Interaction Notes

Note 240

February 1975

Penetration of Electromagnetic Pulses Through Larger Apertures in Shielded Enclosures

R. Mittra
L. Wilson Pearson
Electromagnetics Laboratory
Department of Electrical Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

Abstract

The results of an initial investigation of the Singularity Expansion Method representation of the electromagnetic coupling through a rectangular aperture in a perfectly conducting sheet are reported. The problem is formulated in terms of the coupled Hallén-type integral equations for the dual problem of a rectangular plate. The integral equations are converted to a system of linear algebraic equations by way of the method of moments with subsectionally constant expansion functions and collocation testing. Several techniques used in minimizing the execution time of the computations are described. Some difficulties in accurately approximating the singularities of the system of integral equations by the singularities of the algebraic system are discussed. These difficulties arise because the subsectionally constant representation for the current cannot adequately represent the correct edge singularities in the currents on the plate. A set of pole trajectories indicative of the trends in pole location for the plate is reported. A listing of the pertinent computer code is provided.

This study was performed under subcontract to

The Dikewood Corporation
1009 Bradbury Drive, S.E.
University Research Park
Albuquerque, New Mexico 87106

and has been fully supported by the Defense Nuclear Agency (DNA) under

DNA Subtask EB088
EMP Interaction and Coupling
DNA Work Unit 33
Coupling Characteristics of Apertures

CONTENTS

ILLUSTRATIONS

TABLES

SECTION I

INTRODUCTION

This report presents the results of an investigation for representing the transient electromagnetic coupling through a rectangular aperture by means of the singularity expansion method.

The singularity expansion method, which was introduced by Baum in 1971 (ref. 1), has been subsequently applied to many scatterer geometries. The essence of the singularity expansion method is the representing of the temporal response of a body in terms of the complex natural frequencies for the body.

Taylor et al. point out that the singularity expansion for an aperture in an infinite perfectly conducting screen can be determined in terms of that for the complementary perfectly conducting plate by way of Babinet's principle (ref. 2). This approach was taken in the work reported here. The remaining discussion is directed to the equivalent problem of determining the current distributions on the complementary plate geometry.

Rahmat-Samii and Mittra have derived a coupled pair of Hallén-type integral equations governing the current behavior on the rectangular plate (ref. 3). The work reported here builds on their work by generalizing the integral equations and solution method to the complex frequency plane for the

---

1. Baum, C. E., "On the Singularity Expansion Method for the Solution of Electromagnetic Interaction Problems," Interaction Note 88, Air Force Weapons Laboratory, Kirtland AFB, NM, December 1971.

2. Taylor, C. D., Crow, T. T., and Chen, K-T, "On the Singularity Expansion Method Applied to Aperture Penetration: Part I Theory," Interaction Note 134, Air Force Weapons Laboratory, Kirtland AFB, NM, May 1973.

3. Rahmat-Samii, Y. and Mittra, R., "Integral Equation Solution and RCS Computation of a Thin Rectangular Plate," Interaction Note 156, Air Force Weapons Laboratory, Kirtland AFB, NM, December 1973.

SEM application. The same method-of-moments formulation, as described in (ref. 3), is used, i.e., two-dimensional pulse expansion functions with collocation testing.

In order that the computation time be practical in a problem of this complexity, a great deal of care was given to algorithmic streamlining in the conduct of this work. The streamlining includes maximum exploitation of geometric symmetry, organization of calculations to make use of redundant terms and partial terms occurring in the calculation, and direct algorithmic exploitation of matrix **sparseness.** The end result is a highly efficient computer code. Key features of the algorithms are discussed in this report.

The pulse expansion appears to be inadequate in accurately modeling the rectangular plate. The difficulty, which relates to representing the actual size of the plate, is demonstrated and discussed herein. Remedies for the problem are suggested, but they are outside the scope of the present work.

By holding the zoning scheme invariant while the aspect ratio of the plate was changed, self-consistent pole trajectories for the four fundamental modes were determined. For the reasons cited above, these poles cannot claim to be the exact poles for the body. They are, however, indicative of the trends in the pole behavior for the plate under change in aspect ratio. These results are reported and discussed in this context.

SECTION II

THIN-PLATE INTEGRAL EQUATION FORMULATION FOR COMPLEX WAVENUMBER

Rahmat-Samii and Mittra (ref. 3) give an integral equation formulation for the rectangular plate subject to time-harmonic excitation. Their results may be directly extended to the complex wavenumber case. That is, for the geometry in Figure 1 with exp[st] time dependence, $s = \sigma + j\omega$ complex, and an incident plane-wave magnetic field component

$\overline{H}^i = [H_{ox}^i \hat{u}_x + H_{oy}^i \hat{u}_y + H_{oz}^i \hat{u}_z] \exp[j(k_x x + k_y y + k_z z)]$ the following coupled integral equations result:

$$\int_{-L/2}^{L/2} \int_{-w/2}^{w/2} \begin{Bmatrix} J_x(x,y) \\ J_y(x,y) \end{Bmatrix} K(x,y|x',y')\, dx'\, dy' = \frac{j}{k_z} \begin{Bmatrix} Ho_g^i \\ -Ho_x^i \end{Bmatrix} \exp[j(k_x x + k_y y)]$$

$$+ \frac{\pi}{k} \begin{Bmatrix} -1 \\ -j \end{Bmatrix} \sum_{-\infty}^{\infty} C_n [j^{n+1} \exp[j(n+1)\phi]\, J_{n+1}(-s\rho/c)$$

$$+ \begin{Bmatrix} 1 \\ -1 \end{Bmatrix} j^{n-1} \exp[j(n-1)\phi]\, J_{n-1}(-s\rho/c)] \qquad (1)$$

The kernel is given by

$$K(x,y \mid x',y') = \exp[-sR/c]/R \qquad (2)$$

with

$$R = [(x - x')^2 + (y - y')^2]^{1/2}$$

The $J_x(x,y)$ and $J_y(x,y)$ denote the respective x and y components of current on the plate; $J_n(\zeta)$ denotes the Bessel function of the first kind; the C_n are unknown constants; c is the velocity of light; and (ρ,ϕ) are the polar coordinates for the point (x,y) on the plate. Equation (1) holds for $x \in (-L/2,L/2)$ and $y \in (-w/2,w/2)$, and $z = 0$.

It is pointed out that the two integral equations represented by (1) are

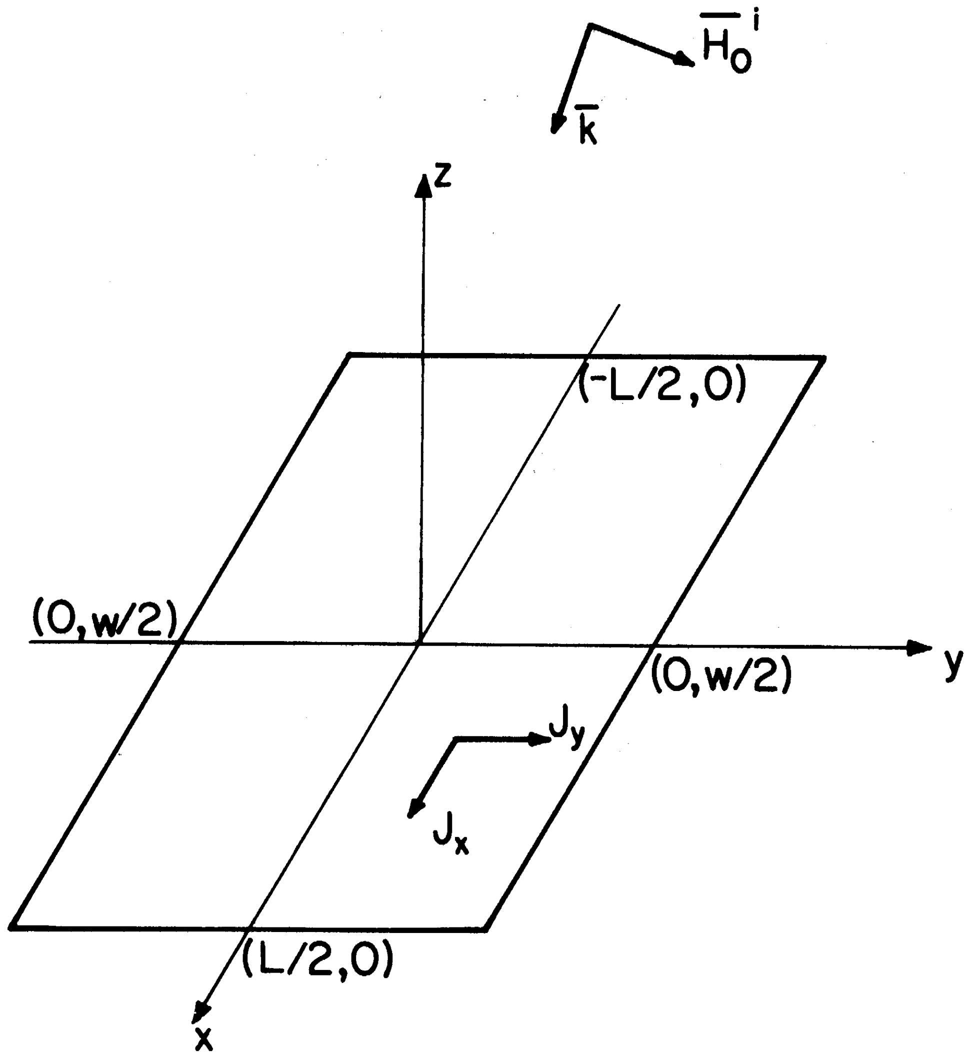


Figure 1. Geometry of the Rectangular Plate

coupled through the C_n in the summation in the right-hand side. This summation is simply a Bessel function expansion of the homogeneous solution to the wave equation which occurs in the derivation of (1). Details of arriving at this expansion are found in (ref. 3). The pair of integral equations (1) is complete in the sense that no additional constraints are needed to correctly specify the currents. It is noteworthy, however, that current solutions to (1) satisfy the Meixner's edge condition (ref. 4). Namely,

$$\left.\begin{array}{lcl} J_x[\pm(L/2 - d),\ y] & \rightarrow & d^{1/2} \\ J_y[\pm(L/2 - d),\ y] & \rightarrow & d^{-1/2} \\ J_x[x, \pm(w/2 - d)] & \rightarrow & d^{-1/2} \\ J_y[x, \pm(w/2 - d)] & \rightarrow & d^{1/2} \end{array}\right\} \quad d \rightarrow 0 \qquad (3)$$

The ability of a numerical solution to approximate the behavior of eqn. (3) is a key point in a subsequent discussion.

---

4. Bladel, J. Van, Electromagnetic Fields, McGraw-Hill, New York, pp. 385-387, 1964.

SECTION III

SYMMETRY CONDITIONS FOR THE NATURAL MODE CURRENTS

The natural frequencies of (1) occur when the complex frequency s is such that there are non-trivial J_x and J_y and the accompanying C_n satisfying (1) for $\overline{H}^i = 0$. Such J_x and J_y solutions are natural mode current solutions for the rectangular plate, and the concomitant value of s is a pole of the plate. The vanishing of incident wave dependence gives rise to symmetry in the integral equations. By discerning the symmetry relations *a priori* and bringing them to bear upon solution procedures, one gains significant computational savings in the numerical solution for poles and natural modes. These symmetry relations are explored in this section.

The excitation-free form of (1) is

$$\int_{-L/2}^{L/2}\int_{-w/2}^{w/2} J_x \; K(x,y|x',y')\; dx'\, dy' = \frac{j\pi c}{s}\sum_{-\infty}^{\infty} C_n \left\{ j^{n+1} \exp[j(n+1)\phi]\; J_{n+1}(-s\rho/c) + j^{n-1}\exp[j(n-1)\phi]\; J_{n-1}(-s\rho/c)\right\} \tag{4a}$$

and

$$\int_{-L/2}^{L/2}\int_{-w/2}^{w/2} J_y \; K(x,y|x',y')\; dx'\, dy' = \frac{\pi c}{s}\sum_{-\infty}^{\infty} C_n \left\{ j^{n+1} \exp[j(n+1)\phi]\; J_{n+1}(-s\rho/c) - j^{n-1}\exp[j(n-1)\phi]\; J_{n-1}(-s\rho/c)\right\} \tag{4b}$$

By using the symmetry of the Bessel function with respect to order, expanding the exponentials by way of Euler's identity, and appropriately adjusting the indices, one arrives at the following equation after some manipulation.

$$\int_{-L/2}^{L/2}\int_{-w/2}^{w/2} J_x \, K \, dx' \, dy'$$

$$= \frac{j\pi c}{s}\sum_{n=0}^{\infty}\Big\{ j^{n+1} d_n^+ [\cos(n+1)\phi \, J_{n-1}(-s\rho/c) - u_{n-1}\cos(n-1)\phi \, J_{n-1}(-s\rho/c)]$$

$$- j^n d_n^- [\sin(n+1)\phi \, J_{n+1}(-s\rho/c) - \sin(n-1)\phi \, J_{n-1}(-s\rho/c)]\Big\} \quad (5a)$$

and

$$\int_{-L/2}^{L/2}\int_{-w/2}^{w/2} J_y \, K \, dx' \, dy'$$

$$= \frac{j\pi c}{s}\sum_{n=0}^{\infty}\Big\{ j^{n+1} d_n^+ [\sin(n+1)\phi \, J_{n+1}(-s\rho/c) + \sin(n-1)\phi \, J_{n-1}(-s\rho/c)]$$

$$+ j^n d_n^- [\cos(n+1)\phi \, J_{n+1}(-s\rho/c) + u_{n-1}\cos(n-1)\phi \, J_{n-1}(-s\rho/c)]\Big\}$$

(5b)

where

$$d_n^{\pm} = C_n \pm C_{-n}$$

and

$$u_n = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

It is noted that the d_n^+ multiply terms containing cosine functions in the J_x equation, while they multiply terms containing sine functions in the J_y equation. The situation is reversed for the d_n^-.

Because of the symmetry properties of the kernel, the integral operator on the left-hand sides of (5) produces a function whose symmetry character is identical to that of the current on which it operates. Then, for a given current symmetry, only part of the $d_n^{\pm}$ on the right-hand side may be non-zero because of the symmetries possessed by the trigonometric terms. Thus, the respective symmetries for J_x and J_y, which are compatible, and the surviving terms in the right-side series may be discerned by 1) postulating a symmetry for J_x, 2) determining from (5a) which right-hand side terms survive so as to be compatible with the J_x symmetry, 3) observing in (5b) the variation which terms have non-zero coefficients, and 4) determining the J_y symmetry conditions compatible with the postulated J_x symmetry conditions.

For example, if J_x is symmetric with respect to the y axis and antisymmetric with respect to the x axis, only $\sin(n+1)\phi$ terms with n even are compatible in (5a). Thus, only d_n^-, n even, may be non-zero. In the right-hand side of (5b), the coefficients multiply $\cos(n+1)\phi$ terms with n even. These cosines sum to functions which are antisymmetric with respect to the y axis and symmetric with respect to the x axis. Stated mathematically, if

$$J_x(x,y) = J_x(-x,y) \tag{6a}$$

and

$$J_x(x,y) = -J_x(x,-y) \tag{6b}$$

then

$$d_n^+ = 0, \qquad \text{for all } n, \tag{6c}$$

$$d_n^- = 0, \qquad n \text{ odd}, \tag{6d}$$

and

$$J_y(x,y) = -J_y(-x,y) \tag{6e}$$

$$J_y(x,y) = J_y(x,-y) \tag{6f}$$

These vector symmetries are in accord with the general symmetry relations given by Baum (ref. 5). The information in (6) may be used to reduce the complexity of the integral equations (4), viz., by (6a,b,e,f) the range of each integration may be halved while by (6c,d) the zero terms of the right-hand side are known a priori:

$$\int_0^{L/2}\int_0^{w/2} J_x\, K^{-+}(x,y|x',y')\, dx'\, dy'$$

$$= \frac{\pi c}{s}\sum_{\substack{n=0\\ n\ \text{even}}}^{\infty} d_n^-\, j^{n-1}\left[\sin(n+1)\phi\, J_{n-1}(-s\rho/c) - \sin(n-1)\phi\, J_{n-1}(-s\rho/c)\right] \tag{7a}$$

---

5. Baum, C. E., "Interaction of Electromagnetic Fields with any Object which has an Electromagnetic Symmetry Plane," Interaction Note 63, Air Force Weapons Laboratory, Kirtland AFB, NM, March 1971.

and

$$\int_0^{L/2}\int_0^{w/2} J_y \, K^{+-}(x,y|x',y') \, dx' \, dy'$$

$$= \frac{\pi c}{s}\sum_{\substack{n=0\\ n \text{ even}}}^{\infty} j^{n+1} \, d_x^- \, [\cos(n+1)\phi \; J_{n+1}(-s\rho/c) + u_{n-1}\cos(n-1)\phi \; J_{n-1}(-s\rho/c)] \tag{7b}$$

where

$$K^{+-}(x,y|x',y') = K(x,y|x',y') - K(x,y|-x',y') + K(x,y|x',-y') - K(x,y|-x',-y') \tag{8a}$$

and

$$K^{-+}(x,y|x',y') = K(x,y|x',y') + K(x,y|-x',y') - K(x,y|x',-y') - K(x,y|-x',-y') \tag{8b}$$

For subsequent reference

$$K^{++}(x,y|x',y') = K(x,y|x',y') + K(x,y|-x',y') + K(x,y|x',-y') + K(x,y|-x',-y') \tag{8c}$$

and

$$K^{--}(x,y|x',y') = K(x,y|x',y') - K(x,y|-x',y') - K(x,y|x',-y') + K(x,y|-x',-y') \tag{8d}$$

are defined as well. Equations (7) are enforced for $z = 0$, $x \in (0,L/2)$ and $y \in (0,w/2)$.

Table 1 summarizes the four symmetry cases which are derived as in the foregoing discussion. By means of this table, four integral equation pairs can be constructed in the spirit of (7) by replacing the kernels in (7) with the appropriate kernels from the table and retaining only the non-vanishing terms in the series expansion.

Figure 2 depicts qualitatively the respective modal current distributions for the lowest frequency natural reasonance exhibiting each symmetry.

Table 1

COMPATIBLE CURRENT SYMMETRY FEATURES

J_x					J_y			
Sym. w.r.t. x axis	Sym. w.r.t. y axis	Kernel	Compatible Trig. Fns.	Coefs. ≠ 0	Compatible Trig. Fns.	Kernel	Sym. w.r.t. x axis	Sym. w.r.t. y axis
sym	sym	K^{++}	cos $2n\phi$	d^{+}_{2n+1}	sin $2n\phi$	K^{--}	anti	anti
sym	anti	K^{+-}	cos $(2n + 1)\phi$	d^{+}_{2n}	sin $(2n + 1)\phi$	K^{-+}	anti	sym
anti	sym	K^{-+}	sin $(2n + 1)\phi$	d^{-}_{2n}	cos $(2n + 1)\phi$	K^{+-}	sym	anti
anti	anti	K^{--}	sin $2n\phi$	d^{-}_{2n+1}	cos $2n\phi$	K^{++}	sym	sym

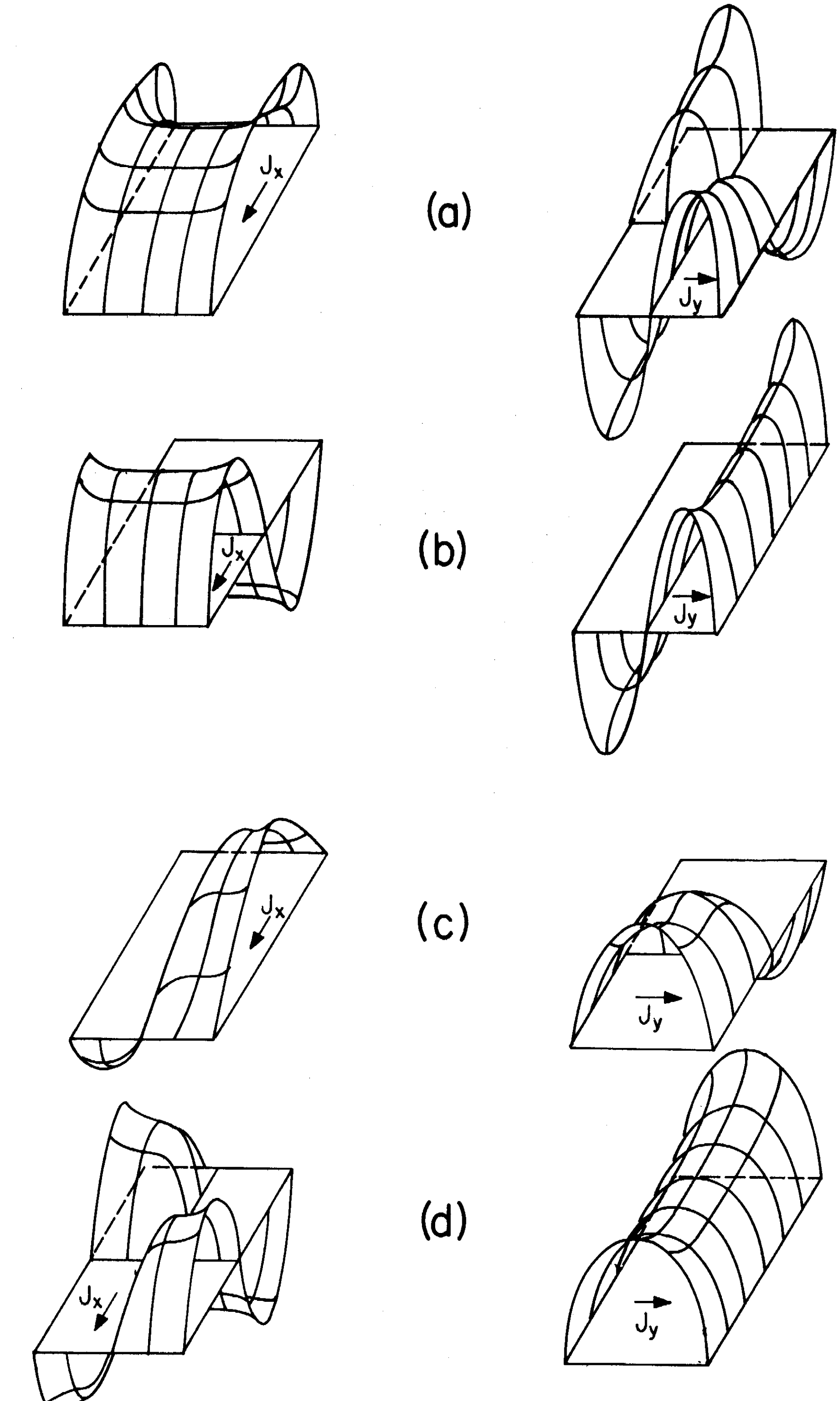


Figure 2. Lowest Order Natural Mode Current Pairs for Each of the Symmetry Cases, a) J_x Symmetric w.r.t. x-Axis and Symmetric w.r.t. y-Axis, b) Symmetric-Antisymmetric, c) Antisymmetric-Symmetric, and d) Antisymmetric-Antisymmetric

SECTION IV

THE NUMERICAL MODEL

The integral equation pair of the form (7) for each of the four symmetry cases can be discretized by the method of moments. In the work reported here, two-dimensional, subsectionally constant expansion functions were used with collocation testing. The zoning scheme is represented in Figure 3.

The unknown currents J_x and J_y were expanded in piecewise constant functions as in (ref. 3) with subsectioning of the form given in Figure 3. Notice that half-width patches are used at the edges of the plate so that match points lie precisely on the edge of the plate. The half-width pulse has proved useful in realizing the actual electrical size of a body in one-dimensional problems (ref. 6). Some numerical experimentation was also done with full-sized pulses on the edges and comparative results are reported in a later section. Some difficulties occur in definition of the edge of the plate in the present formulation because of the presence of two current components which have the asymptotic behavior given in (3). This difficulty is discussed in a later section.

The boundary condition $J_{norm} = 0$ must be enforced on selected patches at the edge of the plate as discussed in (ref. 3). Concomitantly, only as many $d_n^{\pm}$'s are retained in the right-hand side summation in (7) as there are current values preassigned to zero. The shaded patches in Figure 3 indicate the selection of patches where a current component is preassigned a zero value. At the corner patch, both components are preassigned zero values.

---

6. Butler, C. M., "Integral Equation Solution Methods," in "Wire Antennas and Scatterers," Short Course Notes, University of Mississippi, April 1972. (See also IEEE Trans., v. AP-20, pp. 731-736, 1972.)

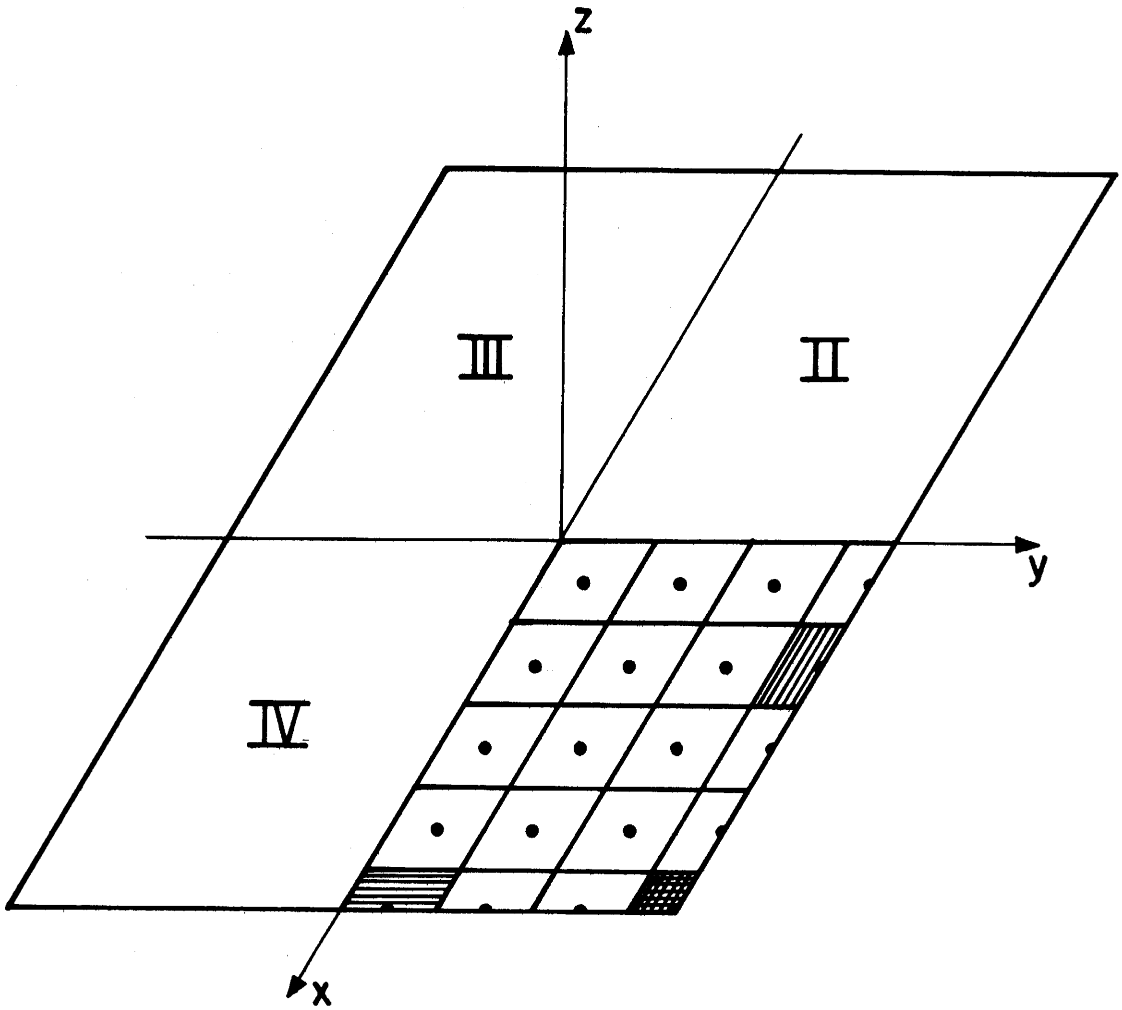


Figure 3. Subsectioning for the Discretization of the Integral Equations

By assigning one match point per expansion patch and by retaining one series expansion term for each current value preassigned in each of the two integral equations, a consistent (i.e. square) system of linear equations results. The truncated summation is taken to the left-hand side so that a homogeneous system results. The matrix organization used to represent these equations is given in Figure 4. Table 2 defines the computer variables noted in Figure 4, primarily for reference purposes in the next section.

The matrix that results is a function of the complex frequency s. A natural resonance occurs when s has a value such that the matrix has a zero determinant; hence, the homogeneous system of equations has a non-trivial solution. The next section explores some algorithmic considerations in the efficient generation and manipulation of the matrix.

$$\begin{bmatrix} \begin{bmatrix} M_x \\ (NI1 \times NJ1) \end{bmatrix} & \begin{bmatrix} 0 \\ (NI1 \times NJ2) \end{bmatrix} & \begin{bmatrix} M_\Sigma \\ (NI1 \\ + NI2 \\ \times NPRE) \end{bmatrix} \\ \begin{bmatrix} 0 \\ (NI2 \times NJ1) \end{bmatrix} & \begin{bmatrix} M_y \\ (NI2 \times NJ2) \end{bmatrix} & \end{bmatrix} \begin{bmatrix} \begin{bmatrix} J_x \\ (NJ1) \end{bmatrix} \\ \begin{bmatrix} J_y \\ (NJ2) \end{bmatrix} \\ \begin{bmatrix} d \\ (NPRE) \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix}$$

Figure 4. Organization of the System of Linear Equations

Table 2

MATRIX FORMULATION PARAMETERS

NI1	Number of match points on the zoned quadrant of the plate.
NI2 = NI1	
NPREJ	Number of patches along the $\|x\| = L/2$ edge where J_x is preassigned to zero.
NPREI	Number of patches along the $\|y\| = w/2$ edge where J_y is preassigned to zero.
NJ1 = NI1-NPREJ	Number of unknown current values in J_x expansion.
NJ2 = NI2-NPREI	Number of unknown current values in J_y expansion.
NJ3, NPRE } = NPREI-NPREJ	Number of preassigned current values (Also the number of coefficients retained in summation).

SECTION V

ALGORITHMIC CONSIDERATIONS IN EVALUATING THE SYSTEM DETERMINANT

Some considerations taken into account in generating the system matrix and evaluating its determinant efficiently are discussed in this section. Since these two operations must be repeatedly carried out for many values of s in the course of determining the natural frequencies of the plate, it is essential that clean, efficient computer programming and coding be used so that execution of the program will be affordable. The volume of code in the algorithms is consistently compromised toward a larger size in order to meet the following two time-efficient objectives:

1. Avoidance of calculating the same quantity twice; and
2. Avoidance of logical decisions, particularly those which might be imbedded in loops.

The program is discussed in the context of the following major segments:

1. Computation of an "interaction matrix";
2. Construction of the non-zero submatrices of the system matrix from the interaction matrix;
3. Computation of the series terms' submatrix; and
4. Determinant evaluation.

The major contribution to the elimination of redundant calculations is the one-time computation of an "interaction matrix" which is made up of the individual kernel integral terms from (2) for all argument combinations which occur in the computation. The subsequent program step then picks, by subscript, entries from this matrix and constructs the appropriate kernel from one of equations (8) according to the symmetry conditions being solved. This procedure can be viewed in terms of the layout given in Figure 5a. The terms in the interaction matrix are those evaluated for the match-point as

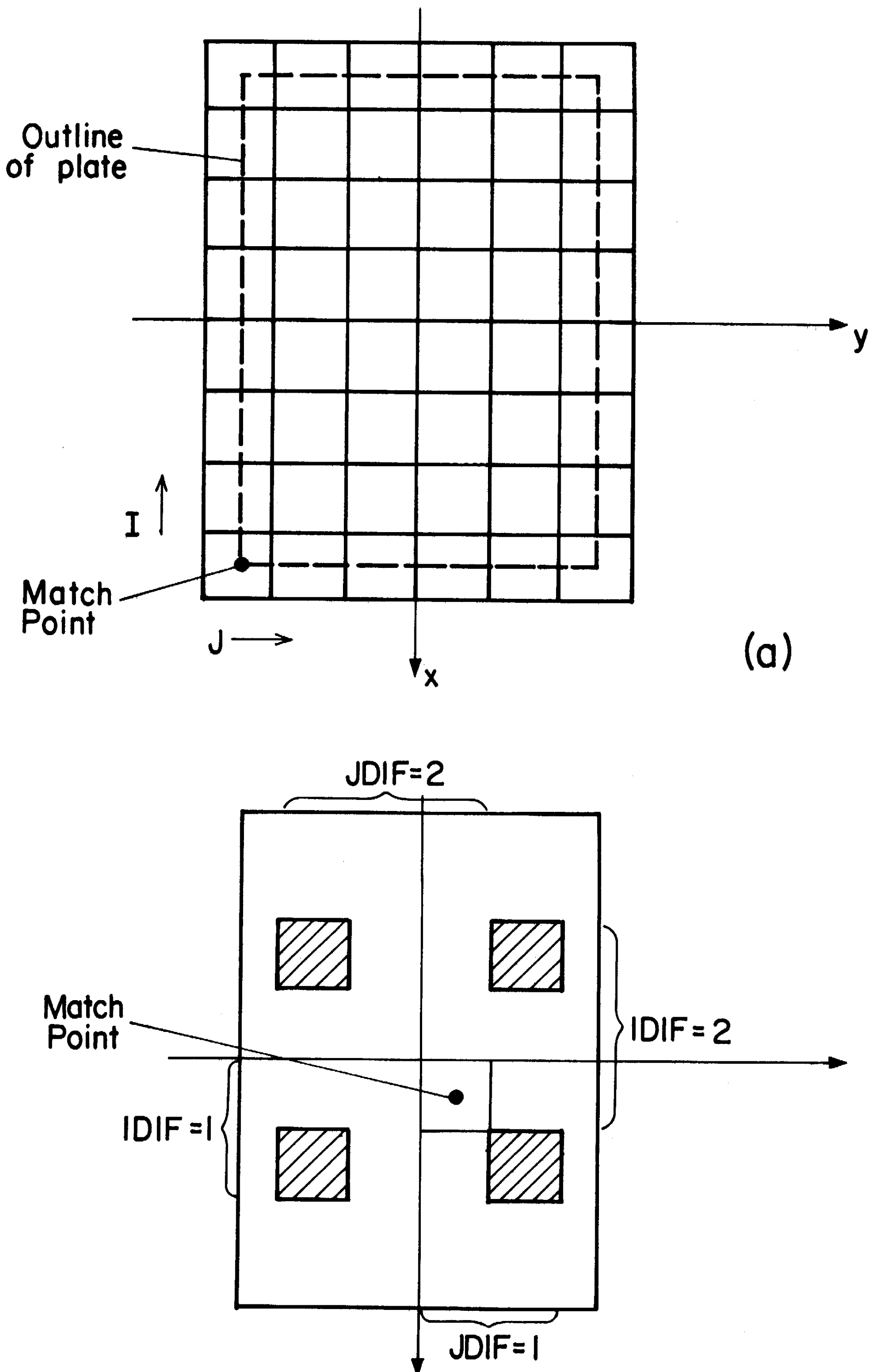


Figure 5. a) Conceptual Zoning for Calculation of the Interaction Matrix, b) Example of the Four Interaction Contributions to a Single Source Term

shown in the lower left with the source patches indexed over the entire plate to generate the matrix. Thus, all geometric relationships which occur in the kernel terms are encompassed in the calculation. Note that all source patches are full patches for this calculation. The effect of half patches at the edges is accounted for by weighting by a factor of 1/2 the edge contributions. The kernel integral appropriate to the symmetry is constructed by summing with correct signs the appropriate elements from the matrix. Figure 5b gives an example of the four source patches entering into one kernel integral.

Differing degrees of sophistication are required in the calculation of the interaction terms depending on the spacing of the patches for which an interaction is being calculated. For the self patch, i.e., the patch in which the match-point resides, the integration of the kernel must be performed analytically because of the integrable singularity in the kernel there. Appendix A gives a series approximation to this integral. The result in Appendix A is evaluated directly in the program. For the patches adjacent to the patch containing the match point, the kernel is a rapidly varying but well-behaved function. The integration over these patches is evaluated numerically by a polynomial approximation. For patches further separated, the kernel is slowly varying and the integral is evaluated approximately as the product of the value of the kernel at the center of the patch and the area of the patch.

Some minor time economy is achieved in the matrix filling algorithm, which is a four-dimensional loop. The economy is found in the form of decision-free indexing, that is, the source contributions from interior patches, from $|x| = L/2$ edge patches, from $|y| = w/2$ edge patches, and from corners take on different forms. Rather than index over all source patches with logical decisions implemented to discriminate among the four cases above, four different loops are used.

The computation of the series term submatrix is relatively straightforward. Because the Bessel-trigonometric products appear in two terms each, they are all precalculated and stored in a vector. The individual terms are then constructed from them.

The determinant evaluation profits significantly from an exploitation of the sparceness of the matrix. Either of two approaches may be taken to the sparce matrix manipulations. One is to separate the matrix algebraically and calculate an inverse as a composite of inverses of terms involving the submatrices. The alternative approach is to attack the matrix directly with Gaussian elimination, and to exploit the sparceness directly in the algorithm. The latter approach was chosen for the present purpose because it is judged to be slightly faster computationally and because in order to determine natural mode solutions for the SEM formulation, the homogeneous system of equations occurring at a pole must be backsolved. The algorithm resulting from the second approach is described in Appendix B.

The determinant evaluation routine itself appears in Appendix C as the function routine CPLATE.

SECTION VI

NUMERICAL CHECKS ON THE ACCURACY OF THE POLES

The results of some numerical checks on the accuracy of the pole location as determined from the numerical model described in Sections II through V are reported. It is shown that the model predicts well the poles for narrow strips possessing essentially thin scatterer resonance properties. Difficulties occur, however, in obtaining self-consistent results under different zone sizes for plates with larger aspect ratios. It is conjectured that the difficulty occurs because the subsectionally constant current representation is inadequate to represent the correct edge behavior for the currents-particularly the singular behavior for the parallel component. The rationale behind this conjecture is discussed.

Initial tests on the accuracy of the model were made for a rectangular strip with a shape ratio $w/L = 1/10$. Such a strip has an approximate equivalent dipole whose diameter-to-length ratio is $1/10\pi$.

Figure 6 gives the results of pole determinations for the first two poles for various numbers of pulses in the expansion of the current and for two different treatments of the edge pulse. The strip was zoned with one pulse across a quadrant. The numbers indicated in the figure are the numbers of pulses along the longitudinal direction of a quadrant. The "half-pulse" results are those obtained by the zone scheme described in Section IV where half-width pulses are placed at the edge so that match points fall at the edge. The "full-pulse" results are those obtained by zoning the plate with equal-sized pulses over the entire quadrant. In the latter case an <u>a posteriori</u> adjustment is made in the data correcting the length of the strip such that the end of the corrected strip lies at the end match-point.

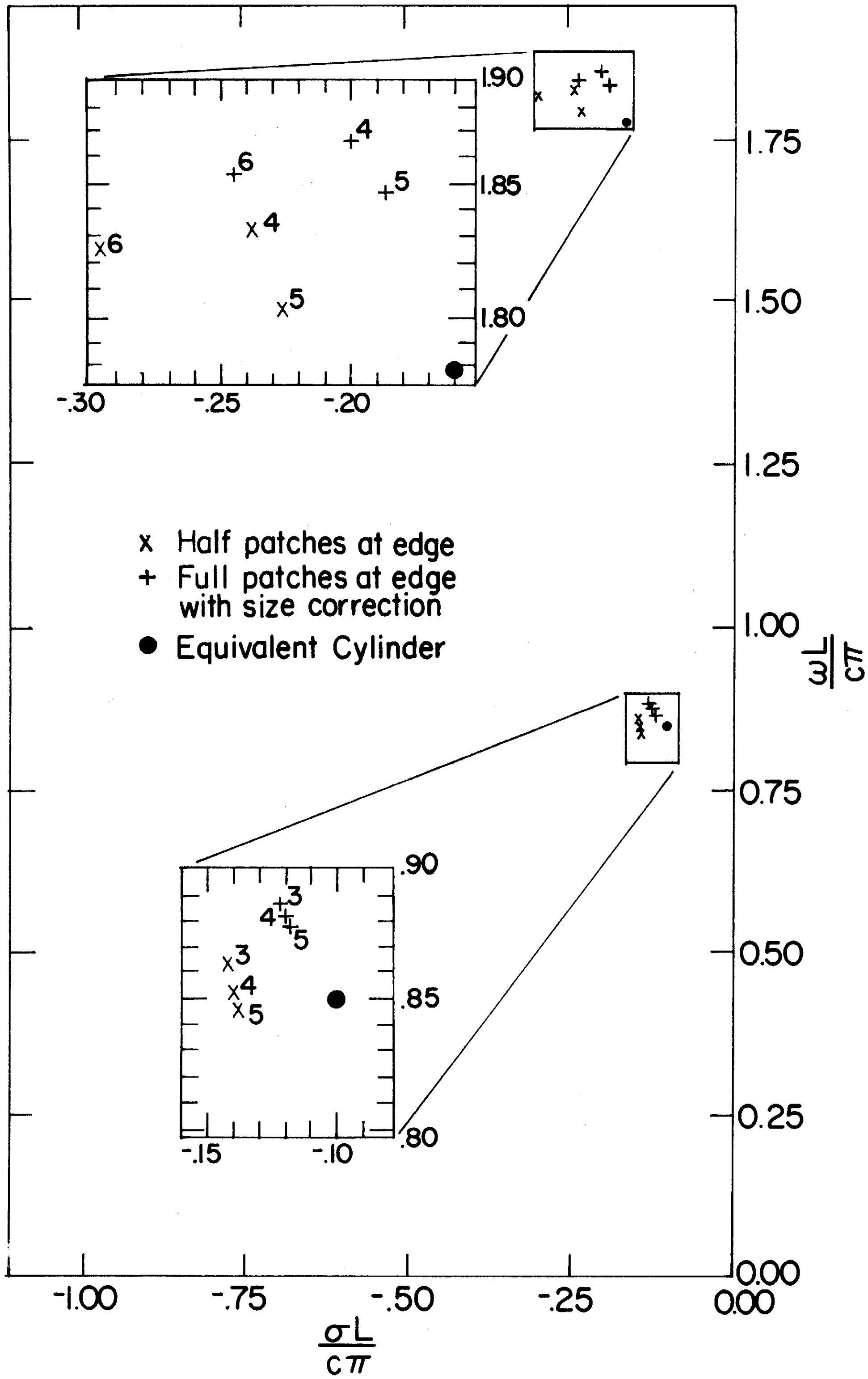


Figure 6. Calculated Pole Locations for Thin-Strip for Varying Numbers of Zones in the x-Direction and Different Edge Treatments (Cylinder Results from Ref. 6)

It is seen that the differences are small both for varying order and increasing pulse density. The NX = 6 results for the second pole show some departure from the trend established by the results for NX = 4 and NX = 5. This is attributable to the fact that the matrix is on the brink of numerical instability for NX = 6. The results for NX = 7, which are not shown, are observed to be meaningless because of the instability manifested.

For comparison purposes, the first two poles for an equivalent cylinder (one whose circumference equals the strip width) are given as found in ref. 7. These results are judged reliable inasmuch as they have been cross-checked among several integral equation formulations and for several method-of-moments schemes. The equivalent radius taken is, of course, an approximation. It is seen that the half-pulse solutions compare slightly more favorably with the cylinder results. Because of this, and moreover, because the a posteriori data adjustment is avoided with the half-pulse scheme, it was used consistently in the remaining solutions.

The stable convergence properties of the numerical model exhibited for the thin-strip solution are not manifested for higher aspect ratios. The reason for the difference is that the strip is essentially a one-dimensional problem and the transverse (y-directed) component of current has little effect on the dominant longitudinal current. For wider structures the coupling is significant and inadequacies in the modeling of the singularities of the currents produce inaccuracies which are highly sensitive to zoning.

Figure 7 shows the results obtained for a pole trajectory as a function of the shape factor w/L where the zoning in the y-direction was adjusted

---

7. Umashankar, K. R., "Transient Scattering by a Thin Wire in Free Space and Above Ground Plane Using the Singularity Expansion Method," Interaction Note 236, August 1974. (See also F. M. Tesche, Interaction Note 102, April 1972.)

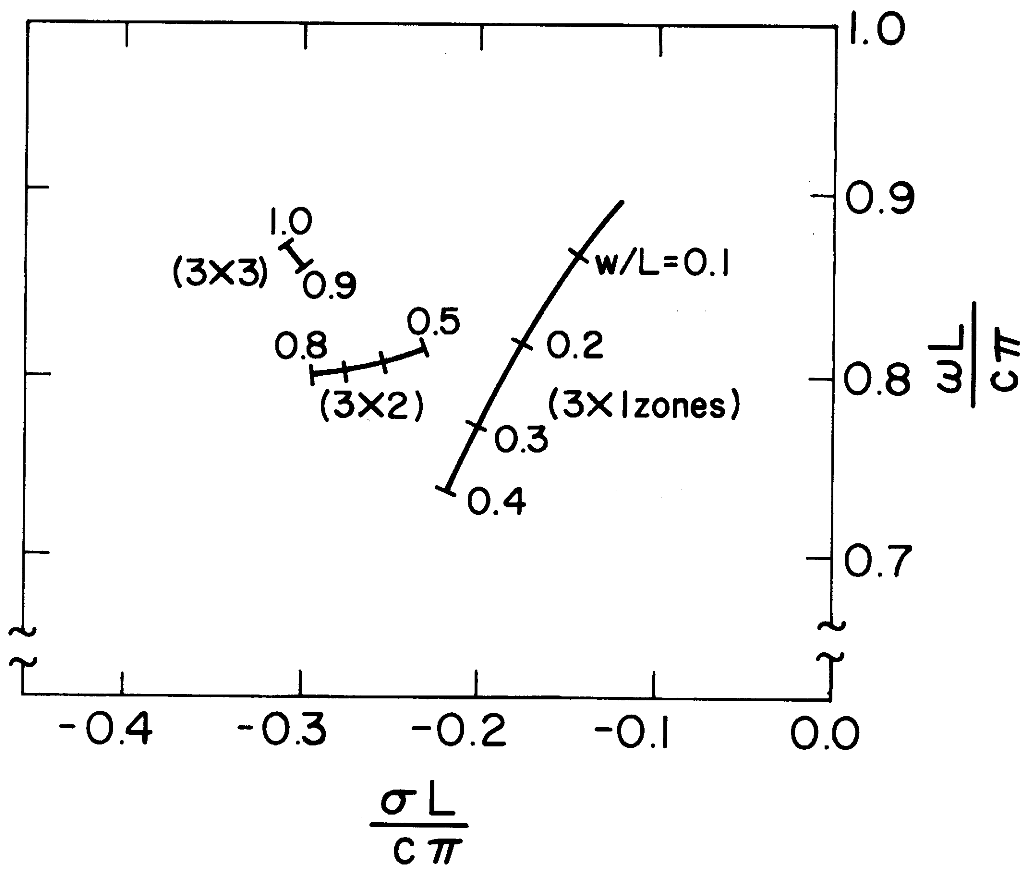


Figure 7. Computer Pole Trajectory Under Varying w/L with Zoning Changes

according to the value of w. It is evident that the solutions are unstable with respect to the zoning on the plate. Attempts to increase the number of zones significantly to improve upon the situation resulted in numerical instabilities in the matrix which cause the pole search iteration to fail.

The reason for the difficulty manifested in Figure 6 is believed to lie in the way that the edge of the plate is defined with the piecewise constant current expansion. Consider the characteristics of the two current components along a line traversing the plate in the y-direction as shown in Figure 8. The correct edge behavior at $|y| = w/2$ is that given in equations (3). The zoning scheme, however, forces $J_x(x,\pm w/2)$ to take a finite value. The current extrapolates to a singular point for some $y > w/2$, i.e., the numerical model represents a plate whose width is greater than w.

If the number of transverse zones is increased as indicated by the dashed curve in Figure 8, the steepness of the edge behavior of J_x is increased, and the extrapolation is characteristic of a narrower plate as compared to the first case. This narrowing of the effective width in the model is reflected in an increased Q (resonance quality factor) as the jumps in Figure 7 indicate.

One is tempted to conclude that a full-width pulse at the edge is a cure for the problem since the point at which the pulse current is defined is shifted relative to the edge as zoning is changed with full-width pulses. The effect of this procedure is to transfer the problem from component of current whose behavior is singular at the edge to the component which goes to zero. With full pulses at the edges, the normal component of current would go to zero interior to the plate rather than at the edge as it properly should.

An approach which is potentially a remedy for this difficulty is discussed in the conclusions.

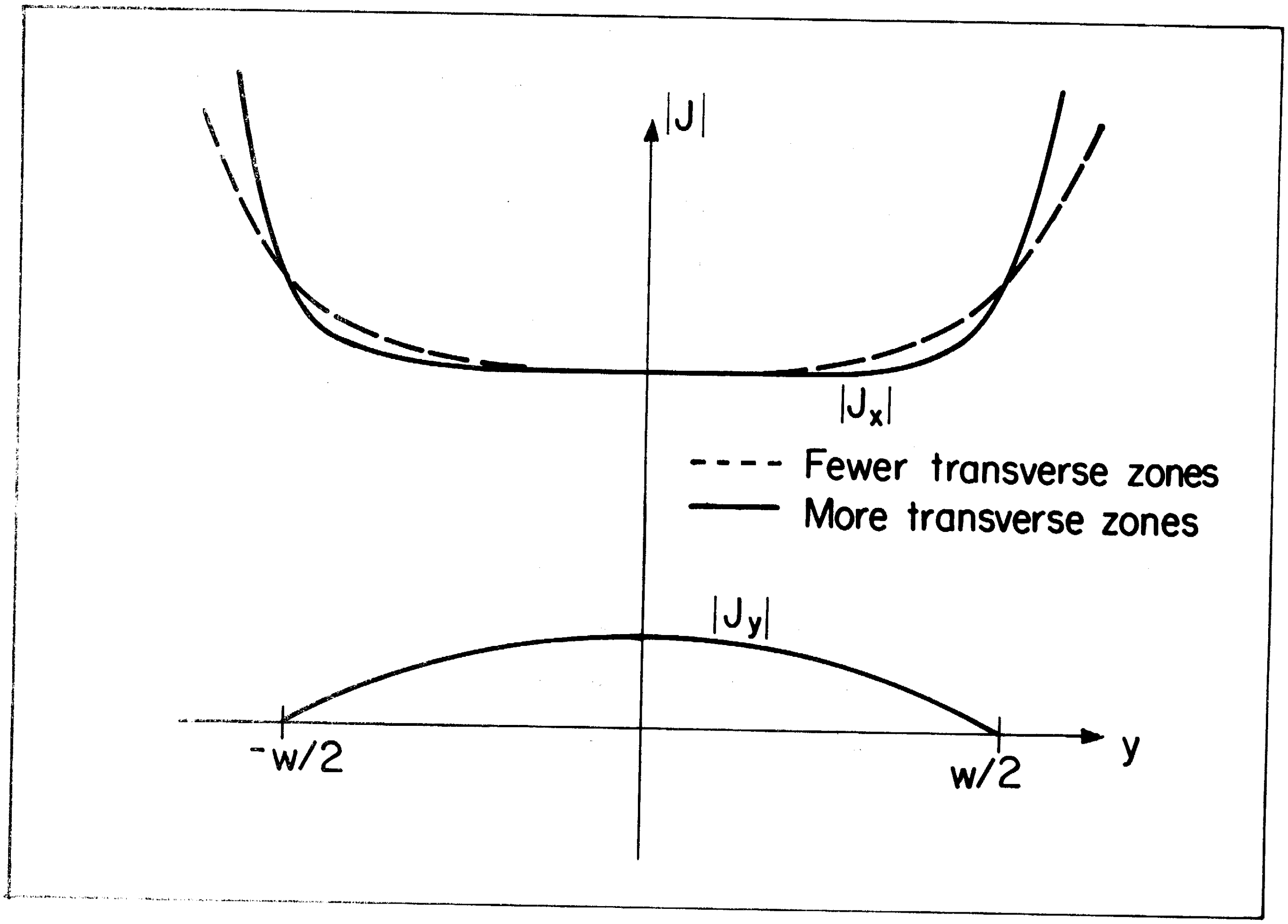


Figure 8. Behavior of Singular Component of Current at the Edge Under Change in Transverse Zoning

SECTION VII

POLE TRAJECTORIES AS A FUNCTION OF SHAPE RATIO

Figure 9 gives the results obtained for pole location for the lowest order pole of each of the symmetries as a function of w/L. Clearly, as the previous section indicates, the results cannot be taken as the correct results for the plate. However, the zoning was fixed for all w/L and the results are expected to reflect the proper trends for these trajectories.

The ++ and +- modes are in essence dipole modes, and their poles show the general lowering of Q as w/L increases. (The ++ indicates that the J_x component is symmetric both with respect to the x and y axes - see Table I.) The -- mode can be thought of as a dipole mode in the transverse direction. As a result it shows a strong frequency dependence on the transverse dimension w. When w/L = 1, the -- mode is identical to the ++ mode rotated spatially 90 degrees. Consequently, the two trajectories coalesce as $w/L \to 1$, when the zoning is 5x5 so as to preserve symmetry in the numerical mode. The failure of the 5x3 zone case is due to the reasons outlined in the previous section.

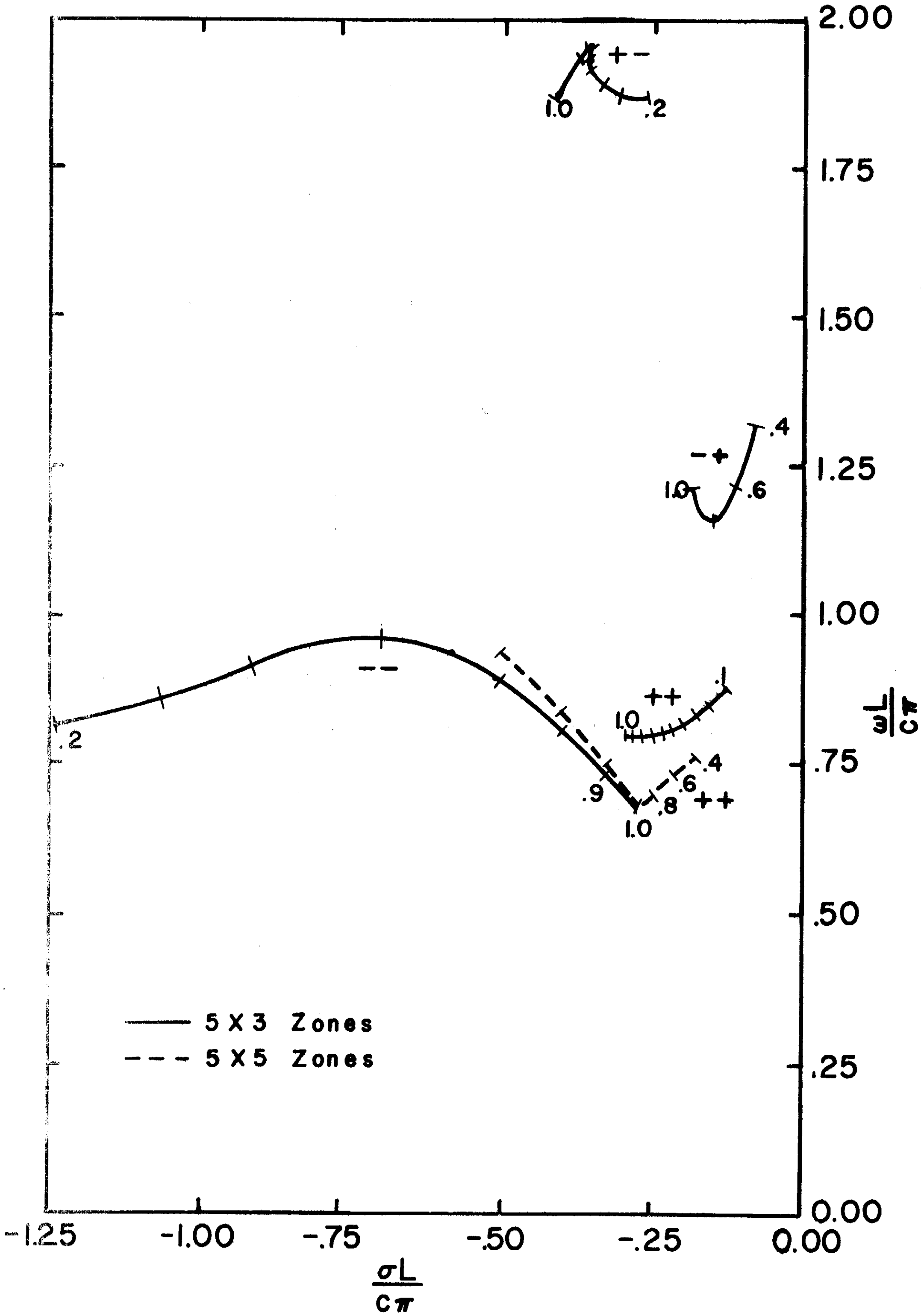


Figure 9. Pole Trajectories as Computed with Zoning Fixed

SECTION VIII

CONCLUSIONS

The application of SEM to the equivalent problems of the perfectly conducting rectangular plate and the rectangular aperture in a perfectly conducting screen has been conducted with partial success. The applicability of SEM and the computational feasibility of determining SEM quantities are demonstrated. It is further demonstrated that by careful program construction, the computational costs of numerical treatment of two-dimensional problems can be made quite reasonable. The cost of generating a matrix and calculating its determinant by the methods discussed herein is less than 10 cents for each value of s.

Difficulties arise in the subsectionally constant current zoning because of a failure to properly model the edge conditions. Whereas Rahmat-Samii and Mittra (ref. 3) obtained good radar cross-section results with such a zoning scheme, the pole locations are highly sensitive to the zoning. Radar cross-section is a far-field quantity, and the integrated effects of the errors are small there. The pole locations, on the other hand, depend strongly on the dimensions of the structure, and it is evident that the plate size must be brought to bear in a precise fashion for the pole locations to be correct.

The edge problem can be remedied by imposing the edge conditions (3) directly in the basis set elements for edge zones.
Davis has done this successfully for the circumferential component of current on a thick cylindrical scatterer (ref. 8). The circumferential current

---

8. Davis, W. A., "Numerical Solutions to the Problem of Electromagnetic Radiation and Scattering by a Finite Cylinder," Ph.D. Thesis, University of Illinois, 1974.

component is singular at the ends of the cylinder. The introduction of the singular basis element will produce a significant complication to the problem in that a second singularity, that of the current, must be integrated analytically in order to implement the model with edge conditions imposed. An independent check on program accuracy is dictated for a problem of this scope before proceeding with the edge condition approach.

APPENDIX A

THE SELF-PATCH INTEGRATION

The term for the interaction matrix for IDIF = JDIF = 0, i.e., where the match point lies at the center of the source patch, can be written

$$I_s = 4 \int_0^{\Delta x/2} \int_0^{\Delta y/2} K(0,0|x',y')\, dx'\, dy' \tag{A1}$$

This presumes a unit amplitude expansion pulse over the patch whose dimensions are Δx and Δy. The symmetry of the kernel with respect to both x' and y' is employed in the forming of (A1). This integral can be transformed to polar coordinates as

$$I_s = 4 \left\{ \int_{\phi=0}^{\tan^{-1}\frac{\Delta y}{\Delta x}} \int_{\rho=0}^{\frac{\Delta x}{2\cos\phi}} \exp[-s\rho/c]\, d\rho\, d\phi \right.$$

$$\left. + \int_{\phi=\tan^{-1}\frac{\Delta y}{\Delta x}}^{\pi/2} \int_{\rho=0}^{\frac{\Delta x}{2\cos\phi}} \exp[-s\rho/c]\, d\rho\, d\phi \right\}$$

$$= -\frac{4c}{s} \left\{ \int_{\phi=0}^{\tan^{-1}\frac{\Delta y}{\Delta x}} [\exp(-s\Delta x \sec\phi/2c) - 1]\, d\phi \right.$$

$$\left. + \int_{\phi=\tan^{-1}\frac{\Delta y}{\Delta x}}^{\pi/2} [\exp(-s\Delta y \csc\phi/2c) - 1]\, d\phi \right\} \tag{A2}$$

If the exponential functions in the integrand are then expanded in McLauren series, the resulting terms of powers of secants and cosecants possess tabulated integrals. The result for three terms retained in the series is

$$I_s \simeq -\frac{4c}{s}\left\{-\frac{s\Delta x}{2c}\cdot 1/2\ \ln q_y + 1/2\left(\frac{s\Delta x}{2c}\right)^2\frac{\Delta y}{\Delta x}\right.$$

$$-\frac{1}{6}\left(\frac{s\Delta x}{2c}\right)^3\frac{\Delta x(\Delta x^2+\Delta y^2)^{1/2}}{2\Delta y^2} - \frac{s\Delta y}{2c}\ 1/2\ \ln q_x$$

$$\left. + 1/2\left(\frac{s\Delta y}{2c}\right)^2\frac{\Delta x}{\Delta y} - 1/6\left(\frac{s\Delta y}{2c}\right)^3\frac{\Delta y(\Delta x^2+\Delta y^2)^{1/2}}{2\Delta y^2}\right\} \tag{A3}$$

where

$$q_{\binom{x}{y}} = \frac{\left[(\Delta x^2+\Delta y^2)^{1/2} + \binom{\Delta x}{\Delta y}\right]}{\left[(\Delta x^2+\Delta y^2)^{1/2} - \binom{\Delta x}{\Delta y}\right]}$$

APPENDIX B

THE SPARSE MATRIX ALGORITHMS

1. Introduction

This Appendix describes the algorithmic approach to minimize the computation time involved in Gaussian elimination triangularization of systems of matrix equations which are "sparsely coupled." The term "sparsely coupled" as applied in this Appendix implies the matrix equation form given in (B1).

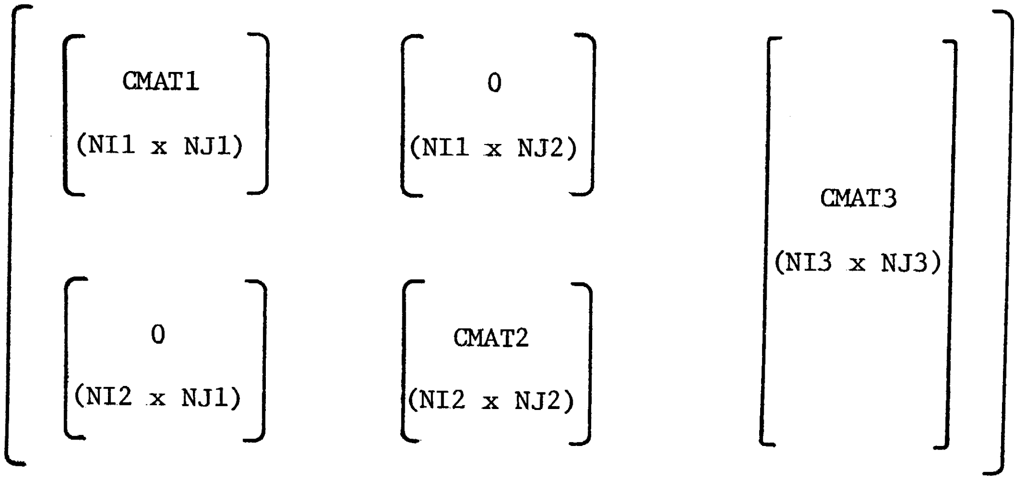
$$[M]\ [X] = \begin{bmatrix} M_1 & 0 & \\ & & M_3 \\ 0 & M_2 & \end{bmatrix} \begin{bmatrix} \\ X \\ \\ \end{bmatrix} = \begin{bmatrix} \\ B \\ \\ \end{bmatrix} \qquad \text{(B1)}$$

It is clear that in this form the sole coupling between the "upper" and "lower" systems of equations is contained in the matrix M_2. Generally, the number of columns in M_2 is small compared with the order of the overall system.

An algebraic approach to exploiting the sparceness of (B1) results in solutions which are given in terms of the several submatrices and their inverses. (See, for example, ref. 9.) It is well-known, however, that it is sufficient for the purposes of determinant calculation and equation solution to triangularize the composite matrix in (B1). The triangularization process involves fewer operations than the diagonalization necessary for the calculation of an inverse.

---

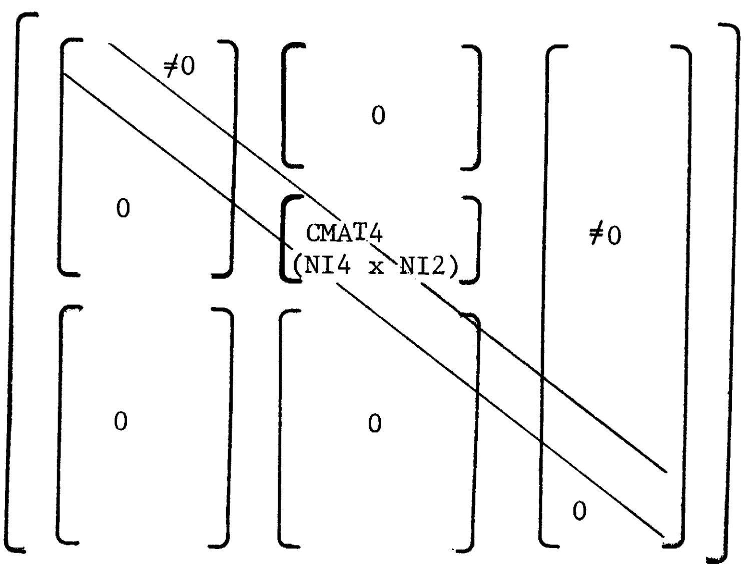
9. Dunaway, O. C., "A Numerical Solution for the Distribution of Time-Harmonic Electromagnetic Fields in an Arbitrary Shaped Aperture in a Ground Screen," M.S. Thesis, University of Mississippi, 1974.



NI3 = NI1 + NI2

NJ3 = NI3 - NJ1 - NJ2

(a)



NI4 = MAX (NI1 - NJ1,0)

(b)

Figure B1. Submatrix Organization for the Sparse Matrix Algorithms, a) the Original Matrix, and b) Triangularized Form with the Generated CMAT4

This Appendix describes an algorithmic exploitation of the sparseness of the composite matrix in (B1). That is, a numerical process is given whereby only the non-zero subelements are stored and operated on, with the computational operations being those which render the composite matrix M upper triangular. The determinant of the composite matrix results directly from this triangularization. A solution for X in (B1) requires a backsolving process involving the triangularized form of M and a vector resulting from applying the elimination operations to the vector B Routines to perform these operations are given as well.

Appendix C gives listings of the routines built on this algorithm. The triangularization routine is named SPRHOM. The backsolving procedure is performed by the entry HOMSLV to the routine SPRSLV. (The name entry SPRSLV backsolves an inhomogeneous system and is not used for present purposes.)

2. The Algorithm

The routine SPRHOM is simply an implementation of the Gaussian elimination procedure with maximum pivot selection applied to the composite matrix M in (B1) viewed as a whole. The sparseness of M is exploited by storing only the non-zero submatrices in (B1) and skipping the operations involving zero elements. The result is a substantial saving in both storage and computation time.

The forms of the input and of the end product for the triangularization are given in Figure (B1) with the sizes of the respective submatrices defined. It is recalled that the fundamental process in the Gaussian elimination procedure is the elimination of all or part of the elements of a column of a matrix with respect to a pivot element, commonly the element of greatest magnitude in the column. That is, for a column under process, the row containing the main diagonal element of the matrix which falls in that column. All or part of the elements not on the main diagonal are "eliminated" or made zero by subtraction of some multiple of the row containing the column maximum. In the triangularization procedure, the part of the column comprising elements lying below the main diagonal after row exchange are eliminated. If the matrix is a part of a system of equations with non-zero right-hand side, the row operations of exchange and subtraction of a constant multiple of another row must be performed on the corresponding elements of the right-hand side vector as well.

The algorithm of the routine SPRHOM operates according to the Gaussian elimination procedure described above. However, the three submatrices CMAT1, CMAT2, and CMAT3 are stored individually. In addition, the routine generates a submatrix CMAT4 in the course of selecting pivots for the columns contained in CMAT2. Further, the "elimination" of elements of submatrices that are zero *a priori* is skipped. The result is significant storage and time economy.

In order to minimize logic decisions in the routine, it is organized to operate sequentially through the partitioned matrix. The steps are as follows (it is convenient to follow the thinking of these steps by tracing the location diagonal of the composite with the aid of Table B1):

a. Perform conventional Gaussian elimination to zero the elements CMAT1(I,J) for I > J, i.e., the elements below the main diagonal of M. Choose maximum pivot elements in conventional fashion. Carry row operations into CMAT3.

b. Create CMAT4 by row swapping with CMAT2 so as to choose maximum pivot elements. Perform elimination to zero CMAT4 elements for I > J and the entire column of CMAT2. The number of rows created in CMAT2 is NI4 = NI1 - NJ1, the amount by which CMAT1 is over-square. Carry row operations across into CMAT3.

c. Choose maximum pivot rows in columns of CMAT2 with J > NI4 and swap with rows given by I = J - NI4 (the rows containing the Jth column diagonal element of the composite). Conduct elimination to zero elements with I > J + NI4. Carry row operations into CMAT3.

d. Conduct conventional pivot selection and elimination on CMAT3 to zero elements CMAT3(I, J) with I > J + NJ1 + NJ2.

In each column elimination operation, the pivot value is multiplied into a product accumulator to produce a value for the determinant of the composite matrix. The sign of this product is changed at each row swap in accord with the rates of matrix algebra row operations.

The backsolving routine SPRSLV with its entry HOMSLV operate in a straightforward manner on the submatrices as reduced by SPRHOM. Details are omitted here, but the routines may be easily followed in a row-by-row flow from the bottom of the composite matrix, if one keeps in mind the row index relations of column 4 of Table B1. The entry HOMSLV assumes a zero determinant value resulted (approximately) from SPRHOM and the last element of the solution vector is picked as unity. (The zero determinant results from a zero falling at the last diagonal location in maximum pivoting Gaussian elimination.) The remainder of the homogeneous solution vector is backsolved conventionally relative to this last element. The vector is then renormalized so that its maximum element is unity.

Table B1

PRIMARY INDEXING QUANTITIES IN THE ALGORITHM

Submatrix	Size of Submatrix[1]	Indices of Main Diag. of Compos.[1]	Relative Row[2] Index of CMAT3 and CRHS
CMAT1	NI1 x NI2	(J,J)	I3 = I
CMAT4	NI1 - NJ1 x NJ2 (can be null)	(J,J)	I3 = I + NJ1
CMAT2	NI2 x NJ2	(J - (NI1 - NJ1), J)	I3 = I + NI1
CMAT3	NI1 + NI2 x NI1 + NI2 - NJ1 - NJ2	(J + NJ1 + NJ2, J)	I3 = I3

1. Quantities given in terms of input parms. to the routine. Related internal quantities are given in Figure B1.

2. Relative to I, the row index of the submatrix in question.

APPENDIX C

PROGRAM LISTINGS

All code compileable on IBM OS/360 and OS/370 FORTRAN levels G or H. The routine ZANLYT and its service routine UERTST is taken from the program library FORTUOI made available by the Computer Services Office, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801. The routines BSLJZ and BSCJZ are taken from the International Mathematical and Statistical Library (IMSL). They may not be reproduced apart from this application program package. The IMSL library is available by subscription from IMSL, Inc., 6100 Hillcroft, Suite 510, Houston, Texas 77036.

```
C     POLE SEARCH PROGRAM FOR S E M FORMULATION OF THIN-PLATE SCATTERER
C     BY L W PEARSON 8/74
C
      IMPLICIT REAL*8(A,B,D-H,O-Z),COMPLEX*16(C)
      COMMON /GEOM/ XSYM,YSYM,W,NX,NY,IPREAS(10),JPREAS(10),NPREI,NPREJ
      INTEGER MES(4,2)/'SYMM','ETRI','C',' ','ANTI','SYMM','ETRI','C'/
      DATA C/(3.D08,0.D0)/,PLUS/'+'/,PI/3.141592653589793/
      DATA HX/'X'/,HY/'Y'/
      EXTERNAL CPLATE
      DIMENSION CX(20),INFER(20)
      LOGICAL LAUTO
  100 READ(5,1,END=999) XSYM,YSYM,NX,NY,WO,WS,WM,CSUNOR,LAUTO
    1 FORMAT(2A1,2X,2I3,5F10.4,T80,L1)
      IMX=1
      IMY=1
      IF(XSYM.NE.PLUS) IMX=2
      IF(YSYM.NE.PLUS) IMY=2
      NW=(WM-WO)/WS
      IF(NW.GT.0) GO TO 105
      NW=-NW
      WS=-WS
  105 IF(WS*NW.LT.WM-WO) NW=NW+1
      DO 200 IW=1,NW
      W=WO+(IW-1)*WS
      IF(.NOT.LAUTO) GO TO 140
C                                      SKIP PAST AUTO ZONING
C
C     ROUTINE TO DETERMINE NO OF EXPANSION PULSES BASED ON ELECTRICAL
C     SIZE OF PLATE
C
      TESTWV=.1885D10/DABS(DIMAG(CSUNOR))
C
C     ///////////////
C
      NPPWVL=6
C
C     ///////////////
C
      FLENX=1/TESTWV
      NX=IDINT(FLENX*NPPWVL)
      IF(DFLOAT(NX).LT.FLENX*NPPWVL) NX=NX+1
      FLENY=W/TESTWV
      NY=IDINT(FLENY*NPPWVL)
      IF(DFLOAT(NY).LT.FLENY*NPPWVL) NY=NY+1
      NX=MIN0(NX,5)
      NY=MIN0(NY,5)
C
C     BEGIN SETUP FOR ALTERNATE EDGE PATCH PREASSIGNMENT
C
  140 NPREI=(NX+2)/3
      NPREJ=(NY+2)/3
      IF(NX-2*NPREI+2.LE.1.AND.NPREI.GT.1) NPREI=NPREI-1
      IF(NY-2*NPREJ+2.LE.1.AND.NPREJ.GT.1) NPREJ=NPREJ-1
      DO 110 I=1,NPREI
      IPREAS(NPREI+1-I)=NX-3*I+3
  110 CONTINUE
      DO 120 J=1,NPREJ
      JPREAS(NPREJ+1-J)=NY-3*J+3
  120 CONTINUE
C                                      LOCATIONS WHERE X-DIRECTED CURREN
C                                      T IS SET TO ZERO GIVEN BY SUBSCRI
```

```
C                                            PTS (NX,JPREAS) AND Y-DIRECTED BY
C                                            (IPREAS,NY)
C
      WRITE(6,2) W,CSUNOR
 2    FORMAT('1ENTER ITERATION',/,'OSHAPE RATIO =',F5.3,5X,
     1'STARTING FREQ =',2D12.4)
      WRITE(6,3)
 3    FORMAT('0',10X,'CUR SYMMETRY',6X,'PULSES',3X,'PREASSIGN LOC''NS')
      WRITE(6,4) HX,(MES(I,IMX),I=1,4),NX,(IPREAS(J),J=1,NPREI)
 4    FORMAT(' ',A1,'-DIR',5X,4A4,I6,5X,10I3)
      WRITE(6,4) HY,(MES(I,IMY),I=1,4),NY,(JPREAS(J),J=1,NPREJ)
      WRITE(6,5)
 5    FORMAT('0',11X,'COMPLEX FREQ',17X,'DETERMINANT')
      CX(1)=CSUNOR
      CALL ZANLYT(CPLATE,1.D-50,4,0,1,1,CX,100,INFER,IER)
      WRITE(6,6) CX(1)
 6    FORMAT('ORETURN FROM ITERATION',/,'OPOLE LOC''N',2E12.4)
      CALL MODE
      CSUNOR=CX(1)
200   CONTINUE
      GO TO 100
999   STOP
      END
```

```
      SUBROUTINE MODE
      IMPLICIT REAL*8(A,B,D-H,O-Z),COMPLEX*16(C)
      COMMON /MAT/ CMATX(25,25),CMATY(25,25),CHOM(50,10),CMAT4(10,25),
     1NPTCHS,NDIM1,NDIMCI,NDIMCJ,NORD
      COMMON /GEOM/ XSYM,YSYM,W,NX,NY,IPREAS(10),JPREAS(10),NPREI,NPREJ
      DIMENSION CPRX(5,5),CPRY(5,5)
      DIMENSION CJ(50)
      NPRE=NPREI+NPREJ
      NPREIM=NPREI-1
      NPREJM=NPREJ-1
      CALL HOMSLV(CMATX,NPTCHS,NPTCHS-NPREJ,NDIM1,NDIM1,
     1            CMATY,NPTCHS,NPTCHS-NPREI,NDIM1,NDIM1,
     2            CHOM,NDIMCI,NDIMCJ,CMAT4,NDIMCJ,NDIM1,CJ,NORD)
      NXM1=NX-1
      NYM1=NY-1
      NSUBS=0
      DO 470 JS=1,NY
      DO 450 IS=1,NXM1
      J=(JS-1)*NX+IS
      CPRX(IS,JS)=CJ(J-NSUBS)
      JM=J-NSUBS
  450 CONTINUE
      J=JS*NX
      IF(JS.NE.JPREAS(NSUBS+1)) GO TO 460
      NSUBS=MIN0(NSUBS+1,NPREJM)
      CPRX(NX,JS)=(0.,0.)
      GO TO 470
  460 CPRX(NX,JS)=CJ(J-NSUBS)
  470 CONTINUE
      DO 500 IS=1,NX
      DO 500 JS=1,NYM1
      J=(JS-1)*NX+IS
      CPRY(IS,JS)=CJ(NPTCHS-NPREJ+J)
  500 CONTINUE
      NSUBS=0
      DO 530 IS=1,NX
      J=NYM1*NX+IS
      IF(IS.NE.IPREAS(NSUBS+1)) GO TO 510
      CPRY(IS,NY)=(0.,0.)
      NSUBS=MIN0(NSUBS+1,NPREIM)
      GO TO 530
  510 CPRY(IS,NY)=CJ(NPTCHS-NPREJ+J-NSUBS)
  530 CONTINUE
      WRITE(6,16)
   16 FORMAT('0*****NATURAL MODE*****',/,'0X-DIRECTED CURRENT')
      DO 540 I=1,NX
      WRITE(6,17) (CPRX(I,J),J=1,NY)
   17 FORMAT('0',5(2D12.4,2X))
  540 CONTINUE
      WRITE(6,18)
   18 FORMAT('0Y-DIRECTED CURRENT')
      DO 550 I=1,NX
      WRITE(6,17) (CPRY(I,J),J=1,NY)
  550 CONTINUE
      WRITE(6,19)
   19 FORMAT('0HOMOGENEOUS EXPANSION COEF''S')
      WRITE(6,17) (CJ(2*NPTCHS-NPRE+I),I=1,NPRE)
      RETURN
      END
```

```
      COMPLEX FUNCTION CPLATE*16(CSUNOR)
C     DETERMINANT EVALUATION ROUTINE FOR HALLEN-TYPE AUGMENTED MOMENT
C     MATRIX FOR THE THIN PLATE SCATTERER
C     FOR S E M APPLICATIONS
C     BY L W PEARSON 8/74
C
      IMPLICIT COMPLEX*16(C),REAL*8(A,B,D-H,O-Z)
      COMMON /GEOM/ XSYM,YSYM,W,NX,NY,IPREAS(10),JPREAS(10),NPREI,NPREJ
      COMMON /MAT/ CMATX(25,25),CMATY(25,25),CHOM(50,10),CMAT4(10,25),
     1NPTCHS,NDIM1,NDIMCI,NDIMCJ,NORD
      REAL*8 DRARG,DIMARG,DRBES(20),DIMBES(20),DUM1(20),DUM2(20),DUM3(20
     1),DUM4(20)
      DIMENSION CINTER(10,10),CINTX(25),CINTY(25),CCOSTM(10),CSINTM(10)
      INTEGER MES(4,2)/'SYMM','ETRI','C',' ','ANTI','SYMM','ETRI','C'/
      DATA C/(3.D08,0.D0)/,PLUS/'+'/,PI/3.141592653589793/
      NDIM1=25
      NDIMCI=50
      NDIMCJ=10
      NDIM=50
C
C     FORMULATION SETUP ROUTINES
C
      IMX=1
      IMY=1
      IF(XSYM.NE.PLUS) IMX=2
      IF(YSYM.NE.PLUS) IMY=2
C                                          IM(X/Y)=2 INDICATES ANTISYMMETRIC
C                                          DISTR OF X-DIRECTED CURRENT WRT X
C                                          /Y AXIS
      NPTCHS=NX*NY
C                                          TOT NO OF CURRENT PATCHES
      NXM1=NX-1
      NYM1=NY-1
      EDGFAC=0.5
C                                          PROPORTIONAL WIDTH OF EDGE PULSES
      EDG2=EDGFAC*EDGFAC
C                                          CORNER FACTOR
      DX=1./(FLOAT(2*NX-2)+2*EDGFAC)
      DY=W/(FLOAT(2*NY-2)+2*EDGFAC)
      NXT2=NX*2
      NYT2=NY*2
      CS=CSUNOR/2/C
C                                          NORMALIZED LAPLACE VARIABLE
      INTPTS=13
      DXINT=DX/12
      DYINT=DY/12
C                                          NUMER INTEG PARMS
      NSYMX=-(-1)**IMX
      NSYMY=-(-1)**IMY
C                                          SIGNED SYMMETRY INDICATORS
      NSMII=NSYMY
      NSMIII=NSYMX*NSYMY
      NSMIV=NSYMX
C                                          SIGNS OF KERNEL FOR EA QUAD'S CON
C                                          TRIBUTION
      NINDX=2
      IF(NSMIII.GT.0) NINDX=1
C                                          NINDX = 1 INDICATES EVEN-EVEN OR
C                                          ODD-ODD SYMMETRY FOR X-DIR CURR
      NSCOS=1
      IF(XSYM.EQ.PLUS) NSCOS=2
```

```
C                                        NSCOS = 2 INDICATES EVEN SYMM WRT
C                                        Y FOR X DIR CURR (I E COSINE EXPA
C                                        NSION OF HOMGENEOUS SOL'N)
C
      NPRE=NPREI+NPREJ
C                                        TOT NO OF PREASSIGNED CURR VALHS
      NPREJM=NPREJ-1
      NPREIM=NPREI-1
      NPREP1=NPRE+1
C     END OF INPUT/SPECIFICATION ROUTINES
C
C     ROUTINE TO FILL INTERACTION MATRIX FROM WHICH MOMENT MATRIX IS
C     CONSTRUCTED
C
      DIAG=DSQRT(DX*DX+DY*DY)
      ALNXTM=2*DLOG((DIAG+DY)/DX)
      ALNYTM=2*DLOG((DIAG+DX)/DY)
      DYBDX=DY/DX
      DXBDY=DX/DY
      CSDX=CS*DX
      CSDY=CS*DY
      CSDX2=CSDX*CSDX
      CSDX3=CSDX*CSDX2
      CSDX4=CSDX2*CSDX2
      CSDY2=CSDY*CSDY
      CSDY3=CSDY*CSDY2
      CSDY4=CSDY2*CSDY2
C                                        COMPONENT TERMS FOR SELF-PATCH SE
C                                        RIES
      CXTERM=-0.5D0*CSDX*ALNXTM+0.5D0*CSDX2*DYBDX-CSDX3*(DXBDY*DIAG/(12*
     1DY)+    ALNXTM/24)+CSDX4*DYBDX*(1+DYBDX*DYBDX/3)/24
      CYTERM=-0.5D0*CSDY*ALNYTM+0.5D0*CSDY2*DXBDY-CSDY3*(DYBDX*DIAG/(12*
     1DX)+    ALNYTM/24)+CSDY4*DXBDY*(1+DXBDY*DXBDY/3)/24
C                                        CALC INDIV SERIES FOR SELF-INTER
      CINTER(1,1)=-2/CS*(CXTERM+CYTERM)
C                                        COMPUTE SELF-INTERACTION
      DO 220 IS=1,2
      XS=(FLOAT(IS)-1.5D0)*DX
      DO 220 JS=1,2
C                                        LOOP TO CALC ADJACENT PATCH INTER
      IF(IS*JS.EQ.1) GO TO 220
      YS=(FLOAT(JS)-1.5D0)*DY
      DO 210 IINT=1,INTPTS
      XP=FLOAT(IINT-1)*DXINT
C                                        NUMER INT WRT X LOOP
      X2TM2=XS+XP
      X2TM2=X2TM2*X2TM2
      DO 200 JINT=1,INTPTS
      YP=FLOAT(JINT-1)*DYINT
C                                        NUMER INT WRT Y LOOP
      Y2TM=YS+YP
      R=DSQRT(X2TM2+Y2TM*Y2TM)
      CINTY(JINT)=CDEXP(-2*CS*R)/R
C                                        EVAL INTEGRAND
 200  CONTINUE
      CALL DWEDDL(CINTY,INTPTS,DYINT,CINTX(IINT))
C                                        INT WRT Y TO YIELD X INTEGRAND
 210  CONTINUE
      CALL DWEDDL(CINTX,INTPTS,DXINT,CINTER(IS,JS))
C                                        INT WRT X
 220  CONTINUE
```

```
      DO 250 IS=1,NXT2
      X2TM2=DFLOAT(IS-1)*DX
      X2TM2=X2TM2*X2TM2
      DO 250 JS=1,NYT2
C                                           LOOPS FOR REMAINDER OF INTERACTIO
C                                           N CALC'ED BY ONE-PT RECTANG RULE
      IF(IS+JS.LT.4.OR.IS.EQ.2.AND.JS.EQ.2) GO TO 250
      Y2TM=FLOAT(JS-1)*DY
      R=DSQRT(X2TM2+Y2TM*Y2TM)
      CINTER(IS,JS)=CDEXP(-2*CS*R)/R*DX*DY
  250 CONTINUE
C     END OF LOOP TO FILL INTERACTION MATRIX
C
C     BEGIN CONSTRUCTION OF MOMENT MATRIX
C
      DO 350 IM=1,NX
      DO 350 JM=1,NY
C                                           INDEXING OF MATCH-POINTS OVER ENT
C                                           IRE QUADRANT
      I=(JM-1)*NX+IM
C                                           ONE-DIM MATCH-PT INDEX GEN'ED
C                                           COL'WISE ALONG X-DIRECTION
      NSUBS=0
      DO 330 JS=1,NYM1
C                                           INDEX OVER SOURCE PATCHES Y-DIR
      JD1=IABS(JM-JS)+1
C                                           1ST AND 2ND QUAD J 'DIFFERENCE
C                                           INDEX'
      JD2=JM+JS
C                                           3RD & 4TH QUAD J 'DIFFERENCE
C                                           INDEX'
C                                           NOTE THAT 'DIFFERENCE INDICES' AR
C                                           E = 'INDEX DIFFERENCE' +1 FOR THE
C                                           SAKE OF FORTRAN INDEXING
      DO 310 IS=1,NXM1
C                                           INDEX OVER SOURCE PATCHES X-DIR
      ID1=IABS(IS-IM)+1
C                                           1ST & 4TH QUAD 'DIFF INDEX'
      ID2=IS+IM
C                                           2ND & 3RD QUAD ' DIFF INDEX'
      J=(JS-1)*NX+IS
C                                           ONE-DIM SOURCE-PT INDEX
      CO=CINTER(ID1,JD1)+NSMIII*CINTER(ID2,JD2)
C                                           SUM OF SOURCE CONT FROM QI & QIII
      CE=NSMII*CINTER(ID2,JD1)+NSMIV*CINTER(ID1,JD2)
C                                           SUM OF SOURCE CONT FROM QII & QIV
      CMATX(I,J-NSUBS)=CO+CE
C                                           SUBMAT ENTRY FOR X-DIR CURR'S
      CMATY(I,J)=CO-CE
C                                           SUBMAT ENTRY FOR Y-DIR CURR'S
C                                           NOTE THAT EVEN Q'S CONT NEGATIVE
C                                           FOR Y-DIR CURR'S
  310 CONTINUE
C
C     END OF SOURCE LOOP FOR INTERIOR PATCHES
C
C     CONSTRUCTION OF SOURCE TERMS FROM ABS(X)=A EDGE FOLLOWS
C
      ID1=IABS(NX-IM)+1
      ID2=NX+IM
      J=JS*NX
```

```
      CO=CINTER(ID1,JD1)+NSMIII*CINTER(ID2,JD2)
C                                            SUM OF SOURCE CONT FROM QI & QIII
      CE=NSMII*CINTER(ID2,JD1)+NSMIV*CINTER(ID1,JD2)
C                                            SUM OF SOURCE CONT FROM QII & QIV
      CMATY(I,J)=(CO-CE)*EDGFAC
C
C                                            SUBMAT ENTRY FOR Y-DIR CURR'S
C                                            NOTE THAT EVEN Q'S CONT NEGATIVE
C                                            FOR Y-DIR CURR'S
      IF(JS.NE.JPREAS(NSUBS+1)) GO TO 325
      NSUBS=MIN0(NSUBS+1,NPREJM)
      GO TO 330
  325 CMATX(I,J-NSUBS)=(CE+CO)*EDGFAC
C                                            SUBMAT ENTRY FOR X-DIR CURR'S
C
C     END ROUTINE FOR ABS(X)=A EDGE TERMS
C
  330 CONTINUE
C
C     END LOOP OVER Y COORD FOR INTERIOR PATCHES
C
C     BEGIN ROUTINE FOR CONSTRUCTION OF SOURCE TERMS FOR ABS(Y)=B EDGE
C
      JD1=IABS(NY-JM)+1
      JD2=NY+JM
      NSUBSJ=NSUBS
      NSUBS=0
      DO 340 IS=1,NXM1
C                                            INDEX DOWN X COORD'S INTERIOR
C                                            PATCHES
      ID1=IABS(IS-IM)+1
      ID2=IS+IM
      J=(NYM1)*NX+IS
      CO=CINTER(ID1,JD1)+NSMIII*CINTER(ID2,JD2)
C                                            SUM OF SOURCE CONT FROM QI & QIII
      CE=NSMII*CINTER(ID2,JD1)+NSMIV*CINTER(ID1,JD2)
C                                            SUM OF SOURCE CONT FROM QII & QIV
      CMATX(I,J-NSUBSJ)=(CE+CO)*EDGFAC
C                                            SUBMAT ENTRY FOR X-DIR CURR'S
      IF(IS.NE.IPREAS(NSUBS+1)) GO TO 335
      NSUBS=MIN0(NSUBS+1,NPREIM)
      GO TO 340
  335 CMATY(I,J-NSUBS)=(CO-CE)*EDGFAC
C                                            SUBMAT ENTRY FOR Y-DIR CURR'S
C                                            NOTE THAT EVEN Q'S CONT NEGATIVE
C                                            FOR Y-DIR CURR'S
  340 CONTINUE
C
C     END ROUTINE FOR ABS(Y)=B EDGE
C
C     CONSTRUCTION OF CORNER SOURCE TERM
C
      ID1=IABS(NX-IM)+1
      ID2=NX+IM
      J=NX*NY
      CO=CINTER(ID1,JD1)+NSMIII*CINTER(ID2,JD2)
C                                            SUM OF SOURCE CONT FROM QI & QIII
      CE=NSMII*CINTER(ID2,JD1)+NSMIV*CINTER(ID1,JD2)
C                                            SUM OF SOURCE CONT FROM QII & QIV
      IF(NY.NE.JPREAS(NPREJ)) CMATX(I,J-NPREJM)=(CE+CO)*EDG2
C                                            SUBMAT ENTRY FOR X-DIR CURR'S
      IF(NX.NE.IPREAS(NPREI)) CMATY(I,J-NPREIM)=(CO-CE)*EDG2
```

```
C                                              SUBMAT ENTRY FOR Y-DIR CURR'S
C                                              NOTE THAT EVEN Q'S CONT NEGATIVE
C                                              FOR Y-DIR CURR'S
  350 CONTINUE
C
C     END OF MOMENT MATRIX INTERACTION CONSTRUCTION
C
C     BEGIN ROUTINE TO ENTER HOMOGENEOUS SOL'N EXPANSION COL'S IN MATRIX
C
  360 NBES=2*NPRE
C                                              HIGHEST ORDER BESSEL FUNCTION IN
C                                              HOMOGENEOUS SOL'N EXPANSION
      IF(NINDX.EQ.2) NBES=NBES-1
C                                              ONE LESS IF EVEN INDEX EXPANSION
      DO 400 IM=1,NX
      X=(FLOAT(IM)-0.5D0)*DX
      DO 400 JM=1,NY
      Y=(FLOAT(JM)-0.5D0)*DY
      I=(JM-1)*NX+IM
C                                              INDEXING THRU MATCH-PTS
      PHI=DATAN(Y/X)
      RHO=DSQRT(X*X+Y*Y)
C                                              POLAR COORD'S OF MATCH-PTS
      DRARG=2*DIMAG(CS)*RHO
      DIMARG=-2*DREAL(CS)*RHO
C                                              ARGUMENT OF BESSEL FN'S
      IF(DABS(DIMARG/DRARG).LT.1.E-20) GO TO 364
C                                              IF REAL ARG SKIP TO REAL BES CALL
      CALL BSCJZ(DRARG,DIMARG,DRBES,DIMBES,NBES,0.D0,16,IERR,DUM1,DUM2,D
     1UM3,DUM4)
C                                              GET TABLE OF BESSEL FUNCTIONS
      GO TO 368
  364 CALL BSLJZ(DRARG,DRBES,NBES,0.D0,16,IERR,DUM1,DUM2)
      CALL ZEROZ(DIMBES,2*(NBES+1))
      CALL ZEROZ(DIMBES,2*(NBES+1))
C                                              SET UP PURE REAL BES FUNCTIONS
  368 CCOSTM(1)=0
      CSINTM(1)=0
C                                              ZERO 1ST TERM COEF  CONSTRUCTION
C                                              VECTORS
      DO 370 II=1,NPREP1
C                                              INDEX THRU CALC OF COEF CONSTR
C                                              VECTOR
      INDX=2*II-NINDX
C                                              CALC SERIES INDEX
      IF(INDX.EQ.0) GO TO 370
C                                              SKIP CALC OF BELOW TERM FOR ZERO
C                                              INDEX - IT WAS SET TO ZERO ABOVE
      ARG=DFLOAT(INDX-1)*PHI
C                                              ARGUMENT OF SIN FN
      CBES=DCMPLX(DRBES(INDX),DIMBES(INDX))
      CCOSTM(II)=DCOS(ARG)*CBES*4*PI
      CSINTM(II)=DSIN(ARG)*CBES*4*PI
C                                              CALC COEFF CONSTRUCTION TERMS
  370 CONTINUE
      DO 380 JJ=1,NPREJ
C                                              LOOP TO REPLACE COL'S FOR PREASSI
C                                              GNED J TERMS
      J=JPREAS(JJ)*NX
C                                              INDEX OF COL BEING REPLACED
      INDX=2*JJ-NINDX
```

```
C                                            SERIES INDEX FOR REPLACING TERM
      GO TO (371,372),NSCOS
C                                            SELECT PROPER SERIES COEF ACCORDI
C                                            NG TO Y SYMMETRY CONDITION
C                                            NSCOS=2 INDICATES COSINES IN X
C                                            CURRENT EQ
 371  CHOM(I,JJ)=-PI/(2*CS)*(0.D0,1.D0)**INDX*(CSINTM(JJ)-CSINTM(JJ+1))
      CHOM(NPTCHS+I,JJ)=-PI/(2*CS)*(0.D0,1.D0)**INDX*(CCOSTM(JJ)+
     1CCOSTM(JJ+1))
      GO TO 380
 372  CHOM(I,JJ)=-PI/(2*CS)*(0.D0,1.D0)**(INDX+1)*(CCOSTM(JJ+1)-
     1CCOSTM(JJ))
      CHOM(I+NPTCHS,JJ)=-PI/(2*CS)*(0.D0,1.D0)**(INDX+1)*(CSINTM(JJ)+
     1CSINTM(JJ+1))
 380  CONTINUE
      DO 390 II=1,NPREI
      J=(NY-1)*NX+IPREAS(II)+NPTCHS
C                                            LOOP TO REPLACE COL'S FOR PREASSI
      JJ=II+NPREJ
C                                            GNED I TERMS
      INDX=2*(II+NPREJ)-NINDX
      GO TO (381,382),NSCOS
 381  CHOM(I,JJ)=-PI/(2*CS)*(0.D0,1.D0)**INDX*(CSINTM(II+NPREJ)-
     1CSINTM(II+NPREJ+1))
      CHOM(I+NPTCHS,JJ)=-PI/(2*CS)*(0.D0,1.D0)**INDX*(CCOSTM(II+NPREJ)+
     1CCOSTM(II+NPREJ+1))
      GO TO 390
 382  CHOM(I,JJ)=-PI/(2*CS)*(0.D0,1.D0)**(INDX+1)*(CCOSTM(II+NPREJ+1)-
     1CCOSTM(II+NPREJ))
      CHOM(I+NPTCHS,JJ)=-PI/(2*CS)*(0.D0,1.D0)**(INDX+1)*
     1(CSINTM(II+NPREJ)+CSINTM(II+NPREJ+1))
 390  CONTINUE
 400  CONTINUE
C
C     END OF MOMENT MATRIX CONSTRUCTION
C
 405  CONTINUE
      CALL SPRHOM(CMATX,NPTCHS,NPTCHS-NPREJ,NDIM1,NDIM1,
     1            CMATY,NPTCHS,NPTCHS-NPREI,NDIM1,NDIM1,
     2            CHOM,NDIMCI,NDIMCJ,CMAT4,NDIMCJ,NDIM1,CDET)
      FRAT=CDABS(CMATX(1,1))
      CPLATE=CDET/FRAT
      WRITE(6,20) CSUNOR,CPLATE
 20   FORMAT(' ',5X,2E12.4,5X,2E12.4)
      RETURN
      END
```

```
      SUBROUTINE SPRHOM(CMAT1,NI1,NJ1,NDIM1I,NDIM1J,CMAT2,NI2,NJ2,NDIM2I
     1,NDIM2J,CMAT3,NDIM3I,NDIM3J,CMAT4,NDIM4I,NDIM4J,CDET)
      IMPLICIT COMPLEX*16(C),REAL*8(A,B,D-H,O-Z)
C
C     SUBROUTINE TO DIAGONALIZE AND CALC DETERMINANT OF A SPARCELY-
C     COUPLED MATRIX
C     BY L W PEARSON 7/74
C     REVISED 5/75
C
      DIMENSION CMAT1(NDIM1I,NDIM1J),CMAT2(NDIM2I,NDIM2J),CMAT3(NDIM3I,
     1NDIM3J),CMAT4(NDIM4I,NDIM4J)
      NI3=NI1+NI2
      NJ3=NI3-NJ2-NJ1
      CALL ZEROZ(CMAT4,4*NDIM4I*NDIM4J)
      CDET=1
C                                    INITIALIZE PRODUCT ACCUMULATOR
      NPR=3
      NJ1M1=NJ1-1
      NJ1L=NJ1
      IF(NJ2+NJ3.GE.1) GO TO 95
      NJ1L=NJ1L-1
      NPR=1
 95   DO 155 M=1,NJ1L
C                                    INDEX ACROSS COL
      MP1=M+1
      FMAX=CDABS(CMAT1(M,M))
      K=M
      IF(MP1.GT.NI1) GO TO 105
      DO 100 I=MP1,NI1
C                                    LOOP TO SEARCH FOR PIVOT IN MTH
C                                    COL
      FCK=CDABS(CMAT1(I,M))
      IF(FCK.LE.FMAX) GO TO 100
      K=I
C                                    IF LARGER ELEMENT FOUND MARK ROW
      FMAX=FCK
C                                    USE NEW LARGE ELEMENT AS COMPARI-
C                                    SON VALUE
 100  CONTINUE
 105  CSTOR=CMAT1(K,M)
C                                    SAVE VAL OF PIVOT ELEMENT
      CDET=CDET*CSTOR
C                                    MULT PIVOT INTO PROD ACCUMULATOR
      IF(K.EQ.M) GO TO 115
C                                    IF PIVOT ON DIAG SKIP ROW EXCH
      CDET=-CDET
C                                    CHANGE SIGN BECAUSE OF ROW EXCH
 107  DO 110 J=M,NJ1
C                                    LOOP TO EXCH DIAG AND PIVOT ROWS
      CSTO=CMAT1(K,J)
      CMAT1(K,J)=CMAT1(M,J)
      CMAT1(M,J)=CSTO
 110  CONTINUE
      IF(NJ3.LT.1) GO TO 115
      DO 112 J=1,NJ3
      CSTO=CMAT3(K,J)
      CMAT3(K,J)=CMAT3(M,J)
      CMAT3(M,J)=CSTO
 112  CONTINUE
 115  CONTINUE
      IF(MP1.GT.NI1) GO TO 155
```

```
      DO 150 I=MP1,NI1
C                                      ELIMINATICN LOOP FOR CMAT1
      CFAC=CMAT1(I,M)/CSTOR
C                                      ELIMINATION FACTOR
      IF(MP1.GT.NJ1) GO TO 125
      DO 120 J=MP1,NJ1
C                                      LOOP ACROSS ROW IN CMAT3
      CMAT1(I,J)=CMAT1(I,J)-CMAT1(M,J)*CFAC
  120 CONTINUE
      IF(NJ3.LT.1) GO TO 150
  125 DO 130 J=1,NJ3
C                                      LOOP ACROSS ROW IN CMAT3
      CMAT3(I,J)=CMAT3(I,J)-CMAT3(M,J)*CFAC
  130 CONTINUE
  150 CONTINUE
  155 CONTINUE
      NI4=NI1-NJ1
      IF(NI4.LE.0) GO TC 290
C
C
C     BEGIN ROUTINE TO CREATE/'DIAGONALIZE' CMAT4
C
      NPIV=NI4
      IF(NI4.GT.NJ2) NPIV=NJ2
      DO 250 M=1,NPIV
C                                      INDEX ACROSS COL FOR CMAT4 DIAG
      MP1=M+1
      FMAX=CDABS(CMAT2(1,M))
      K=1
      IF(NI2.LT.2) GO TO 205
      DO 200 I=2,NI2
C                                      LOOP TO SEARCH FOR PIVOT IN MTH
C                                      COL
      FCK=CDABS(CMAT2(I,M))
      IF(FCK.LE.FMAX) GO TC 200
      K=I
C                                      IF LARGER ELEMENT FOUND MARK ROW
      FMAX=FCK
C                                      USE NEW LARGE ELEMENT AS COMPARI-
C                                      SON VALUE
  200 CONTINUE
  205 CSTOR=CMAT2(K,M)
C                                      SAVE VAL OF PIVOT ELEMENT
      CDET=CDET*CSTOR
C                                      MULT PIVOT INTO PROD ACCUM
      CDET=-CDET
C                                      CHANGE SIGN OF DETERM BECAUSE OF
C                                      EXCHANGE FROM CMAT2 TO CMAT4
      DO 210 J=M,NJ2
C                                      LOOP TO EXCHANGE DIAG AND PIVOT R
C                                      ROWS
      CSTO=CMAT4(M,J)
      CMAT4(M,J)=CMAT2(K,J)
      CMAT2(K,J)=CSTO
  210 CONTINUE
      K3=K+NI1
      M3=NJ1+M
      IF(NJ3.LT.1) GO TO 213
      DO 212 J=1,NJ3
      CSTO=CMAT3(K3,J)
      CMAT3(K3,J)=CMAT3(M3,J)
```

```
      CMAT3(M3,J)=CSTO
  212 CONTINUE
  213 IF(NI2.LT.1) GO TO 290
  235 DO 250 I=1,NI2
C                                    LOOP TO CARRY ELIMINATION INTO
C                                    CMAT2
      I3=NI1+I
      CFAC=CMAT2(I,M)/CSTOR
      IF(MP1.GT.NJ2) GO TO 242
      DO 240 J=MP1,NJ2
C                                    LOOP ACROSS ROW OF CMAT2
      CMAT2(I,J)=CMAT2(I,J)-CMAT4(M,J)*CFAC
  240 CONTINUE
      IF(NJ3.LT.1) GO TO 250
  242 DO 245 J=1,NJ3
C                                    LOOP ACROSS ROW OF CMAT3
      CMAT3(I3,J)=CMAT3(I3,J)-CMAT3(M3,J)*CFAC
  245 CONTINUE
  250 CONTINUE
C
C     END ROUTINE TO 'DIAGONALIZE' CMAT4
C
  290 IF(NI4.GE.NJ2) GO TO 390
C                                    IF DIAGONAL DOES NOT PASS THRU
C                                    SKIP DIAGONALIZATION FOR CMAT2
C
C     BEGIN ROUTINE TO 'DIAGONALIZE' CMAT2
C
      NI4P1=NI4+1
      NJ2L=NJ2
      IF(NJ3.GE.1) GO TO 295
      NJ2L=NJ2L-1
      NPR=2
  295 DO 350 M=NI4P1,NJ2L
      MI=M-NI4
      M3=MI+NI1
      MP1=M+1
      MIP1=MI+1
      FMAX=CDABS(CMAT2(MI,M))
      K=MI
      IF(MIP1.GT.NI2) GO TO 305
      DO 300 I=MIP1,NI2
C                                    LOOP TO SEARCH FOR PIVOT IN MTH
C                                    COL
      FCK=CDABS(CMAT2(I,M))
      IF(FCK.LE.FMAX) GO TO 300
      K=I
C                                    IF LARGER ELEMENT FOUND MARK ROW
      FMAX=FCK
C                                    USE NEW LARGE ELEMENT AS COMPARI-
C                                    SON VALUE
  300 CONTINUE
  305 CSTOR=CMAT2(K,M)
C                                    SAVE VAL OF PIVOT ELEMENT
      K3=K+NI1
      CDET=CDET*CSTOR
C                                    MULT PIVOT INTO PROD ACCUMULATOR
      IF(K.EQ.MI) GO TO 315
C                                    IF PIVOT ON DIAG SKIP ROW EXCH
      CDET=-CDET
C                                    CHANGE SIGN BECAUSE OF ROW EXCH
```

```
      DO 310 J=M,NJ2
      CSTO=CMAT2(K,J)
      CMAT2(K,J)=CMAT2(MI,J)
      CMAT2(MI,J)=CSTO
  310 CONTINUE
      IF(NJ3.LT.1) GO TO 315
      DO 312 J=1,NJ3
C                                      LOOP FOR EXCH IN CMAT3
      CSTO=CMAT3(K3,J)
      CMAT3(K3,J)=CMAT3(M3,J)
      CMAT3(M3,J)=CSTO
  312 CONTINUE
  315 CONTINUE
      IF(MIP1.GT.NI2) GO TO 390
      DO 350 I=MIP1,NI2
C                                      ELIMINATION LOOP
      I3=I+NI1
      CFAC=CMAT2(I,M)/CSTOR
      IF(MP1.GT.NJ2) GO TO 335
      DO 330 J=MP1,NJ2
C                                      LOOP ACROSS ROW OF CMAT2
      CMAT2(I,J)=CMAT2(I,J)-CMAT2(MI,J)*CFAC
  330 CONTINUE
  335 IF(NJ3.LT.1) GO TO 350
      DO 345 J=1,NJ3
C                                      LOOP ACROSS ROW IN CMAT3
      CMAT3(I3,J)=CMAT3(I3,J)-CMAT3(M3,J)*CFAC
  345 CONTINUE
  350 CONTINUE
C
C     BEGIN ROUTINE TO 'DIAGONALIZE' CMAT3
C
  390 NJ3M1=NJ3-1
      IF(NJ3M1.LT.1) GO TO 455
      DO 450 M=1,NJ3M1
C                                      INDEX ACROSS COL
      MP1=M+1
      MI=M+NJ1+NJ2
      MIP1=MI+1
      FMAX=CDABS(CMAT3(MI,M))
      K=MI
      IF(MIP1.GT.NI3) GO TO 405
      DO 400 I=MIP1,NI3
C                                      LOOP TO SEARCH FOR PIVOT IN MTH
C                                      COL
      FCK=CDABS(CMAT3(I,M))
      IF(FCK.LE.FMAX) GO TO 400
      K=I
C                                      IF LARGER ELEMENT FOUND MARK ROW
      FMAX=FCK
C                                      USE NEW LARGE ELEMENT AS COMPARI-
C                                      SO VALUE
  400 CONTINUE
  405 CSTOR=CMAT3(K,M)
C                                      SAVE VAL OF PIVOT ELEMENT
      CDET=CDET*CSTOR
C                                      MULT PIVOT INTO PROD ACCUMULATOR
      IF(K.EQ.MI) GO TO 415
C                                      IF PIVOT ON DIAG SKIP ROW EXCH
      CDET=-CDET
C                                      CHANGE SIGN BECAUSE OF ROW EXCH
```

```
      DO 410 J=M,NJ3
C                                         LOOP TO EXCH DIAG AND PIVOT ROWS
      CSTO=CMAT3(K,J)
      CMAT3(K,J)=CMAT3(MI,J)
      CMAT3(MI,J)=CSTO
  410 CONTINUE
  415 CONTINUE
      DO 450 I=MIP1,NI3
C                                         ELIMINATION LOOP
      CFAC=CMAT3(I,M)/CSTOR
      DO 445 J=MP1,NJ3
C                                         LOOP ACROSS ROW IN CMAT3
      CMAT3(I,J)=CMAT3(I,J)-CMAT3(MI,J)*CFAC
  445 CONTINUE
  450 CONTINUE
  455 GO TO (461,462,463), NPR
  461 CDET=CDET*CMAT1(NI1,NJ1)
      RETURN
  462 CDET=CDET*CMAT2(NI2,NJ2)
      RETURN
  463 CDET=CDET*CMAT3(NI3,NJ3)
      RETURN
C                                         MULT LAST ELEMENT INTO DETERM
      END
```

```
      SUBROUTINE SPRSLV(CMAT1,NI1,NJ1,NDIM1I,NDIM1J,CMAT2,NI2,NJ2,NDIM2I
     1,NDIM2J,CMAT3,NDIM3I,NDIM3J,CMAT4,NDIM4I,NDIM4J,CRHS,CSOLN)
C
C     SUBROUTINE TO BACKSOLVE A TRIANGULARIZED SYSTEM OF SPARCELY-
C     COUPLED LINEAR EQUATION
C     BY L W PEARSON 7/74
C     REVISED 5/75
C
C     STORAGE FORM COMPATIBLE WITH THE TRIANGULARIZATION ROUTINE  SPARCE
C
C     THE ENTRY 'HOMSLV' BELOW ALLOWS THE SOLUTION FOR NATURAL VECTORS
C     OF HOMOGENEOUS SYSTEMS PROVIDED THE DETERMINANT OF THE SYSTEM IS
C     ZERO
C
      IMPLICIT COMPLEX*16(C),REAL*8(A,B,D-H,O-Z)
      DIMENSION CMAT1(NDIM1I,NDIM1J),CMAT2(NDIM2I,NDIM2J),CMAT3(NDIM3I,N
     1DIM3J),CMAT4(NDIM4I,NDIM4J),CRHS(NDIM3I),CSOLN(NDIM3I)
      LOGICAL LHOM
C
C     SETUP FOR INHOMOGENEOUS SYSTEM
C
      LHOM=.FALSE.
C                                        SET INDICATOR FOR INHOM ENTRY
      NI3=NI1+NI2
C                                        NO ROWS IN COUPLING SUBMATRIX
      NJ3=NI3-NJ1-NJ2
C                                        NO OF COLS
      NI4=NI1-NJ1
C                                        NO ROWS IN SECONDARY COUPLING
C                                        SUBMATRIX
      ND2=NJ2-NI4
C                                        NO OF DIAGONAL TERMS OF MATRIX
C                                        IN CMAT2
      NPR=3
      IF(NJ3.LT.1) NPR=2
C                                        SET INDICATOR FOR NULL CMAT3
C                                        DEGENERACY
      IF(NJ3+NJ2.LT.1) NPR=1
C                                        SET INDICATOR FOR NULL CMAT2 &
C                                        CMAT3
      GO TO (81,82,83) , NPR
C                                        GO MAKE FIRST DIVISION FOR RIGHT-
C                                        MOST MATRIX
 81   CSOLN(NI3)=CRHS(NI3)/CMAT1(NI1,NJ1)
      GO TO 100
 82   CSOLN(NI3)=CRHS(NI3)/CMAT2(NI2,NJ2)
      GO TO 100
 83   CSOLN(NI3)=CRHS(NI3)/CMAT3(NI3,NJ3)
C                                        SOLVE FOR 'LAST' UNKNOWN
      GO TO 100
C                                        GO TO SOLN ROUTINES
C
C     END OF SETUP FOR INHOM SYSTEM
C
C     BEGIN ENTRY/SETUP FOR HOMOGENEOUS SYSTEM
C
      ENTRY HOMSLV(CMAT1,NI1,NJ1,NDIM1I,NDIM1J,CMAT2,NI2,NJ2,NDIM2I,NDIM
     12J,CMAT3,NDIM3I,NDIM3J,CMAT4,NDIM4I,NDIM4J,CSOLN,NORD)
C
      LHOM=.TRUE.
C                                        LOGICAL INDICATOR FOR HOMOGEN SYS
```

```
      NI3=NI1+NI2
      NJ3=NI3-NJ1-NJ2
      NI4=NI1-NJ1
      ND2=NJ2-NI4
      CSOLN(NI3)=1
C                                     ASSIGN ARBITRARY ELEMENT IN SOL'N
C
C     END SETUP FOR HOMCGENECUS ENTRY
C
C     BEGIN BACKSOLVE FOR EQUATIONS INVOLVING ONLY CMAT3 (LAST NJ3 EQS)
C
  100 FMAX=CDABS(CSOLN(NI3))
      IMAX=NI3
      IF(NJ3.LT.2) GO TO 200
C                                     SKIP ROUTINE IF ONLY LAST VARIABL
C                                     COUPLES (IT WAS SOLVED/ASSIGNED
C                                     ABOVE)
      DO 150 IC=2,NJ3
      ICM1=IC-1
      I=NI3-IC+1
      I=NI3-IC+1
C                                     CALC MATRIX ROW INDX FROM
C                                     COMPLEMENTARY INDX
      JD3=I-NJ1-NJ2
C                                     COL INDX FOR CMAT3 WHICH DEFINES
C                                     DIAG OF MATRIX
      CSUM=0
      DO 110 J3C=1,ICM1
C                                     LOOP TO ACCUM NEGATIVE SUM OF
C                                     PREVIOUSLY CALC'D UNKNS
      J3=NJ3+1-J3C
C                                     COL OF COEF IN CMAT3
      J=NI3+1-J3C
C                                     ROW OF UNKN IN CSOLN
      CSUM=CSUM-CMAT3(I,J3)*CSOLN(J)
  110 CONTINUE
      IF(.NOT.LHOM) CSUM=CSUM+CRHS(I)
C                                     ADD R H S TO SUM
      CSOLN(I)=CSUM/CMAT3(I,JD3)
C                                     DIVIDE BY DIAG COEF
      IF(CDABS(CSOLN(I)).LE.FMAX) GO TO 150
      FMAX=CDABS(CSOLN(I))
      IMAX=I
C                                     CHECK FOR MAX ELEMENT
  150 CONTINUE
C
C     BEGIN ROUTINE TO SOLVE FOR ELEMENTS INVOLVING CMAT3 & CMAT2
C
  200 IF(NJ3.GE.NI2) GO TO 300
C                                     SKIP ROUTINE IF DIAG DOES NOT
C                                     PASS THRU CMAT2
      DO 250 IC=1,ND2
      ICM1=IC-1
      I2=NI2-NJ3+1-IC
      I3=NI3-NJ3+1-IC
      JD2=NJ2+1-IC
      NCM1=NJ3+IC-1
      CSUM=0
      IF(NJ3.LT.1) GO TO 215
      DO 210 JC=1,NJ3
C                                     LOOP TO SUM CONTRIB FROM CMAT3
```

```
      J3=NJ3+1-JC
      J=NI3+1-JC
      CSUM=CSUM-CMAT3(I3,J3)*CSOLN(J)
 210  CONTINUE
 215  IF(ICM1.LT.1) GO TO 225
C                                        SKIP IF NO TERMS CONTRIB FR CMAT2
      DO 220 J2C=1,ICM1
      J2=NJ2+1-J2C
      J=NI3-NJ3+1-J2C
      CSUM=CSUM-CMAT2(I2,J2)*CSOLN(J)
 220  CONTINUE
 225  IF(.NOT.LHOM) CSUM=CSUM+CRHS(I3)
      CSOLN(I3)=CSUM/CMAT2(I2,JD2)
      IF(CDABS(CSOLN(I3)).LE.FMAX) GO TO 250
      FMAX=CDABS(CSOLN(I3))
      IMAX=I
 250  CONTINUE
C
C     BEGIN ROUTINE TO SOLVE FOR ELEMENTS INVOLVING CMAT3 &CMAT4
C
 300  IF(NI4.LT.1) GO TO 400
      DO 350 IC=1,NI4
      I4=NI4+1-IC
      JD4=I4
      I3=NI1+1-IC
      CSUM=0
      IF(NJ3.LT.1) GO TO 315
      DO 310 J3C=1,NJ3
      J3=NJ3+1-J3C
      J=NI3+1-J3C
      CSUM=CSUM-CMAT3(I3,J3)*CSOLN(J)
 310  CONTINUE
 315  NSUBS=ND2+IC-1
C                                        NO OF NON-DIAG CMAT4 EL'S IN EQ
      IF(NSUBS.LT.1) GO TO 325
      DO 320 J4C=1,NSUBS
      J4=NJ2+1-J4C
      J=NI3-NJ3+1-J4C
      CSUM=CSUM-CMAT4(I4,J4)*CSOLN(J)
 320  CONTINUE
 325  IF(.NOT.LHOM) CSUM=CSUM+CRHS(I3)
      CSOLN(I3)=CSUM/CMAT4(I4,I4)
      IF(CDABS(CSOLN(I3)).LE.FMAX) GO TO 350
      FMAX=CDABS(CSOLN(I3))
      IMAX=I3
 350  CONTINUE
C
C     BEGIN ROUTINE TO SOLVE EQ'S INVOLVING CMAT3 & CMAT1
C
 400  IF(NJ1.LT.1) GO TO 455
      DO 450 IC=1,NJ1
      I=NJ1+1-IC
      ICM1=IC-1
      CSUM=0
      IF(NJ3.LT.1) GO TO 415
      DO 410 J3C=1,NJ3
      J3=NJ3+1-J3C
      J=NI3+1-J3C
      CSUM=CSUM-CMAT3(I,J3)*CSOLN(J)
 410  CONTINUE
 415  IF(ICM1.LT.1) GO TO 425
```

```
      DO 420 JC=1,ICM1
      J=NJ1+1-JC
      CSUM=CSUM-CMAT1(I,J)*CSOLN(J)
  420 CONTINUE
  425 IF(.NOT.LHOM) CSUM=CSUM+CRHS(I)
      CSOLN(I)=CSUM/CMAT1(I,I)
      IF(CDABS(CSOLN(I)).LE.FMAX) GO TO 450
      FMAX=CDABS(CSCLN(I))
      IMAX=I
  450 CONTINUE
C
C
C     END OF SOLUTION
C
  455 IF(.NOT.LHOM) RETURN
C                                        RETURN IF INHOM SYSTEM
C
C     BEGIN NORMALIZATION ROUTINE FOR NATURAL VECTOR FOR HOMOGENEOUS
C     CASE
C
      CSCALE=1./CSOLN(IMAX)
      DO 500 I=1,NI3
      CSOLN(I)=CSOLN(I)*CSCALE
  500 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE COPYZ(X,Y,N)
      DIMENSION X(1),Y(1)
      DO 100 I=1,N
      X(I)=Y(I)
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ZEROZ(IARRAY,N)
      DIMENSION IARRAY(1)
      DO 100 I=1,N
      IARRAY(I)=0
100   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DWEDDL(FCN,N,DELTA,VINT)
      IMPLICIT REAL*8(A-H,O-Z)
      COMPLEX*16 FCN,C,VINT
      DIMENSION FCN(N)
      DIMENSION COEF(6)
      DATA COEF/2.D0,5.D0,1.D0,6.D0,1.D0,5.D0/
      IF((N-1)/6*6.EQ.N-1) GO TO 100
      WRITE(6,1)
1     FORMAT('0INCORRECT POINTS TO WEDDLE')
      A=1/0
100   CONTINUE
      VINT=0
      DO 200 J=1,N
      JCOEF=J-((J-1)/6)*6
      VINT=VINT+COEF(JCOEF)*FCN(J)
200   CONTINUE
      VINT=(VINT-FCN(1)-FCN(N))*(0.3D0,0.D0)*DCMPLX(DELTA,0.D0)
      RETURN
      END
```

```
C.ZANLYT........D...............................................................
C
C   FUNCTION            - DETERMINATION OF ZEROS OF AN ANALYTIC COMPLEX
C                           FUNCTION USING MULLER'S METHOD WITH
C                           DEFLATION
C   USAGE               - CALL ZANLYT (F,EPS,NSIG,KN,NGUESS,N,X,ITMAX,
C                           INFER,IER)
C   PARAMETERS   F      - A FUNCTION SUBPROGRAM, F(Z), WRITTEN BY THE
C                           USER SPECIFYING THE EQUATION WHOSE ROOTS
C                           ARE TO BE FOUND.  F MUST BE TYPE-NAMED AS
C                           FOLLOWS - COMPLEX FUNCTION F*16 (Z)
C                EPS    - 1ST STOPPING CRITERION.  A ROOT Z  IS ACCEPTED
C                           IF ABSOLUTE VALUE OF F(Z) .LE. EPS (INPUT)
C                NSIG   - 2ND STOPPING CRITERION.  A ROOT IS ACCEPTED
C                           IF TWO SUCCESSIVE APPROXIMATIONS TO A GIVEN
C                           ROOT AGREE IN THE FIRST NSIG DIGITS. (INPUT)
C                             NOTE. IF EITHER OR BOTH OF THE STOPPING
C                             CRITERIA ARE FULFILLED, THE ROOT IS
C                             ACCEPTED.
C                KN     - THE NUMBER OF KNOWN ROOTS WHICH MUST BE STORED
C                           IN X(1),...,X(KN), PRIOR TO ENTRY TO ZANLYT
C                NGUESS - THE NUMBER OF INITIAL GUESSES PROVIDED. THESE
C                           GUESSES MUST BE STORED IN X(KN+1),...,
C                           X(KN+NGUESS) AND NGUESS MUST BE SET EQUAL
C                           TO ZERO IF NO GUESSES ARE PROVIDED. (INPUT)
C                N      - THE NUMBER OF NEW ROOTS TO BE FOUND BY
C                           ZANLYT (INPUT)
C                X      - A LONG-WORD COMPLEX VECTOR ARRAY OF LENGTH
C                           .GE. 3*(KN+N). X(1),...,X(KN) ON INPUT
C                           MUST CONTAIN ANY KNOWN ROOTS. X(KN+1),...,
C                           X(KN+N) ON INPUT MAY, AT THE USER'S OPTION,
C                           CONTAIN INITIAL GUESSES FOR THE N NEW
C                           ROOTS WHICH ARE TO BE COMPUTED. ON OUTPUT,
C                           X(KN+1),..., X(KN+N) CONTAIN EITHER A ROOT
C                           CORRECT TO WITHIN A CONVERGENCE CRITERON
C                           OR THE VALUE(12345678.12345678D+0,12345678.
C                           12345678D+0) INDICATIVE OF A FAILURE TO
C                           ACHIEVE THE SPECIFIED CONVERGENCE FOR THAT
C                           ROOT, SAY X(KN+J). IN THE LATTER CASE, THE
C                           MOST RECENT APPROXIMATION TO X(KN+J) IS
C                           AVAILABLE IN X(ISUB), WHERE ISUB=2*(KN+N)+J
C                ITMAX  - THE MAXIMUM ALLOWABLE NUMBER OF ITERATIONS
C                           PER ROOT (INPUT)
C                INFER  - AN INTEGER VECTOR OF LENGTH .GE. KN+N.  ON
C                           OUTPUT INFER(J) CONTAINS THE NUMBER OF
C                           ITERATIONS USED IN FINDING THE J-TH ROOT
C                           WHEN CONVERGENCE WAS ACHIEVED.  IF
C                           CONVERGENCE WAS NOT OBTAINED IN ITMAX
C                           ITERATIONS, INFER(J) WILL CONTAIN ITMAX+1
C                           (OUTPUT)
C                IER    - ERROR PARAMETER (OUTPUT)
C                         WARNING ERROR = 32 + N
C                           N = 1 FAILURE TO CONVERGE WITHIN ITMAX
C                           ITERATIONS FOR ONE OF THE (N) NEW ROOTS TO
C                           BE FOUND
C   PRECISION           - DOUBLE
C   REQ'D IMSL ROUTINES - UERTST
C   AUTHOR/IMPLEMENTOR  - C. G. JOHNSON/L. L. WILLIAMS
C   LANGUAGE            - FORTRAN
C...............................................................................
C   LATEST REVISION     - SEPTEMBER 1, 1971
```

```
C
      SUBROUTINE ZANLYT   (F,EPS,NSIG,KN,NGUESS,N,X,ITMAX,INFER,IER)
      COMPLEX*16          X(1),ONE,D,DD,DEN,DI,FPRT,FRT,
     1                    H,RT,T1,T2,T3,TEM,X0,X1,X2,BI,F,XX
      DOUBLE PRECISION    QZ,EPS,EPS1
      DIMENSION           INFER(1)
      IER = 0
      ONE = (1.0D+00,0.0D+00)
      EPS1 = 10.0D+00**(-NSIG)
      ICONJ = 0
      IBOMB = 0
C                                  SET NUMBER OF ITERATIONS
      MB1 = KN+1
      MB2 = KN+N
      LSTART = MB2+1
      MPG = MB1+NGUESS
      DO 2 I = MPG,MB2
    2    X(I) = (0.D+0,0.D+0)
      L = MB1
      IF (KN .EQ. 0) GO TO 5
      DO 3 I = 1,KN
         INFER(I) = 0
         ITEMP = MB2+I
         X(ITEMP) = X(I)
         ITEMP = MB2+ITEMP
    3 X(ITEMP) = X(I)
    5 JK = 0
      QZ = CDABS(X(L))
      IF (QZ .LE. 1.0D-15) GO TO 25
C                                  ROOT ESTIMATE NOT EQUAL TO ZERO
   10 RT = (.9D+00,0.0D+00)*X(L)
      ASSIGN 15 TO NN
      GO TO 135
   15 X0 = FPRT
      RT = (1.1D+00,0.0D+00)*X(L)
      ASSIGN 20 TO NN
      GO TO 135
   20 X1 = FPRT
      H = X(L)-RT
      RT = X(L)
      ASSIGN 40 TO NN
      GO TO 135
C                                  ROOT ESTIMATE EQUAL TO ZERO
   25 RT = -ONE
      ASSIGN 30 TO NN
      GO TO 135
   30 X0 = FPRT
      RT = ONE
      ASSIGN 35 TO NN
      GO TO 135
   35 X1 = FPRT
      RT = (0.0D+00,0.0D+00)
      H = -ONE
      ASSIGN 40 TO NN
      GO TO 135
   40 X2 = FPRT
   45 D = (-0.5D+00,0.0D+00)
C                                  BEGIN MAIN ALGORITHM
   50 DD = ONE + D
      T1 = X0*D*D
      T2 = X1*DD*DD
```

```
      XX = X2*DD
      T3 = X2*D
      BI = T1-T2+XX+T3
      DEN = BI*BI-(4.0D+0,0.D+0)*(XX*T1-T3*(T2-XX))
C                                        USE DENOMINATOR OF MAXIMUM AMPLITUDE
      T1 = CDSQRT(DEN)
      T2 = BI + T1
      T3 = BI - T1
      QZ = CDABS(T2) - CDABS(T3)
      IF (QZ .GE. 0) GO TO 60
   55 DEN = T3
      GO TO 65
   60 DEN = T2
C                                        TEST FOR ZERO DENOMINATOR
   65 QZ = CDABS(DEN)
      IF (QZ .GT. 1.D-15) GO TO 75
   70 DEN = ONE
   75 DI = ((-2.0D+00,0.0D+00)*XX)/DEN
      H = DI * H
      RT = RT + H
C                                        CHECK CONVERGENCE OF THE FIRST KIND
      QZ = CDABS(H/RT)
      IF (QZ .LE. EPS1) GO TO 100
   80 ASSIGN 85 TO NN
      GO TO 135
   85 QZ = CDABS(FPRT)-CDABS(X2*(10.0D0,0.0D0))
      IF (QZ .LT. 0.D+0) GO TO 95
C                                        TAKE REMEDIAL ACTION TO INDUCE
C                                          CONVERGENCE
   90 DI = DI*(0.5D+00,0.0D+00)
      H = H*(0.5D+00,0.0D+00)
      RT = RT-H
      GO TO 135
   95 X0 = X1
      X1 = X2
      X2 = FPRT
      D = DI
      GO TO 50
C                                        A ROOT HAS BEEN FOUND
  100 FRT = F(RT)
  105 X(L) = RT
      ITEMP = MB2+L-IBOMB
      X(ITEMP) = RT
      ITEMP = MB2+MB2+L
      X(ITEMP) = RT
C                                        CHECK TO SEE IF COMPLEX-CONJUGATE
C                                        IS ALSO A ROOT
      IF (CDABS(F(DCONJG(X(L)))) .GT. 10.D+0*CDABS(FRT)) GO TO 115
      QZ = CDABS(X(L)- DCONJG(X(L)))
      IF (ICONJ .NE. 0 .OR. QZ .LT. 1.0D-8) GO TO 115
      ISTART = L+2
      INSER1 = L+1
         DO 110 INSERT = ISTART,MB2
         X(INSERT) = X(INSER1)
  110    INSER1 = INSERT
      X(L+1) = DCONJG(X(L))
      ICONJ = 1
      GO TO 120
  115 ICONJ = 0
  120 CONTINUE
  125 INFER(L) = JK
```

```
      L = L+1
      IF (L .LE. MB2) GO TO 5
C                                   RETURN TO CALLING PROGRAM
  130 GO TO 185
  135 JK = JK+1
      IF (JK .GT. ITMAX) GO TO 180
  140 FRT = F(RT)
      FPRT = FRT
C                                   TEST TO SEE IF FIRST ROOT IS BEING
C                                      DETERMINED
      IF (L .EQ. 1) GO TO 160
      IF (L .LE. IBOMB+1) GO TO 160
C                                   COMPUTE DENOMINATOR FOR MODIFIED
C                                      FUNCTION
  145 LIMUP = MB2+L-IBOMB-1
      DO 150 I = LSTART,LIMUP
         TEM = RT - X(I)
         QZ = CDABS(TEM)
         IF (QZ .LT. 5.0D-15) GO TO 175
  150 FPRT = FPRT/TEM
C                                   CHECK CONVERGENCE OF THE SECOND KIND
  160 QZ = CDABS(FRT)
      IF (QZ .GE. EPS) GO TO 170
  165 QZ = CDABS(FPRT)
      IF (QZ .LT. EPS) GO TO 105
  170 GO TO NN,(15,20,30,35,40,85)
  175 RT = RT * (1.000001D+0,0.0D+0)
      GO TO 135
C                                   WARNING ERROR, ITMAX = MAXIMUM
  180 IER = 33
      INFER(L) = ITMAX + 1
      IBOMB = IBOMB + 1
      X(L) = (12345678.12345678D+0,12345678.12345678D+0)
      ITEMP = MB2 + MB2 + L
      X(ITEMP) = RT
      L = L+1
      IF (L .LE. MB2) GO TO 5
  185 IF (IER .EQ. 0) GO TO 9005
 9000 CONTINUE
      CALL UERTST(IER,'ZANLYT')
 9005 RETURN
      END
```

```
C.UERTST...............................................................
C
C   FUNCTION            - ERROR MESSAGE GENERATION
C   USAGE               - CALL UERTST(IER,'NAMEXX')
C   PARAMETERS   IER    - ERROR PARAMETER. TYPE + N  WHERE
C                           TYPE= 128 IMPLIES TERMINAL ERROR
C                                  64 IMPLIES WARNING WITH FIX
C                                  32 IMPLIES WARNING
C                           N    = ERROR CODE RELEVANT TO CALLING ROUTINE
C                NAMEXX - NAME OF THE CALLING ROUTINE
C   AUTHOR/IMPLEMENTER  - PEDER SVENDSEN
C   LANGUAGE            - FORTRAN
C......................................................................
C   LATEST REVISION     - JANUARY 19, 1971
C
      SUBROUTINE UERTST(IER,NAME)
C
      DIMENSION            ITYP(5,4),IBIT(4)
      INTEGER*2            NAME(3)
      INTEGER              WARN,WARF,TERM,PRINTR
      EQUIVALENCE          (IBIT(1),WARN),(IBIT(2),WARF),(IBIT(3),TERM)
      DATA       ITYP      /'WARN','ING ','    ','    ','    ',
     *                      'WARN','ING(','WITH',' FIX',')   ',
     *                      'TERM','INAL','    ','    ','    ',
     *                      'NON-','DEFI','NED ','    ','    '/,
     *           IBIT      / 32,64,128,0/
      DATA       PRINTR    / 6/
      IER2=IER
      IF (IER2 .GE. WARN) GO TO 5
C                                      NON-DEFINED
      IER1=4
      GO TO 20
    5 IF (IER2 .LT. TERM) GO TO 10
C                                      TERMINAL
      IER1=3
      GO TO 20
   10 IF (IER2 .LT. WARF) GO TO 15
C                                      WARNING(WITH FIX)
      IER1=2
      GO TO 20
C                                      WARNING
   15 IER1=1
C                                      EXTRACT 'N'
   20 IER2=IER2-IBIT(IER1)
C                                      PRINT ERROR MESSAGE
      WRITE (PRINTR,25) (ITYP(I,IER1),I=1,5),NAME,IER2
   25 FORMAT(' *** I M S L(UERTST) ***  ',5A4,4X,3A2,4X,I2)
      RETURN
      END
```

```
      SUBROUTINE BSLJZ(X,FJ,NMAX,A,ND,IERR,FJAPRX,RR)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION FJ(1),FJAPRX(1),RR(1)
      NMAXT=NMAX
      IF(NMAXT.GE.0)GO TO 30
      IF(DABS(A).LE.1.0D-15)GO TO 10
      GO TO 20
 10   IERR=4
      RETURN
 20   NMAXAB=IABS(NMAXT)
      NMAXT=1
 30   IF(A.GT.0.0)GO TO 40
      IF(DABS(A).LE.1.0D-15)GO TO 40
      IERR=1
      RETURN
 40   IF(A.LT.1.0)GO TO 70
      IERR=2
      RETURN
 70   IF(X.GT.0.0)GO TO 130
      IERR=3
      RETURN
 130  IERR=0
      EPSLON=.5D0*10.**(-ND)
      NMP1=NMAX+1
      DO 160 N=1,NMP1
 160  FJAPRX(N)=0.0
      SUM=(X/2.)**A/DGAMMA(1.+A)
      D1=2.3026D0*ND+1.3863D0
      IF(NMAXT.LE.0)GO TO 230
      Y=.5D0*D1/NMAXT
      CALL TZ(Y,TANS)
      R=NMAXT*TANS
      GO TO 240
 230  R=0.0
 240  Y=.73576D0*D1/X
      CALL TZ(Y,TANS)
      S=1.3591D0*X*TANS
      IF(R.GT.S)GO TO 280
      NU=1+IDINT(S)
      GO TO 290
 280  NU=1+IDINT(R)
 290  M=0
      FL=1.
      LIMIT=(NU/2)
 320  M=M+1
      FL=FL*(M+A)/(M+1.D0)
      IF(M.LT.LIMIT)GO TO 320
      N=2*M
      R=0.0
      S=0.0
 390  DENOM=2.*(A+N)/X-R
      IF(DABS(DENOM).LE.1.0D-15)DENOM=DENOM+1.0D-15
 430  R=1./DENOM
      NMOD2=MOD(N,2)
      IF(NMOD2.NE.0)GO TO 480
      FL=FL*(N+2.D0)/(N+2.*A)
      FLMBDA=FL*(N+A)
      GO TO 490
 480  FLMBDA=0.0
 490  S=R*(FLMBDA+S)
      IF(N.LE.NMAXT)RR(N)=R
```

```
      N=N-1
      IF(N.GE.1)GO TO 390
      FJ(1)=SUM/(1.+S)
      IF(NMAXT.EQ.0)GO TO 570
      DO 560 N=1,NMAXT
560   FJ(N+1)=RR(N)*FJ(N)
570   DO 640 N=1,NMP1
      IF(DABS((FJ(N)-FJAPRX(N))/FJ(N)).LE.EPSLON)GO TO 640
      DO 610 M=1,NMP1
610   FJAPRX(M)=FJ(M)
      NU=NU+5
      GO TO 290
640   CONTINUE
      IF(NMAX.GE.0)RETURN
      FJ(2)=2.*A*FJ(1)/X-FJ(2)
      IF(NMAXAB.EQ.1)RETURN
      DO 650 N=2,NMAXAB
650   FJ(N+1)=2.*(A-N)*FJ(N)/X-FJ(N-1)
      RETURN
      END
```

```
      SUBROUTINE BSCJZ(X,Y,U,V,NMAX,A,ND,IERR,UAPPRX,VAPPRX,RR1,RR2)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION U(100),V(100),UAPPRX(100),VAPPRX(100),RR1(100),
     1          RR2(100)
      IF(A.GE.0.0)GO TO 40
      IERR=1
      RETURN
40    IF(A.LT.1)GO TO 70
      IERR=2
      RETURN
70    IF(X.GT.0.0)GO TO 110
      IF(DABS(Y).LE.1.0D-14)GO TO 90
      GO TO 110
90    IERR=3
      RETURN
110   IF(NMAX.GE.0)GO TO 140
      IERR=4
      RETURN
140   IERR=0
      EPSLON=.5D0*10.**(-ND)
      NMP1=NMAX+1
      DO 200 N=1,NMP1
      UAPPRX(N)=0.0
200   VAPPRX(N)=0.0
      Y1=DABS(Y)
      RZ2=X**2+Y**2
      RZ=DSQRT(RZ2)
      IF(DABS(X).LE.1.0D-14)GO TO 290
      PHI=DATAN2(Y1,X)
      IF(X.LT.0.0) PHI=3.141592653589793D0 + PHI
      GO TO 300
290   PHI=1.570796326794896D0
300   C=DEXP(Y1)*(RZ/2.)**A/DGAMMA(1.+A)
      SUM2=A*PHI-X
      SUM1=C*DCOS(SUM2)
      SUM2=C*DSIN(SUM2)
      D1=2.3026D0*ND+1.3863D0
      IF(NMAX.GT.0)GO TO 380
      R=0.0
      GO TO 390
380   PARAM=.5D0*D1/NMAX
      CALL TZ(PARAM,TANS)
      R=NMAX*TANS
390   S=1.3591D0*RZ
      PARAM=.73576D0*(D1-Y1)/RZ
      CALL TZ(PARAM,TANS)
      IF(Y1.LT.D1)S=S*TANS
      IF(R.GT.S)GO TO 450
      NU=1+IDINT(S)
      GO TO 460
450   NU=1+IDINT(R)
460   N=0
      FL=1.
      C1=1.
      C2=0.
500   N=N+1
      FL=FL*(N+2.*A)/(N+1.D0)
      C=-C1
      C1=C2
      C2=C
      IF(N.LT.NU)GO TO 500
```

```
      R1=0.0
      R2=0.0
      S1=0.0
      S2=0.0
  610 C=(2.*(A+N)-X*R1+Y1*R2)**2+(X*R2+Y1*R1)**2
      R1=(2.*(A+N)*X-RZ2*R1)/C
      R2=(2.*(A+N)*Y1+RZ2*R2)/C
      FL=FL*(N+1.D0)/(N+2.*A)
      C=2.*(N+A)*FL
      FLAMB1=C*C1
      FLAMB2=C*C2
      C=C1
      C1=-C2
      C2=C
      S=R1*(FLAMB1+S1)-R2*(FLAMB2+S2)
      S2=R1*(FLAMB2+S2)+R2*(FLAMB1+S1)
      S1=S
      IF(N.GT.NMAX)GO TO 770
      RR1(N)=R1
      RR2(N)=R2
  770 N=N-1
      IF(N.GE.1)GO TO 610
      C=(1.+S1)**2+S2**2
      U(1)=(SUM1*(1.+S1)+SUM2*S2)/C
      V(1)=(SUM2*(1.+S1)-SUM1*S2)/C
      IF(NMAX.EQ.0)GO TO 850
      DO 840 N=1,NMAX
      U(N+1)=RR1(N)*U(N)-RR2(N)*V(N)
  840 V(N+1)=RR1(N)*V(N)+RR2(N)*U(N)
  850 IF(Y.LT.0.0)GO TO 860
      GO TO 880
  860 DO 870 N=1,NMP1
  870 V(N)=-V(N)
  880 DO 950 N=1,NMP1
      TEMP1=(U(N)-UAPPRX(N))**2
      TEMP1=TEMP1+(V(N)-VAPPRX(N))**2
      TEMP1=TEMP1/(U(N)**2+V(N)**2)
      IF(TEMP1.LE.EPSLON)GO TO 950
      DO 920 M=1,NMP1
      UAPPRX(M)=U(M)
  920 VAPPRX(M)=V(M)
      NU=NU+5
      GO TO 460
  950 CONTINUE
      RETURN
      END
      SUBROUTINE TZ(Y,TANS)
      REAL*8 Y,Z,P,TANS,DLOG
      IF(Y.GT.10.0)GO TO 40
      P=.00005794100*Y-.0017614800
      P=Y*P+.0208645D0
      P=Y*P-.12901300
      P=Y*P+.85777D0
      TANS=Y*P+1.10125D0
      RETURN
   40 Z=DLOG(Y)-.775D0
      P=(.775D0-DLOG(Z))/(1.0+Z)
      TANS=Y/((1.+P)*Z)
      RETURN
      END
```

104 (74)

REFERENCES

1. Baum, C.E., "On the Singularity Expansion Method for the Solution of Electromagnetic Interaction Problems," Interaction Note 88, Air Force Weapons Laboratory, Kirtland AFB, NM, December 1971.

2. Taylor, C. D., Crow, T. T., and Chen, K-T, "On the Singularity Expansion Method Applied to Aperture Penetration: Part I Theory," Interaction Note 134, Air Force Weapons Laboratory, Kirtland AFB, NM, May 1973.

3. Rahmat-Samii, Y. and Mittra, R., "Integral Equation Solution and RCS Computation of a Thin Rectangular Plate," Interaction Note 156, Air Force Weapons Laboratory, Kirtland AFB, NM, December 1973.

4. Bladel, J. Van, Electromagnetic Fields, McGraw-Hill, New York, pp. 385-387, 1964.

5. Baum, C. E., "Interaction of Electromagnetic Fields with any Object which has an Electromagnetic Symmetry Plane," Interaction Note 63, Air Force Weapons Laboratory, Kirtland AFB, NM, March 1971.

6. Butler, C. M., "Integral Equation Solution Methods," in "Wire Antennas and Scatterers," Short Course Notes, University of Mississippi, April 1972. (See also IEEE Trans., v. AP-20, pp. 731-736, 1972.)

7. Umashankar, K.R., "Transient Scattering by a Thin Wire in Free Space and Above Ground Plane Using the Singularity Expansion Method," Interaction Note 236, August 1974. (See also F.M. Tesche, Interaction Note 102, April 1972.)

8. Davis, W. A., "Numerical Solutions to the Problem of Electromagnetic Radiation and Scattering by a Finite Cylinder," Ph.D. Thesis, University of Illinois, 1974.

9. Dunaway, O. C., "A Numerical Solution for the Distribution of Time-Harmonic Electromagnetic Fields in an Arbitrary Shaped Aperture in a Ground Screen," M.S. Thesis, University of Mississippi, 1974.

☆U.S. GOVERNMENT PRINTING OFFICE: 1975—A-0298/(42) 21