theta + Br^\lambda \sin\lambda\theta, \quad (21a)$$

where A and B are arbitrary constants, and λ is arbitrary. The appropriate boundary conditions for a sector of angle α , with one side $\theta = 0$, are the corresponding zero conditions,

$$\begin{aligned} u &= 0; \quad \theta = 0 \\ u &= 0; \quad \theta = \alpha, \end{aligned} \quad (22)$$

as these give rise to an eigenvalue problem.

On applying conditions (22) to eqn.(21a) we obtain

$$\begin{aligned} 0 &= A \\ 0 &= Br^\lambda \sin\lambda\alpha, \end{aligned}$$

a solution to which, other than the trivial one $u \equiv 0$, is given by

$$\sin\lambda\alpha = 0 \quad (23)$$

This is called the eigenvalue equation for the problem, and its solution is

$$\lambda\alpha = k\pi,$$

where k is any integer. On denoting the above discrete set of values - called eigenvalues - for λ by λ_k we obtain as solutions

$$u_k = Br^{\lambda_k} \sin\lambda_k\theta; \quad \lambda_k = k\pi/\alpha, \quad (24)$$

where u_k are called the eigenfunctions for the problem.

Similar eigenfunctions* - here called Vertex Functions - must be incorporated in "mix" (8) from all vertices $j = 1, j'; (j'=q')$. Accordingly the k^{th} eigenfunction, or vertex function, for the j^{th} vertex is denoted by the symbol V_{kj} , where

$$V_{kj} \equiv u_k = B_{kj} [r^{\lambda_k} \sin\lambda_k\theta]_j \quad (25)$$

the subscript j outside the square bracket denotes that the corresponding origin of coordinates is at vertex j, with $\theta = 0$ along side j. The corresponding arbitrary constant is written as B_{kj} , since it can change with k and j:

Note that V_{kj} is a point-function, and this can be emphasised, when required, by writing it as $V_{kj}(P)$.

* Eqn.(20) is invariant under a rotation of axes, and accordingly on taking the polar axis $\theta_j = 0$ along the j^{th} side with pole at vertex j, harmonic solutions (21), and eigenfunctions (24) apply to the vertex j.

The permissible range of λ_k values is determined by the physics of the problem. Negative values of λ_k cannot be admitted as these would require infinite temperatures at the corresponding vertices, but can infinite temperature derivatives (at $r_j = 0$) be admitted? Consider the heat flux \bar{q} , where

$$\bar{q} = -\beta \nabla u_k = -\beta \left\{ \frac{\partial u}{\partial r} \hat{r} + \frac{1}{r} \frac{\partial u}{\partial \theta} \hat{\theta} \right\} \quad (26)$$

The flow through any circular arc of angle α , due to each V_{kj} , is given by

$$\begin{aligned} q_j &= \int_0^\alpha \frac{\partial V_{kj}}{\partial r} r d\theta \\ &= \lambda_k r^{\lambda_k} \int_0^\alpha \sin \lambda_k \theta d\theta \\ &= r^{\lambda_k} \{1 - \cos \lambda_j \alpha\}, \end{aligned} \quad (27)$$

and this is infinite - and thus physically inadmissible - if $\lambda_k < 0$. Accordingly in (25) we require that

$$\lambda_k > 0. \quad (28)$$

The corresponding heat flux, as given by (26) on substituting for V_{kj} , is of order $r^{\lambda_k - 1}$, and if $0 < \lambda_k < 1$ the flux is infinite at $r = 0$, but this is admissible since the nett heat flow through the vertex is zero.

Note that, if λ_k is not an integer, all roots $\lambda_k > 0$ have infinite derivatives of order p and higher, where $\lambda_k - p < 0$. Accordingly all functions V_{kj} are singularity functions in the sense that all their derivatives above a certain order are singular at $r = 0$.

When a temperature discontinuity occurs at any point P on the boundary, or at any vertex, the corresponding discontinuity can be incorporated in "mix" (8) by including a zero order harmonic solution to eqn.(20) for each such point given by

$$u_0 = B_0 \theta \quad (29)$$

[b] EDGE FUNCTIONS (half-plane problem)

Consider the solution in half plane $y_j' \geq 0$, where the edge-axes for j^{th} side denoted (x_j', y_j') are taken respectively along and perpendicular to the j^{th} side and are at an angle ϕ_j to the reference axis (x, y) .

On transforming eqn.(7) to the above edge-axes with the aid of the relations:

$$\begin{aligned}x'_j &= (x-x_j)\cos\phi_j + (y-y_j)\sin\phi_j \\y'_j &= -(x-x_j)\sin\phi_j + (y-y_j)\cos\phi_j \\ \frac{\partial}{\partial x} &= \hat{x} \cdot \nabla = \hat{x} \cdot \left(\hat{x}'_j \frac{\partial}{\partial x'_j} + \hat{y}'_j \frac{\partial}{\partial y'_j} \right) \\ &= \cos\phi_j \frac{\partial}{\partial x'_j} - \sin\phi_j \frac{\partial}{\partial y'_j} \\ \frac{\partial}{\partial y} &= \sin\phi_j \frac{\partial}{\partial x'_j} + \cos\phi_j \frac{\partial}{\partial y'_j},\end{aligned}\tag{30}$$

we obtain

$$L\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right) = L'\left(\frac{\partial}{\partial x'_j}, \frac{\partial}{\partial y'_j}\right),\tag{31}$$

where the coefficients in L' may depend on ϕ_j .

Since Laplace's equation is invariant under a rotation of axes, then eqn.(7) is invariant and hence transforms to

$$\left(\frac{\partial^2}{\partial x'^2_j} + \frac{\partial^2}{\partial y'^2_j}\right)u = 0\tag{32}$$

Solutions of eqn.(32) that are trigonometric in x'_j will be of the form

$$u = F(y')\sin(mx' + \alpha),\tag{33}$$

and on substituting in (32) it follows that

$$\frac{d^2 F}{dy'^2_j} - m^2 F = 0,$$

the solution to which is

$$F(y'_j) = A e^{-my'_j} + B e^{my'_j}$$

On seeking functions that could represent the propagation inwards of the decaying effects of boundary adjustments on side j , we must exclude the positive exponential part. Accordingly

$$u = A_{mj} e^{-my'_j} \sin(mx' + \alpha),\tag{33a}$$

where the arbitrary constant A may change with m and j and is written as A_{mj} . We define, as in Quinlan (1), on setting $\alpha=0$, Edge-Functions for straight edges in a Laplace problem:

$$E_{mj} = A_{mj} e^{-my'_j} \sin mx'_j\tag{34}$$

where further analysis is required to show that m must be restricted to the discrete set (36).

As in the case of V_{kj} this is a point function and can, if required, be written as $E_{mj}(P)$.

EDGE-FUNCTIONS (half-strip problem)

Recent developments in Edge-Functions show that it is useful to consider the basic problem (b) as approaching the solution in a semi-finite strip on the side AB, ($y'_j = 0$) as base, with infinite sides $x'_j = 0$ and $x'_j = a_j$. Since the other sides of the boundary are assumed to have little effect on the solution along A^*B^* , it follows that the insertion of the semi-infinite sides, with appropriate boundary conditions, will likewise have little effect on the solution along A^*B^* .

Accordingly in the case of sine series expansion (12), we set up the half-strip problem, for Fig. 2 under, with the boundary conditions

$$\begin{aligned} u &= 0 ; \quad x'_j = 0, \quad x'_j = a_j \\ u &\rightarrow 0 \text{ as } y'_j \rightarrow \infty \\ u &= G(x'_j) ; \quad y'_j = 0, \quad 0 \leq x'_j \leq a_j. \end{aligned} \tag{35}$$

We seek a solution, in the usual manner, to eqn.(32) that is trigonometric in x'_j - corresponding to the pair of zero boundary conditions - in the form

$$u = F(y'_j) \sin(mx'_j + \alpha)$$

where, as in (33a), on solving for $F(y'_j)$ and applying condition $u \rightarrow 0$ as $y'_j \rightarrow \infty$, we obtain

$$u = A_{mj} e^{-my'_j} \sin(mx'_j + \alpha) \tag{35a}$$

Conditions $u = 0$ when $x'_j = 0$ and $x'_j = a_j$ require

$$\alpha = 0 ; \quad m = \pi M/a_j ; \quad M = 1, 2, \dots \tag{36}$$

The solution for an actual half-strip follows, in the usual way, on taking the superposition

$$u = \sum_{M=1}^{\infty} A_{mj} e^{-my'_j} \sin mx'_j ; \quad m = \pi M/a_j \tag{37}$$

where the remaining condition (35) requires that

$$G(x'_j) = \sum_{M=1}^{\infty} A_{mj} \sin mx'_j ;$$

the coefficients A_{mj} for a Fourier sine series expansion being given by

$$A_{mj} = \frac{2}{a_j} \int_0^{a_j} G(x'_j) \sin mx'_j dx'_j \tag{38}$$

This analysis indicates that Edge-Functions of type (37), as defined in (34) but restricted to the discrete values $m = \pi M/a_j$, should be in solution "mix" (8). However the coefficients A_{mj} must be obtained by some scheme, as in (9), that allows for the effects of the other boundaries - rather than A_{mj} as given by (38) using the crude mathematical half-strip model. It might reasonably be expected that, when $G(x'_j)$ is not identically zero, the actual calculated coefficients would be of the same order as these resulting from (38).

If expansion (12) is truncated at L harmonics, it follows that the series (37) should be truncated similarly at $M = L$, or in solution "mix" (8), we should include as many Edge-Functions from each side as we have significant - i.e. retained - harmonics on that side.

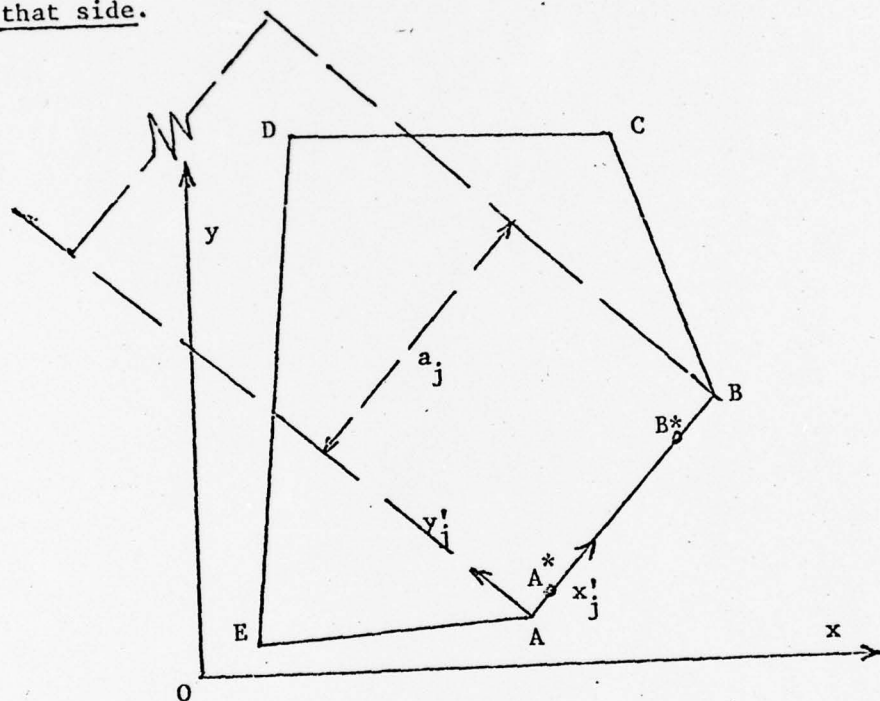


Fig. 2

It is evident that there is a distinct advantage in using the above half-strip model in place of the half-plane model in 20(b).

It remains to provide additional functions equal to the total number of vertex eqs.(19) in the problem - and the obvious sources to provide these are the Vertex Functions V_{kj} . It has been found in practice that it does not matter how many are taken from each of the various vertices, provided that sufficient of the lower eigenvalues $\lambda_1, \lambda_2, \lambda_3 \dots$ are included from each vertex to imbed its asymptotic behaviour into the solution form.

EXAMPLE

It is a simple programming task to set up matrix (17), using eqs.(16) and (19) with truncation at any specified level L, for a polygonal region with continuous temperatures on the boundary. A simple program, LAPEX, is given in Appendix A(1), with some explanatory captions.

Appendix A(2) gives results for a quadrilateral region with zero temperatures on three edges, while on the remaining edge $q = 1$ condition (2) is specified by

$$u = g_1(x,y) = t(1 - t) ; \quad t = x'_1/a_1 \quad (37)$$

The integrals in eqs.(16) are evaluated by the trapezoidal rule, noting that the integrand is zero at both ends of the range since $\sin \eta x'_q$ is then zero. This anticipates the replacement of Harmonic Matching by the much more economical Harmonic Fitting Method as given in Section 5.

The equations are grouped in successive harmonic sets $N = 1, N = 2 \dots \dots N = L$, the equations from each side being arranged consecutively in each set. The columns for the Edge-Functions and the rows for the corresponding harmonic equations are arranged to intersect on the diagonal of the coefficients matrix. This produces diagonally dominant matrix for the E.F. coefficients - the characteristic matrix form for E.F.M. The four vertex equations are arranged in rows after the harmonic equations, and the columns for the corresponding vertex functions follow those for the edge-functions. The right hand side is put up as the final column of the matrix, and the attached simple solution subroutine, Qsolve, solves the matrix using Gaussian elimination.

With a view to their extension in Appendix B simple subroutines EDGEF and POLW are introduced for the evaluation of the respective Edge and Vertex Functions at any point (x_k, y_k) .

The Boundary Residuals - the difference between the values computed from the solution form (8) as obtained and the prescribed boundary values - are computed and printed at a specified MDIV points on each side. Their root mean square is obtained and printed as a suitable overall measure of how well the boundary conditions are satisfied on each side.

A single data card specifying points $A(u_1, v_1)$ and $B(u_2, v_2)$ and MDIV, instructs the program to compute the temperature at each of the MDIV equidistant points on the line AB. In the production phase use is made of that part of the program set up for the calculation of the boundary residuals, merely by cutting the j-cycle to a single pass, corresponding to $j = 1$. An indicator N code = 1 indicates the production phase; the boundary residuals phase being denoted by Ncode = 0. A blank card, which reads MDIV as zero, terminates the program.

SECTION 2

Mixed Boundary Conditions and Full Harmonic Matching

We now proceed to generalise to mixed boundary conditions, where either u or $\frac{\partial u}{\partial y}_q$ can be specified on any edge segment, for regions with or without holes. If the boundary identity (11) for u is expanded in a full Fourier series in the range $(0, a_q)$, eqn. (12) is replaced by the series

$$\psi_q = \sum_{N=0}^{\infty} B_N \cos nx'_q + \sum_{N=1}^{\infty} C_N \sin nx'_q \equiv 0 ; n = 2\pi N/a_q \quad (39)$$

Truncation at L harmonics gives, on zeroing the relevant harmonics,

$$\begin{aligned} B_N &= 0 ; C_N = 0 ; N = 1 \text{ to } L \\ B_0 &= 0 ; \text{ (zero harmonic term)} \end{aligned} \quad (40)$$

This process will be called Full Harmonic Matching, and the corresponding $2L + 1$ eqs., analogous to eqs. (16), can be written as

$$\sum_{i=1}^I A_i \int_0^a u_i^c(x'_q) \cos(nx'_q + \alpha_k) dx'_q - \int_0^a g_q(x'_q) \cos(nx'_q + \alpha_k) dx'_q = 0, \quad (41)$$

for the sequence $N = 0 ; N = 1$ with $k = 1$ and $k = 2, \dots$

$\dots ; N = L$ with $k = 1$ and $k = 2$; where

$$\alpha_1 = 0 ; \alpha_2 = \pi/2$$

Each side, including each hole, contributes its set of harmonic equations, and the corresponding cosine and sine harmonics from all sides are grouped together in harmonic sets $N = 0, 1, \dots, L$, to constitute the rows of the coefficients matrix.

Again as in (18) convergence of series (39) to the order $1/N^3$ can be insured by setting, in accordance with Fourier series theory,

$$\left. \begin{aligned} \psi_q(x'_q) = 0 \\ \frac{\partial \psi_q}{\partial x'_q}(x'_q) = 0 \end{aligned} \right\} x'_q = 0 \text{ and } x'_q = a_q \quad (43)$$

A similar analysis applies on any segment q on which the normal derivative $\frac{\partial u}{\partial y'_q}$ is prescribed, in which u is replaced by $\frac{\partial u}{\partial y'_q}$ in the boundary identity (10) and $\psi_q(x'_q)$ is defined accordingly.

Four vertex equations follow from eqs.(43) for each boundary segment, though care must be taken, as in subroutine COLMAT in Appendix B, to omit equations that are redundant - as is the case when u is continuous at a vertex.

The computation of eqs.(43) for all possible boundary conditions requires that the following derivatives:

$$\frac{\partial}{\partial y'_q}; \quad \frac{\partial}{\partial x'_q}; \quad \frac{\partial^2}{\partial x'_q \partial y'_q}, \quad (44)$$

be available for all functions in solution (8).

We now proceed to reexamine the basic problems, and to obtain the necessary derivatives.

[a] Vertex Functions for the j^{th} vertex

When mixed boundary conditions occur on the sides of sector in Fig.3, we identify four possibilities, distinguished by the indicator $NVEX = NVER(j)$, where

(i) Case I: $NVER(j) = 1$ corresponding to conditions $u = 0$ on both sides j and $(j-1)$ of sector, which as in (25) gives

$$A = 0; \quad B = 1; \quad \lambda_k = k\pi/\alpha; \quad k = 1, 2, \dots \quad (45)$$

(ii) Case II: $NVER(j) = 2$ corresponding to conditions $u = 0$ on side j on which $\theta = 0$, and $\frac{\partial u}{\partial n} = 0$ on side $j-1$ on which $\theta = \alpha$; \hat{n} denotes the normal to side $\theta = \alpha$ and can be denoted by the unit vector $\hat{\theta}$.

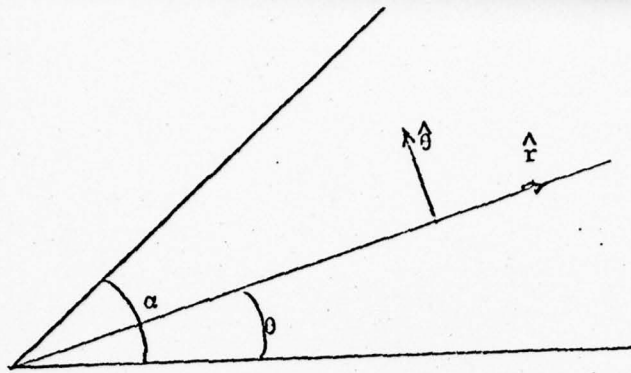


Fig. 3

Since in polar coordinates

$$\frac{\partial u}{\partial n} = \hat{\theta} \cdot \nabla u = \hat{\theta} \cdot \left[\frac{\partial u}{\partial r} \hat{r} + \frac{1}{r} \frac{\partial u}{\partial \theta} \hat{\theta} \right] \quad (46)$$

it follows on applying to eqn.(21) that

$$\frac{\partial u}{\partial n} = \lambda r^{\lambda-1} [-A \sin \lambda \theta + B \cos \lambda \theta] \quad (47)$$

Accordingly on analysing the sector conditions, as in (22), we obtain

$$A = 0 ; B = 1 ; \cos \lambda \alpha = 0, \quad (48)$$

the eigenvalues of which are

$$\lambda_k = k\pi/\alpha - \pi/2\alpha ; k = 1, 2, \dots \quad (49)$$

and the corresponding corner functions are given by form (21) on using the above values for A, B, and λ .

(iii) Case III: $NVER(j) = 3$ corresponding to conditions $\frac{\partial u}{\partial n} = 0$ on $\theta = 0$, and $u = 0$ on $\theta = \alpha$. Results follow similarly and are given in table I under.

(iv) Case IV: $NVER(j) = 4$ corresponding to conditions $\frac{\partial u}{\partial n} = 0$ on $\theta = 0$, and $\frac{\partial u}{\partial n} = 0$ on $\theta = \alpha$. All results are collected together in table I under, in a form that facilitates programming.

NVEX	λ_k	A	B
1	$k\pi/\alpha$	0	1
2	$k\pi/\alpha - \pi/(2\alpha)$	0	1
3	$k\pi/\alpha - \pi/(2\alpha)$	1	0
4	$k\pi/\alpha$	1	0

TABLE I - VERTEX FUNCTIONS

The corresponding Vertex Functions for j^{th} vertex are denoted by V_{kj} where

$$V_{kj} \equiv u = r^{\lambda_k} [A \cos \lambda_k \theta + B \sin \lambda_k \theta]; \quad (50)$$

The physics of the problem requires that λ_k be positive. The normal derivative for any direction \hat{y}'_q follows, as in (46):

$$\begin{aligned} \frac{\partial u}{\partial y'_q} &= \hat{y}'_q \cdot \nabla u = \hat{y}'_q \cdot \left[\frac{\partial u}{\partial r} \hat{r} + \frac{1}{r} \frac{\partial u}{\partial \theta} \hat{\theta} \right] \\ &= \{-A \sin(\lambda' \theta + \phi_{qj}) + B \cos(\lambda' \theta + \phi_{qj})\} \lambda r^{\lambda'-1}; \quad \lambda' = \lambda - 1 \\ &\quad \phi_{qj} = \phi_q - \phi_j \end{aligned} \quad (51)$$

on noting that

$$\begin{aligned} \hat{y}'_q \cdot \hat{r} &= \cos(\pi/2 + \phi_{qj} - \theta) = -\sin(\phi_{qj} - \theta) \\ \hat{y}'_q \cdot \hat{\theta} &= \cos(\phi_{qj} - \theta) \end{aligned}$$

where \hat{y}'_q is a unit vector, normal to direction \hat{x}'_q , which makes an angle ϕ_q with y axis, or an angle $\pi/2 + \phi_{qj}$ with the j th side.

[b] (Full) Edge-Functions

Since expansion (39) involves a full Fourier Series expansion on the side AB in Fig.1 with periodic extension to all other points on the infinite line of which AB is part, it is appropriate to introduce periodicity into boundary conditions (35) for the half-strip problem of Fig.2. Accordingly we take the boundary conditions on the infinite sides of the strip as

$$u(0, y'_j) = u(a_j, y'_j); \quad y'_j \geq 0, \quad (52)$$

which on applying to solution (35a) require

$$m = 2\pi M/a_j \quad (53)$$

Superposition (37) must then be replaced by

$$u = A_0 + \sum_{M=1}^{\infty} A_{mj} e^{-my'_j} \cos mx'_j + \sum_{M=1}^{\infty} B_{mj} e^{-my'_j} \sin mx'_j, \quad (54)$$

where Fourier integrals, similar to (38), follow for A_{mj} and B_{mj} in an actual half-strip problem. As previously we associate the cosine and sine edge-functions in (54) with the corresponding cosine and sine harmonics, (M corresponds to N) in expansion (39) - and arrange that the corresponding columns and rows intersect on the diagonal of the coefficients matrix.

The above cosine and sine edge-functions are combined in the following definition:

(Full) Edge-Functions, for side j.

$$E_{mj} = E_{mj}^1 + E_{mj}^2 ; E_{mj}^k = A_{mj}^k e^{-m_j y_j'} \cos(m_j x_j' + \alpha_k); k = 1, 2$$

$$m_j = 2\pi M/a_j ; \alpha_1 = 0, \alpha_2 = \pi/2, \quad (55)$$

where $k = 1$ and $k = 2$ give the respective cosine and sine edge-functions. The term (Full) is usually omitted, as expansion (39) is more economical computerwise than the sine series expansion in (12).

If truncation on any side j in expansion (39) is taken at L, then L (Full) Edge-functions from side j should be included in solution "mix" (8) to match up the corresponding harmonics for $N = 1, 2, \dots, L$. This leaves the zero harmonics in (41) - the total number for the problem being tallied as NZERO - unmatched by corresponding edge-functions, and an additional NZERO Vertex Functions must be provided to match these. The equations for the zero harmonics are inserted, as rows in the matrix, directly after harmonic set L.

As in eqn.(51) the required normal derivative for the direction \hat{y}'_q follows as:

$$\begin{aligned} \frac{\partial E_{mj}^k}{\partial y'_q} &= \hat{y}'_q \cdot \nabla(E_{mj}^k) \\ &= \hat{y}'_q \cdot \left\{ \hat{x}'_j \frac{\partial E_{mj}^k}{\partial x'_j} + \hat{y}'_j \frac{\partial E_{mj}^k}{\partial y'_j} \right\} \\ &= -A_{mj}^k m_j e^{-m_j y_j'} \cos(m_j x_j' + \phi_{qj} + \alpha_k); \phi_{qj} = \phi_q - \phi_j, \end{aligned} \quad (56)$$

since

$$\hat{y}'_q \cdot \hat{x}'_j = -\sin \phi_{qj} ; \hat{y}'_q \cdot \hat{y}'_j = \cos \phi_{qj}.$$

[c] Curved Boundaries Including Holes

The polar solutions of Laplace's eqn. in (21) apply to closed curves - where θ acts as a parameter for points in the boundary with a range $(0, 2\pi)$ - provided that λ is an integer, say $\lambda = k$. Positive values for k correspond to an outer boundary, and negative values to an inner one.

Development follows as above for (Full) Edge-Functions, where the range of 2π for θ corresponds to a_j for x'_j . Accordingly when truncation is at L, for any closed curve we must include in set(8) the functions

$$\sum_{j=1}^{j^*} \left\{ \sum_{k=0}^L P_{kj} \right\} \quad (57)$$

where the asterisk denotes that the terms only apply to closed curved boundaries j arising in the problem, with

$$P_{kj} = A_{Mj} r^{Mj} \cos(M_j \theta) + B_{Mj} r^{Mj} \sin(M_j \theta)$$

$$M = 1, 2, 3, \dots$$

$$M_j = M, \text{ if } j \text{ relates to a curved outer boundary} \quad (58)$$

$$= -M, \text{ if } j \text{ relates to a curved inner boundary}$$

When $k = 0$, we require the zero order harmonics

$$P_{0j} = A_{0j} + B_{0j} \log r \quad (59)$$

where the $\log r$ term is omitted if the section is solid.

The origin for the coordinates (r, θ) can be taken at any suitable point within the corresponding closed curve, and could be taken at different locations if both inner and outer boundaries are closed curves.

The normal derivative $\frac{\partial P_{kj}}{\partial y'_q}$ follows from eqn.(51) on putting $\lambda = N_k$.

Note: Functions (58) may be thought of as Edge-Functions for the corresponding curved boundaries, and must be inserted in the appropriate rows in the harmonic sets in the Matrix.

[d] Logarithmic Singularities at Vertices: Log-Vertex Functions

As noted in the analysis of vertex functions (21) infinite derivatives of order p , where $p > \lambda_k$, occur when λ_k is not an integer. On performing a limiting analysis as $\lambda_k \rightarrow K^*$, where K^* is an integer, it can be shown that the solution to eqn.(20) denoted by V_{kj}^* , for $\lambda = \lambda_k = K^*$, where

$$V_{kj}^* = \frac{\partial}{\partial \lambda} [Ar^\lambda \cos \lambda \theta + Br^\lambda \sin \lambda \theta], \quad (60)$$

$$= [Ar^\lambda \cos \lambda \theta + Br^\lambda \sin \lambda \theta] \log r$$

$$- \lambda Ar^\lambda \sin \lambda \theta + \lambda Br^\lambda \cos \lambda \theta, \quad (61)$$

satisfies the following conditions:-

(i) Function (60) satisfies Laplace eqn.(20) since the derivative w.r.t. the parameter λ of any solution (21) of eqn.(20) also satisfies the same equation.

(ii) the coefficient of $\log r$ - the singular part of the solution - is zero on the sides $\theta = 0$ and $\theta = \alpha$, since A , B and λ are determined in Table I for the specified boundary conditions on the angle. However, in the interior of the angle, $0 < \theta < \alpha$, the coefficient of $\log r$ is not zero, and hence the solution provides logarithmic singular behaviour within the angle.

(iii) an analysis, as in (27), of the total heat flow from term V_{kj}^* through any circular arc isolating the vertex, gives - on interchanging the operations of differentiation and integration -

$$q_j = \frac{\partial}{\partial \lambda} [r^\lambda (1 - \cos \lambda \alpha)]$$

$$= r^\lambda [(1 - \cos \lambda \alpha) \log r + \lambda \sin \lambda \alpha], \lambda = \lambda_k ; \quad (62)$$

and consequently form (60) is physically admissible in solution "mix"(8) provided condition (28), $\lambda_k > 0$, is satisfied.

Note: Functions V_{kj}^* will subsequently be called LOG-VERTEX FUNCTIONS.

[e] Derivatives required in set (44)

The derivative $\frac{\partial}{\partial x'_q}$ follows from those given in eqs.(51) and (56) on replacing ϕ_{qj} by $\phi_{qj} - \pi/2$, since the direction \hat{x}'_q is $\pi/2$ behind that of \hat{y}'_q .

On applying the operator

$$\frac{\partial}{\partial x'_q} = \hat{x}'_q \cdot \nabla,$$

to eqs.(51) and (56) respectively we obtain

$$\frac{\partial^2 V_{kj}}{\partial x'_q \partial y'_q} = \{-A \sin(\lambda'' \theta + 2\phi_{qj}) + B \cos(\lambda'' \theta + 2\phi_{qj})\} \lambda' \lambda'' r^{\lambda''},$$

$$\lambda'' = \lambda - 2; \lambda' = \lambda - 1$$

$$\frac{\partial^2 E_{kj}}{\partial x'_q \partial y'_q} = A_{mj}^k m_j^2 e^{-m_j y'_j} \sin(m_j x'_j + 2\phi_{qj} + \alpha_k) \quad (63)$$

Derivatives of functions (58) follow from these in (63) on putting $\lambda = N_k$.

Difficulties are encountered in deriving analytical expressions for derivatives for Vertex Eqs.(43) in the case of curved boundary segments, as shown in section 3(a), and numerical differentiation is recommended in these cases.

Accordingly to enable the program LAPGEN in Appendix 2 to deal with all boundary segments, whether straight or curved, numerical differentiation is used for derivatives $\frac{\partial \psi_q}{\partial x_q}$ required in Vertex Eqs.(45)

for all straight line segments. This is based on the two-point formula

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}, \quad (64)$$

and the corresponding computation - involving the computation of two point values - we shall refer to as a "two-point" equation. Similarly analytical expressions for the derivatives of V_{kj}^* in (60) are best obtained by computing numerically, using formula (64), the $\frac{\partial}{\partial \lambda}$ of the corresponding derivatives of V_{kj} . These are given in POLW, where the increment $\Delta \lambda = \text{delta}$ is read in.

PROGRAM LAPGEN

The program in Appendix A(1) now requires to be expanded to include:

(i) Examine all vertex eqs. for redundancy, and - for convenience - set up all information to enable the corresponding point-eqs. to be put up in rows of matrix, after those for the zero harmonics. The assigned subscripted variables are - position, (XFIX,YFIX); associated slope, BFIX; side, JFIX; function indicator MBFIX; location subscript, (LOC); and position on side, (DFIX). The number of equations is tallied by NFP.

(ii) Total no. Vertex Functions NORF required is

$$\text{NORF} = \text{NFP} + \text{NZERO} \quad (66)$$

where NZERO is total no. zero harmonics (= no. of sides NS in a Laplace problem).

(iii) Provide for distribution of NORF Vertex Functions between the vertices of polygon, the number required at j^{th} vertex being computed as NAD(j). Locate any integer values for λ_k and - arrange to include the corresponding Log-Vertex functions (60).

(iv) Set up, in subscripted locations, the information required for computing the corresponding column sets, NP, of the matrix:- function type NFN; vertex location NA; associated A and B from Table 1 in coef(1,NP), coef(2,NP); and eigenvalue λ_k in E. It is economical to put up functions (52) and (57) two columns at the time - corresponding to each value of M - and each two such columns is said to constitute a column set. Vertex functions (51) are put up in single columns, or column sets of one.

(v) Put up right hand side of matrix as its last column set, LP.

The above tasks are performed in subroutine COLMAT in Appendix B(1).

Function indicators, MT, are introduced, in accordance with the following table:

MT	1	2	11	12	3
Function	u	$\frac{\partial u}{\partial y_q}$	$\frac{\partial u}{\partial x_q}$	$\frac{\partial^2 u}{\partial x_q \partial y_q}$	v

Table 2: Function Indicators

Note The harmonic conjugate v of u is given by MT = 3.

A more elaborate solution program SOLCOR is required, which uses Gaussian elimination on the harmonic eqs., and single pivoting to locate the maximum element in the corresponding column for zero harmonic and point-eqs. These latter equations, since they are matched by Vertex functions, cannot be put up in a diagonally dominant pattern in the matrix.

SOLCOR solves coefficients eqs.(17) not only for the specified truncation level at L harmonics, but can also obtain, as a by product - requiring only a few percent increase in the overall computer time - the solutions for respective truncations at L-1, L-2 and L-3 harmonics.

The indicator LS specifies the number of solutions required where $LS \leq 4$ in the present program. Accordingly, exact solutions are obtained to LS mathematical models, of increasing complexity, for the given problem.

Provision has to be made to provide the derivatives listed in set (44) for the point-fns. computed in subroutines EDGEF and POLW. Subroutines POLC and PMAP are added to provide respective Curved Edge Functions for circular or elliptic boundary segments and/or corresponding shaped holes.

Consequential adjustments are required in LAPEX in the information input, and in the putting up of the matrix. Points on ellipses are located, if required, by the usual parametric coordinates $x = a \cos \phi$, $y = b \sin \phi$, and are computed, together with corresponding slopes, in subroutine ELPS.

Provision is made to adjust the number of integration points, NOK(j), on any side by reading in a proportionality factor NHS(j). These are roughly proportional to the lengths of the corresponding boundary segments, and

$$NOK(j) = 2 * FSET * L * NHS(j) + 3 \quad (67)$$

where FSET is a factor ≥ 1 . Fset increases proportionally the number of integration points on all sides, as is required later when using Harmonic Fitting, as developed in Section 5, in place of Harmonic Matching.

An indicator NBDY is used to indicate the type of boundary conditions (2) arising in the problem. Thus NBDY = 1 indicates that on the j^{th} side - where the occurrence of non-zero boundary conditions is indicated by the subscripted variable NBY(j) = 1 - the boundary conditions are of the polynomial form

$$g_j(x,y) = \sum_{k=1}^6 C(j,k)t^{k-1}, \quad (65)$$

where $t = x_j^i/a_j$. The required coefficients C(j,k) are read in for each side which has NBY(j) = 1.

Computation of point values of (65) is done in a subroutine POLP, which also provides numerical coefficients - without requiring to read them in as in the case of NBDY = 1 - for the torsion problem as indicated by NBDY = 2. Additional cases can be inserted, corresponding to values of NBDY > 2, in POLP as the user requires.

The checking of the boundary residuals is as before except that it is done for LS truncation levels.

Production is based on one data card for each line of production, but is now done in a special subroutine PRODN, which enables various operations indicated by NCODE = 1,2,3 - differentiations, principal stress computations and integrations - to be performed. The user can readily incorporate into PRODN any further facilities that may be required.

The expanded program, LAPGEN, with explanatory captions is attached in appendix B(1), and an illustrative example is given in Appendix B(2). A user guide to inputting data on the black box principle is given in Appendix C.

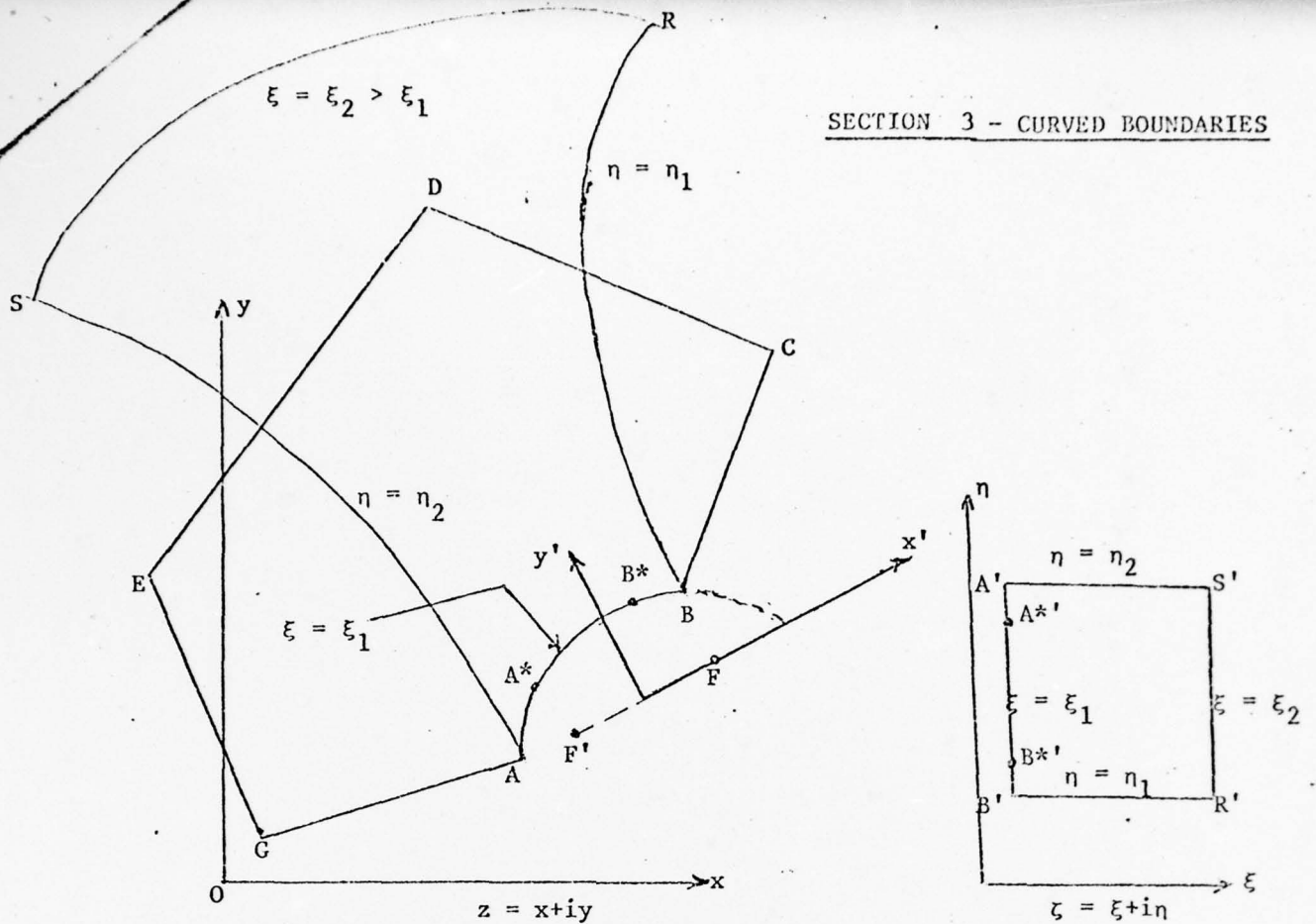


Fig. 4

(a) Curved Edge-Functions for Curved Boundary Segment AB

If the mapping

$$\zeta = F(z)$$

(68)

maps the rectilinear region $A'B'R'S'$ in the ζ -plane onto the curvilinear region $ABRS$ in the z -plane, then any solution of Laplace's eq. (7) in the ζ -plane satisfies Laplace's eq. in the z -plane.

Analogous to the half-strip approximation (35) to basic problem (b) at (20), the solution near A^*B^* - where the other sides are assumed to have little effect - has the characteristics of the solution in $ABRS$ in the region A^*B^* , when, as in (52), we set

$$u(\xi, \eta_1) = u(\xi, \eta_2) ; \quad \xi \geq \xi_1,$$

(69)

as the conditions on sides BR and AS respectively.

These conditions can be satisfied by requiring that u be periodic in η ,

the period being $\eta_2 - \eta_1$. In the transformed problem in the ζ -plane, the solution in the region of A^*B^* is characterised by edge-functions, analogous to functions (55), which may be written in the form

$$C_{mj} = C_{mj}^1 + C_{mj}^2; \quad C_{mj}^k = A_{mj}^k e^{-m_j \xi_j} \cos(m_j \eta_j + \alpha_k); \quad k = 1, 2$$

$$m_j = 2\pi M / (\eta_2 - \eta_1); \quad \alpha_1 = 0, \quad \alpha_2 = \pi/2 \quad (70)$$

where the mapped plane ζ corresponding to the j^{th} side, when curved, is denoted by

$$\zeta_j = \xi_j + i\eta_j, \quad (71)$$

and may be termed the ζ_j -plane.

In subsequent work it is advantageous to work with the complex variable ζ_j , and to write function (70) in the form

$$C_{mj} = \text{Re}\{A_{mj} e^{-m_j \zeta_j}\} = e^{-m_j \xi_j} [A'_{mj} \cos(m_j \eta_j) - A''_{mj} \sin(m_j \eta_j)] \quad (72)$$

where A_{mj} is a complex constant

$$A_{mj} = A'_{mj} - iA''_{mj} \quad (73)$$

and Re denotes the real part.

Accordingly, the appropriate functions C_{mj} must be introduced into solution mix (8) for each curved edge-segment.

We now proceed to find the derivatives of C_{mj} as required in set (44).

Analogous to mapping (68) we can write the mapping for curved side j as

$$\zeta_j = F_j(z) \quad (74)$$

Since C_{mj} is a function of z and \bar{z} , say

$$C_{mj} = G(z, \bar{z}); \quad z = x+iy; \quad \bar{z} = x-iy$$

we require the operators

$$\begin{aligned} \frac{\partial}{\partial x'_q} &= e^{i\phi_q} \frac{\partial}{\partial z} + e^{-i\phi_q} \frac{\partial}{\partial \bar{z}} \\ \frac{\partial}{\partial y'_q} &= ie^{i\phi_q} \frac{\partial}{\partial z} - ie^{-i\phi_q} \frac{\partial}{\partial \bar{z}} \end{aligned} \quad (75)$$

The operator $\frac{\partial}{\partial x'_q}$ follows on noting, from fig.1, that the coordinates

of any point $P(x, y)$ are given by:

$$\begin{aligned} x &= x'_q + x'_q \cos \phi_q - y'_q \sin \phi_q \\ y &= y'_q + x'_q \sin \phi_q + y'_q \cos \phi_q, \end{aligned} \quad (76)$$

where the coordinates of the q^{th} vertex are (x_q, y_q) . On operating on $G(z, \bar{z})$ it follows that

$$\frac{\partial G}{\partial x'_q} = \frac{\partial G}{\partial z} \frac{\partial z}{\partial x'_q} + \frac{\partial G}{\partial \bar{z}} \frac{\partial \bar{z}}{\partial x'_q},$$

and on substituting

$$\frac{\partial z}{\partial x'_q} = \frac{\partial x}{\partial x'_q} + i \frac{\partial y}{\partial x'_q} = e^{i\phi_q}; \quad \frac{\partial \bar{z}}{\partial x'_q} = e^{-i\phi_q},$$

the operator for $\frac{\partial}{\partial x'_q}$ is obtained. The operator $\frac{\partial}{\partial y'_q}$ follows similarly.

Accordingly we obtain

$$\begin{aligned} \frac{\partial C_{mj}}{\partial y'_q} &= \text{Re}[A_{mj} \frac{\partial}{\partial y'_q} (e^{-m_j \zeta_j})] \\ &= \text{Re}[-im_j A_{mj} e^{i\phi_q} \frac{d\zeta_j}{dz} e^{-m_j \zeta_j}] \end{aligned} \quad (78)$$

The parameter η_q , for segment q , now corresponds to x'_q - the parameter for points on q^{th} line segment - as used in harmonic matching eqs.(41). Analogous to the vertex eqs.(43), we require derivatives w.r.t. η_q for all functions in solution "mix" and for all their normal derivatives. These are difficult to compute and are best obtained by numerical differentiation using "two-point" eqs. formula (64) as set up in LAPGEN.

(b) Example: Elliptic indentation

If AB in Fig.4 is a segment of an ellipse, with principal axes x' and y' , the required mapping is

$$z' = c \cosh \zeta; \quad z' = x' + iy' \quad (79)$$

from which the point correspondences are

$$\begin{aligned} x' &= c \cosh \xi \cos \eta \\ y' &= c \sinh \xi \sin \eta \end{aligned} \quad (80)$$

The mapping is rendered single valued by taking the limits $\xi(0, \infty)$ and $\eta(0, 2\pi)$. The corresponding singular points in the z -plane are at $F'(-c, 0)$, $F(c, 0)$ with a cut from $-c$ to c on the x' axis.

On eliminating η from eqs(80) it follows that

$$\frac{x'^2}{c^2 \cosh^2 \xi} + \frac{y'^2}{c^2 \sinh^2 \xi} = 1, \quad (81)$$

or any line $\xi = \xi_0$ in the ξ -plane corresponds to an ellipse in the z' -plane with semi-axes $a = c \cosh \xi_0$, $b = c \sinh \xi_0$; $c^2 = a^2 - b^2$. Similarly $\eta = \eta_0$ corresponds to a branch of a hyperbola that has the same foci as the ellipse. Then, it is easily shown that AERS is mapped onto A'B'R'S' in a one-to-one manner.

To obtain point-values for the coefficients of A'_{mj} and A''_{mj} from C_{mj} , or its derivatives, we require to find ξ and η for any point $P(x', y')$ in the domain Ω - where (x', y') are coordinates of P w.r.t. principal axes for elliptic indentation. Equation (81) gives the quadratic

$$c^2 T^2 + T(c^2 - x'^2 - y'^2) - y'^2 = 0; \quad T = \sinh^2 \xi,$$

from which

$$2c^2 T = -(c^2 - x'^2 - y'^2) + \sqrt{(c^2 - x'^2 - y'^2)^2 + 4c^2 y'^2}, \quad (82)$$

the positive square root being taken since $T > 0$. When $y' = 0$ then

$$\begin{aligned} T &= x'^2/c^2 - 1; & x'^2 &\geq c^2 \\ &= 0; & x'^2 &< c^2 \end{aligned} \quad (82a)$$

It follows that

$$\xi = \sinh^{-1}(\sqrt{T}) = \log_e(\sqrt{T} + \sqrt{1+T}), \quad (83)$$

and eqs (80) then give

$$\eta = \tan^{-1} \left(\frac{y' \cosh \xi}{x' \sinh \xi} \right), \quad (84)$$

where η must be adjusted, for quadrant location of P , to lie in $(0, 2\pi)$

A suitable subroutine, PMAP, similar to EDGEF and POLC in Appendix B, is included. The values for η_1 and η_2 are obtained from (84).

Note that singular points of mappings (74) are not allowable in the domain of solution (8), which must be regular at all points other than the singular points specified as in (3). If the singular points of any mapping occur in the domain Ω of the problem then either (i) split Ω into elements to ensure that the domain of any set of curved edge-functions does not include the corresponding singular points of the transformation, or (ii) superpose other suitable functions, as over in (e), to neutralise any unwanted singularities.

(c) Circular Indentations

This follows as in (b) on using the transformation

$$z = e^{\zeta} \quad (85)$$

Alternatively, if AB in Fig.4 is an arc of a circle, the corresponding basic problem can be taken as that for a curvilinear region bounded by radii $r = r_1$, $r = r_2$; and radial lines $\theta = \theta_1$, $\theta = \theta_2$, where $r_2 \rightarrow \infty$

The solution then follows in form (57) on setting

$$M_j = -2\pi M / (\theta_2 - \theta_1) \quad (86)$$

(d) Smooth Inner Boundaries

If the inner boundary is a smooth curve the corresponding Curved Edge Functions, analogous to polar functions (58), follow from eq.(82), on adding the periodicity requirement

$$\eta_1 = 0 ; \quad \eta_2 = 2\pi \quad (87)$$

(e) Solid Elliptical Section

Let us examine the mapping

$$z = c \cosh \zeta, \quad (88)$$

as in (b), in the case of a region bounded by a solid ellipse, shown in Fig. 5.

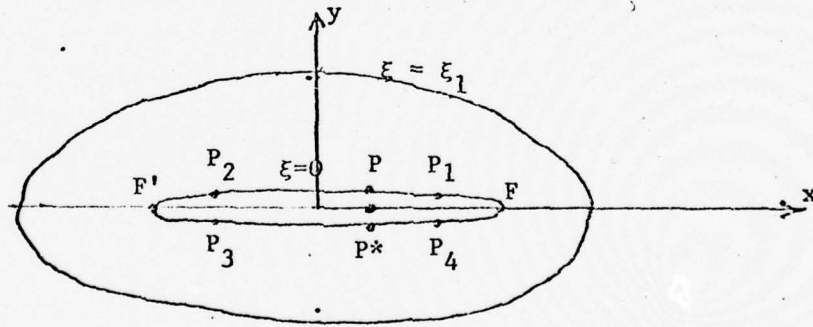


Fig. 5

The corresponding Curved Edge-Functions for $\xi = \xi_1$, analogous to those in (72), that decay inwards from the boundary, are

$$C_M = \text{Re}\{\Lambda_M e^{M\zeta}\} = e^{M\xi} [\Lambda_M' \cos M\eta - \Lambda_M'' \sin M\eta] \quad (89)$$

These require a cut FF' in the z -plane, which corresponds to the limiting ellipse $\xi = 0$. On taking $\xi(0, \infty)$, $\eta(0, 2\pi)$ to render mapping single-valued η is discontinuous on the cut, having values, at respective neighbouring points, (i) of $\eta = \alpha$ and $\eta = 2\pi - \alpha$ at P_1 and P_4 and (ii) of $\eta = \pi - \alpha$ and $\eta = \pi + \alpha$ at P_2 and P_3 where $0 \leq \alpha \leq \pi/2$.

It follows that $\cos M\eta$ is continuous on cut FF' but $\sin M\eta$ is discontinuous as it changes sign - and accordingly C_M is discontinuous. The functions got on replacing M by $-M$ in (89)

$$C_{-M} = \text{Re}\{A_{-M} e^{-M\xi}\} = e^{-M\xi} [A'_{-M} \cos M\eta + A''_{-M} \sin M\eta], \quad (90)$$

have a similar discontinuity on FF' , and on superposing, the resulting discontinuity on $\xi = 0$ is

$$2(A''_M - A''_{-M}) \sin M\alpha,$$

and this reduces to zero on setting

$$A''_{-M} = A''_M \quad (91)$$

Analogous to eq. (78), and noting from eq. (88) that

$$\frac{d\xi}{dz} = \frac{1}{c \sinh \xi}, \quad (92)$$

the discontinuity between points P and P^* in the derivation $\frac{\partial}{\partial y_q}$ of $C_M + C_{-M}$ is, on $\xi = 0$:

$$\begin{aligned} & \text{Re}[iMe^{i\phi_q} \cdot \left(\frac{1}{ic \sin \eta}\right) \{A_M e^{iM\eta} - A_{-M} e^{-iM\eta}\}]_{P}^{P^*} \\ &= \frac{2M}{c} \text{Re}[e^{i\phi_q} (A_M - A_{-M}) \frac{\cos M\eta}{\sin \eta} + (A_M + A_{-M}) \frac{\sin M\eta}{\sin \eta}]_{P}^{P^*} \end{aligned} \quad (93)$$

and hence the discontinuity in the term $\cos M\eta/\sin \eta$ vanishes if

$$A_M = A_{-M} \quad (94)$$

Also note that at the singular points of mapping (88) the function $C_M + C_{-M}$ and its derivative $\frac{\partial}{\partial y_q}$ is everywhere finite since limit of $(\sin M\eta/\sin \eta) \rightarrow M$ at points F' , $\eta = \pi$, and F , $\eta = 0$ or 2π .

Similarly all the higher derivatives of $C_M + C_{-M}$ can be shown to be finite and continuous at all points within and on the elliptical boundary.

If a region is bounded internally and externally by ellipses, then the curved Edge Functions for the outer bounded must be rendered finite and continuous by superposing the corresponding C_{-M} when the foci F and F' of the outer ellipse are in the region of the problem.

(f) Approximate Curved Edge-Functions

If the mapping (79) is used to map* the side A'B of the rectangle A'B'R'S' onto an elliptic arc in the z-plane that is not too different from the curved boundary segment AB, then the functions C_{mj} as got from the elliptic mapping approximates pretty well the characteristics of the solution in region A B** and can be included in solution "mix" (8).

Points on the actual boundary can be expressed in terms of ξ and η , where η is in (η_1, η_2) . The parameter η can be used for the actual boundary points, the corresponding ξ being where the curve η intersects the actual boundary - and η_q , for segment q, then corresponds to parameter x'_q in eqs.(51) for harmonic matching.

Accordingly on exact mapping (79) is not essential, and in most practical cases a "fitted" elliptic mapping is sufficient.

* In practice this would require the fitting, by a least squares criterion, of an elliptic arc - involving five parameters (axes, centre, orientation) - to the prescribed arc AB. We shall refer to the resulting mapping as a "fitted" mapping.

The solution "mix" for eq.(1) can then be written as

$$\begin{aligned} u &= u_0 + u(i) \\ &= u_0 + u(ii) \end{aligned} \quad (93)$$

for regions I and II respectively, where u_0 denotes all the functions that are common to both regions and are continuous, and with continuous derivatives, across the cut.

The functions $u(i)$ and $u(ii)$, which are confined to regions I and II respectively, must then be matched across the cut to give continuity, in a least squares sense, across the cut for u and $\frac{\partial u}{\partial n}$.

Since the Edge-Functions from DC and DE produce a discontinuity across AD, this must be negated by superposing functions that are characteristic of a line, AB, of discontinuities in u and $\frac{\partial u}{\partial n}$. Analogous to the half-strip model (52) the solution characteristics in region A*D* are similar to those for an infinite strip bounded by infinite sides perpendicular to AD through A and D respectively.

On taking edge-axes x' and y' for AD as shown in Fig.6, we require that the solution $u(x',y')$ satisfy the following boundary conditions for the strip:

- (i) $u(x',y') \rightarrow 0$ as $y' \rightarrow \infty$ and $y' \rightarrow -\infty$,
- (ii) $u(0,y') = u(a',y')$, for all points on $x'=0$ and $x'=a'$, where $a' = AD$.

Since $u(x',y')$ satisfies Laplace's eqn. suitable forms for u , in semi-finite strips $y' > 0$ and $y' < 0$, analogous to forms (54), are, with $m = 2\pi M/a'$:

$$\begin{aligned} u &= \sum e^{-my'} \{A_m \cos mx' + B_m \sin mx'\} , \quad y' > 0 \\ &\quad \sum e^{my'} \{A'_m \cos mx' + B'_m \sin mx'\} , \quad y' < 0 \end{aligned} \quad (94)$$

In an actual strip problem, if the discontinuities in u and $\frac{\partial u}{\partial y}$, were specified as $f(x')$ and $g(x')$ respectively, it follows that

$$\begin{aligned} f(x') &= \sum \{(A_m - A'_m) \cos mx' + (B_m - B'_m) \sin mx'\} \\ g(x') &= \sum \{-m(A_m + A'_m) \cos mx' + -m(B_m + B'_m) \sin mx'\}, \end{aligned}$$

and on using Fourier coefficients formulae the coefficients in (94) follow.

However in an actual problem, as in Fig. 6, the coefficients must be determined as in (11) which allows for the influence of all sides, and functions (94) must be included in solution "mix" (8).

These latter can be identified as Edge-Functions of type (55) for the upper and lower side of the cut AD, where the upper side has indicator j and the lower has $(j + 1)$.

(b) SINGULARITY ON EDGE

Consider a singularity, as in (3),

$$u = Q \delta(\eta - \eta_0)$$

acting at the point P, parameter η_0 , on curved boundary segment AB in Fig. 4. On taking ABRS as the basic region with conditions (69), and superposing solutions (72), to satisfy the conditions on AB, $\xi = \xi_1$, we require

$$Q \delta(\eta - \eta_0) = \sum e^{-m\xi_1} [A_{mj}^{(1)} \cos(m\eta) + A_{mj}^{(2)} \sin(m\eta)]. \quad (95)$$

The coefficients follow, on taking a summation $M = 0$ to $M = \infty$, as

$$A_{mj}^{(1)} e^{-m\xi_1} = \frac{Q}{\eta_2 - \eta_1} \int_{\eta_1}^{\eta_2} \delta(\eta - \eta_0) \cos(m\eta) d\eta,$$

which gives for $A_{mj}^{(1)}$ and, similarly, for $A_{mj}^{(2)}$

$$A_{mj}^{(1)} = \frac{Q e^{m\xi_1} \cos(m\eta_0)}{\eta_2 - \eta_1} \quad (96)$$

$$A_{mj}^{(2)} = \frac{Q e^{m\xi_1} \sin(m\eta_0)}{\eta_2 - \eta_1}$$

Series (95) provides a useful particular integral (4) for above type of boundary singularity. Since its value is known, $Q\delta(\eta - \eta_0)$, on AB, the series only requires to be evaluated at interior points (ξ, η) where $\xi > \xi_1$, and it is then dominated by the negative exponential factor $e^{-m(\xi - \xi_1)}$.

Note (i) that all quantities $\xi_1, \eta_1, \eta_2, \eta_0$ and m relate to side j , but subscript j is omitted here, as not necessary.

(ii) If the singularity acts on a straight side we require the obvious replacements

$$\eta \Rightarrow x'_q; \quad \xi_1 = 0; \quad \eta_1 = 0; \quad \eta_2 = a_1; \quad \eta_0 = a,$$

the singularity being at the point $x'_q = a$.

SECTION 5 - DISCRETE ORTHOGONAL FUNCTIONS

If the function $y=f(x)$ is given as the set of n discrete points x_λ in the interval (a,b) , by

$$y_\lambda = f(x_\lambda) ; \lambda = 1, n \quad (97)$$

and $\{\phi_k(x)\}$ is a set of m discrete orthogonal functions, then we can approximate $f(x)$ by the polynomial

$$\bar{y} = \sum_{k=1}^m c_k \phi_k(x) \quad (98)$$

by minimising the sum of the squares, η^2 , of the weighted errors - weight factors w_λ - at the n data points x_λ :

$$\begin{aligned} \eta^2 &= \sum_{\lambda=1}^n w_\lambda (y_\lambda - \bar{y}_\lambda)^2 \\ &= \sum_{\lambda=1}^n w_\lambda [y_\lambda - \sum_{k=1}^m c_k \phi_k(x_\lambda)]^2 \end{aligned} \quad (99)$$

where \bar{y}_λ is the value of \bar{y} at x_λ .

On minimising η^2 with respect to the unknowns c_j ; $j=1, \dots, m$ by setting

$$\frac{\delta(\eta^2)}{\delta c_j} = 0 ; j = 1, \dots, m ; n \geq m ;$$

we obtain

$$\sum_{\lambda=1}^n w_\lambda [y_\lambda - \sum_{k=1}^m c_k \phi_k(x)] \phi_j(x_\lambda) = 0 \quad (100)$$

If the set $\{\phi_k(x)\}$ is orthogonal over the discrete points x_λ with weight factors w_λ , this means that

$$\sum_{\lambda=1}^n w_\lambda \phi_k(x_\lambda) \phi_j(x_\lambda) = N_j \delta_k^j ; \quad (101)$$

where N_j is the 'norm' of the function ϕ_j , and δ_k^j is the Kronecker delta. Accordingly, equation (A-5) gives on replacing subscript j by k :

$$c_k = \frac{1}{N_k} \sum_{\lambda=1}^n w_\lambda y_\lambda \phi_k(x_\lambda) \quad (102)$$

If the set $\phi_k(x)$ is complex, then $\phi_j(x)$ in (101) and (102) must be replaced by its complex conjugate $\phi_j^*(x)$ since N_j must be real giving

$$\sum_{\lambda=1}^n w_{\lambda} \phi_k(x_{\lambda}) \phi_j^*(x_{\lambda}) = N_k \delta_k^j$$

$$c_k = \frac{1}{N_k} \sum_{\lambda=1}^n w_{\lambda} y_{\lambda} \phi_k^*(x_{\lambda}), \quad (103)$$

on noting that the right hand side of (98) must be real which implies that $c_k \phi_k(x)$ must be real, and that η^2 is real.

Two limiting cases for n are of special interest:

(i) If $n=m$, then the obvious minimum condition for η^2 is that each of the errors $(y_{\lambda} - \bar{y}_{\lambda})$ should be zero, or the approximating curve \bar{y} should pass through the set of points $(x_{\lambda}, y_{\lambda})$, $\lambda=1, \dots, n$, and hence y and \bar{y} would be point-matched, or collocated, at the above set of points. If a unique solution exists it is necessary that $n \geq m$.

(ii) If $n \rightarrow \infty$ and $m \rightarrow \infty$, then the discrete set x_{λ} becomes the continuous x in the interval (a, b) , and (98) becomes an expansion in the infinite set of orthogonal functions $\{\phi_k(x)\}$. When $\phi_k(x)$ is complex, the coefficients follow from (103) as

$$c_k = \frac{1}{N_k} \int_a^b w(x) y(x) \phi_k^*(x) dx, \quad (104)$$

and the condition for orthogonality of the function $\phi_k(x)$ in (a, b) follows from (102) as

$$N_k \delta_k^j = \int_a^b w(x) \phi_k(x) \phi_j^*(x) dx, \quad (105)$$

where $w(x)$ is the associated weight factor in (a, b) .

Discrete Fourier Polynomials

If $\phi_k(x) = e^{ikx}$ in (98) it is easily shown that these functions satisfy the orthogonality relations

$$\int_{-\pi}^{\pi} e^{ikx} e^{-ijk} dx = 2\pi \delta_k^j; \quad i = \sqrt{-1}, \quad (106)$$

the quadrature expression for which is

$$\text{Lt}_{n \rightarrow \infty} \sum_{\lambda=-n}^n w_{\lambda} e^{ikx_{\lambda}} e^{-ijx_{\lambda}} = 2\pi \delta_k^j, \quad (107)$$

where the weight factors are w_λ , the origin for x being taken at the mid-point of the interval $(-\pi, \pi)$, which is divided for convenience into $2n$ equal parts.

Result (107) shows that the functions e^{ikx_λ} are orthogonal over the infinite set of discrete points x_λ in $(-\pi, \pi)$ with weight factors w_λ . We now ask can some w_λ and distribution of points x_λ be found, resulting from the application of some quadrature formula to (106), which will satisfy relation (107) for finite values of n and thereby provide a method of deducing a set of discrete orthogonal polynomials from a corresponding set of continuous orthogonal functions.

On investigating the trapezoidal formula, with its associated weight factors

$$\begin{aligned} w_\lambda &= \frac{1}{2} ; \quad \lambda = \pm n \\ &= 1 ; \quad |\lambda| < n \end{aligned} \quad (108)$$

and taking the equidistant distribution for x

$$x_\lambda = \frac{\lambda\pi}{n} ; \quad \lambda = -n(1)n,$$

we are led to examine the finite analogue, S , of series (107):

$$S = \sum_{\lambda=-n}^n w_\lambda e^{i(k-j)x_\lambda} = \sum_{\lambda=-n}^n w_\lambda e^{i\lambda\theta}, \quad (109)$$

where

$$\theta = \pi(k-j)/n.$$

On substituting into (109) for w_λ from (108) and applying the formula for summing $(2n+1)$ terms of a geometrical progression, it follows, when $k \neq j$, that

$$S = -\frac{1}{2}(e^{-in\theta} + e^{in\theta}) + e^{-in\theta} \frac{1 - e^{i(2n+1)\theta}}{1 - e^{i\theta}}, \quad (110)$$

where

$$e^{in\theta} = e^{i\pi(k-j)} = \cos(k-j)\pi = \pm 1 ; \quad k \neq j,$$

depending on whether $(k-j)$ is even or odd. Accordingly it follows that $S = 0$.

If $k = j$, then the series (109) reduces to

$$S = \sum_{\lambda=-n}^n w_\lambda = 2n, \quad (111)$$

and hence for all k and j

$$S = \sum_{\lambda=-n}^n w_\lambda e^{ikx_\lambda} e^{-ijx_\lambda} = 2n \delta_k^j, \quad (112)$$

which establishes condition (103) with $N_k = 2n$ for the functions e^{ikx_λ} when k is any integer, including zero. It follows from (103) that

$$c_k = \frac{1}{2n} \sum_{\lambda=-n}^n w_\lambda y_\lambda e^{-ikx_\lambda}$$

and if we take a correspondingly balanced polynomial form for (98) by writing

$$\bar{y} = \sum_{k=-m}^m c_k e^{ikx}, \quad (114)$$

it follows from (113) that

$$c_{-k} = c_k^*,$$

where c_k^* is the complex conjugate of c_k , and on writing

$$c_k = a_k + ib_k$$

series (114) reduces to the real series

$$\bar{y} = c_0 + 2 \sum_{k=1}^m a_k \cos kx - 2 \sum_{k=1}^m b_k \sin kx, \quad (115)$$

where on taking real and imaginary parts of result (113), we obtain

$$2a_k = \frac{1}{n} \sum_{\lambda=-n}^n w_\lambda y_\lambda \cos kx_\lambda$$

$$2b_k = -\frac{1}{n} \sum_{\lambda=-n}^n w_\lambda y_\lambda \sin kx_\lambda \quad (116)$$

On setting $x = x' - \pi$, we easily show that (115) and (116) are invariant in x' . Hence the above formulae hold for the x' interval $(0, 2\pi)$, and this can be mapped onto the interval $(0, c)$ by setting $x' = 2\pi x'/c$, replacing kx' by $2\pi kx''/c$ Or, on dropping the dashes, this means that (114) and (116) apply to the interval $(0, c)$ on replacing k by $2\pi k/c$.

Trigonometric Interpolation Series

Since boundary identity (39) can be approximate by the truncated series

$$\psi(x'_q) = \sum_{N=0}^L B_N \cos nx'_q + \sum_{N=1}^L C_N \sin nx'_q, \quad (117)$$

in the interval, $0 \leq x'_q \leq a_q$; $n = 2\pi N/a_q$, we might regard the trigonometric series on the r.h.s. as a trigonometric interpolation series for $\psi'(x'_q)$ in the specified interval. The fitting coefficients then follow by discrete least squares as in eq.(98), the coefficients being given by formulae (116).

We observe that the series (115) is similar to a Fourier series, truncated at m terms, for the interval $(-\pi, \pi)$ and that the summation formulae (116) for the Fourier coefficients $2a_k$ and $-2b_k$ are what would be obtained by evaluating the corresponding Fourier series integrals, analogous to these in (16), by the trapezoidal rule, involving the division of the interval $(-\pi, \pi)$ into $2n$ equal parts. Hence a working rule for the harmonic equations in sets (16) or (41):

Evaluate all Fourier integrals by the trapezoidal rule

This, in effect, substitutes series (116) for the corresponding integrals, as is required when harmonic matching is replaced by discrete least squares minimisation of the boundary residuals on each boundary segment.

SECTION 6 - DISTINCTIVE FEATURES

The distinctive features of the Edge-Function Method, as illustrated by the examples in Appendices A and B and the formulation in the present paper, are:

(a) Algebra:

The functions introduced in solution "mix" (8) may appear complicated at first sight, but an effective algorithmic method of controlling the resulting algebra - called the Computer Form Method - was developed by Quinlan [12,13]. Accordingly, Edge-Functions, Vertex Functions and any of their derived functions like normal slopes, moments or shears, can, on calling the appropriate subroutine, be obtained as readily as any trigonometric or exponential function.

(b) Programming:

A systems approach to programming, based on a main program Quinp together with 18 subroutines - each with its own definite task to perform - is given in [4]. This has been used to advantage by subsequent research workers, Tai, Nash, Dashmukh, O'Callaghan and others, to considerably simplify and shorten the programming tasks arising in extending the Edge-Function Method to problems of greater complexity, in vibrations and shallow shells. The program Quinp [4] and its accompanying program-description is still adequate, though it is proposed to issue shortly an up-dated version to include cuts, cracks and reentrant angles.

The program LAPGEN given in Appendix B(1) illustrates the chief features of QUINP. If one is familiar with LAPGEN, the various parts of QUINP should then be readily understandable.

(c) Computing-Time:

Considerable computing time is saved by using the discrete - rather than the continuous - least squares method of minimising the boundary residuals, as developed in Section 5, thus saving considerable time for setting up the coefficients matrix. The rows are arranged in the coefficients matrix so that the equations for each harmonic are grouped together in successive bands, thereby producing a strongly diagonalised system with a considerable saving in solution time. Moreover this arrangement of the matrix enables the solutions corresponding to a lesser number of harmonics to be deduced without any appreciable increase in computer time. Accordingly, as a routine, solution vectors corresponding to terminating the boundary identities at L-3, L-2, L-1 and L harmonics respectively are computed and tested for each problem.

(d) Acceptability of Solution:

Each solution vector provides an "exact" solution to a problem governed by the same differential equation but with slightly different boundary conditions to those specified. Such solutions may be regarded as mathematical models of the physical problem. In each case the difference - termed the Boundary Residuals - between the boundary values as computed and the specified values, is computed and reported through its approximate root mean square value - r.m.s. - on each side of the boundary and for each boundary condition. The set of r.m.s. values provides a simple yet comprehensive reliability test to enable an engineer to decide whether, or not, to accept the results.

(e) Convergence Demonstration:

A practical "convergence" demonstration is provided by the routine provision in each problem of several solutions, corresponding to increasing harmonics and matrix size, with r.m.s. values presented for the resulting boundary residuals. These invariably decrease rapidly as the number of harmonics used increased. Likewise production results (e.g. normal slopes, shears, harmonic conjugate, etc.) as required at interior points, are always computed for a number of different harmonics and hence as in examples given in appendix B their "convergence" can be seen at a glance. No other competing system - finite element, finite differences or boundary integral - can offer this effective comparison of the effect on the results of increased computer time expenditure, without involving a very considerable increase in computer time over that which would be required for a one-shot solution based on the largest matrix size involved.

(f) No pre-computer processing of the problem is required

As can be seen from Appendix C and from the data cards at the end of each example, no pre-computer processing is required. Only the geometrical and load data and material's moduli are required, together with one control card. These do not require any knowledge of E.F.M. for their preparation. Accordingly E.F.M. can be operated completely as a "Black Box". This is in sharp contrast to the intricate element networks required in finite element and boundary integral methods. Production is based on a single data card for each production set, requiring the computation of a specified function at a specified number of equidistant points on a specified line.

REFERENCES FOR THE EDGE-FUNCTION METHOD

1. Quinlan, P. M. (1964), "The Torsion of an Irregular Polygon", Proc. Roy. Soc., Vol. 282A.
2. Quinlan, P. M. (1968), "Polygonal and Swept-Back plates with Cut-Outs and Column Supports", OAR Research Applications Conference, Washington, 1968.
3. Kelly, G. V. and Quinlan, P. M. (1967), "The Method for Boundary Value Problems", OAR Research Report.
4. Quinlan, P. M. (1971), "Final Scientific Report on Grant AF EOAR-69-0049", With Computer Program QUINP, as an appendix.
5. O'Callaghan, M. J. (1972) "The Edge-Function Method for Linear Boundary Value Problems with Discontinuous Boundary Conditions", Ph.D. Thesis University College, Cork.
6. Quinlan, P. M. (1974), "The Edge-Function Method In Elastostatics" "Studies in Numerical Analysis", Festschrift to Professor Lanczos, Academic Press.
7. Quinlan, P. M. (1974), "The Edge-Function Method" 16th British Colloquium Theoretical Mechanics, St. Andrews.
8. Desmukh, R. S. (1973), "The Edge-Function Method Applied to Moderately Thick Plates and Shallow Shells", Ph.D. Thesis, University of Massachusetts.
9. Quinlan, P. M. (1975), "The Edge-Function Method for Cracks and Stress Concentrations in Two Dimensional Problems"; To be published in Int. J. Num. Math. Eng.
10. Tai, I. H. and Nash, W. A. (1973), "Vibrations of Thin-Plates - A New Approach", AFOSR TR-74-0789. University of Massachusetts.
11. O'Callaghan, M. J., Nash, W. A. and Quinlan, P. M. (1975) "Vibration Analysis of Thin Shallow Spherical Shells Using The Edge-Function Method"; AFOSR TR 72-2340.
12. Quinlan, P. M. (1965), "The λ -Method for Rectangular Plates, Proc. Roy. Soc., Vol. 288A.
13. Quinlan, P. M. (1965), "The λ -Method for Aelotropic Rectangular and Skew Plates", Proc. Roy. Irish Acad.
14. Sokolnikoff, I. S. "Mathematical Theory of Elasticity", McGraw-Hill, 1956.

APPENDIX A - LAPEX

Program LAPEX consists of a simple main program LAPEX with four subroutines EDGEF, QPOLAR, POLW and QSOLVE(NE). The program is described briefly at the end of Section 1 and is attached together with illustrative example (37) as Appendices A(1) and A(2).

The corresponding data cards are

(1) CONTROL CARD

L, Ns, Ndiv, Mdiv, Nprog

where

L = Truncation level

Ns = No sides

Ndiv = No. divisions used in integral evaluation

Mdiv = No. of check points on sides

Nprog = Example no.

(2) COORDINATES VERTICES OF POLYGON

x(j), y(j)

(3) COEFFICIENTS FOR BOUNDARY CONDITIONS (37) on sides

c(j,k) (k = 1,4) - for each side

(4) PRODUCTION

$U_1, V_1, U_2, V_2, Mdiv,$

for function u at Mdiv equidistant points on line (U_1, V_1) to (U_2, V_2) .

It is urged that the reader should become fully conversant with LAPEX, before proceeding to its fuller development in LAPGEN in Appendix B(2).

- (5) Two Minor Subroutines are included: QPOLAR, to determine polar coordinates, and GAUSS to determine division points and weight factors for both Harmonic Fitting and Gaussian Integration. The program LAPGEN follows with numerous explanatory captions, the references being to the main paper.

An illustrative example is appended, as Appendix B(2), to attached program LAPGEN, Appendix B(1). This deals with the torsion of a quadrilateral section with an elliptical cavity, Fig. 7, for four different truncation levels.

An effort has been made to make the output in Appendix B(2) self explanatory. The Boundary residuals are less than 1%, and the differences between computed quantities for levels $LL = 2$ and $LL = 4$ are seen to be considerably less.

Solutions of acceptable engineering accuracy are provided by truncation level $LL = 1$, involving only 53 equations with a time requirement on any computer of less than 150% of the time it would require to solve 53 linear equations using Gaussian elimination.

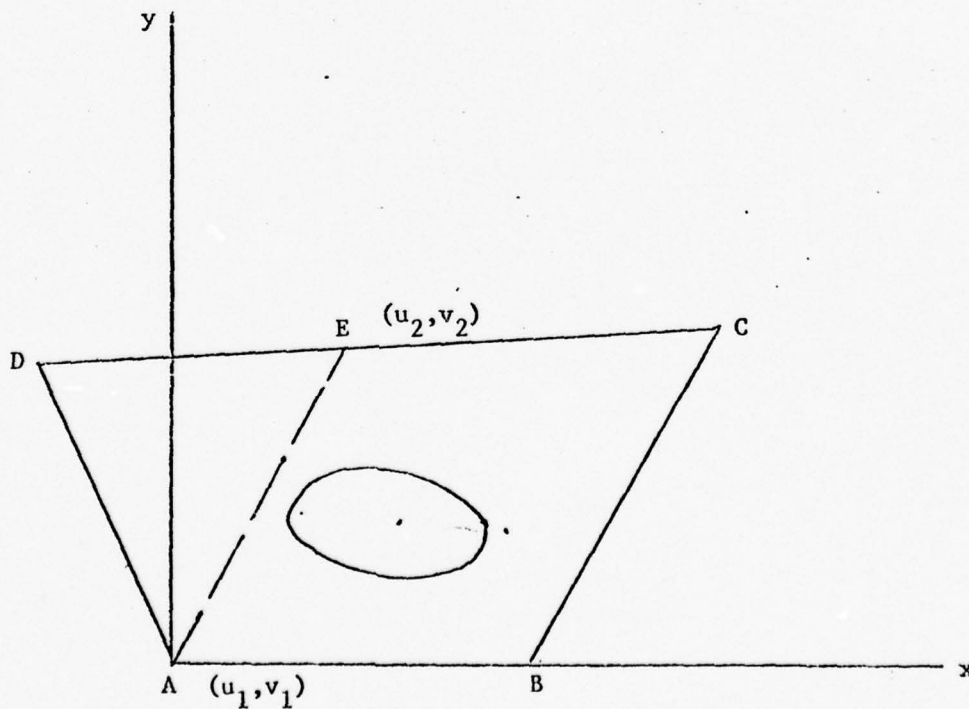


Figure 7.

APPENDIX B - LAPGEN

This program consists of main program LAPGEN and 10 subroutines consisting of

- (1) COLMAT Sets up data for points at vertex j in the vertex equations taking care to eliminate any redundant equations. It then assigns the functions for solution "mix" (8), and sets up the necessary data for their evaluation at any point. Assigned points and functions are then printed out as a useful aid in checking.
- (2) POINT FUNCTIONS for indicators $MT = 1, 2, 3$
 - (a) EDGEF Evaluates coeffs of unknowns E_{mj}^1 and E_{mj}^2 for Edge-Functions in eqs (55) and (56).
 - (b) POLW Evaluates Vertex Function V_{kj} in eq (50) and also Log-Vertex Functions V_{kj}^* , eq (60), where they replace V_{kj} .
 - (c) POLC Evaluates Harmonic Polars, eq (58), including the zero polar $\log r$.
 - (d) PMAP Evaluates curved Edge-Functions based on section 3 for mapping $z = c \cosh w$ as given by eq (89) including the special case of a solid ellipse.
 - (e) POLP Evaluates r.h.s. of matrix. Particular integrals not yet included. Provision is made for boundary conditions of types $NBDY = 1$ and $NBDY = 2$.
- (3) SOLCOR Solver routine for LS truncation levels, based on single pivoting for columns using the relevant equations.
- (4) PRODN Arranges for production, other than computation of results for $MT = 1, 2, 3$ as indicated by $Ncode = 1$. This is divided into two categories: $Ncode = 2$, and $Ncode = 3$ corresponding to differentiation and integration respectively as the main operations involved. The several cases in each category are designated by the indicator NOFN. Provision is made for torsional rigidity, shear stress, resultant stress and shear lines for the torsional problem developed in Sokolnikoff [14]. Other cases can be added as they arise, and thus a comprehensive program for Laplace/Poisson problems can be built up.

APPENDIX C - DATA INPUT FOR LAPGEN

The data sets required are

(1) CONTROL CARD

L, NS, NB, NPROG, NBDY, LS, NMAT, NPRIN, MDIV, DELTA, FSET, GMOD
where

L = Maximum no. harmonics - if set as percentage of optimum,
program will set corresponding L.

NS = No. boundary segments

NB = No. closed boundaries

NPROG = No. assigned to problem

NBDY \Rightarrow Type indicator for non-zero boundary conditions
= 1 indicates polynomial $\sum_{k=1}^6 C(j,k)t^{k-1}$

= 2 indicates torsion conditions as set in POLP.

LS = No. Comparative solutions required

NMAT \Rightarrow Matrix print out given if set NMAT = 1

NPRIN \Rightarrow Gives print out of residuals on boundary if set NPRIN = 1

MDIV \Rightarrow No. of checking points required on boundary

DELTA = Increment for use in numerical differentiation

FSET = Degree of smoothing required

GMOD = Torsional rigidity; if dealing with torsion problem
read in value for GMOD.

Note L, LS, MDIV, DELTA and FSET: if any of these are left blank, program will provide an appropriate value. Consequently only NS and NB must be specified.

(2) DATA FOR EACH SEGMENT

MB(J), NHS(J), NBY(J), NTYP(J), X(J), Y(J)

where

MB(J) = Function indicator MT as defined in Table 2

NHS(J) = Proportionality factor to regulate no. fitting points
per side, approximately proportional to length. If
left blank, will be set to one.

NBY(J) \Rightarrow Indicates if set to one that non-zero boundary conditions
of type NBDY = 1 occur on side j, and corresponding coeffs.
must be read in as data set (4) under.

NTYP(J) \Rightarrow Boundary curve type indicator for segment J where:
 = 0 indicates straight line
 = 1 indicates curved indentation
 = 2 indicates curved mound
 = 3 indicates closed curve

X(J),Y(J) \Rightarrow Coordinates of vertex J, where appropriate

(3) ADDITIONAL DATA CARD FOR EACH CURVED SEGMENT

AE(J), BE(J), XE(J), YE(J), ZE(J),

where for elliptic, or circular boundaries

Semi-axes are AE, BE; centre (XE,YE) and inclination of major axis ZE (as multiple of $\pi/2$).

(4) NON-ZERO BOUNDARY COEFFS. FOR NBDY = 1

C(J,K), (K = 1,6), for sides J with indicator NBY(J) = 1.

(5) PRODUCTION - one card per production set, unless when NCODE = 2 or NCODE = 3 a second card must follow to describe the function that is sought.

$U_1, V_1, U_2, V_2, B XK, MDIV, MT, J, NCODE, NOFN, NAXIS,$

where production is for

- (i) SEGMENT (U_1, V_1) to (U_2, V_2) or if J is specified it is for segment J of boundary
- (ii) B XK = angle associated with production (i.e. shear axis) as a multiple of $\pi/2$.
- (iii) MDIV = No. equidistant points at which results are required.
- (iv) NCODE \Rightarrow Indicates type of production required
 - = 1 : Functions MT = 1,2,3 as in Table 2
 - = 2 : Functions involving some differentiations
 - = 3 : Functions involving integration
- (v) NOFN \Rightarrow Indicates different functions available for each of the production types NCODE = 2 and NCODE = 3.

APPENDICES A(1) + A(2)

```

CJOB
C PROGRAM LAPEX FOR SIMPLE LAPLACE PROBLEM
C
1   COMMON X(11),Y(11),C(10,4),A(10),B(10),ANG(10),E(100),RMS(40),
    1G(100,101) ,SOL(100)
2   COMMON NFU(101),NA(101),L,NS,NDIV,MDIV,NPROC
3   DIMENSION HOLD(80,81)
4   DIMENSION DUM(10432),IJK(207)
5   EQUIVALENCE(DUM(1),X(1)),(IJK(1),NFU(1))
6   DO 1234 K=1,10432
7   1234 DUM(K)=0
8   DO 1235 K=1,207
9   1235 IJK(K)=0
C
C   SECTION 1. READ AND PRINT DATA
C
10  WRITE(6,799)
11  WRITE(6,7899)
12  7899 FORMAT(1H ,5X, ' PRINT OUT OF DATA CARDS FOR HEAT FLOW IN A QUAD
    1LATERAL ',/, ' TEMPERATURES GIVEN ON EDGES ')
13  PI=3.14159265
14  READ(5,8000) L,NS,NDIV,MDIV,NPROC
15  WRITE(6,5000) L,NS,NDIV,MDIV,NPROC
16  DO 1 J =1,NS
17  READ(5,8001) X(J),Y(J)
18  1 WRITE(6,5001) X(J),Y(J)
19  DO 2 J=1,NS
20  READ(5,8001)(C(J,K),K=1,4)
21  2 WRITE(6,5001)(C(J,K),K=1,4)
22  WRITE(6,799)
23  8000 FORMAT(20I3)
24  8001 FORMAT(7E11.4)
25  5000 FORMAT(14 ,20I3)
26  5001 FORMAT(1H ,7E11.4)
27  NE =L*NS +NS
28  NEE =NE+1
29  NEP =NS
30  X(NS+1) =X(1)
31  Y(NS+1) =Y(1)
C   -----IF NDIV & MDIV LEFT BLANK SET TO USUAL NOS.
32  IF(NDIV)4,3,4
33  3 NDIV=L+2
34  4 IF(MDIV)6,5,6
35  5 MDIV=2*L+3
36  6 CONTINUE
C
C   SECTION 2. SIDE LENGTHS A(J),SLOPES B(J) AND ANG(J)
C
37  DO 12 J =1,NS
38  XA=X(J+1)-X(J)
39  YA=Y(J+1)-Y(J)
40  CALL QPOLAR(XA,YA,RR,BA,PI)
41  B(J)=BA
42  12 A(J) = RR
43  DO 1550 J =1,NS
44  IF(J-1)15,13,14
45  13 ANG(1) =PI-B(1) +B(NS)
46  GO TO 15
47  14 ANG(J) =PI-B(J)+B(J-1)
48  15 IF(ANG(J)-2*PI)1550,1550,1540

```

```

49     1540 ANG(J)=ANG(J)-2*PI
50     1550 CONTINJE
      C
      C SECTION 3. ARRANGE FUNCTIONS
      C
51         NP = 0
52         DO 21 M =1,L
53         DO 21 J =1,NS
54         NP =NP +1
55         NA(NP)=J
56         NFN(NP) =1
57     21  E(NP) =M*PI/A(J)
58         DO 22 J=1,NS
59         NP =NS*L+J
60         NA(NP)=J
61         NFN(NP)=4
62     22  E(NP)=PI/ANG(J)
63         NE = NP
64         LP =NP +1
65         NFN(LP)=3
66         NA(LP) = 10
      C
      C SECTION 4. SET UP COLUMNS OF MATRIX G(I,J)
      C HARMONICS ON CYCLE I=1 AND PTS ON I=2
      C
67         DO 50 NP =1,LP
68         JE =NA(NP)
69         NFN =NFN(NP)
70         EM=E(NP)
71         XJE=X(JE)
72         YJE=Y(JE)
73         BJE=B(JE)
74         DO 50 I=1,2
75         DO 50 J =1,NS
76         KK =NDIV-I
77         BA=B(J)
78         IF(I-1)31,31,30
79     30  KK =1
80     31  DO 50 K =1,KK
81         DK =0
82         IF(I-1)33,32,33
83     32  DK =K-1
84         DK=DK/KK
85         XK=X(J)+DK*(X(J+1)-X(J))
86         YK=Y(J)+DK*(Y(J+1)-Y(J))
87         GO TO 34
88     33  XK=X(J)
89         YK=Y(J)
90     34  IF(NFN-3)35,36,37
91     35  CALL QEDSEF(XK,YK,XJE,YJE,BJE,EM,VAL)
92         GO TO 38
93     36  VAL =C(J,1)+C(J,2)*DK+C(J,3)*DK**2+C(J,4)*DK**3
94         GO TO 38
95     37  CALL QPDLW(XK,YK,XJE,YJE,BJE,EM,VAL,PI)
96     38  IF(I-1)40,40,39
97     39  JX =NS*L+J
98         G(JX,NP)=VAL
99         GO TO 50
100    40  DO 41 M =1,L
101         JR=(M-1)*NS+J
102    41  G(JR,NP)=G(JR,NP)+VAL*SIN(PI*M*DK)

```

```

103      50  CONTINUE
      C
      C      SECTION 5.  SOLVE MATRIX G(I,J)
      C      NEXT PROCEED TO CHECK RESIDUALS AT PTS ON SIDES AND
      C      SET THEIR R.M.S, RMS(J), WITH NCODE=0 SET NCODE=1 FOR PRODUCTION
      C      AT LINE 99
104      WRITE(6,799)
105      342  DO 348 JR=1,NE
106          DO 348 JC=1,NEE
107      348  HDLD(JR,JC)=G(JR,JC)
108          CALL QSOLVE(NE)
109      352  WRITE(6,5019)
110      5019  FORMAT(1H ,10X, ' PRINT OUT OF RESIDUALS FOR EQUATIONS IF NE<50 '
111          DO 350 JR=1,NE
112          XX=0.00E 00
113          DO 349 JC=1,NEE
114      349  XX=XX+HDLD(JR,JC)*SOL(JC)
115      350  WRITE(6,5007) JR,XX
116          WRITE(6,799)
117      353  NSJ=NS
118          NCODE=0
119          LPP=LP
120          MMDIV=MDIV-1
121      200  WRITE(6,799)
122          DO 80 J=1,NSJ
123          YY=0
124          DO 70 K=1,MDIV
125          XX=0
126          DK=K-1
127          DK=DK/MMDIV
128          IF(NCODE)201,201,202
129      202  XK=J1 +DK*(U2-U1)
130          YK=V1+DK*(V2-V1)
131          GO TO 203
132      201  XK=X(J)+DK*(X(J+1)-X(J))
133          YK=Y(J)+DK*(Y(J+1)-Y(J))
134      203  DO 68 NP=1,LPP
135          JE=NA(NP)
136          NFN=NFU(NP)
137          EM=E(NP)
138          XJE=X(JE)
139          YJE=Y(JE)
140          BJE=B(JE)
141          IF(NFN-3)65,66,67
142      65  CALL QEDGEF(XK,YK,XJE,YJE,BJE,EM,VAL)
143          GO TO 68
144      66  VAL=C(J,1)+C(J,2)*DK+C(J,3)*DK**2+C(J,4)*DK**3
145          GO TO 68
146      67  CALL QPOLW(XK,YK,XJE,YJE,BJE,EM,VAL,PI)
      C
      C      CUMULATE XX FOR KTH POINT ON SIDE J
      C
147      68  XX=XX+VAL*SJL(NP)
148          G(J,K)=XX
149      70  YY=YY+XX**2
150      80  RMS(J)=YY
151          IF(NCODE) 204,204,205
152      205  WRITE(6,5009)MDIV
153      5009  FORMAT(1H ,4X, ' TEMPERATURES AT M= ',I3, ' PTS ON LINE U1,V1 ')
154          GO TO 206

```

```

155 204 WRITE(6,5003)
156 5003 FORMAT(1H ,4X,' ROOT MEAN SQR RESIDUALS ON SIDE J ')
157 WRITE(6,5004)(J,J=1,NS)
158 5004 FORMAT(1H ,3X,8(10X,I3))
159 WRITE(6,5005)(RMS(J),J=1,NS)
160 5005 FORMAT(1H ,4X,8(2X,E11.4))
161 799 FDRMAT(1H ,6X,' *****')
162 WRITE(6,5006)MDIV
163 5006 FORMAT(1H ,6X,' PRINT OF RESID AT MDIV=',I3,' PTS ON SIDE J ',/
1 (NOTE: EVERY SECOND VALUE SHOULD BE ZERO IF MDIV LEFT BLANK) '
164 WRITE(6,5004)(J,J=1,NS)
165 206 DO 90 K=1,MDIV
166 90 WRITE(6,5007)K,(G(J,K),J=1,NSJ)
167 5007 FORMAT(1H ,14,2X,7(2X,E11.4))
168 READ(5,8008)U1,V1,U2,V2,MDIV
169 WRITE(6,799)
170 IF(MDIV)100,100,99
171 99 WRITE(6,5011)
172 5011 FORMAT(1H ,10X,' PRODUCTION AT POINTS ON GIVEN LINE ',/)
173 WRITE(6,5010)
174 5010 FORMAT(1H ,10X,'PRINT OUT OF DATA CARD U1,V1,U2,V2,M ',/)
175 WRITE(6,5008)U1,V1,U2,V2,MDIV
176 NCODE=1
177 LPP=LP-1
178 MMDIV=MDIV-1
179 NSJ=1
180 GO TO 200
181 8008 FORMAT(4E11.4,I3)
182 5008 FORMAT(14 ,13X,4E11.4,I3)
183 100 CALL EXIT
184 STOP
185 END

186 SUBROUTINE QPOLAR(XA,YA,RR,BA,PI)
187 RR=SQRT(XA**2+YA**2)
188 IF(ABS(XA)-0.000001)7,8,8
189 7 XA=XA+0.000001
190 8 BA=ATAN(YA/XA)
191 IF(XA)9,9,10
192 9 BA=BA+PI
193 GO TO 12
194 10 IF(YA)11,12,12
195 11 BA=BA+2*PI
196 12 RETURN
197 END

198 SUBROUTINE QEDGEF(XK,YK,XJE,YJE,BJE,EM,VAL)
199 XXX=(XK-XJE)*COS(BJE)+(YK-YJE)*SIN(BJE)
200 YYK=-(XK-XJE)*SIN(BJE)+(YK-YJE)*COS(BJE)
201 VAL=EXP(-EM*YYK)*SIN(EM*XXX)
202 RETURN
203 END

204 SUBROUTINE QPOLW(XK,YK,XJE,YJE,BJE,EM,VAL,PI)
205 XA=XK-XJE
206 YA=YK-YJE
207 CALL QPOLAR(XA,YA,RR,BA,PI)
208 BA=BA-3JE
209 C -----ENSURE ANGLE BA>0
IF(BA-0.1E-06)28,29,29

```

```

210      28  BA=BA+2*PI
211      29  VAL=RR*EM*SIN(EM*BA)
212      RETURN
213      END

214      SUBROUTINE QSOLVE(NE)
215      COMMON X(11),Y(11),C(10,4),A(10),B(10),ANG(10),E(100),RMS(40),
IG(100,101),SOL(100)
216      COMMON NFU(101),NA(101),L,NS,NDIV,MDIV,NPROG
217      LP=NE+1
218      DO 60 JR=1,NE
219      XX=G(JR,JR)
220      XX=1/XX
221      DO 52 JC=1,LP
222      52  G(JR,JC)=G(JR,JC)*XX
223      DO 60 JRR=1,NE
224      IF(JRR-JR)53,60,53
225      53  XG=G(JRR,JR)
226      DO 54 JCC=1,LP
227      54  G(JRR,JCC)=G(JRR,JCC)-XG*G(JR,JCC)
228      60  CONTINUE
229      DO 61 JR=1,NE
230      61  SOL(JR)=G(JR,LP)
231      WRITE(6,5002)NPROG
232      5002 FORMAT(1H,6X,' SOLUTION VECTOR NPROG =',I3)
233      DO 62 K=1,NE
234      62  WRITE(6,5001)SOL(K)
235      WRITE(6,799)
236      5001 FORMAT(1H,7E11.4)
237      799  FORMAT(1H,6X,' ***** ' )
238      SOL(LP)=-1
239      RETURN
240      END

```

CENTRY

PRINT OUT OF DATA CARDS FOR HEAT FLOW IN A QUADRILATERAL
TEMPERATURES GIVEN ON EDGES

10 4 0 0 12

```

0.0000E 00 0.0000E 00
0.1000E 01 0.0000E 00
0.1200E 01 0.1000E 01
-0.2000E 00 0.9000E 00
0.0000E 00 0.1000E 01-0.1000E 01 0.0000E 00
0.0000E 00 0.0000E 00 0.0000E 00 0.0000E 00
0.0000E 00 0.0000E 00 0.0000E 00 0.0000E 00
0.0000E 00 0.0000E 00 0.0000E 00 0.0000E 00

```

SOLUTION VECTOR NPROG = 12

```

0.2662E 00
0.2023E-02
-0.3286E-01
0.1710E-01
0.3065E-02
0.4452E-02
-0.1160E-02
-0.6627E-02
0.1156E-01
-0.1502E-02

```

-0.8570E-03
0.2122E-02
0.1344E-02
0.6985E-03
-0.1084E-02
-0.1299E-02
0.2991E-02
-0.1387E-02
-0.6895E-03
0.5526E-03
0.7245E-03
0.1698E-03
-0.6242E-03
-0.5971E-03
0.1198E-02
-0.8892E-03
-0.4160E-03
0.5790E-05
0.3750E-03
0.4651E-04
-0.3333E-03
-0.4517E-03
0.4704E-03
-0.4280E-03
-0.1954E-03
-0.1480E-03
0.1171E-03
0.9718E-05
-0.1055E-03
-0.1898E-03
0.6103E-02
0.5069E-02
-0.1729E-02
0.3992E-02

PRINT OUT OF RESIDUALS FOR EQUATIONS IF NE<50

1	-0.1717E-04
2	0.1531E-05
3	0.2519E-05
4	0.1665E-05
5	-0.3651E-06
6	-0.2272E-06
7	-0.5902E-07
8	0.2161E-06
9	-0.8605E-06
10	0.2049E-06
11	0.6138E-06
12	-0.3725E-07
13	-0.6706E-07
14	-0.1341E-06
15	0.2581E-06
16	0.1267E-06
17	0.0000E 00
18	0.1155E-06
19	0.1607E-07
20	0.1502E-06
21	-0.1118E-06
22	0.1602E-06
23	-0.1075E-06
24	0.1509E-06

25 -0.4680E-07
 26 0.9313E-07
 27 0.4155E-07
 28 0.6799E-07
 29 0.2095E-08
 30 -0.3725E-08
 31 -0.2467E-07
 32 0.4680E-07
 33 -0.5355E-08
 34 0.2678E-07
 35 0.2968E-07
 36 0.2445E-07
 37 0.1141E-07
 38 -0.4889E-08
 39 -0.1483E-07
 40 0.4784E-08
 41 0.4098E-07
 42 -0.2570E-06
 43 0.1562E-06
 44 0.1215E-06

ROOT MEAN SQR RESIDUALS ON SIDE J

	1	2	3	4
	0.1407E-04	0.1977E-03	0.9401E-05	0.1761E-05

PRINT OF RESID AT MDIV= 23 PTS ON SIDE J

(NOTE: EVERY SECOND VALUE SHOULD BE ZERO IF MDIV LEFT BLANK)

	1	2	3	4
1	0.4098E-07	-0.2570E-06	0.1562E-06	0.1215E-06
2	-0.3601E-02	0.3486E-02	0.3017E-02	0.8766E-03
3	-0.1371E-05	0.1006E-06	0.2033E-06	0.1766E-06
4	0.7303E-03	-0.5569E-03	-0.4892E-03	0.1432E-03
5	-0.1609E-05	0.4470E-07	0.3243E-06	0.1890E-06
6	-0.3209E-03	0.2344E-03	0.1939E-03	0.7926E-03
7	-0.1848E-05	0.4470E-07	0.3160E-06	0.2414E-06
8	0.1851E-03	-0.1448E-03	-0.1045E-03	-0.2162E-03
9	-0.1788E-05	0.1416E-06	0.2359E-06	0.3180E-06
10	-0.1335E-03	0.1127E-03	0.6717E-04	0.1083E-03
11	-0.2205E-05	0.1676E-06	0.2327E-06	0.3616E-06
12	0.1041E-03	-0.1047E-03	-0.4717E-04	-0.6162E-04
13	-0.2205E-05	0.9686E-07	0.3078E-06	0.3465E-06
14	-0.9924E-04	0.1147E-03	0.3617E-04	0.3357E-04
15	-0.2265E-05	0.1192E-06	0.3774E-06	0.2384E-06
16	0.1009E-03	-0.6323E-02	-0.2742E-04	-0.6463E-05
17	-0.1907E-05	0.1974E-06	0.3782E-06	0.1974E-06
18	-0.1324E-03	-0.6755E-02	0.2074E-04	-0.3259E-04
19	-0.1967E-05	0.1914E-06	0.2898E-06	0.1267E-06
20	0.2153E-03	-0.8380E-02	-0.7443E-05	0.1266E-03
21	-0.1311E-05	0.1399E-06	0.1927E-06	0.4470E-07
22	-0.5616E-03	-0.5416E-02	-0.5271E-04	-0.5134E-03
23	-0.2570E-06	0.1562E-06	0.1404E-06	0.4098E-07

PRODUCTION AT POINTS ON GIVEN LINE

PRINT OUT OF DATA CARD U1,V1,U2,V2,M

0.0000E 00 0.0000E 00 0.1000E 01 0.5000E 00 10

TEMPERATURES AT M= 10 PTS ON LINE U1,V1

1 0.4098E-07

2 0.8386E-01
3 0.1264E 00
4 0.1408E 00
5 0.1353E 00
6 0.1165E 00
7 0.9024E-01
8 0.6177E-01
9 0.3552E-01
10 0.1452E-01

CORE USAGE OBJECT CODE= 11832 BYTES, ARRAY AREA= 68475 BYTES, TOTAL AREA
DIAGNOSTICS NUMBER OF ERRORS= 0, NUMBER OF WARNINGS= 0, NUMBE
COMPILE TIME= 0.00 SEC, EXECUTION TIME= 0.00 SEC, WATFIV - JUL 1973 V1L4

APPENDICES B(1)+B(2)

```

CJOB
C *****
C LAPSEN PROGRAM FOR GENERAL LAPLACE PROBLEMS APPENDIX B(1)
C *****
1 COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,10)
CG(99,100),SOL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(50)
C,XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),GMOD,DFIX(50),TUR(4)
2 COMMON BFIX(50),ZE(11),F(4),COEF(2,100),PI,DELTA,XCAR(40),D(50)
3 COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPROG,NE,LS
4 COMMON NROW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM
5 COMMON VJK(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50)
6 COMMON NFP,NZERO,NSS,NB,NOEL(10),LOC(50),NBDY,NFPOL
7 DIMENSION HOLD(40,41)
8 DIMENSION PADDY(11492),IJK(470)
9 EQUIVALENCE (PADDY(1),X(1)),(IJK(1),NFU(1))
10 DO 1234 I=1,11492
11 PADDY(1)=0.0E+00
12 1234 CONTINUE
13 DO 1235 I = 1,470
14 1235 IJK(I)=0
C
C SECTION 1. READ IN AND PRINT OUT DATA.
C
15 WRITE(6,799)
16 WRITE(6,799)
C ----- CONTROL CARD
17 READ(5,8000)L,NS,NB,NPROG,NBDY,LS,NMAT,NPRIN,MDIV,DELTA,FSET,GMC
18 WRITE(6,7990)NPROG
19 WRITE(6,799)
20 READ(5,7991)(XCAR(I),I=1,40)
21 WRITE(6,4991)(XCAR(I),I=1,40)
22 WRITE(6,799)
23 WRITE(6,5101)
24 WRITE(6,5000)L,NS,NB,NPROG,NBDY,LS,NMAT,NPRIN,MDIV,DELTA,FSET,GMC
25 PI=3.14159265
26 WRITE(6,799)
27 WRITE(6,4990)NS
28 WRITE(6,5002)
29 NCARDS=3
30 DO 1 J = 1,NS
C ----- SEGMENT DATA
31 READ(5,7999)MB(J),NHS(J),NBY(J),NTYP(J),X(J),Y(J)
32 NCARDS=NCARDS+1
C ----- NCARD TALLIES THE DATA CARDS AS READ IN.
33 1 WRITE(6,4999)J,MB(J),NHS(J),NBY(J),NTYP(J),X(J),Y(J)
34 DO 878 J=1,NS
C ----- IF NHS(J) LEFT BLANK SET TO 1
35 IF(NHS(J))878,877,878
36 877 NHS(J) = 1
37 878 CONTINUE
38 879 WRITE(6,799)
39 INDIC=0
40 DO 1233 J=1,NS
41 IF(NTYP(J))1233,1233,1230
C ----- ADDITIONAL DATA FOR CURVED SIDES
42 1230 READ(5,7998)AE(J),BE(J),XE(J),YE(J),ZE(J)
43 NCARDS=NCARDS+1
44 IF(INDIC)1232,1232,1231
45 1232 WRITE(6,799)
46 WRITE(6,4992)

```

```

47         INDIC=1
48         WRITE(5,5103)
49     1231 WRITE(6,4998)AE(J),BE(J),XE(J),YE(J),ZE(J)
50         ZE(J)=0.5*ZE(J)
51     1233 CONTINUE
C         ----- IF DELTA & FSET & NB LEFT BLANK SET TO USUAL VALUES
52         IF(DELTA)1237,1236,1237
53     1236 DELTA = 0.1000E-03
54     1237 IF(FSET)1239,1238,1239
55     1238 FSET = 1.0000
56     1239 IF(NB)1240,1240,1241
57     1240 NB = 1
58     1241 INDIC=0
59     1244 IF(NBDY-2)1245,1244,1244
60     1244 DO 1246 J=1,NS
61     1246 NBY(J)=1
62         GO TO 1247
63     1245 DO 2 J=1,NS
64         IF(NBY(J))2,2,3
C         ----- READ COEFFS. C(J,K), (K=1,6) FOR NBDY=1
65     3 READ(5,8001)(C(J,K),K=1,6)
66         NCARDS=NCARDS+1
67         IF(INDIC)1243,1242,1243
68     1242 INDIC=1
69         WRITE(5,799)
70         WRITE(6,4993)
71     1243 WRITE(6,5001)J,(C(J,K),K=1,6)
72     2 CONTINUE
73         WRITE(5,799)
C         ----- NSS = NO. SHARP CORNERS ON BOUNDARY
C         NOK(J) = NO. INTEGRATION PTS.
C         NRDW(J) = WHERE HARMS. FOR SEGMENT J BEGIN IN EACH SET
C         NVER(J) = VERTEX NO. FOR VERTEX J
74     1247 NSS=NS
75         IF(NTYP(NS)-3)5,4,4
76     4 NSS=NS-1
77         IF(NS-2)5,10,5
78     10 NSS=0
79     5 NRDW(1)=1
80         NH=0
81         DO 9 J=1,NS
82         NOK(J)=2*L*NHS(J)*FSET+3-NTYP(J)/3
83         NVER(J)=0
84         NRDW(J+1)=NRDW(J)+2*NHS(J)
85         NH=NH+2*NHS(J)
86         IF(NTYP(J)-3)6,9,9
87     6 JJ=J-1
88         IF(J-1)7,7,8
89     7 JJ=NSS
90     8 NVER(J)=2*(MB(J)-1)+MB(JJ)
91     9 CONTINUE
C
C         SECTION 2.           SIDE LENGTHS,A(J), SLOPES,B(1,J) & B(2,J)
C                               VERTEX ANGLES,ANG(J),FOR POLYGON.
C                               ENSURE ANG(J) IN INTERVAL (0,2*PI)
92         IF(NSS)16,16,11
93     11 X(NSS+1)=X(1)
94         Y(NSS+1)=Y(1)
95     DO 12 J = 1,NS
96         IF(NTYP(J))1510,1510,1520

```

```

97      1520 CALL ELPS(J,0,0)
98      GO TO 12
99      1510 XA=X(J+1)-X(J)
100     YA=Y(J+1)-Y(J)
101     CALL QPOLAR(XA,YA,RR,BA,PI)
102     B(1,J)=BA
103     B(2,J)=BA
104     A(J)=RR
105     12 CONTINUE
106     DO 1560 J=1,NSS
107     IF(J-1)15,13,14
108     13 ANG(1)=PI-B(1,1)+B(2,NSS)
109     GO TO 15
110     14 ANG(J)=PI-B(1,J)+B(2,J-1)
111     15 IF(ANG(J)-2*PI)1550,1550,1540
112     1540 ANG(J)=ANG(J)-2*PI
113     1550 IF(ANG(J))1555,1560,1560
114     1555 ANG(J)=ANG(J)+2*PI
115     1560 CONTINUE
C      ----- ADJUST L & LS TO APPROPRIATE VALUE IF NECESSARY
116     IF(L-25)18,17,17
117     17 LMAX=(100-NFP)/NS
118     L=(LMAX*L)/100
119     18 IF(L-2)19,20,20
120     19 L=2
121     20 IF(L-LS)21,2500,2500
122     21 LS=L
123     2500 IF(LS)22,22,23
124     22 LS=2
125     GO TO 20
126     23 IF(LS-4)2501,2501,24
127     24 LS=4
128     2501 CONTINUE
C      ----- CALL SUBROUTINE COLMAT TO SET UP FIXED POINTS
C      AND ASSIGN THE CORRESPONDING COLUMNS OF MATRIX
129     16 CALL COLMAT
C
C      SECTION 3.  SET UP COLUMNS NP OF MATRIX; POINTS
C      ON CYCLE I=2 AND HARMONICS ON I=1.
C      EVALUATE INTEGRALS BY TRAPEZOIDAL RULE WITH NOK(J) PTS. PER SID
130     NCODE=0
131     JCC=0
132     NPP=(NH*L)/2
133     DO 51 NP=1,LP
134     NCOLS=1
135     IF(NP-NPP)25,25,26
136     25 NCOLS=2
C      ----- COLUMN DATA AS SET UP IN COLMAT
137     26 JE=NA(NP)
138     NFN=NFJ(NP)
139     EM=E(NP)
140     XJE=X(JE)
141     YJE=Y(JE)
142     BJE=B(1,JE)
143     JRD=NH*L
144     JRP=JRD+NZERO
145     DO 501 I=1,2
146     JJ=NFP
147     NVEL=0
148     IF(I-1)28,27,28

```

```

149      27 JJ=NS
150      28 DO 500 J=1, JJ
151          IF(I-1) 31, 30, 31
152      30 KK=NDK(J)
153          CALL GAUSS(KK, NCODE)
154          MT=M3(J)
155          JP=J
156          GO TO 329
157      31 KK=1
158          M=LDC(J)
159          MT=MRFIX(M)
160          IF(MT-10) 329, 328, 328
161      328 KK=2
162          MT=MT-10
163      329 DO 50 <=1, KK
164          IF(I-2) 32, 33, 33
165      C      ----- INTEGRATION POINTS AND WEIGHT FACTORS
166      32 DK=D(K)
167          WK=W(K)
168          JNB=NTYP(J)
169      C
170      C      -----DATA FOR POINTS ON ELLIPTIC CURVE NO. NNEL
171      330 IF(JN3) 330, 330, 331
172      331 IF(<-1) 332, 332, 333
173      332 IF(NP-1) 333, 334, 333
174      334 NNEL=NNEL+1
175          NOEL(J)=NNEL
176          CALL ELPS(J, NNEL, KK)
177          NEL=NOEL(J)
178          XK=XEL(NEL, K)
179          YK=YEL(NEL, K)
180          BXK=BEL(NEL, K)
181          GO TO 34
182      C
183      C      -----DATA FOR POINTS ON STRAIGHT LINE SEGMENTS.
184      330 XK=X(J)+DK*(X(J+1)-X(J))
185          YK=Y(J)+DK*(Y(J+1)-Y(J))
186          BXK=B(1, J)
187          GO TO 34
188      C
189      C      -----DATA FOR FIXED POINTS AS ASSIGNED IN COLMAT.
190      33 4=LDC(J)+K-1
191          XK=XFIX(M)
192          YK=YFIX(M)
193          JP=JFIX(M)
194          BXK=BFIX(M)
195          DK=DFIX(M)
196      C      ----- CALL SUBROUTINES FOR POINT VALUES
197      34 GO TO (35, 35, 36, 37, 370, 370), NFN
198      35 CALL EDGEF(XK, YK, BXK, XJE, YJE, BJE, EM, PI, MT, VA, VB)
199          GO TO 38
200      36 CALL PDLF(XK, YK, BXK, DK, MT, NCODE, JP, VA)
201          NCOLS=1
202          GO TO 38
203      37 CA=CDEF(1, NP)
204          CB=CDEF(2, NP)
205          CALL PDLW(XK, YK, BXK, XJE, YJE, BJE, EM, CA, CB, MT, VA, PI, DELTA)
206          GO TO 38
207      370 XJE=XE(JE)
208          YJE=YE(JE)
209          BJE=ZE(JE)

```

```

202          IF(NFN-6)371,372,372
203      372 CALL PMAP(XK,YK,BXK,XJE,YJE,BJE,EM,MT,JE,VA,VB)
204          GO TO 38
205      371 CALL PDLC(XK,YK,BXK,XJE,YJE,BJE,EM,MT,VA,VB,PI)
206      38 F(1)=VA
207          F(2)=V3
C          ----- COMPUTE MATRIX ENTRIES AND INSERT IN G(JR,JX)
208          IF(I-1)41,41,39
C          ----- POINT OR 'TWO-POINT' EQS.
209      39 JR=JRP+J
210          DO 40 JC=1,NCOLS
211          JX=JCC+JC
212      40 G(JR,JX)=G(JR,JX)-F(JC)*((-1)**K)
213          GO TO 50
214      41 JR=JRD+J
C          ----- ZERO HARMONIC EQS.
215          DO 42 JC=1,NCOLS
216          JX=JCC+JC
217      42 G(JR,JX)=G(JR,JX)+WK*F(JC)
218          NN=0
219          MM=NHS(J)
C          ----- COSINE AND SINE HARMONIC EQS.
220          DO 44 MS=1,L
221          DO 44 M=1,MM
222          NN=NN+1
223          DO 44 NEG=1,2
224          BETA=(NEG-1)*PI*0.5E+00
225          JR=(MS-1)*NH+NROW(J)+(2*(M-1))+NEG-1
226          CC=COS(2*PI*DK*NN+BETA)
227          DO 44 JC=1,NCOLS
228          JX=JCC+JC
229      44 G(JR,JX)=G(JR,JX)+WK*CC*F(JC)
230          50 CONTINUE
231          500 CONTINUE
232          501 CONTINUE
233          51 JCC=JCC+NCOLS
234          NEE=NE+1
C          ----- PRINT OUT OF COEFFS. MATRIX IF NMAT SET TO 1
235          IF(NMAT)342,342,341
236      341 WRITE(6,799)
237          WRITE(6,5018)
238          DO 347 JC=1,NEE
239          WRITE(6,5016)JC
240          WRITE(6,5017)(G(JR,JC),JR=1,NE)
241      347 CONTINUE
242          WRITE(6,799)
243      342 IF(NE-40)343,343,351
244      343 DO 348 JR=1,NE
245          DO 348 JC=1,NEE
246      348 HOLD(JR,JC)=G(JR,JC)
247          WRITE(6,799)
C          ----- CALL SUBROUTINE TO SOLVE MATRIX FOR LS SOLUTIONS
C          AND CHECK RESULTS IF NE<41
248      351 CALL SOLCOR
249          IF(NE-40)352,352,353
250      352 WRITE(6,799)
251          WRITE(6,5019)LS
252          WRITE(6,5004)(LL,LL=1,LS)
253          DO 3500 JR=1,NE
254          DO 350 LL=1,LS

```

```

255          F(LL)=0.00E+00
256          DO 349 JC=1,NEE
257          349 F(LL)=F(LL)+HOLD(JR,JC)*SOL(LL,JC)
258          350 CONTINUE
259          3500 WRITE(6,5007)JR,(F(LL),LL=1,LS)
260          WRITE(6,799)
C
C          ----- SECTION 4
C          NEXT PROCEED TO CHECK RESIDUALS AT POINTS K ON SIDES J,(J=1,NS)
C          AND GET THEIR R.M.S.,RMS(J),WITH NCODE=0 FOR LEVELS OF SOLUTION
C          LL=1,LS. PROGRAM SIMILAR TO COEFFS. MATRIX PROG. LABELS 25-40.
261          WRITE(6,799)
262          353 NSJ=NS
263          NSA=1
264          NSIDE=1
C          ----- INDICATORS INTRODUCED SO THAT CHECK PART OF PROGRAM CAN BE
C          USED AGAIN FOR PRODUCTION--SEE LABEL 2360.
265          NCY=1
266          NNHOLD=0
267          200 WRITE(6,799)
268          NNEL=0
269          DO 80 NN=1,NCY
270          DO 80 J=NSA,NSJ
271          NOEL(J)=0
C          ----- IF MDIV LEFT BLANK SET TO SUITABLE NO. TO CHECK AT
C          'FIXED' POINTS AND MID-POINTS INBETWEEN
272          IF(MDIV)354,354,355
273          354 MDIV=2*NDK(J)-1
274          DO 356 LL=1,LS
275          356 RMS(LL,J)=0
276          CALL GAUSS(MDIV,NCODE)
277          JNB=NTYP(J)
278          DO 221 K=1,MDIV
279          G(1,K)=0
280          G(2,K)=0
281          G(3,K)=0
282          G(4,K)=0
283          DK=D(K)
284          IF(NSIDE)202,202,201
285          202 XK=U1+DK*(U2-U1)
286          YK=V1+DK*(V2-V1)
287          GO TO 210
288          201 IF(JNB)209,209,199
289          199 IF(K-1)203,203,206
290          203 NVEL=NVEL+1
291          NOEL(J)=NVEL
292          CALL ELPS(J,NVEL,MDIV)
293          206 NEL=NOEL(J)
294          XK=XEL(NEL,K)
295          YK=YEL(NEL,K)
296          BXK=BEL(NEL,K)
297          GO TO 208
298          209 XK=X(J)+DK*(X(J+1)-X(J))
299          YK=Y(J)+DK*(Y(J+1)-Y(J))
300          BXK=B(1,J)
301          208 IF(NCODE)204,204,210
302          204 MT=M3(J)
303          210 JCC=0
304          IF(NCY-2)361,360,361
305          360 BXK=BHOLD(NN)

```

```

306      361 XFIX(K)=XK
307      YFIX(K)=YK
308      BFIX(K)=3XK
309      DO 219 NP=1,LP
310      JE=VA(NP)
311      NFN=NFJ(NP)
312      NCOLS=1
313      IF(NP-NPP)211,211,212
314      211 NCOLS=2
315      212 EM=E(NP)
316      XJE=X(JE)
317      YJE=Y(JE)
318      BJE=B(1,JE)
319      GO TO (213,213,214,215,216,216),NFN
320      213 CALL EDGEF(XK,YK,BXK,XJE,YJE,BJE,EM,PI,MT,VA,VB)
321      GO TO 217
322      214 CALL PDLP(XK,YK,BXK,DK,MT,NCODE,J,VA)
323      NCOLS=1
324      GO TO 217
325      215 CA=CDEF(1,NP)
326      CB=CDEF(2,NP)
327      CALL PDLW(XK,YK,BXK,XJE,YJE,BJE,EM,CA,CB,MT,VA,PI,DELTA)
328      GO TO 217
329      216 XJE=XE(JE)
330      YJE=YE(JE)
331      BJE=ZE(JE)
332      IF(NFN-6)2216,2217,2217
333      2217 CALL PMAP(XK,YK,BXK,XJE,YJE,BJE,EM,MT,JE,VA,VB)
334      GO TO 217
335      2216 CALL POLC(XK,YK,BXK,XJE,YJE,BJE,EM,MT,VA,VB,PI)
336      217 F(1)=VA
337      F(2)=VB
C      ----- COMPUTE COEFFS. IN COL. JX FOR POINT K, WITH CORRESPONDING
C      SOLUTION SOL(LL,LX) AND FIND CORRESPONDING R.M.S. FOR EACH
C      SIDE J AND TRUNCATION LEVEL LL.
338      DO 218 JC=1,NCOLS
339      JX=JCC+JC
340      DO 218 LL=1,LS
341      218 G(LL,K)=G(LL,K)+F(JC)*SOL(LL,JX)
342      219 JCC=JCC+NCOLS
343      IF(NCODE)2219,2219,221
344      2219 DO 220 LL=1,LS
345      220 RMS(LL,J)=RMS(LL,J)+G(LL,K)**2
346      221 CONTINUE
C
C      SECTION 5. PRINT OUT AND PRODUCTION ARRANGEMENTS
C      ----- IF NCODE>1 FURTHER PROCESSING AND PRINT OUT DONE IN
C      SUBROUTINE PRODN.
347      IF(NCODE-1)2222,235,2223
348      2223 CALL PRODN(J,NSJ,NCODE,NOFN,NNHOLD,NHOLD)
349      GO TO 80
350      2222 DO 222 LL=1,LS
351      222 RMS(LL,J)=SQRT(RMS(LL,J)/MDIV)
352      GO TO 234
353      235 WRITE(6,5009)MT,MDIV
354      GO TO 236
355      234 IF(NPRIN)80,80,2224
356      2224 WRITE(6,799)
357      WRITE(6,5006)MDIV,J
358      NPRIN=NPRIN-1

```

```

359      236 WRITE(5,5004)(LL,LL=1,LS)
360      DD 79 K=1,MDIV
361      79 WRITE(5,5007)K,(G(LL,K),LL=1,LS)
362      80 CONTINUE
363      IF(NCODE)2340,2340,2360
364      2340 WRITE(6,799)
365      WRITE(6,5003)
366      WRITE(6,5004)(LL,LL=1,LS)
367      DD 2350 J=1,NS
368      2350 WRITE(6,5005)J,(RMS(LL,J),LL=1,LS)
C      -----DATA CARD FOR PRODUCTION NCODE = 1 INVOLVES COMPUTING
C      FUNCTIONS MT =1,2,3. EITHER COORDS.(U1,V1),(U2,V2) OF LINE, O
C      ITS SIDE NO.J (WHERE IT IS A BOUNDARY SEGMENT) ARE SPECIFIED
C      -----NCODE = 2 INVOLVES DIFFERENTIATION AND
C      NCODE = 3 INTEGRATIONS FOR OPERATION NOFN.
369      2360 READ(5,8008)U1,V1,U2,V2,BXK,MDIV,MT,J,NCODE,NOFN,NAXIS
370      NCARDS=NCARDS+1
371      IF(MDIV+NCODE)100,100,99
372      99 WRITE(6,799)
373      IF(NCODE-1)2361,2361,2362
374      2362 IF(NHOLD)2364,2364,2365
375      2364 READ(5,7991)(XCAR(I),I=1,20)
376      NCARDS=NCARDS+1
377      WRITE(6,4991)(XCAR(I),I=1,20)
378      GO TO 2365
379      2361 WRITE(6,5011)
380      2365 IF(NCODE-3)2369,2368,2369
381      2368 IF(NCFN-1)2369,2370,2369
382      2369 WRITE(6,5010)
383      WRITE(6,5008)U1,V1,U2,V2,BXK,MDIV,MT,J,NCODE,NOFN,NAXIS
384      2370 IF(NAXIS)2366,2366,2367
385      2366 NAXIS=1
386      2367 BXK=BXK*PI*0.5E+00
387      NSIDE=J
388      NSJ=1
389      NCFN=1
390      NSA=1
391      NHOLD=0
392      92 IF(J)95,94,93
393      94 IF(NCODE-2)95,490,96
394      96 GO TO(97,98,98),NOFN
C      ----- NCODE = 3 ; NOFN = 1, TORSIONAL RIGIDITY.
C      NOFN > 1 AVAILABLE , NOT YET SPECIFIED.
395      97 NSIDE=1
396      NSJ=NS
397      MT=1
398      NHOLD=NS
399      MDIV=24
400      GO TO 95
401      98 CONTINUE
402      GO TO 95
403      490 GO TO(492,491,493),NOFN
C      ----- NCODE = 2 ; NOFN = 1, SHEAR STRESS IN TORSION PROBLEM.
C      NOFN = 2, RESULTANT STRESS,AND ITS DIRECTION;
C      INVOLVES TWO CYCLES NN =1,2.
C      NOFN = 3, SHEAR LINES IN TORSION.
404      491 BHOLD(1)=0
405      BHOLD(2)=0.5*PI
406      NCFN=2
407      MT=2

```

```

408          NHDJ=2
409          GO TO 95
410          492 MT=2
411          GO TO 95
412          493 MT=3
413          GO TO 95
414          93 NSA=J
415          NSJ=J
416          GO TO 94
417          95 GO TO 200
418          100 WRITE(6,799)
C          ----- FINAL OPERATION : PRINT OUT OF NCARD DATA CARDS FOR
C          PROBLEM.
419          WRITE(5,5120)NCARDS,NPROG
420          DO 120 N=1,NCARDS
421          READ(5,7991)(XCAR(I),I=1,20)
422          120 WRITE(6,4991)(XCAR(I),I=1,20)
423          WRITE(6,799)
C
C          ----- FORMAT STATEMENTS
C
424          7990 FORMAT(1H ,10X,'APPENDIX B(2) DATA PRINT OUT FOR NPROG =',I3,/)
425          7991 FORMAT(20A4)
426          4991 FORMAT(1H ,20A4)
427          5101 FORMAT(1H ,' L NS NB NPROG NBDY LS NMAT NPRIN MDIV DELTA
          1 FSET GMDD')
428          8000 FORMAT(9I3,3E11.4)
429          5000 FORMAT(1H ,9(I3,2X),3E11.4)
430          4990 FORMAT(1H ,10X,'DATA FOR NS =',I3,' SIDES',/)
431          5002 FORMAT(1H ,' J M3(J) N4S(J) NBY(J) NTYP(J) X(J) Y(J)'
432          4992 FORMAT(1H ,10X,'SUPPLEMENTARY DATA FOR ELLIP./CIRC. BDRIES',/)
433          5103 FORMAT(1H ,' AE(J) BE(J) XE(J) YE(J) ZE(J)'
434          7998 FORMAT(5E11.4)
435          4998 FORMAT(1H ,5E11.4)
436          7999 FORMAT(4I3,3E11.4)
437          4999 FORMAT(1H ,I4,4(I5,3X),3(E11.4,1X))
438          4993 FORMAT(1H ,'J-SIDE COEFFS. C(J,K) FOR NON-ZERO BDRY. CONDIRS. CAS
          1 NBRV=1')
439          8001 FORMAT(7E11.4)
440          5001 FORMAT(1H ,I4,3X,6(E11.4,1X))
441          5018 FORMAT(1H ,6X,' PRINT OF MATRIX BY COLS IF SET NMAT =1 ',/)
442          5016 FORMAT(1H ,10X,'PRINT OUT OF COLUMN JC=',I3,/)
443          5017 FORMAT(1X,8E11.4)
444          5019 FORMAT(1H ,4X,'PRINT OUT OF RESIDUAL FOR EQUATION NO. N IF NEK41
          C LS =',I3,' TRUNC. LEVELS'/)
445          5009 FORMAT(1H ,4X,'FUNCTIONS MT =',I3,' AT N=MDIV = ',I3,' PTS. ON L
          CNE (J1,V1),(U2,V2) FOR TRUNC. LEVELS LL=1,LS ' )
446          5006 FORMAT(1H ,3X,'PRINT OUT OF RESIDUALS AT MDIV=',I3,' POINTS ON SI
          CE J =',I3,' FOR TRUNC. LEVELS LL',/)
447          5007 FORMAT(1H ,I4,2X,4(2X,E11.4))
448          5003 FORMAT(1H ,6X,'ROOT MEAN SQUARE OF BOUNDARY RESIDUALS ON SIDE N=
          C FOR TRUNC. LEVELS LL=1,LS ' )
449          5004 FORMAT(1H ,' NO.=N',4(6X,'LL =',I3))
450          5005 FORMAT(1H ,I4,4X,4(2X,E11.4))
451          5011 FORMAT(1H ,10X,'PRODUCTION AT POINTS ON GIVEN LINE',/)
452          5010 FORMAT(1H ,'DATA U1 V1 U2 V2 BXK
          1 MDIV MT J NCODE NOFN NAXIS')
453          8008 FORMAT(5E11.4,5I3,I2)
454          5008 FORMAT(1H ,4X,5E11.4,6I5)
455          799 FORMAT(1H ,6X,' ***** ' )

```

```

456 5120 FORMAT(1H , ' PRINT OUT OF NCARDS = ',I3,' FORMATTED DATA CARDS FC
1 CASE NPROG=',I3,/)
457 STOP
458 END

```

C
C
C
C
C

```

459 SUBROUTINE COLMAT
460 COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,10)
CG(99,100),SDL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(50)
C,XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),GMOD,DFIX(50),TOR(4)
461 COMMON BFIX(50),ZE(11),F(4),COEF(2,100),PI,DELTA,XCAR(40),D(50)
462 COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPROG,NE,LS
463 COMMON VROW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM
464 COMMON NIX(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50)
465 COMMON NFP,NZERO,NSS,NB,NOEL(10),LOC(50),NBDY,NFPOL

```

C
C
C

SECTION 1. TO SET UP DATA FOR POINTS AT VERTEX J IN THE
VERTEX EQUATIONS.

```

466 NE=0
467 NR=0
468 NJDRF=0
469 NP=NH*L/2
470 NELIM=0
471 NJDRF=0
472 IF(NSS)101,101,200
473 200 DO 201 J=1,NSS
474 NVEX=NVER(J)
475 IF(NVEX)201,201,1
476 1 JJ 2 K=1,4
477 2 NEQ(K)=1
478 JOMIT=1
479 TEST=ABS(COS(ANG(J)))
480 IF(TEST-0.00001)3,3,4
481 3 JOMIT=0
C -----EXAMINE THE VARIOUS TYPES , NVEX , OF CONDITIONS AT J.
C MAXIMUM OF 4 VERTEX EQS.- IF REQUIRE TO OMIT ONE EQ. SET
C CORRESPONDING INDICATOR NEQ(K) TO 0.
482 4 GO TO (51,5,6,6),NVEX
483 51 JJ=J-1
484 IF(JJ)52,52,53
485 52 JJ=NSS
486 53 XX=C(JJ,1)+C(JJ,2)+C(JJ,3)+C(JJ,4)+C(JJ,5)+C(JJ,6)-C(J,1)
487 IF(ABS(XX)-0.000001)54,54,55
488 54 NEQ(3)=0
489 GO TO 9
490 55 NP=NP+1

```

C
C
C
C
C
C

DISCONTINUITY IN FUNCTION (NVEX=1) AT ANY VERTEX REQUIRES
USE OF CORRECTIVE THETA FUNCTION IN COL NP OF MIX AND SET UP
CORR. COL. DATA INDICATED BY SETTING E(NP)=2001. THESE ARE
TALLIED BY NJDRF.

```

491 NA(NP)=J
492 NFU(NP)=4
493 E(NP)=2001
494 NJDRF=NJDRF+1

```

```

495      GO TO 9
496      5 NEQ(2)=JDMIT
497      GO TO 9
498      6 NEQ(4)=JDMIT
499      9 DO 20 K=1,4
500      IF(NEQ(K))20,20,10
501      10 NR=NR+1
502      NE=NE+1
503      LJC(NE)=NR
504      JJ=J
505      GO TO (11,83,13,13),K
C      -----SET UP DATA FOR POINT,(K=1,K=3) AND 'TWO-POINT',(K=2,K=4)
C      VERTEX EQ. NE IN SUBSCRIPTED LOCATIONS NR. NE TALLIES EQS.
506      11 XFIX(NR)=X(J)
507      YFIX(NR)=Y(J)
508      BFIX(NR)=B(1,J)
509      MBFIX(NR)=MB(J)
510      JFIX(NR)=J
511      DFIX(NR)=0
512      GO TO 20
513      13 JJ=J-1
514      IF(JJ)14,14,15
515      14 JJ=NSS
516      15 IF(K-3)82,82,83
517      82 XFIX(NR)=X(J)
518      YFIX(NR)=Y(J)
519      BFIX(NR)=B(2,JJ)
520      JFIX(NR)=JJ
521      MBFIX(NR)=MB(JJ)
522      DFIX(NR)=1.0
523      GO TO 20
524      83 NR=NR-1
C      ----- 'TWO-POINT' EQS. ON STRAIGHT AND ELLIPTIC SEGMENTS.
525      DO 100 M=1,2
526      NR=NR+1
527      DK=((-1)**M)*DELTA
528      IF(K-2)85,85,84
529      84 DK=1-DK
530      85 DFIX(NR)=DK
531      JFIX(NR)=JJ
532      MBFIX(NR)=MB(JJ)+10
533      IF(NTYP(JJ))86,86,87
534      86 XFIX(NR)=X(JJ)+DK*(X(JJ+1)-X(JJ))
535      YFIX(NR)=Y(JJ)+DK*(Y(JJ+1)-Y(JJ))
536      BFIX(NR)=B(1,JJ)
C      ----- SIMILAR TO SUBROUTINE ELPS.
537      GO TO 100
538      87 Z1=PAR(1,JJ)+DK*(PAR(2,JJ)-PAR(1,JJ))
539      ZZ=ZE(JJ)
540      X1=COS(Z1)
541      Y1=SIN(Z1)
542      IF(Y1)89,88,89
543      88 Y1=0.1D-10
544      89 Z5=-BE(JJ)*X1/(AE(JJ)*Y1)
545      ADD=PI
546      IF(Z1->1)91,91,90
547      90 ADD=2*PI
548      91 Z6=ATAN(Z5)+ADD
549      ADD=0
550      IF(NTYP(JJ)-2)93,92,93

```

```

551      92 ADD=PI
552      93 3FIX(NR)=Z6+ZZ+ADD
553          X1=X1*AC(JJ)
554          Y1=Y1*BE(JJ)
555          XFIX(NR)=XE(JJ)+X1*COS(ZZ)-Y1*SIN(ZZ)
556          YFIX(NR)=YE(JJ)+X1*SIN(ZZ)+Y1*COS(ZZ)
557      100 CONTINJE
558          20 CONTINJE
559      201 CONTINJE
C          REMOVE INDETERMINACY IN NEUMANN PROBLEM BY PUTTING
C          FUNCTION TO ZERO AT LAST VERTEX, AND OMITTING THE LAST
C          VELIM EQS. AS LEFT BY COL. SELECTION PROCEDURE IN SOLCOR
560      101 VELIM=1
561          DO 103 J=1,NS
562          IF(M3(J)-2)102,103,103
563      102 VELIM=0
564      103 CONTINUE
565          IF(VELIM)105,105,104
566      104 NR=NR+1
567          NE=NE+1
568          LOC(NE)=NR
569          XFIX(NR)=X(2)
570          YFIX(NR)=Y(2)+BE(1)
571          MBFIX(NR)=1
572          DFIX(NR)=0
573          JFIX(NR)=10
C          ----- REMOVE INDETERMINACY IN HARMONIC CONJUGATE EPSI,MT=3, BY
C          SETTING EPSI=0 AT VERTEX 1, OR END OF MAJOR AXIS OF ELLIPS
574      105 NR=NR+1
575          NE=NE+1
576          LOC(NE)=NR
577          JFIX(NR)=10
578          DFIX(NR)=0
579          MBFIX(NR)=3
580          XFIX(NR)=X(1)+AC(1)
581          YFIX(NR)=Y(1)
582      210 NFP=NE
583          JRF=NR
584          NZERO=NS
585          NE=NE+NZERO
586          NORF=NE-JNRF
C          ----- NORF = TOTAL NO. OF VERTEX FUNCTIONS REQUIRED.
C          NZERO = TOTAL NO. OF ZERO HARMONIC EQS.
587          IF(NTYP(1))23,23,21
588      21 NP=NP+1
C
C          SET UP COLS. FOR CONSTANT TERM FOR ZERO HARMONIC FOR EXTERIOR
C          ELLIPSE & LOG FUNCTION FOR EACH ELLIPTIC INDENTATION
C          INDICATED BY SETTING E(NP)=1000
589          NA(NP)=1
590          NFU(NP)=5
591          E(NP)=0
592          NORF=NORF-1
593          IF(NS-2)24,23,23
594      23 DO 240 J=2,NS
595          IF(NTYP(J))240,240,230
596      230 NP=NP+1
597          NA(NP)=J
598          NFU(NP)=5
599          IF(AC(J)-BE(J)-0.100E-04)232,232,231

```

```

600 231 NFU(NP)=6
601 232 E(NP)=1000
602     NDRF=NDRF-1
603 240 CONTINUE

```

```

C
C
C
C
C

```

SECTION 2.

SET UP DATA FOR, AND ASSIGN, FUNCTIONS
FOR COLUMNS OF MATRIX AS IN TABLE 1 -VERTEX
FUNCTIONS.

```

604 24 IF(NDRF)37,37,25
605 25 NNN=NDRF/NSS
606     NN=NDRF-NNN*NSS
607     DO 30 J=1,NSS
608     NAD(J)=NNN
609     IF(J-NN)29,29,30
610 29 NAD(J)=NNN+1
611 30 CONTINUE
612     DO 40 J=1,NSS

```

```

C

```

----- NAD(J) = NO. OF VERTEX FUNCTIONS REQUIRED FROM VERTEX J

```

613     KK=NAD(J)
614     DO 40 K=1,KK
615     NP=NP+1
616     NVEX=NVER(J)
617     ALFA=ANG(J)
618     EE=K*PI/ALFA
619     EF=EE-PI/(2*ALFA)
620     GO TO (31,32,33,34),NVEX
621 31 CA=0
622     CB=1
623     CE=EE
624     GO TO 35
625 32 CA=0
626     CB=1
627     CE=EF
628     GO TO 35
629 33 CA=1
630     CB=0
631     CE=EF
632     GO TO 35
633 34 CA=1
634     CB=0
635     CE=EE
636 35 COEF(1,NP)=CA
637     COEF(2,NP)=CB
638     VA(NP)=J
639     E(NP)=CE
640     VFU(NP)=4

```

```

C
C
C

```

CHECK FOR INTEGER VALUES OF LAMDA=CE, AND INDICATE BY
INCREASING STORAGE VALUE E(NP) BY 1000.

```

641     VX=CE+0.0000001
642     IF(CE-VX-0.0001)36,36,40
643 36 E(NP)=CE+1000
644 40 CONTINUE

```

```

C

```

SET UP COLUMN LP FOR R.H.S. OF EQUATION.

```

645     LP=NP+1
646     VFU(LP)=3
647     VA(LP)=10
648 37 NP=0

```

```

C

```

C SECTION 3. SET UP COLUMN SETS NP, EACH COMPRISING TWO
 C COLS., MATCHING TWO EQS. FOR HARMONIC SETS M
 C ON SIDE J .
 C

```

649      DD 50 MS=1,L
650      DD 50 J=1,NS
651      MM=VHS(J)
652      DD 50 M=1,MM
653      NP=NP+1
654      ME=(MS-1)*MM+M
655      NA(NP)=J
C ----- FULL EDGE FNS.(55)
656      IF(NTYP(J))41,41,42
657      41 NFU(NP)=1
658      E(NP)=2*PI*ME/A(J)
659      GO TO 44
C ----- POLAR NFU = 5, OR CURVED EDGE FNS. NFU = 6
660      42 NFU(NP)=5
661      XX=ABS(AE(J)-BE(J))-0.1*AE(J)
662      IF(XX)46,46,45
663      45 NFU(NP)=6
664      46 E(NP)=ME*2*PI/ABS(PAR(2,J)-PAR(1,J))
665      IF(NTYP(J)-2)43,44,48
666      48 IF(J-1)44,44,43
667      43 E(NP)=-E(NP)
668      44 NE=NE+2
669      50 CONTINUE
C
C      PRINT OUT OF DETAILS FOR NFP FIXED POINTS AND LP COLUMN SETS.
C
670      WRITE(6,902)NFP
671      WRITE(6,799)
672      WRITE(6,903)
673      ZZ=NH*L+NZERO+0.5000
674      DD 904 K=1,JRF
675      ZZ=ZZ+1-0.4999*(MBFIX(K)/10)
676      KK=ZZ
677      904 WRITE(6,905)XFIX(K),YFIX(K),BFIX(K),DFIX(K),JFIX(K),MBFIX(K),KK
678      WRITE(6,799)
679      WRITE(6,906)LP
680      WRITE(6,907)
681      DD 908 K=1,LP
682      908 WRITE(6,909)NFU(K),NA(K),E(K),K
683      WRITE(6,799)
684      WRITE(6,910)NE,NH,L
685      WRITE(6,799)
686      902 FORMAT(1H ,10X, ' DETAILS OF NFP=',I3,' FIXED POINTS ASSIGNED IN
687      1 COLMAT')
687      799 FORMAT(11X, '*****',/)
688      903 FORMAT(7X, ' XFIX      YFIX      BFIX      DFIX      JFIX
689      1MBFIX      EQ NO. ')
689      905 FORMAT(1X,4E11.4,3(7X,I3))
690      906 FORMAT(7X,'DETAILS OF LP=',I3,' SETS OF FNS ASSIGNED IN COLMAT',/
691      907 FORMAT(1X, '      NFU      NA      E(NP)      NP ')
692      909 FORMAT(1X,2(4X,I3),2X,E11.4,I5)
693      910 FORMAT(1H ,4X,' NO. OF EQUATIONS IN MATRIX NE =',I3,/
694      1 ,4X, 'NO. OF EQUATIONS IN HARMONIC SET NH =',I3,/
695      2 ,4X, 'NO. OF HARMONIC SETS L= ',I3)
694      RETJRN
695      END

```

C
C
C
C
C
C
C

EDGEF. EDGE FUNCTIONS, EQS. (55)+(56).
VA AND VB STORE COEFFS. OF UNKNOWNNS FOR FUNCTION MT=1,2,OR 3.

```
696            SUBROUTINE EDGEF(XK,YK,BXK,XJE,YJE,BJE,EM,PI,MT,VA,VB)
697            XXX=(XK-XJE)*COS(BJE)+(YK-YJE)*SIN(BJE)
698            YYK=- (XK-XJE)*SIN(BJE)+(YK-YJE)*COS(BJE)
699            SSS=BXK-BJE
700            EMM=EXP(-EM*YYK)
701            IF(MT-2)1,2,4
702            4 VA=EMM*SIN(EM*XXX)
703            VB=EMM*COS(EM*XXX)
704            GO TO 3
705            1 VA=EMM*COS(EM*XXX)
706            VB=-EMM*SIN(EM*XXX)
707            GO TO 3
708            2 VA=-EM*EMM*COS(EM*XXX+SSS)
709            VB= EM*EMM*SIN(EM*XXX+SSS)
710            3 RETJRN
711            END
```

C
C
C
C
C
C

POLW. VERTEX FUNCTIONS , TABLE 1 , VA STORES
COEFF. (50) OF UNKNOWNN .

```
712            SUBROUTINE POLW(XK,YK,BXK,XJE,YJE,BJE,EM,CA,CB,MT,VA,PI,DELTA)
713            XA=XK-XJE
714            YA=YK-YJE
715            CALL QPOLAR(XA,YA,RR,BA,PI)
716            ----- GIVES POLAR COORDS. (RR,BA) FOR POINT.
717            BA=BA-BJE
718            DDEL=10*DELTA
719            IF(BA+DDEL)28,28,29
720            28 BA=BA+2*PI
721            29 SS=BXK-BJE
722            IF(EM-2000)32,32,30
723            30 VA=BA
724            IF(MT-2)50,31,41
725            41 VA=-ALOG(RR)
726            GO TO 50
727            31 VA=COS(SS-BA)/RR
728            GO TO 50
729            32 NPS=1+EM/1000
730            ----- LOG-VERTEX FNS. EQS.(60) INDIC. EM>1000 ; NPS=2
731            GOT BY NUMERICAL DIFF. INTERVAL DELTA
732            GIVES VERTEX FUNCTIONS WHEN NPS =1 .
733            EM=EM-1000*(NPS-1)
734            XX=0
735            DO 10 K=1,NPS
736            EN=EM
737            IF(NPS-2)45,44,45
738            44 EN=EM*(1+((-1)**K)*DELTA)
739            45 EMM=EN-MT+1
```

50	0.00000E 00	0.00000E 00	0.65030E-04	0.66308E-04
51	0.00000E 00	0.00000E 00	0.00000E 00	0.16098E-04
52	0.00000E 00	0.00000E 00	0.00000E 00	-0.16523E-03
53	0.00000E 00	0.00000E 00	0.00000E 00	0.40161E-04

```

736      IF(MT-3)33,36,36
737      36 EMM=EV-MT+3
738      33 RE=0
739      IF(RR-0.00001)20,20,19
740      19 RE=XX**EMM
741      20 IF(MT-2)1,2,3
742      3 XX=XX+((-1)**K)*(CA*SIN(EM*BA)-CB*COS(EM*BA))*RE
743      GO TO 10
744      1 XX=XX+((-1)**K)*(CA*COS(EM*BA)+CB*SIN(EM*BA))*RE
745      GO TO 10
746      2 XX=XX+((-1)**K)*(-CA*SIN(EMM*BA+SS)+CB*COS(EMM*BA+SS))*EN*RE
747      10 CONTINUE
748      VA=XX
749      50 RETURN
750      END

```

C
C
C
C
C

PMAP MAPPING Z=C COSH(W) CURVED EDGE-FUNCTIONS SEC.
VA AND VB STORE COEFFS. OF UNKNOWNS IN EQ(89) FOR MT=1,2,OR 3.

```

751      SUBROUTINE PMAP(XK,YK,BXK,XJE,YJE,BJE,EM,MT,JE,VA,VB)
752      COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,10)
      CS(99,100),SOL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(50)
      C,XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),SMOD,DFIX(50),TOR(4)
753      COMMON BFIX(50),ZE(11),F(4),COEF(2,100),PI,DELTA,XCAR(40),D(50)
754      COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPROG,NE,LS
755      COMMON NROW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM
756      COMMON NDK(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50)
757      COMMON NFP,NZERO,NSS,NB,NOEL(10),LOC(50),NBDY,NFPOL
758      XX=XK-XJE
759      YY=YK-YJE
760      ZK=BJE
761      VB3=0
762      VAA=0

```

C

```

----- EQUATIONS (82) , (83) , (84) .
763      XA=XX*COS(ZK)+YY*SIN(ZK)
764      YA=-XX*SIN(ZK)+YY*COS(ZK)
765      CC=AE(JE)**2-BE(JE)**2
766      C1=(CC-XA**2-YA**2)
767      T=(-C1+SQRT(C1**2+4*CC*(YA**2)))/(2*CC)
768      IF(ABS(YA)-0.1E-06)480,480,482
769      480 T=-1+XA*XA/CC
770      IF(T)481,481,482
771      481 T=0
772      482 T1=SQRT(T)+SQRT(1+T)
773      XI=ALOG(T1)
774      XA=XA*(EXP(2*XI)-1)
775      YA=YA*(EXP(2*XI)+1)
776      CALL QPOLAR(XA,YA,RR,BA,PI)
777      ETA=BA

```

C

C

```

----- SET UP EQS(89)+(90) FOR SOLID ELLIPSE.
778      VA=0
779      VB=0
780      NCY=1
781      IF(JE-1)1,1,3
782      1 IF(NTYP(1)-3)3,2,3
783      2 NCY=2
784      3 DO 510 N=1,NCY

```

```

785         IF(EM-999)6,6,490
786     490 IF(MT-2)491,502,492
787     491 VAA=ALOG(X1)
788         GO TO 504
789     492 VAA=3A
790         GO TO 504
791         6 IF(N-2)5,4,4
792         4 EM=-EM
793         5 ETAM=BA*EM
794         XIM=XI*EM
795         FA=EXP(XIM)
796         FB=COS(ETAM)
797         FC=SIN(ETAM)
798         IF(MT-2)501,502,503
799     501 VAA=FA*FB
800         VBB=-FA*FC
801         GO TO 504
802     502 X1=SIN(ETA)
803         X2=COS(ETA)
804         X3=EXP(X1)
805         X4=1/X3
806         X5=X3-X4
807         X6=X3+X4
808         DX=0.5*(X6*X6-4*X2*X2)
809         IF(A3S(DX)-0.1E-07)505,505,506
810     505 DX=0.1E-07
811     506 CD=SQRT(CC)
812         D1=X5*X2/(CD*DX)
813         D2=-X6*X1/(CD*DX)
814         IF(EM-999)494,494,493
815     493 X2=COS(3XK)
816         X1=SIN(3XK)
817         X3=X1*X1+ETA*ETA
818         X4=X1/X3
819         X5=ETA/X3
820         VAA=-X1*(X4*D1+X5*D2)+X2*(-X4*D2+X5*D1)
821         GO TO 504
822     494 FQJ=3XK-BJE
823         FB=COS(ETAM-FQJ)
824         FC=SIN(ETAM-FQJ)
825         FA=FA*EM
826         VAA=FA*(-D1*FC+D2*FB)
827         VBB=FA*(-D1*FB-D2*FC)
828         GO TO 504
829     503 VAA=FA*FC
830         VBB=FA*FB
831     504 VA=VA+VAA
832     510 VB=VB+VBB
833         RETJRN
834         END

```

C
C
C
C
C

QPOLAR.

GIVES POLAR CORRDS.(RR,BA) WHERE BA CONTINUOUS
IN (0,2*PI).

```

835     SUBROUTINE QPOLAR(XA,YA,RR,BA,PI)
836     RR=SQRT(XA**2+YA**2)
837     IF(A3S(XA)-0.00001)7,8,8
838     7 XA=XA+0.00001

```

```

839      8 BA=ATAN(YA/XA)
840      IF(XA)9,9,10
841      9 BA=BA+PI
842      GO TO 12
843      10 IF(YA)11,12,12
844      11 BA=BA+2*PI
845      12 RETJRN
846      END

```

```

C
C
C
C
C
C
C
C
C

```

```

ELPS      GIVES (1) PARAMETERS PAR(1,J) AND PAR(2,J) AND
SLOPES B(1,J) AND B(2,J) AT END POINTS OF BDRY.
SEGMENT FOR KK=0 ;
(2) COORDS. (XK,YK) AND SLOPE BXK AT KK
POINTS ON SEGMENT, KK>0. STORED IN XEL,YEL,BEL.

```

```

847      SUBROUTINE ELPS(J,NNEL,KK)
848      COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,10)
CG(99,100),SOL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(50)
C,XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),GMOD,DFIX(50),TOR(4)
849      COMMON BFIX(50),ZE(11),F(4),COEF(2,100),PI,DELTA,XCAR(40),D(50)
850      COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPRG,NE,LS
851      COMMON NROW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM
852      COMMON NDK(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50)
853      COMMON NFP,NZERO,NSS,NB,NOEL(10),LDC(50),NBDY,NFPOL
854      IF(KK)300,300,330

```

```

C ----- PARAMETERS OF END POINTS.

```

```

855      300 IF(NTYP(J)-3)302,301,301
856      301 PAR(1,J)=0
857      PAR(2,J)=2*PI
858      A(J)=2*PI
859      GO TO 500
860      302 IF(BE(J))303,303,304
861      303 BE(J)=AE(J)
862      304 DO 315 K=1,2
863      JJ=J+K-1
864      XX=X(JJ)-XE(J)
865      YY=Y(JJ)-YE(J)
866      ZK=ZE(J)
867      XA=XX*COS(ZK)+YY*SIN(ZK)
868      YA=-XX*SIN(ZK)+YY*COS(ZK)
869      XXA=BE(J)*XA
870      YYA=AE(J)*YA
871      CALL SPOLAR(XXA,YYA,RR,BA,PI)
872      XA=COS(BA)
873      YA=SIN(BA)
874      IF(YA)433,432,433
875      432 YA=YA+0.000001
876      433 BAA=ATAN(-BE(J)*XA/(AE(J)*YA))
877      IF(YA)435,435,434
878      434 BAA=BAA+PI
879      GO TO 437
880      435 IF(XA)436,436,437
881      436 BAA=3AA+2*PI
882      437 PAR(K,J)=BA
883      B(K,J)=BAA+ZK
884      IF(K-2)315,311,311
885      311 PP=PAR(2,J)-PAR(1,J)

```

```

886      IF(NTYP(J)-2)315,312,314
887      312 IF(P)315,315,313
888      313 PAR(2,J)=PAR(2,J)-2*PI
889      GO TO 315
890      314 IF(P)316,315,315
891      316 PAR(2,J)=PAR(2,J)+2*PI
892      315 CONTINUE
893      A(J)=PAR(2,J)-PAR(1,J)
894      GO TO 500
C      ----- KK POINTS ON SEGMENT J.
895      330 DO 340 K=1,KK
896      DK=D(K)
897      XX=PAR(1,J)+DK*(PAR(2,J)-PAR(1,J))
898      XA=COS(XX)
899      YA=SIN(XX)
900      IF(YA)333,332,333
901      332 YA=YA+0.000001
902      333 BA=ATAN(-BE(J)*XA/(AE(J)*YA))
903      IF(YA)335,335,334
904      334 BA=BA+PI
905      GO TO 337
906      335 BA=BA+2*PI
907      337 CONTINUE
908      ZK=ZE(J)
909      XA=XA*AE(J)
910      YA=YA*BE(J)
911      XEL(NVEL,K)=XE(J)+XA*COS(ZK)-YA*SIN(ZK)
912      YEL(NVEL,K)=YE(J)+XA*SIN(ZK)+YA*COS(ZK)
913      BEL(NVEL,K)=BA+ZK
914      340 CONTINUE
915      500 RETURN
916      END

```

C
C
C
C

SOLCOR.

SOLVER ROUTINE FOR LS TRUNCATION LEVELS.

```

917      SUBROUTINE SOLCOR
918      DIMENSION JRSTOP(100),HOLD(40,40)
919      COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,10)
CG(99,100),SDL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(50)
C,XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),GMOD,DFIX(50),TOR(4)
920      COMMON BFIX(50),ZE(11),F(4),CDEF(2,100),PI,DELTA,XCAR(40),D(50)
921      COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPRG,NE,LS
922      COMMON VRDW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM
923      COMMON NDK(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50)
924      COMMON NFP,NZERC,NSS,NB,NOEL(10),LOC(50),NBDY,NFPOL
925      NDE=2
926      LRJW=NE-NH*L
927      LLRJDW=LRJW+1
928      LSET=L*(NH/NDE)
929      NEE=NE+1
C      ----- MATRIX ARRANGED IN LSET SETS OF PAIRS OF HARMONIC
C      EQS. FOR SUCCESSIVE SIDES IN EACH HARMONIC SET ; FOLLOWED
C      BY LOW OTHER (ZERO HARMONICS AND VERTEX) EQS.. THE LAST
C      ROW OF SET CONTAINING EQ. NO. MM IS JRSTOP(MM).
C
930      MM=0
931      DO 3 K=1,LSET
932      DO 3 M=1,NDE

```

```

933      MM=MM+1
934      3 JRSTDP(MM)=NDE*K
935      MMA=MM+1
936      DO 4 MM=MMA,NE
937      4 JRSTDP(MM)=NE
938      KS=L-LS+1
939      JR2=0
940      NEQH=NH*L
941      _R3W=_R3W+1
C      ----- SOLUTION FOR TRUNCATION AT KL HARMONICS
942      DO 300 KL=KS,L
943      JR1=JR2+1
944      JR2=<L*NH
945      JR3=JR2+LR0W-1
946      DO 240 JRA=JR1, JR3
C      ----- ELIM. OF UNKNOWN JC TO TRIANGULATE MATRIX USING
C      PIVOT SELECTION FOR ROWS JC TO JRS--APPROPRIATE ROWS FOR
C      PIVOTING FOR COL JC .
947      JC=JRA
948      IF(JRA-JR2-1)201,190,200
949      190 DO 191 JCC=1,LLROW
950      DO 191 JRR=1,LR0W
951      JCX=NEQH+JCC
952      JRX=NEQH+JRR
C      ----- HOLD AS REQUIRED WHEN SOLVING FOR NEXT KL TRUNC. LEVEL.
953      191 HOLD(JRR,JCC)=G(JRX,JCX)
954      200 JC=NEQH+JRA-JR2
955      201 XX=0
956      JRS=JRSTDP(JC)
957      DO 210 JR=JC,JRS
958      YY=ABS(G(JR,JC))
959      IF(XX-YY)202,210,210
960      202 XX=YY
961      JRM=JR
962      ZZ=G(JR,JC)
963      210 CONTINJE
964      DO 216 JCC=JC,NEE
965      XX=G(JC,JCC)
966      G(JC,JCC)=G(JRM,JCC)
967      216 G(JRM,JCC)=XX
968      JC1=JC+1
969      ZZ=1/ZZ
970      DO 220 JCC=JC,NEE
971      220 G(JC,JCC)=G(JC,JCC)*ZZ
972      DO 230 JRR=JC1,NE
973      XG=G(JRR,JC)
974      DO 230 JCC=JC,NEE
975      230 G(JRR,JCC)=G(JRR,JCC)-XG*G(JC,JCC)
976      240 CONTINJE
C      ----- BACK SUBSTITUTE IN TRIANG. MATRIX FOR SOLN. FOR
C      TRUNC. AT KL HARMONICS.
977      LL=<L-<S+1
978      SOL(LL,NEE)=-1
979      ZZ=ABS(G(NE,NE))
980      IF(ZZ-0.1000E-06)1240,1240,1241
981      1240 NELIM=1
982      GO TO 222
983      1241 SOL(...,NE)=G(NE,NEE)/G(NE,NE)
C      ----- OMTS LAST NELIM EQUATIONS ; NELIM SET IN COLMAT
C      MEANS OF CUTTING OUT REDUNDANT EQS.

```

```

984      IF(NELIM)229,229,222
985      222 DO 223 K=1,NELIM
986          M=NE-K+1
987          SOL(LL,NE)=0
988      223 G(M,M+1)=0
989      229 JR3=JR3+1
990          DO 250 K=2,JR3
991              M=NE-K+1
992              M1=M+1
993              IF(M-NEQH)242,242,243
994      242 M=M-NEQH+JR2
995      243 XX=0
996          DO 245 MM=M1,NEE
997              MA=MM
998              IF(M4-NEQH)241,241,245
999      241 MA=MM-NEQH+JR2
1000      245 XX=XX+SOL(LL,MA)*G(M,MA)
1001          SOL(LL,M)=-XX
1002          IF(K-LROW)250,246,250
1003      246 DO 247 JCC=1,LLROW
1004          DO 247 JRR=1,LROW
1005              JCX=NEQH+JCC
1006              JRX=NEQH+JRR
1007      247 G(JRX,JCX)=HOLD(JRR,JCC)
1008      250 CONTINUE
1009      300 CONTINUE
C
1010      350 WRITE(6,799)
1011          WRITE(6,110)
1012          WRITE(6,799)
1013          WRITE(6,111)LS
1014          WRITE(6,112)
1015          WRITE(6,115)(LL,LL=1,LS)
1016          DO 351 J = 1,NE
1017              WRITE(6,101)J,(SOL(K,J),K=1,LS)
1018      351 CONTINUE
1019      110 FORMAT(140,30X,' SOLUTION VECTORS',/)
1020      111 FORMAT(1H ,' SOLUTIONS OF COEFS. MATRIX FOR NO OF SETS LS =',I4
1021      112 FORMAT(1H ,' THE DIFF. LEVELS OF TRUNC. ARE DENOTED BY LL=1,LS')
1022      115 FORMAT(1H ,' COEF.',4(3X,' LL =',I3,2X))
1023      101 FORMAT(1H ,I4,2X,4(D12.5,2X))
1024      799 FORMAT(1H ,25X,25H*****//)
1025          RETURN
1026          END

```

```

C
C
C      P J L C .                H A R M O N I C   P O L A R S .
C

```

```

1027      SUBROUTINE POLC(XK,YK,BXK,XJE,YJE,BJE,EM,MT,VA,VB,PI)
1028          VA = 0
1029          VB = 0
1030          XA=XK-XJE
1031          YA=YK-YJE
1032          CALL QPOLAR(XA,YA,RR,BA,PI)
1033          SS = BXK-BJE
1034          BA=BA-3JE
1035          IF(EM-999)4,4,1
C
1036      1 IF(MT-2)2,3,11

```

```

1037      11 VA=BA
1038      GO TO 9
1039      2 VA=ALOG(RR)
1040      GO TO 9
1041      3 VA=-SIN(SS-BA)/RR
1042      GO TO 9
1043      4 IF(EM)6,5,6
C          ----- CASE EM = 0 , U = CONSTANT .
1044      5 VA=0
1045      IF(MT-1)9,15,9
1046      15 VA = 1
1047      GO TO 9
C          ----- POLAR HARMONICS N & -N          FQS.(58).
1048      6 IF(MT-2)7,8,20
1049      20 XX=RR**EM
1050      VA=XX*SIN(EM*BA)
1051      VB=-XX*COS(EM*BA)
1052      GO TO 9
1053      7 XX=RR**EM
1054      VA=XX*COS(EM*BA)
1055      VB=XX*SIN(EM*BA)
1056      GO TO 9
1057      8 EMM = EM-1
1058      XX=(RR**EMM)*EM
1059      VA=-XX*SIN(EMM*BA+SS)
1060      VB=XX*COS(EMM*BA+SS)
1061      9 RETJRN
1062      END
C
C
C          POLP      PARTICULAR SOLNS. (NOT YET INCLUDED), & R.H.S.
C
1063      SUBROUTINE POLP(XK,YK,BXK,DK,MT,NCODE,JP,VA)
1064      COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,10)
CG(99,100),SDL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(50)
C, XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),GMOD,DFIX(50),TOR(4)
1065      COMMON BFIX(50),ZE(11),F(4),COEF(2,100),PI,DELTA,XCAR(40),D(50)
1066      COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPROG,NE,LS
1067      COMMON NROW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM
1068      COMMON NDK(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50)
1069      COMMON NP,NZERO,NSS,NB,NOEL(10),LOC(50),NBDY,NFPOL
1070      VA=0
1071      IF(NCODE)3,10,3
1072      10 J=JP
1073      IF(NBY(JP))3,3,1
1074      1 IF(NBDY-2)4,2,3
1075      4 VA=C(J,1)+C(J,2)*DK+C(J,3)*DK**2+C(J,4)*DK**3+C(J,5)*DK**4+C(J,6)*
C*5
1076      GO TO 3
1077      2 IF(MT-2)3,5,3
1078      5 VA=-XK*COS(BXK)-YK*SIN(BXK)
C          ----- R.H.S. FOR TORSION PROBLEM
1079      3 RETJRN
1080      END
C
C
C          PRODV      SUBROUTINE FOR SPECIAL OUTPUTS  NCODE=2 & NCODE=3,
C                    FOR CASES DESIGNATED BY NOFN = 1,2,3 .

```

```

C
1081      SUBROUTINE PRODN(J,NSJ,NCODE,NOFN,NNHOLD,NHOLD)
1082      COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,10)
      CS(99,100),SDL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(50)
      C,XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),GMOD,DFIX(50),TOR(4)
1083      COMMON BFIX(50),ZE(11),F(4),COEF(2,100),PI,DELTA,XCAR(40),D(50)
1084      COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPROC,NE,LS
1085      COMMON VROW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM
1086      COMMON NOK(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50)
1087      COMMON NFP,NZERO,NSS,NB,NOEL(10),LOC(50),NBDY,NFPOL
1088      NNHOLD=NNHOLD+1
1089      LLA=I
1090      JUMP=0
1091      IF(NCODE-2)5,5,8

```

```

C ----- NCODE=2 MAY INVOLVE DIFFERENTIATIONS FOR
C CASES NOFN = 1,2,3 GIVEN .

```

```

1092      5 DJ 6 LL=1,LS
1093      DO 6 K=1,MDIV
1094      XK=XFIX(K)
1095      YK=YFIX(K)
1096      BXK=BFIX(K)
1097      GO TO(51,51,52),NOFN
1098      51 XX=1/(TOR(LL)*GMOD)
1099      XKG=COS(2*BXK)*XX
1100      XKF=(XK*COS(BXK)-YK*SIN(BXK))*XX
1101      NHOLD=NOFN
1102      IF(NOFN-2)54,53,54
1103      52 XKG=1
1104      XKF=-0.5*(XK*XK+YK*YK)
1105      NHOLD=1
1106      GO TO 54
1107      53 JLL=NNHOLD*4+LL
1108      GO TO 6
1109      54 JLL=LL
1110      6 G(JLL,K)=G(LL,K)*XKG+XKF
1111      IF(NHOLD-NNHOLD)20,7,20
1112      7 NHOLD=0
1113      NHOLD=0
1114      GO TO(70,55,70),NOFN

```

```

C ----- PROCESSING FOR NOFN = 2 FOR RESULTANT STRESS AND ITS
C DIRECTION.

```

```

1115      55 DJ 60 LL=1,LS
1116      DO 60 K=1,MDIV
1117      ZA=G(LL+4,K)
1118      ZB=G(LL+8,K)
1119      G(LL,K)=SQRT(ZA*ZA+ZB*ZB)
1120      ZC=ATAN(ZB/ZA)
1121      IF(ZC)56,60,60
1122      56 ZC=ZC+PI
1123      60 G(LL+12,K)=ZC*180/PI
1124      WRITE(6,790)
1125      JUMP=1
1126      WRITE(6,799)
1127      GO TO 70
1128      61 WRITE(6,791)
1129      JUMP=0
1130      LLA=13
1131      WRITE(6,799)
1132      GO TO 70

```

```

C ----- RESULTS PRINT OUT
1133 70 WRITE(6,792)
1134 WRITE(6,793)(LL,LL=1,LS)
1135 DO 80 <=1,MDIV
1136 LLB=LLA+LS-1
1137 80 WRITE(6,794)XFIX(K),YFIX(K),(G(LX,K),LX=LLA,LLB)
C ----- NCODE=3 INVOLVING INTEGRATIONS FOR FN. NOFN=1
C SPACE AVAILABLE FOR OTHER CASES DESIGNATED BY NOFN = 2,3
1138 IF(JJMP-1)20,61,20
1139 8 DO 15 <=1,MDIV
1140 XK=XFIX(K)
1141 YK=YFIX(K)
1142 BXK=BFIX(K)
1143 GO TO(11,12,13),NOFN
1144 11 IF(M3(J)-1)15,15,9
1145 9 C1=COS(3XK)
1146 S1=SIN(3XK)
1147 XKG=-XK*C1-YK*S1
1148 XKF=(S1*XK**3-C1*YK**3)/3.0
1149 WK=W(K)
1150 FW=1.0
1151 IF(J-1)18,18,16
1152 16 IF(NTY?(J)-3)18,17,18
1153 17 FW=-1.0
1154 18 WK=WK*0.5*A(J)*FW*NAXIS
1155 GO TO 13
1156 12 CONTINUE
1157 13 DO 14 LL=1,LS
1158 14 TOR(LL)=TOR(LL)+(G(LL,K)*XKG+XKF)*K
1159 15 CONTINUE
1160 IF(VHOLD-NNHOLD)20,10,20
1161 10 NNHOLD=0
C ----- PRINT OUT OF RESULTS FOR NCODE=3 AND NOFN=1.
1162 NHOLD=0
1163 WRITE(6,796)
1164 WRITE(6,797)
1165 WRITE(6,798)(TOR(LL),LL=1,LS)
1166 790 FORMAT(1H ,10X,'RESULTANT STRESSES AT SPECIFIED POINTS')
1167 791 FORMAT(1H ,4X,'DIRECTION OF RESULTANT STRESS WITH OX IN DEGREES')
1168 792 FORMAT(1H ,10X,'POINT CO-ORDS. (XK,YK) TRUNCATION LEVEL LL=1,LS')
1169 793 FORMAT(1H ,24X,4(5X,'LL=',I3))
1170 794 FORMAT(1H ,6(1X,E11.4))
1171 796 FORMAT(1H ,10X,'VALUE OF TORSIONAL RIGIDITY FOR TRUNC. LEVEL LL')
1172 797 FORMAT(1H ,10X,LL=1 LL=2 LL=3 LL=4')
1173 798 FORMAT(1H ,4(2X,E11.4))
1174 799 FORMAT(1H ,6X,'*****')
1175 20 RETURN
1176 END
C
C
C SUBROUTINE GAUSS FOR PT. LOCATIONS D(K) & WT. FACTORS W(K)
C
1177 SUBROUTINE GAUSS(N,NCODE)
1178 COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,10),
CG(99,100),SOL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(50),
C,XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),GMOD,DFIX(50),TOR(4)
1179 COMMON BFIX(50),ZE(11),F(4),COEF(2,100),PI,DELTA,XCAR(40),D(50)
1180 COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPRDG,NE,LS
1181 COMMON NROW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM

```

```

1182 COMMON NDK(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50)
1183 COMMON NFP,NZERO,NSS,NB,NOEL(10),LDC(50),NBDY,NFPOL
1184 IF(NCODE-3)57,54,57

```

```

C
C 24 POINT SUBDIVISION GAUSSIAN QUADRATURE ; NCODE = 3 .
C

```

```

1185 54 IF(N-24)57,55,57
1186 55 D(1) = .00240639
1187 D(2) = .01263572
1188 D(3) = .03086272
1189 D(4) = .05679223
1190 D(5) = .08999900
1191 D(6) = .12993790
1192 D(7) = .17595317
1193 D(8) = .22728926
1194 D(9) = .28310325
1195 D(10) = .34247866
1196 D(11) = .40444056
1197 D(12) = .46797156
1198 W(1) = .01234123
1199 W(2) = .02853139
1200 W(3) = .04427744
1201 W(4) = .05929858
1202 W(5) = .07334648
1203 W(6) = .08619016
1204 W(7) = .09761865
1205 W(8) = .10744427
1206 W(9) = .11550567
1207 W(10) = .12167047
1208 W(11) = .12583746
1209 W(12) = .12793820
1210 DD 56 < = 1,12
1211 <1 = 25-K
1212 D(<1) = 1.0-D(K)
1213 56 W(K1) = W(K)
1214 GO TO 52

```

```

C
C EQJAL SUBDIVISIONS FOR HARMONIC FITTING ; NCODE < 3.

```

```

1215 57 M = N-1
1216 Z1 = M
1217 Z1 = 1./Z1
1218 DD 58 < = 1,N
1219 Z2 = <-1
1220 D(<) = Z2*Z1
1221 W(<) = 2.0*Z1
1222 IF(K-1)59,59,60
1223 59 W(1) = Z1
1224 GO TO 58
1225 60 IF(<-N)58,61,61
1226 61 W(N) = Z1
1227 58 CONTINUE
1228 62 IF(N-50)901,901,902
1229 902 N = 50
1230 64 CONTINUE
1231 901 RETURN
1232 END

```

CENTRY

```

*****
*****

```

APPENDIX B(2) DATA PRINT OUT FOR NPROG = 7

 TORSION OF QUADRILATERAL PLATE WITH AN ELLIPTICAL HOLE
 ALL EDGES FULLY FIXED

LAP14880
 LAP14890

 L NS NB NPROG NBDY LS NMAT NPRIN MDIV DELTA FSET GMOD
 6 5 2 7 2 4 0 1 0 0.0000E 00 0.0000E 00 0.1000E 05

 DATA FOR NS = 5 SIDES

J	MB(J)	NHS(J)	NBY(J)	NTYP(J)	X(J)	Y(J)
1	2	1	0	0	0.0000E 00	0.0000E 00
2	2	1	0	0	0.1000E 01	0.0000E 00
3	2	1	0	0	0.1200E 01	0.1000E 01
4	2	1	0	0	-0.2000E 00	0.9000E 00
5	2	1	0	3	0.0000E 00	0.0000E 00

 SUPPLEMENTARY DATA FOR ELLIP./CIRC. BDRIES

AE(J)	BE(J)	XE(J)	YE(J)	ZE(J)
0.2000E 00	0.1500E 00	0.6000E 00	0.5000E 00	0.0000E 00

DETAILS OF NFP= 18 FIXED POINTS ASSIGNED IN COLMAT

XFIX	YFIX	BFIX	DFIX	JFIX	MBFIX	EQ NO.
0.0000E 00	0.0000E 00	0.0000E 00	0.0000E 00	1	2	66
-0.1000E-03	0.0000E 00	0.0000E 00	-0.1000E-03	1	12	67
0.1000E-03	0.0000E 00	0.0000E 00	0.1000E-03	1	12	67
0.0000E 00	0.0000E 00	0.4931E 01	0.1000E 01	4	2	68
0.1979E-04	-0.8923E-04	0.4931E 01	0.1000E 01	4	12	69
-0.2003E-04	0.9000E-04	0.4931E 01	0.9999E 00	4	12	69
0.1000E 01	0.0000E 00	0.1373E 01	0.0000E 00	2	2	70
0.1000E 01	-0.1000E-03	0.1373E 01	-0.1000E-03	2	12	71
0.1000E 01	0.1000E-03	0.1373E 01	0.1000E-03	2	12	71
0.1000E 01	0.0000E 00	0.0000E 00	0.1000E 01	1	2	72
0.1000E 01	0.0000E 00	0.0000E 00	0.1000E 01	1	12	73
0.9999E 00	0.0000E 00	0.0000E 00	0.9999E 00	1	12	73
0.1200E 01	0.1000E 01	0.3213E 01	0.0000E 00	3	2	74
0.1200E 01	0.1000E 01	0.3213E 01	-0.1000E-03	3	12	75
0.1200E 01	0.1000E 01	0.3213E 01	0.1000E-03	3	12	75
0.1200E 01	0.1000E 01	0.1373E 01	0.1000E 01	2	2	76
0.1200E 01	0.1000E 01	0.1373E 01	0.1000E 01	2	12	77
0.1200E 01	0.9999E 00	0.1373E 01	0.9999E 00	2	12	77
-0.2000E 00	0.9000E 00	0.4931E 01	0.0000E 00	4	2	78
-0.2000E 00	0.9001E 00	0.4931E 01	-0.1000E-03	4	12	79
-0.2000E 00	0.8999E 00	0.4931E 01	0.1000E-03	4	12	79
-0.2000E 00	0.9000E 00	0.3213E 01	0.1000E 01	3	2	80
-0.2001E 00	0.9000E 00	0.3213E 01	0.1000E 01	3	12	81
-0.1999E 00	0.9000E 00	0.3213E 01	0.9999E 00	3	12	81
0.1000E 01	0.0000E 00	0.0000E 00	0.0000E 00	10	1	82
0.0000E 00	0.0000E 00	0.0000E 00	0.0000E 00	10	3	83

 DETAILS OF LP= 54 SETS OF FNS ASSIGNED IN COLMAT

NFU	NA	E(NP)	NP
1	1	0.6283E 01	1
1	2	0.6161E 01	2

1	3	0.4477E	01	3
1	4	0.6815E	01	4
6	5	-0.1000E	01	5
1	1	0.1257E	02	6
1	2	0.1232E	02	7
1	3	0.8953E	01	8
1	4	0.1363E	02	9
6	5	-0.2000E	01	10
1	1	0.1885E	02	11
1	2	0.1848E	02	12
1	3	0.1343E	02	13
1	4	0.2045E	02	14
6	5	-0.3000E	01	15
1	1	0.2513E	02	16
1	2	0.2464E	02	17
1	3	0.1791E	02	18
1	4	0.2726E	02	19
6	5	-0.4000E	01	20
1	1	0.3142E	02	21
1	2	0.3081E	02	22
1	3	0.2238E	02	23
1	4	0.3408E	02	24
6	5	-0.5000E	01	25
1	1	0.3770E	02	26
1	2	0.3697E	02	27
1	3	0.2686E	02	28
1	4	0.4089E	02	29
6	5	-0.6000E	01	30
6	5	0.1000E	04	31
4	1	0.1756E	01	32
4	1	0.3511E	01	33
4	1	0.5257E	01	34
4	1	0.7022E	01	35
4	1	0.8778E	01	36
4	1	0.1053E	02	37
4	2	0.1777E	01	38
4	2	0.3553E	01	39
4	2	0.5330E	01	40
4	2	0.7107E	01	41
4	2	0.8884E	01	42
4	2	0.1066E	02	43
4	3	0.2413E	01	44
4	3	0.4825E	01	45
4	3	0.7238E	01	46
4	3	0.9651E	01	47
4	3	0.1206E	02	48
4	4	0.2207E	01	49
4	4	0.4414E	01	50
4	4	0.6621E	01	51
4	4	0.8828E	01	52
4	4	0.1104E	02	53
3	10	0.0000E	00	54

NO. OF EQUATIONS IN MATRIX NE = 83
 NO. OF EQUATIONS IN HARMONIC SET NH = 10
 NO. OF HARMONIC SETS L= 6

SOLUTION VECTORS

SOLUTIONS OF COEFS. MATRIX FOR NO OF SETS LS = 4
 THE DIFF. LEVELS OF TRJNC. ARE DENOTED BY LL=1,LS

COEF.	LL = 1	LL = 2	LL = 3	LL = 4
1	0.98202E 00	0.98790E 00	0.10667E 01	0.10816E 01
2	-0.12209E 01	-0.10006E 01	-0.78331E 00	-0.68970E 00
3	-0.95259E 00	-0.34239E 00	-0.81472E 00	-0.76961E 00
4	-0.13042E 01	-0.14352E 01	-0.16070E 01	-0.16660E 01
5	-0.59695E 00	0.18468E 01	0.35011E 01	0.37497E 01
6	-0.14145E 02	-0.12207E 02	-0.10846E 02	-0.98086E 01
7	0.12652E 01	0.57255E 00	-0.14702E 00	-0.49969E 00
8	-0.94909E 00	-0.62094E 00	-0.29889E 00	-0.94453E-01
9	0.80594E-02	0.78857E-02	0.79028E-02	0.79200E-02
10	-0.40240E-01	-0.40287E-01	-0.40262E-01	-0.40210E-01
11	0.40181E-02	0.69710E-02	0.11042E-01	0.12609E-01
12	-0.18461E-01	-0.21662E-01	-0.24232E-01	-0.25972E-01
13	0.25959E-02	0.78425E-02	0.11652E-01	0.14338E-01
14	-0.21136E-01	-0.22965E-01	-0.25235E-01	-0.26269E-01
15	-0.11504E-01	-0.17149E-01	-0.30149E-01	-0.45522E-01
16	-0.15835E 00	-0.16781E 00	-0.18283E 00	-0.18570E 00
17	0.35026E-02	-0.43402E-02	-0.12983E-01	-0.16568E-01
18	-0.12326E-01	-0.87582E-02	-0.47824E-02	-0.21340E-02
19	-0.91725E-02	-0.91372E-02	-0.91465E-02	-0.91467E-02
20	-0.29329E-02	-0.27659E-02	-0.27954E-02	-0.28060E-02
21	0.19763E-03	0.47228E-03	0.84903E-03	0.10161E-02
22	-0.25906E-02	-0.34893E-02	-0.42350E-02	-0.46625E-02
23	0.90407E-03	0.16583E-02	0.22809E-02	0.27398E-02
24	-0.21414E-02	-0.23650E-02	-0.25742E-02	-0.26869E-02
25	-0.19492E-02	-0.33051E-02	-0.51392E-02	-0.72371E-02
26	-0.10626E-01	-0.13178E-01	-0.16012E-01	-0.17088E-01
27	-0.91864E-04	-0.77083E-03	-0.15673E-02	-0.18437E-02
28	-0.16595E-02	-0.14351E-02	-0.11194E-02	-0.92495E-03
29	-0.83256E-02	-0.83142E-02	-0.82783E-02	-0.82797E-02
30	-0.13836E-01	-0.13881E-01	-0.13883E-01	-0.13877E-01
31	0.00000E 00	0.55856E-04	0.13347E-03	0.17534E-03
32	0.00000E 00	-0.10163E-02	-0.12482E-02	-0.13767E-02
33	0.00000E 00	0.33821E-03	0.50581E-03	0.63882E-03
34	0.00000E 00	-0.52475E-03	-0.57492E-03	-0.60342E-03
35	0.00000E 00	-0.65517E-03	-0.10631E-02	-0.15626E-02
36	0.00000E 00	-0.21357E-02	-0.28411E-02	-0.31415E-02
37	0.00000E 00	-0.15067E-03	-0.29666E-03	-0.33667E-03
38	0.00000E 00	-0.43145E-03	-0.38806E-03	-0.36739E-03
39	0.00000E 00	-0.13468E-02	-0.13431E-02	-0.13385E-02
40	0.00000E 00	-0.12292E-01	-0.12297E-01	-0.12299E-01
41	0.00000E 00	0.00000E 00	0.28043E-04	0.45787E-04
42	0.00000E 00	0.00000E 00	-0.44818E-03	-0.49395E-03
43	0.00000E 00	0.00000E 00	0.11167E-03	0.16515E-03
44	0.00000E 00	0.00000E 00	-0.18122E-03	-0.19122E-03
45	0.00000E 00	0.00000E 00	-0.23341E-03	-0.41334E-03
46	0.00000E 00	0.00000E 00	-0.71644E-03	-0.81336E-03
47	0.00000E 00	0.00000E 00	-0.59593E-04	-0.64913E-04
48	0.00000E 00	0.00000E 00	-0.15110E-03	-0.15033E-03
49	0.00000E 00	0.00000E 00	-0.23869E-02	-0.23825E-02

50	0.00000E 00	0.00000E 00	0.65030E-04	0.66308E-04
51	0.00000E 00	0.00000E 00	0.00000E 00	0.16098E-04
52	0.00000E 00	0.00000E 00	0.00000E 00	-0.16523E-03
53	0.00000E 00	0.00000E 00	0.00000E 00	0.40161E-04
54	0.00000E 00	0.00000E 00	0.00000E 00	-0.60772E-04
55	0.00000E 00	0.00000E 00	0.00000E 00	-0.12581E-03
56	0.00000E 00	0.00000E 00	0.00000E 00	-0.22240E-03
57	0.00000E 00	0.00000E 00	0.00000E 00	-0.23417E-05
58	0.00000E 00	0.00000E 00	0.00000E 00	-0.54019E-04
59	0.00000E 00	0.00000E 00	0.00000E 00	-0.18667E-03
60	0.00000E 00	0.00000E 00	0.00000E 00	-0.17865E-04
61	-0.48392E-05	-0.33006E-05	-0.37253E-08	0.17136E-05
62	-0.64967E 00	-0.53527E 00	-0.64365E 00	-0.65110E 00
63	0.11853E 01	0.12693E 01	0.12626E 01	0.11622E 01
64	-0.11175E 01	-0.12530E 01	-0.12635E 01	-0.11532E 01
65	0.18150E 01	0.18471E 01	0.18632E 01	0.17811E 01
66	-0.46143E 00	-0.48994E 00	-0.51034E 00	-0.49356E 00
67	0.12514E 00	0.13851E 00	0.15257E 00	0.15437E 00
68	-0.53071E 00	-0.57256E 00	-0.59676E 00	-0.61077E 00
69	0.74352E 01	0.78382E 01	0.83432E 01	0.81987E 01
70	-0.46564E 01	-0.31518E 01	-0.20572E 01	-0.16009E 01
71	-0.13993E 01	-0.62454E 00	0.10731E-02	0.26625E 00
72	-0.91244E 00	-0.67918E 00	-0.50376E 00	-0.40875E 00
73	-0.10359E 00	-0.60605E-01	-0.22254E-01	-0.77015E-03
74	0.80109E 01	0.72289E 01	0.68057E 01	0.64581E 01
75	-0.34598E 00	-0.20922E 00	-0.10339E 00	-0.84905E-01
76	0.12713E 01	0.11016E 01	0.97750E 00	0.89642E 00
77	-0.30128E-01	-0.29446E-01	-0.30445E-01	-0.32503E-01
78	0.26388E-01	0.19068E-01	0.12103E-01	0.85009E-02
79	-0.57160E 01	-0.50588E 01	-0.46934E 01	-0.44428E 01
80	-0.27428E 01	-0.25124E 01	-0.24300E 01	-0.23495E 01
81	-0.74386E 00	-0.64149E 00	-0.58405E 00	-0.54462E 00
82	-0.19363E 00	-0.18339E 00	-0.18572E 00	-0.18225E 00
83	0.00000E 00	0.00000E 00	0.00000E 00	0.00000E 00

PRINT OUT OF RESIDUALS AT MDIV= 29 POINTS ON SIDE J = 1 FOR TRUNC. LEVELS LL

NO.=N	LL = 1	LL = 2	LL = 3	LL = 4
1	0.5311E-04	-0.1498E-03	-0.1801E-03	-0.3495E-03
2	0.3940E-01	0.3444E-01	0.2900E-01	0.2486E-01
3	0.2619E-01	0.1105E-01	0.2396E-02	-0.9006E-04
4	-0.1436E-02	-0.1296E-01	-0.1234E-01	-0.8153E-02
5	-0.1903E-01	-0.1479E-01	-0.4116E-02	0.3517E-03
6	-0.1929E-01	-0.1478E-02	0.6689E-02	0.4152E-02
7	-0.6915E-02	0.1010E-01	0.5720E-02	0.4178E-04
8	0.7640E-02	0.9735E-02	-0.2372E-02	-0.2242E-02
9	0.1515E-01	0.2779E-03	-0.5403E-02	0.3151E-03
10	0.1228E-01	-0.7916E-02	-0.4861E-03	0.1844E-02
11	0.2326E-02	-0.7284E-02	0.4691E-02	-0.3593E-04
12	-0.8443E-02	0.2494E-03	0.2923E-02	-0.1450E-02
13	-0.1294E-01	0.7238E-02	-0.2528E-02	0.2944E-03
14	-0.9037E-02	0.6882E-02	-0.4076E-02	0.1494E-02
15	0.6198E-03	-0.1469E-03	0.3118E-03	-0.8970E-04
16	0.9760E-02	-0.7015E-02	0.4268E-02	-0.1429E-02
17	0.1309E-01	-0.6632E-02	0.2415E-02	0.3245E-03
18	0.7390E-02	0.5659E-03	-0.3007E-02	0.1733E-02
19	-0.2834E-02	0.7825E-02	-0.4193E-02	-0.8345E-04
20	-0.1246E-01	0.7623E-02	0.1032E-02	-0.1896E-02
21	-0.1426E-01	-0.6124E-03	0.5673E-02	0.3985E-03

22	-0.6174E-02	-0.9458E-02	0.2240E-02	0.2795E-02
23	0.8077E-02	-0.9071E-02	-0.5546E-02	-0.1926E-03
24	0.1941E-01	0.2506E-02	-0.5902E-02	-0.4252E-02
25	0.1798E-01	0.1465E-01	0.4647E-02	0.2989E-03
26	0.4478E-03	0.1196E-01	0.1191E-01	0.8770E-02
27	-0.2529E-01	-0.1069E-01	-0.2365E-02	0.8251E-04
28	-0.3690E-01	-0.3176E-01	-0.2668E-01	-0.2419E-01
29	0.4882E-03	0.7324E-03	0.9612E-03	0.1038E-02

ROOT MEAN SQUARE OF BOUNDARY RESIDUALS ON SIDE N=J FOR TRUNC. LEVELS LL=1,

NO.=N	LL = 1	LL = 2	LL = 3	LL = 4
1	0.1585E-01	0.1172E-01	0.8747E-02	0.6980E-02
2	0.9986E-02	0.5661E-02	0.3932E-02	0.3123E-02
3	0.1942E-01	0.9760E-02	0.6485E-02	0.4515E-02
4	0.1045E-01	0.5795E-02	0.3882E-02	0.3332E-02
5	0.4041E-02	0.3850E-03	0.4144E-04	0.4069E-04

PRODUCTION AT POINTS ON GIVEN LINE

DATA	U1	V1	U2	V2	BXK	MDIV	MT	J	NCD
	0.0000E 00	0.0000E 00	0.5000E 00	0.9500E 00	0.0000E 00	11	1	0	1

FUNCTIONS MT = 1 AT N=MDIV = 11PTS. ON LINE (U1,V1),(U2,V2) FOR TRUNC. LE

NO.=N	LL = 1	LL = 2	LL = 3	LL = 4
1	-0.5676E 00	-0.5677E 00	-0.5674E 00	-0.5671E 00
2	-0.5754E 00	-0.5755E 00	-0.5753E 00	-0.5752E 00
3	-0.5835E 00	-0.5896E 00	-0.5894E 00	-0.5894E 00
4	-0.6056E 00	-0.6058E 00	-0.6056E 00	-0.6056E 00
5	-0.6227E 00	-0.6229E 00	-0.6227E 00	-0.6227E 00
6	-0.6409E 00	-0.6411E 00	-0.6410E 00	-0.6410E 00
7	-0.6612E 00	-0.6614E 00	-0.6613E 00	-0.6613E 00
8	-0.6829E 00	-0.6830E 00	-0.6829E 00	-0.6829E 00
9	-0.7060E 00	-0.7061E 00	-0.7060E 00	-0.7060E 00
10	-0.7337E 00	-0.7336E 00	-0.7336E 00	-0.7336E 00
11	-0.7686E 00	-0.7691E 00	-0.7688E 00	-0.7690E 00

PRODUCTION AT POINTS ON GIVEN LINE

DATA	U1	V1	U2	V2	BXK	MDIV	MT	J	NCD
	0.0000E 00	0.0000E 00	0.5000E 00	0.9500E 00	0.0000E 00	11	2	0	1

FUNCTIONS MT = 2 AT N=MDIV = 11PTS. ON LINE (U1,V1),(U2,V2) FOR TRUNC. LE

NO.=N	LL = 1	LL = 2	LL = 3	LL = 4
1	0.5311E-04	-0.1498E-03	-0.1801E-03	-0.3495E-03
2	-0.2970E 00	-0.2969E 00	-0.2973E 00	-0.2977E 00
3	-0.3895E 00	-0.3898E 00	-0.3901E 00	-0.3904E 00
4	-0.4306E 00	-0.4309E 00	-0.4311E 00	-0.4313E 00
5	-0.4476E 00	-0.4479E 00	-0.4480E 00	-0.4482E 00
6	-0.4555E 00	-0.4556E 00	-0.4555E 00	-0.4558E 00
7	-0.4613E 00	-0.4612E 00	-0.4613E 00	-0.4614E 00
8	-0.4639E 00	-0.4641E 00	-0.4641E 00	-0.4642E 00
9	-0.4743E 00	-0.4735E 00	-0.4737E 00	-0.4738E 00
10	-0.5050E 00	-0.5012E 00	-0.5025E 00	-0.5021E 00
11	-0.5465E 00	-0.5558E 00	-0.5518E 00	-0.5542E 00

SHEAR LINES CHI=CONST. CHI=EPSI-0.5*(XK*XK YK*YK)

LAP14990

DATA	U1	V1	U2	V2	BXK	MDIV	MT	J	NCD
	0.0000E 00	0.0000E 00	0.5000E 00	0.9500E 00	0.0000E 00	11	3	0	2

POINT CO-ORDS. (X<,Y<) TRUNCATION LEVEL LL=1,LS

	LL = 1	LL = 2	LL = 3	LL = 4	
0.0000E 00	0.0000E 00	0.5722E-04	0.6104E-04	0.7534E-04	0.7153E-04
0.5000E-01	0.9500E-01	0.2181E-01	0.2227E-01	0.2244E-01	0.2247E-01
0.1000E 00	0.1900E 00	0.5765E-01	0.5807E-01	0.5821E-01	0.5823E-01
0.1500E 00	0.2850E 00	0.9274E-01	0.9318E-01	0.9331E-01	0.9333E-01
0.2000E 00	0.3800E 00	0.1211E 00	0.1216E 00	0.1217E 00	0.1217E 00
0.2500E 00	0.4750E 00	0.1387E 00	0.1393E 00	0.1394E 00	0.1394E 00
0.3000E 00	0.5700E 00	0.1419E 00	0.1424E 00	0.1426E 00	0.1426E 00
0.3500E 00	0.6550E 00	0.1276E 00	0.1283E 00	0.1285E 00	0.1285E 00
0.4000E 00	0.7600E 00	0.9699E-01	0.9777E-01	0.9789E-01	0.9792E-01
0.4500E 00	0.8550E 00	0.5123E-01	0.5180E-01	0.5197E-01	0.5200E-01
0.5000E 00	0.9500E 00	-0.9176E-02	-0.9931E-02	-0.9242E-02	-0.9396E-02

TORSIONAL RIGIDITY FOR TRUNC. LEVELS LL=1,LS

LAP15010

VALUE OF TORSIONAL RIGIDITY FOR TRUNC. LEVEL LL

LL=1	LL=2	LL=3	LL=4
0.1654E 00	0.1653E 00	0.1653E 00	0.1652E 00

RESULTANT STRESS & ITS DIRECTION

LAP15030

DATA	U1	V1	U2	V2	BXK	MDIV	MT	J	NCO
	0.0000E 00	0.0000E 00	0.5000E 00	0.9500E 00	0.0000E 00	11	0	0	2

RESULTANT STRESSES AT SPECIFIED POINTS

POINT CO-ORDS. (XK,YK) TRUNCATION LEVEL LL=1,LS

	LL = 1	LL = 2	LL = 3	LL = 4	
0.0000E 00	0.0000E 00	0.5626E-07	0.1989E-06	0.2118E-06	0.3186E-06
0.5000E-01	0.9500E-01	0.1981E-03	0.1935E-03	0.1984E-03	0.1985E-03
0.1000E 00	0.1900E 00	0.2241E-03	0.2243E-03	0.2243E-03	0.2245E-03
0.1500E 00	0.2850E 00	0.2051E-03	0.2056E-03	0.2056E-03	0.2057E-03
0.2000E 00	0.3800E 00	0.1654E-03	0.1658E-03	0.1659E-03	0.1660E-03
0.2500E 00	0.4750E 00	0.1246E-03	0.1248E-03	0.1249E-03	0.1250E-03
0.3000E 00	0.5700E 00	0.1160E-03	0.1160E-03	0.1161E-03	0.1162E-03
0.3500E 00	0.6650E 00	0.1628E-03	0.1628E-03	0.1629E-03	0.1629E-03
0.4000E 00	0.7600E 00	0.2457E-03	0.2454E-03	0.2456E-03	0.2455E-03
0.4500E 00	0.8550E 00	0.3417E-03	0.3440E-03	0.3436E-03	0.3438E-03
0.5000E 00	0.9500E 00	0.4369E-03	0.4545E-03	0.4471E-03	0.4505E-03

DIRECTION OF RESULTANT STRESS WITH OX IN DEGREES

POINT CO-ORDS. (XK,YK) TRUNCATION LEVEL LL=1,LS

	LL = 1	LL = 2	LL = 3	LL = 4	
0.0000E 00	0.0000E 00	0.1248E 03	0.1171E 03	0.1210E 03	0.1316E 03
0.5000E-01	0.9500E-01	0.1389E 03	0.1388E 03	0.1389E 03	0.1391E 03
0.1000E 00	0.1900E 00	0.1414E 03	0.1414E 03	0.1415E 03	0.1416E 03
0.1500E 00	0.2850E 00	0.1458E 03	0.1458E 03	0.1458E 03	0.1459E 03
0.2000E 00	0.3800E 00	0.1548E 03	0.1548E 03	0.1548E 03	0.1548E 03
0.2500E 00	0.4750E 00	0.1753E 03	0.1752E 03	0.1752E 03	0.1753E 03
0.3000E 00	0.5700E 00	0.3279E 02	0.3276E 02	0.3275E 02	0.3274E 02
0.3500E 00	0.6650E 00	0.6497E 02	0.6492E 02	0.6490E 02	0.6489E 02
0.4000E 00	0.7600E 00	0.7946E 02	0.7956E 02	0.7953E 02	0.7952E 02
0.4500E 00	0.8550E 00	0.8442E 02	0.8483E 02	0.8470E 02	0.8474E 02
0.5000E 00	0.9500E 00	0.8631E 02	0.8574E 02	0.8598E 02	0.8582E 02