

AD-A050 256

NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA

F/6 20/14

'INVERT', A COMPUTER PROGRAM FOR OBTAINING D-REGION ELECTRON DE--ETC(U)

NOV 77 D G MORFITT, C H SHELLMAN

DNA-MIPR-77-521

UNCLASSIFIED

NOSC-IR-782

NL

1 of 2

AD
A050256



AD A 050256

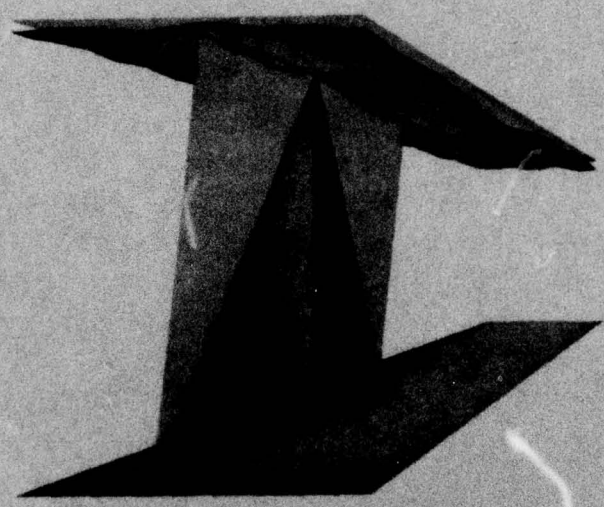
12 &

"INVERT", A COMPUTER PROGRAM FOR OBTAINING D-REGION ELECTRON DENSITY PROFILES FROM VLF REFLECTION COEFFICIENTS.

Interim Report No. 782

D.G. Morfitt and C.H. Shellman

AD No. [unclear]
DDC FILE COPY



DDC
RECEIVED
FEB 22 1978
A

**NAVAL OCEAN SYSTEMS CENTER
ELECTROMAGNETIC PROPAGATION DIVISION
SAN DIEGO, CALIFORNIA 92152**

30 NOVEMBER 1977

This work sponsored by the Defense Nuclear Agency under:
Subtask Code S99QAXHB042 and Work Unit 20
NOSC Problem MP20

Approved for public release; distribution is unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Interim Report No. 782	2. GOVT ACCESSION NO. NOSC-IR-782	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) INVERT, A Computer Program For Obtaining D-Region Electron Density Profiles from VLF Reflection Coefficients		5. AUTHOR(S) Research Rep.
6. PERFORMING ORG. REPORT NUMBER		7. CONTRACT OR GRANT NUMBER(S) DNA MIPR 77-521, 78-504
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Ocean Systems Center 271 Catalina Blvd. San Diego, CA 92152		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DNA S99QAXH B042 NOSC MP20 B042
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Nuclear Agency, RAAE Washington, D.C. 20305		12. REPORT DATE 30 November 1977
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) DNA-MIPR-77-521, DNA-MIPR-78-504		13. NUMBER OF PAGES 192 188p.
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution is unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) VLF Propagation Reflection coefficients Electron density profiles Computer program Ionosphere Full-wave Inversion		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) INVERT is a FORTRAN computer program designed to study the feasibility of obtaining electron density distributions of the D-region of the ionosphere. These electron density profiles would be derived from measurements of ionospheric reflection coefficients. The radio propagation frequencies would be limited to the VLF band, particularly 3-20 kHz. This report contains a discussion of the analytical approach taken in INVERT, a FORTRAN listing of the program, instructions for using the program and some sample calculations using simulated data. The program has not been used with real data.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102 LF 014 6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

393 259

JOB

TABLE OF CONTENTS

		Page #
	ABSTRACT	1
I	INTRODUCTION	2
II	INVERSION THEORY	5
	A. Reflection Coefficient Data	5
	B. Parameterization of the Electron Density Profile	6
	C. Computed Reflection Coefficients	6
	D. Fundamentals of the Inversion Procedure	7
	E. Assumptions Applied in the Inversion Procedure	11
	F. The Complex Reflection Coefficient Plane	15
	G. The Transformation Functions, $g(\omega_i)$ and $G(\omega_i)$	17
	H. Interpolation in the $R(\omega)$ and $r(\omega)$ Planes to Obtain $(dR/d\omega)$ and $(dr/d\omega)$	20
	I. Continuous Phase Terms, ϕ_g and ϕ_G , in the r and R Planes	22
	J. The Inversion Equation	26
III	SEQUENCE OF PROGRAM EXECUTION	29
IV	STRUCTURE AND DESCRIPTION OF THE INVERT COMPUTER PROGRAM	43
	A. Program Structure	43
	B. Description of Computer Program Routines	43
	C. Identification of NAMELIST Variables	49

TABLE OF CONTENTS Continued

	Page #
V EXECUTION OF INVERT USING SIMULATED DATA	55
A. Errors	55
B. The Auxiliary Program, FULLWAVE	55
C. The Auxiliary Program, RANDOM	61
D. Examples of INVERT Execution for Simulated Data	63
VI COMMENTS ON EXECUTION OF INVERT FOR MEASURED DATA	96
VII REFERENCES	98
APPENDICES	
A. FORTRAN LISTING OF THE "FULLWAVE" COMPUTER PROGRAM	99
B. FORTRAN LISTING OF THE "RANDOM" COMPUTER PROGRAM	126
C. FORTRAN LISTING OF THE "INVERT" COMPUTER PROGRAM	128

ABSTRACT

INVERT is a FORTRAN computer program designed to study the feasibility of obtaining electron density distributions of the D-region of the ionosphere. These electron density profiles would be derived from measurements of ionospheric reflection coefficients. The radio propagation frequencies would be limited to the VLF band, particularly 3-20 kHz. This report contains a discussion of the analytical approach taken in INVERT, a FORTRAN listing of the program, instructions for using the program and some sample calculations using simulated data. The program has not been used with real data.

ADDRESS FOR	
NTIS	Write Desk <input checked="" type="checkbox"/>
DDC	But. Serv. <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
DISTRIBUTION AVAILABILITY CODES	
Dist.	AVAIL. ORD. OR SPECIAL
A	

I. INTRODUCTION

The nature of the lowest ionosphere, the D-region, has been the objective of Government sponsored research for more than a decade. The principal interest stems from the global requirement for reliable strategic communications at very low radio frequencies (VLF 3-30 kHz).

Numerical methods are presently available for calculating VLF systems performance capabilities if the height distributions of the electron density and the collision frequency of the lower ionosphere are known. The problem is, however, to develop a procedure that will result in a reasonable representation of these required profiles.

Radio sounding at VLF is an often-used method for exploring the D-region. This region is considered to range from 50 km to about 100 km above the earth's surface. Because of the low-electron densities characteristic of these ionospheric heights, usual methods, such as high frequency (hf) ionosounds provide no useful information. For this same reason, data from rocket soundings also provide little detail at these heights.

The measured data parameters, acquired from VLF ionospheric sounders, are the ionospheric reflection coefficients. With special sounding techniques, such as described in reference 1, it is possible to obtain reflection coefficient data simultaneously at many frequencies over the VLF band. Also, since the theory of radio wave reflection from the ionosphere is well developed, reference 2, it is possible to calculate the characteristic reflection coefficients of radio waves propagated in models of the ionosphere. The goal has been to obtain the height distributions of the ionospheric factors by the simultaneous application of these two procedures, (i.e. sounder measurements and theoretical calculations). Such an approach would be known as profile inversion.

Early work in ionospheric inversion used trial-and-error techniques to deduce profiles. That is, values for the ionospheric properties were assumed and the ionospheric reflection coefficients calculated; these results were then compared with experimental values, and if the agreement was not satisfactory, the calculations were performed again with different values used for the ionospheric properties. The goodness of fit between calculations and experimental data were determined by a subjective estimate of the investigator.

The approach taken in the INVERT computer program is to attempt an analytical determination of the ionospheric profiles without the use of trial-and-error. The procedure begins with ionospheric reflection coefficient data and seeks to deduce the required electron density profile by finding the rate of change of the reflection coefficients with respect to changes at each point in the profile. Ideally the end result of the INVERT inversion procedure will be a profile which contains all the detail justified by the data, but which does not contain spurious data which would represent fitting to error in the data.

The fundamental ideas of the INVERT inversion technique have been previously described in reference 3. For this reason, several references will be made to that report rather than to reproduce the material here.

This report is divided into six major sections other than the introduction. Section II presents a discussion of the theory used in the inversion process. Section III traces the execution of the INVERT procedure. Section IV gives a description of the INVERT computer program. Section V discusses execution of INVERT for simulated data while Section VI is for measured data. The Appendices A, B, and C give FORTRAN listings of the computer programs.

The notation used throughout this report is as described below, where in general the subscript convention used is that j and j' refer to the "n" layers of the electron density profile while i and i' refer to the "m" data parameters. Note that most equations are written so that subscripted variables are thought of as matrices or vectors.

The notation is:

Square matrix:

$$\vec{A} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} = \underset{j \downarrow}{j \uparrow} \begin{pmatrix} & j' \rightarrow \\ & a_{jj'} \\ & \end{pmatrix} \quad (A)$$

Rectangular matrix:

$$\vec{B} = \begin{bmatrix} b_{11} & \dots & b_{1m} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ b_{n1} & \dots & b_{nm} \end{bmatrix} = \underset{j \downarrow}{i \rightarrow} \begin{bmatrix} & i \\ & b_{ji} \\ & \end{bmatrix} \quad (B)$$

Column vector:

$$\vec{C} = \begin{pmatrix} c_1 \\ \vdots \\ \vdots \\ c_n \end{pmatrix} = \underset{j \downarrow}{j \rightarrow} \begin{pmatrix} c_j \end{pmatrix} \quad (C)$$

Row vector:

$$\vec{D} = \overbrace{d_1 \dots d_n} = \overbrace{d_{j'}}^{j' \rightarrow} \quad (D)$$

II. INVERSION THEORY

The inversion problem is to begin with reflection coefficient data and deduce the responsible electron density profile.

A. Reflection Coefficient Data

Very low frequency (VLF) reflection coefficient data may be obtained from steep incidence sounders. Reflection of a VLF wave at near-vertical incidence is described by the reflection matrix:

$$\tilde{R} = \begin{pmatrix} {}_{\parallel}R_{\parallel} & {}_{\perp}R_{\parallel} \\ {}_{\parallel}R_{\perp} & {}_{\perp}R_{\perp} \end{pmatrix} \quad (1)$$

where the first subscript refers to the polarization of the up going wave with respect to the plane of incidence and the second subscript refers to the polarization of the down coming wave. Note that the matrix elements are complex numbers.

The reflection coefficient ratio used in the inversion procedure is given in equation (66) of reference 3 as:

$$R(\omega_i) = \left\{ \frac{- {}_{\perp}R_{\perp}(\theta) + j {}_{\perp}R_{\parallel}(\theta)}{- {}_{\perp}R_{\perp}(\theta) - j {}_{\perp}R_{\parallel}(\theta)} \right\}_i \quad (2)$$

where $i = 1, \dots, m$

The terms of equation (2) are further identified as:

ω_i -- The transmitted frequency.

m -- The total number of transmitted frequencies.

θ -- The incident angle of the radio wave on the ionosphere.

j -- $\sqrt{-1}$

$R(\omega_i)$ -- The value of the reflection ratio at the i -th frequency as obtained from data.

The reflection coefficients of equations (1) and (2) are functions of the earth's magnetic field. As the sounder site approaches the geomagnetic

equator, the cross term (i.e. ${}_{\perp}R_{\parallel}$) tends to zero and thus the reflection ratio, $R(\omega_i)$, becomes equal to one. When this situation occurs, the inversion technique yields no information. The most useful data for determining profiles is obtained, therefore, at locations with large magnetic dip angles (i.e. the magnetic poles).

B. Parameterization of the Electron Density Profile

The procedure is to represent the electron density profile by a series of short exponential segments. The common logs of the electron densities at the heights, z_j , where the segments join, are taken to be the unknown parameters of the profile and are denoted by:

$$\alpha_j = \alpha(z_j) = \log_{10} N_e(z_j) \quad (3)$$

where $j = 1, 2, \dots, n$.

and where N_e is the electron density (electrons/cc) at height z (km).

C. Computed Reflection Coefficients

The ionospheric reflection coefficients given in equation (1) may be computed for any given electron density profile by the "full-wave" method given in reference 4. These reflection coefficients are the result of the integration:

$$R = \int_b^a \left(\frac{dR}{dz} \right) dz \quad (4)$$

As discussed previously for measured data, a useful representation of the computed reflection coefficient is:

$$r(\omega_i) = \left\{ \frac{-{}_{\perp}R_{\perp}(\theta) + j {}_{\perp}R_{\parallel}(\theta)}{-{}_{\perp}R_{\perp}(\theta) - j {}_{\perp}R_{\parallel}(\theta)} \right\}_i \quad (5)$$

$i = 1, \dots, m$

In this instance the reflection coefficients, ${}_1R_{\perp}$ and ${}_1R_{\parallel}$ are computed from the integration of equation (4) and thus " $r(\omega_i)$ " is the computed value of the reflection coefficient parameter at the i -th frequency.

D. Fundamentals of the Inversion Procedure

The approach taken in searching for a best-fit electron density profile to data was originally presented in reference 5. The basic requirement is that the profile is to be linear (on a log scale) unless the data gives information to justify a given degree of detail. To carry out this approach, two functions are defined. The first is a measure of deviation of computed reflection coefficients from measured data, defined as:

$$s = \sum_i^m \left\{ (r_i - R_i) / \sigma_i \right\}^2 \quad (6)$$

$$i = 1, \dots, m$$

where

m -- is the total number of propagation frequencies.

r_i -- are the computed values of the reflection coefficient ratio from equation (5).

R_i -- are the "measured" reflection coefficients in the form of equation (2).

σ_i -- are the uncertainties in the "measured" values. These are also defined as the "expected" errors.

The second function is a measure of detail, or curvature, in the profile and is defined as:

$$c = \int_b^a \left\{ \frac{d^2 \alpha(z)}{dz^2} \right\} dz \quad (7)$$

where z is the height variable, $\alpha(z)$ is the common log of the electron density profile at height, z , and the integral is taken over the range of the profile.

Equation (7) defines the entire profile curve, $\alpha(z)$. In practice $\alpha(z)$ can only be determined at a finite number of points, $(\alpha(z_j) = \alpha_j)$, and therefore the form of the profile is given by equation (3). For simplicity it will be assumed that the α_j 's are equally spaced at intervals, Δz . The second derivative at z_j is then approximated by:

$$\frac{d^2\alpha}{dz^2} = \left[\left(\frac{\alpha_{j-1} - \alpha_j}{\Delta z} \right) - \left(\frac{\alpha_j - \alpha_{j+1}}{\Delta z} \right) \right] \Delta z \quad (8)$$

In application, the curvature function "c" of equation (7) is expressed in terms of summation over profile segments. The equation becomes:

$$c \approx \left\{ \sum_{j=2}^{n-1} (\alpha_{j-1} - 2\alpha_j + \alpha_{j+1})^2 \right\} (\Delta z)^{-3} \quad (9)$$

where n is the number of points α_j .

The expression in brackets represents the second derivative at height z_j . The slab summation form of "c" approaches the integral of equation (7) as the slabs are made sufficiently thin.

Equation (9) can be written in matrix notation as:

$$c \approx (\Delta z)^{-3} \cdot \overbrace{\alpha_{j'}}^{j' \rightarrow} \cdot \underset{j \downarrow}{j'} \left(C_{j',j} \right) \cdot \underset{j \downarrow}{j} \left(\alpha_j \right) \quad (10)$$

Equation (13) can be re-written (as derived in section II-C of reference 3) as:

$$\underline{K}(\lambda, r(\omega_i)) \cdot \vec{(\Delta\alpha_j)} = \vec{T}(\lambda, \alpha_j^0, r(\omega_i), R(\omega_i)) \quad (14)$$

where: $j = 1, \dots, n$ for the segments of the electron density profile.

and where:

\underline{K} = The coefficient matrix, and is a function of the trade-off parameter, " λ "; and also a function of the computed ionospheric reflection coefficients $r(\omega_i)$.

\vec{T} = A vector which is a function of the above " λ ", and r . It is also a function of the measured ionospheric reflection coefficients, $R(\omega_i)$, and a function of the previous electron density profile, α_j^0 .

$\vec{(\Delta\alpha_j)}$ = A vector which represents the modifications to be added to each previous electron density profile during each step of the iteration scheme.

ω_i = The propagation frequency, $i=1, \dots, m$.

The solutions to equation (14) are obtained by an iterative procedure. First, an initial profile, α_j^0 ($j=1, \dots, n$) is chosen and a value is assigned to the trade-off parameter, " λ ". These values, along with functions of the full-wave calculations, $r(\omega_i)$, and data values, $R(\omega_i)$, are substituted and a solution $\vec{(\Delta\alpha_j)}$ is obtained as:

$$\vec{(\Delta\alpha_j)} = \left(\underline{K}^{-1} \right) \cdot \vec{T} \quad (15)$$

These values of the $(\Delta\alpha_j)$'s are added to the (α_j^0) 's to get a new profile $(\alpha_j^{\rightarrow})_{new}$. The sequence then continues with $(\alpha_j^{\rightarrow})_{new}$ assuming the role of (α_j^0) ,

and a new value assigned to " λ ". This leads to a new solution of $(\Delta\vec{\alpha}_j)$ and thus to a new electron density profile given by the relation:

$$(\vec{\alpha}_j)_{\text{new}} = (\vec{\alpha}_j^0)_{\text{previous}} + (\Delta\vec{\alpha}_j) \quad (16)$$

E. Assumptions Applied in the Inversion Procedure

In the inversion procedure several assumptions concerning the ionospheric parameters are made. In particular all parameters will be considered to be known except the electron density profile, which will be unknown, and is to be found from the data. The geomagnetic field parameters will be taken to be known, and the magnetic azimuth will be restricted to be either 90° or 270° . As is the usual case at VLF, ions are considered to have negligible effect on propagation and are omitted from the model. The collision frequency profile is taken to be known and is constrained to be of exponential form:

$$\nu = \nu_0 \exp(-az) \quad (17)$$

Additional assumptions (i.e. constraints) on the form of the desired electron density profile are listed below and are illustrated in figure 1 where the electron density (N_e) is shown as a function of ionospheric height in terms of $\log_{10} N_e$.

(1) "Stop" constraints are introduced at both the top and bottom of the electron density profile. These correspond to limiting values of electron density which the profile may assume at these heights. At the top of the profile the "stop" defines the minimum value of electron density which the profile may assume. At the bottom, the "stop" defines the maximum value for that height.

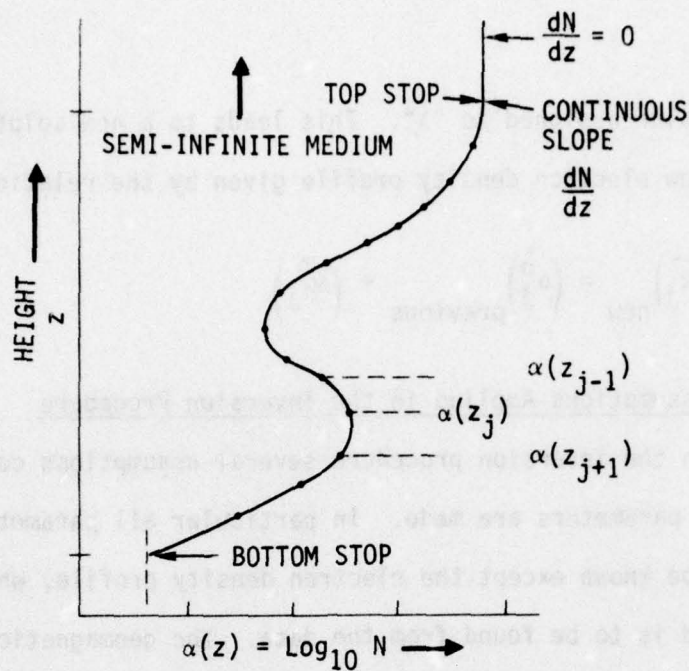


Figure 1. Segmented Electron Density Profile,
 $\alpha(z_j)$'s Showing Constraints.

(2) It is assumed that there are no electrons below the chosen lowest height of the profile. Above the chosen top height of the profile the electron density is assumed to be of constant value equal to the electron density at the chosen top height. This latter property is usually referred to as a semi-infinite medium.

(3) The profile is constrained to be reasonably smooth but otherwise consistent with the reflection coefficient data. Note that this implies that the profile approaches a slope of $(dN/dz) = 0$ near the top height which is just below the semi-infinite medium.

Inequalities are used as constraints on the end points of the $\alpha(z)$ profile. At the top of the profile the electron density is constrained to be at least as great as a chosen fixed value. At the bottom height of the

profile, the electron density is constrained to be no greater than a chosen fixed value. Thus:

$$\begin{aligned}\alpha(a) &= \alpha(z_1) \geq \alpha_{\text{Top Stop}} \\ \alpha(b) &= \alpha(z_n) \leq \alpha_{\text{Bot Stop}}\end{aligned}\tag{18}$$

In addition, in order to ensure a continuous slope, (dN_e/dz) , leading upward to the semi-infinite medium, the slope at the top of the profile ($z = a = z_1$) is constrained to approach that of the semi-infinite medium. In terms of the segmented $\alpha(z)$ curve, that is, the segmented electron density profile, this is implemented by adding the term:

$$(\alpha_2 - \alpha_1)^2 (\Delta z)^{-3}\tag{19,a}$$

to the summation representing the curvature function. That is, "c" is now defined as:

$$c = \left\{ (\alpha_2 - \alpha_1)^2 + \sum_{j=2}^{n-1} (\alpha_{j-1} - 2\alpha_j + \alpha_{j+1})^2 \right\} (\Delta z)^{-3}\tag{19,b}$$

Compare this result to equation (9).

With these sets of constraints, the equation to be solved is a modified form of equation (13). In particular the matrix:

$$j' \downarrow \begin{pmatrix} & & j \\ (1+p_T) & -1 & 0 \dots 0 \\ -1 & +1 & 0 \dots 0 \\ 0 & 0 & 0 \dots 0 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \dots p_B \end{pmatrix}\tag{20}$$

is added to the unmodified C matrix of equation (11). That is, the original C matrix terms are modified to be:

$$\left\{ \begin{array}{l} C_{11} = 1 + (1 + p_T) = 2 + p_T \\ C_{12} = -2 + (-1) = -3 \\ C_{21} = -2 + (-1) = -3 \\ C_{22} = 5 + (+1) = 6 \\ C_{nn} = 1 + p_B = 1 + p_B \end{array} \right. \quad (21)$$

The new matrix C^{new} is defined as

$$C^{\text{new}} = \begin{pmatrix} (2+p_T) & -3 & 1 & 0 & \dots & \dots & \dots \\ -3 & 6 & -4 & 1 & 0 & & \\ 1 & -4 & 6 & -4 & 1 & & \circ \\ & & & & & & \\ \circ & & & & & & \\ & & & 1 & -4 & 6 & -4 & 1 \\ & & & & & 1 & -4 & 5 & -2 \\ & & & & & & & 1 & -2 & (1+p_B) \end{pmatrix} \quad (22)$$

A vector, \vec{V} , is also defined to be:

$$j' \downarrow (V_{j', \text{new}}) = j' \downarrow \begin{pmatrix} p_T \cdot \alpha_{\text{Top Stop}} \\ \vdots \\ p_B \cdot \alpha_{\text{Bot Stop}} \end{pmatrix} \quad (23)$$

and is a consequence of the added constraints.

The values of the inequalities of equation (18) are defined as follows:

- (a) P_T $\left\{ \begin{array}{l} = 0, \text{ if in the solution for the } (\Delta\alpha_j)\text{'s of equation (15), the} \\ \text{top end of the profile is inclined to move to the right} \\ \text{of the stop.} \\ = \text{ a very large positive value if the top end of the profile} \\ \text{is inclined to move to the left of the stop.} \end{array} \right.$
- (b) P_B $\left\{ \begin{array}{l} = 0 \text{ if, in the solution for the } (\Delta\alpha_j)\text{'s of equation (15) the} \\ \text{bottom end of the profile is inclined to move to the left} \\ \text{of the stop.} \\ = \text{ a very large value if the bottom end of the profile is} \\ \text{inclined to move to the right of the stop.} \end{array} \right.$

F. The Complex Reflection Coefficient Plane

Calculation performed using different electron density profiles will produce different values for the set of reflection coefficient ratios, $r(\omega_i)$.

The reflection coefficient ratio values are illustrated in figure 2 as plotted in the complex R (or r) plane. Here the frequency, ω , serves as the parametric variable. The data values, $R(\omega_i)$ are shown in comparison to two sets of $r(\omega_i)$ values. The curve $[r(\omega)]_1$ results from applying equation (4) and equation (5) to a given electron density profile. The curve $[r(\omega)]_2$ presents the results obtained from a second electron density profile. It is pointed out, that because of the constraints placed on the trial electron density profiles, not all forms of the $r(\omega)$ curve can be generated simply by varying the profile.

The problem to be solved with the inversion procedure is to find that electron density profile, which when used in the full-wave computation of reflection coefficients, will produce a match to the data, $R(\omega_i)$.

Such a match is illustrated in figure 3. Because of error in the data

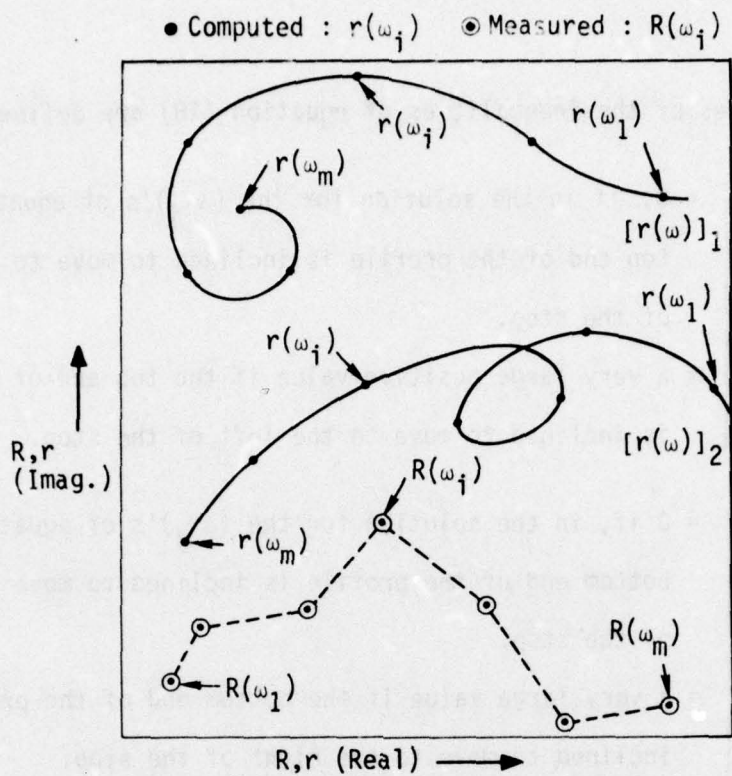


Figure 2. Data Values $R(\omega_i)$ and Computed Values $r(\omega_i)$ as Plotted in the Complex Plane.

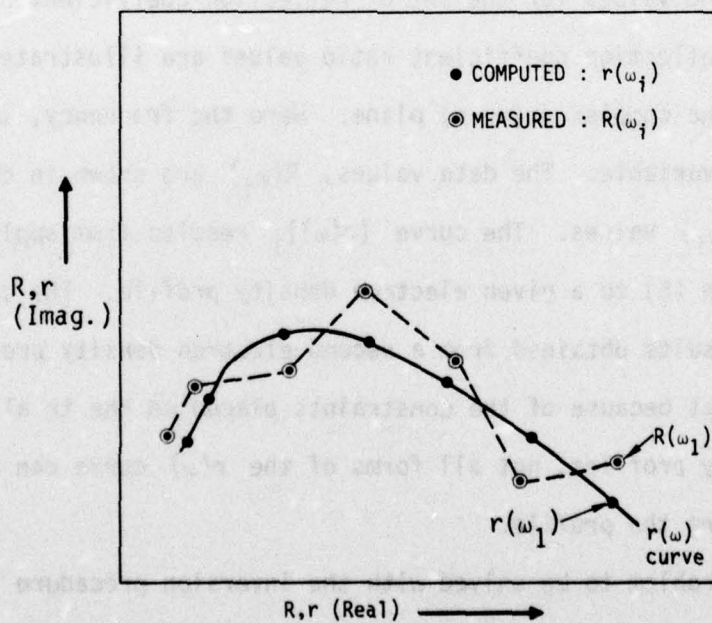


Figure 3. Matched Comparison of $R(\omega_i)$ Reflection Coefficient Data and $r(\omega)$ Computations.

and because of the constraints put on the ionospheric model, the data cannot be matched exactly.

The fact that an attempt is made to match the curves in real and imaginary parts of $R(\omega_i)$ and $r(\omega_i)$ implies that the spacing, in the R-r plane between frequency points (ω_i and ω_{i+1} , etc.) of $r(\omega_i)$ and $r(\omega_{i+1})$, etc., must also match the spacing between $R(\omega_i)$ and $R(\omega_{i+1})$, etc.

G. The Transformation Functions, $g(\omega_i)$ and $G(\omega_i)$

In order to make convergence of equation (14) more likely, the following function of the ionospheric reflection coefficients is introduced. (See the discussion on pp 28 in section II-C of reference 3.) Define:

$$g_1 = r(\omega_1)$$

$$g_i = \ln \left(\frac{dr}{d\omega} \right)_i \quad (24)$$

$$i = 1, \dots, m$$

where

r_i is given by equation (5).

ω is the propagation frequency (kHz).

m is the total number of data frequencies.

Equation (24) may also be written as:

$$g_i = \ln \left| \left(\frac{dr}{d\omega} \right)_i \right| + j(\phi_g)_i \quad (25)$$

$$i = 2, \dots, m$$

and

$$(\phi_g)_i = ((\phi_g)_i \pm 2n_i \pi) \quad (26)$$

where (ϕ_g) is the principal part of the phase term and lies in the interval $-\pi < \phi_i \leq \pi$.

The value of n_i , in the phase term, is chosen for each frequency so as to ensure that the phase, ϕ_i , is continuous along the $r(\omega)$ curve.

The phase angle $(\phi_g)_i$ may also be written as:

$$(\phi_g)_i = \text{ARCTAN} \left(\frac{\text{Imag. } (dr/d\omega)_i}{\text{Real } (dr/d\omega)_i} \right) \quad (27)$$

The definition of the transformation function, as given by equation (25), allows for a complete definition of the $r(\omega)$ curve in terms of spacing (i.e., $|dr/d\omega|$), phase (i.e., ϕ) and the location of one point on the $r(\omega)$ curve, where for example the one frequency point is chosen to be the lowest frequency, ω_1 . An $r(\omega)$ curve illustrating the "g" transformation is given in figure 4.

The transformation function for the data values, $R(\omega_i)$, is also introduced as:

$$\left. \begin{aligned} G_1 &= R(\omega_1) \\ G_i &= 1n \left(\frac{dR}{d\omega} \right)_i \\ \text{or} \\ G_i &= 1n \left| \left(\frac{dR}{d\omega} \right)_i \right| + j(\phi_G)_i \\ i &= 2, \dots, m \end{aligned} \right\} \quad (28)$$

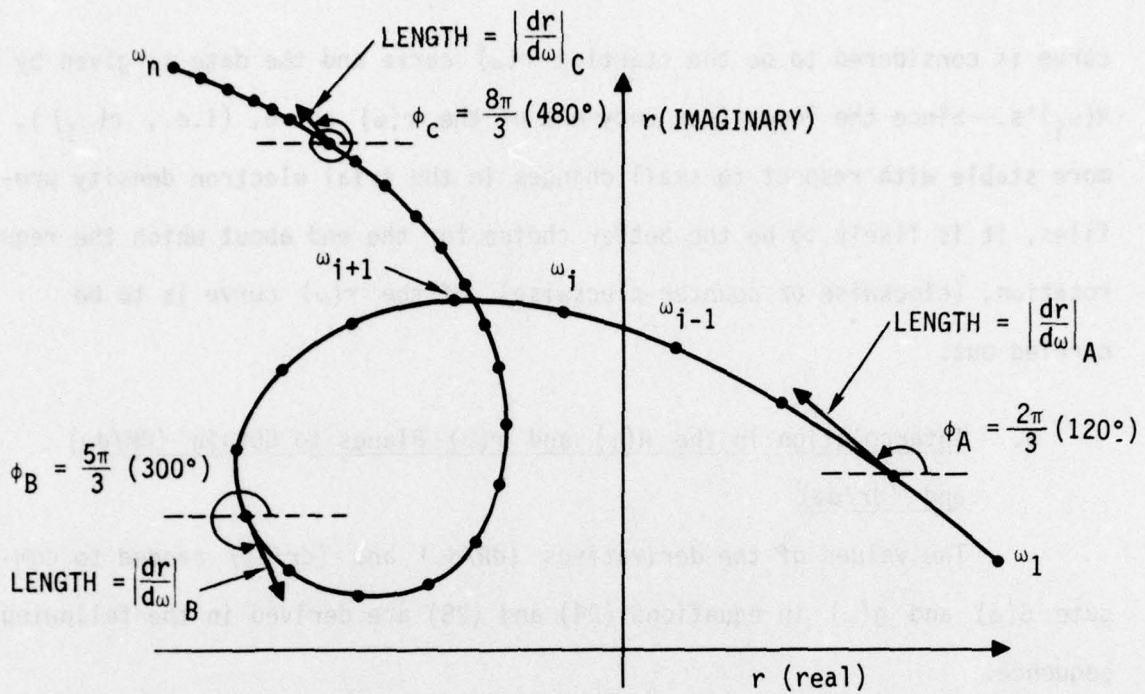


Figure 4. The $r(\omega)$ Curve Illustrating the Transformation Function, $g(\omega_i) = \ln (dr/d\omega)_i + j\phi_i$ (i.e., Spacing and Phase).

The phase, $(\phi_G)_i$ is written as:

$$(\phi_G)_i = \left((\phi_G)_i \pm 2 (n_G)_i \pi \right) \quad (29)$$

and

$$(\phi_G)_i = \text{ARCTAN} \left(\frac{\text{Imag. } (dR/d\omega)_i}{\text{Real } (dR/d\omega)_i} \right) \quad (30)$$

To obtain a match between an $r(\omega)$ curve and the data curve $R(\omega)$, as exemplified in figure 3, the required $r(\omega)$ curve may need to be completely reversed from the $r(\omega)$ curve obtained from the starting solution of the inversion scheme. This is the situation portrayed in figure 2 where the $[r(\omega)]_2$

curve is considered to be the starting $r(\omega)$ curve and the data is given by the $R(\omega_i)$'s. Since the lower frequency end of the $r(\omega)$ curve, (i.e., $r(\omega_1)$), is more stable with respect to small changes in the trial electron density profiles, it is likely to be the better choice for the end about which the required rotation, (clockwise or counter-clockwise), of the $r(\omega)$ curve is to be carried out.

H. Interpolation in the $R(\omega)$ and $r(\omega)$ Planes to Obtain $(dR/d\omega)$ and $(dr/d\omega)$

The values of the derivatives $(dR/d\omega)$ and $(dr/d\omega)$ needed to compute $G(\omega)$ and $g(\omega)$ in equations (24) and (28) are derived in the following sequence.

The set of transmitted frequencies, ω_i , ($i = 1, \dots, m$) is assumed to be a subset of possible frequencies, $\bar{\omega}_k$, ($k = 1, \dots, K$). K is larger than m . A reflection coefficient ratio (similar to $r(\omega_i)$) will be denoted for the frequency set, $\bar{\omega}_k$, by $\rho(\bar{\omega}_k)$. The defining equation is:

$$\rho(\bar{\omega}_k) = \sum_{i=1}^m b_{k,i} \cdot R(\omega_i) \quad (31)$$

Where the $\rho(\bar{\omega}_k)$'s are linear combinations of the $R(\omega_i)$'s with coefficients $(b_{k,i})$. The $\rho(\bar{\omega}_k)$ values are obtained by interpolation of the $R(\omega_i)$ values and equation (31) is the interpolation equation. The $(b_{k,i})$ are defined in section II-C of reference 3. The $(b_{k,i})$ are not functions of the numerical values of the $R(\omega_i)$ but depend only on the spacing of the ω_i frequencies and on the number of frequencies m . At any frequency value where ω_i is equal to an $\bar{\omega}_k$, the value of $\rho(\bar{\omega}_k)$ will be equal to the value of $R(\omega_i)$.

Figure 5 shows the R-plane representation of the interpolated values, $\rho(\bar{\omega}_k)$, and the data value, $R(\omega_i)$.

The derivative at any one interpolation point, $\bar{\omega}_k$, is defined in terms of the change in value of $\rho(\bar{\omega})$ at the interpolation points preceding and following that point. That is:

$$\left(\frac{d\rho(\omega)}{d\omega}\right)_k = \frac{(\rho(\bar{\omega}_{k+1}) - \rho(\bar{\omega}_{k-1}))}{2 \cdot \Delta\omega} \quad (32)$$

where $\Delta\omega$ is the interpolation increment in frequency.

Using equation (31) equation (32) is also written:

$$\left(\frac{d\rho(\omega)}{d\omega}\right)_k = (b_{k+1,i} - b_{k-1,i}) R(\omega_i) / 2\Delta\omega \quad (33)$$

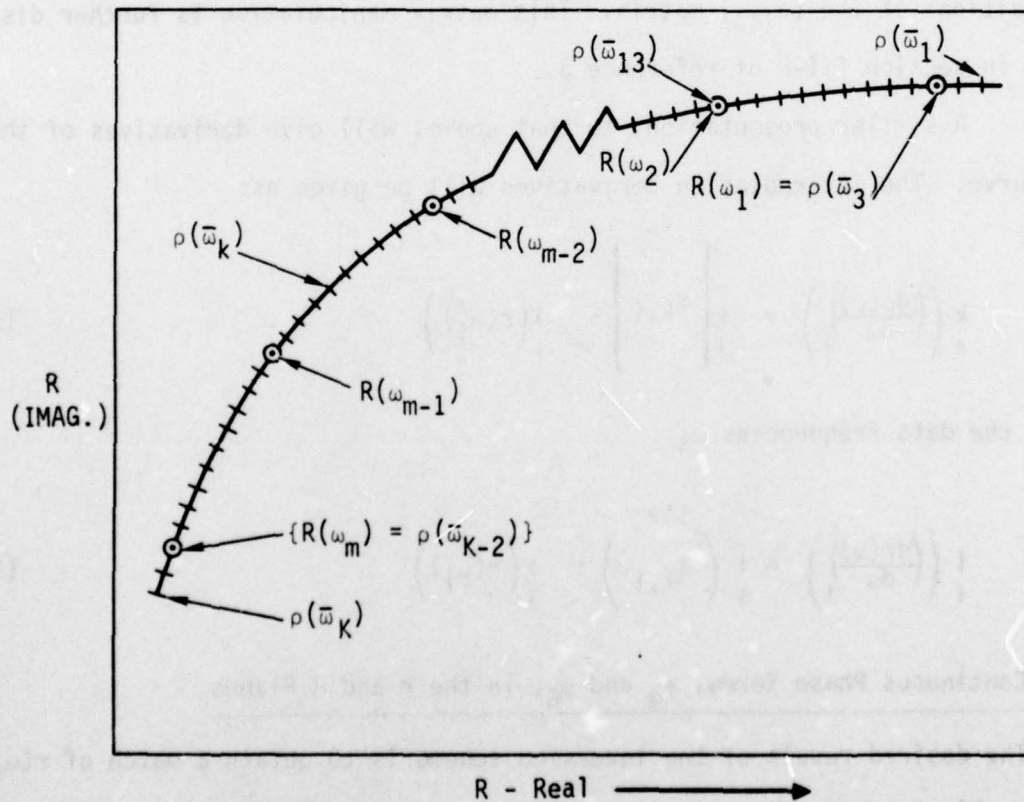


Figure 5. R-Plane Representation of Interpolated Values of $\rho(\omega_k)$ and Data Values $R(\omega_i)$.

and in matrix notation

$$\left. \begin{aligned} \underset{\downarrow}{k} \left(\left(\frac{d\rho(\omega)}{d\omega} \right)_k \right) &= \underset{\downarrow}{k} \left[\begin{matrix} i \rightarrow \\ \beta_{k,i} \end{matrix} \right] \cdot \underset{\downarrow}{i} \left(R(\omega_i) \right) \end{aligned} \right\} \quad (34)$$

where $\beta_{k,i} = (b_{k+1,i} - b_{k-1,i})$

At the data frequencies, ω_i , equation (34) simplifies to:

$$\underset{\downarrow}{i} \left(\left(\frac{dR(\omega)}{d\omega} \right)_i \right) = \underset{\downarrow}{i} \left(\begin{matrix} i \rightarrow \\ a_{i,i'} \end{matrix} \right) \cdot \underset{\downarrow}{i'} \left(R(\omega_i) \right) \quad (35)$$

where the $a_{i,i'}$ matrix is obtained from the $\beta_{k,i}$ matrix of equation (34) by placing only those rows of the $(\beta_{k,i})$ matrix which correspond to ω_i values into row positions of the $(a_{i,i'})$ matrix. This matrix manipulation is further discussed in section III-H of reference 3.

A similar presentation, to that above, will give derivatives of the $r(\omega)$ curve. The interpolation derivatives will be given as:

$$\underset{\downarrow}{k} \left(\left(\frac{dr(\omega)}{d\omega} \right)_k \right) = \underset{\downarrow}{k} \left[\begin{matrix} i \rightarrow \\ \beta_{k,i} \end{matrix} \right] \cdot \underset{\downarrow}{i} \left(r(\omega_i) \right) \quad (36)$$

and at the data frequencies ω_i :

$$\underset{\downarrow}{i} \left(\left(\frac{dr(\omega)}{d\omega} \right)_i \right) = \underset{\downarrow}{i} \left(\begin{matrix} i \rightarrow \\ a_{i,i'} \end{matrix} \right) \cdot \underset{\downarrow}{i'} \left(r(\omega_i) \right) \quad (37)$$

I. Continuous Phase Terms, ϕ_g and ϕ_G , in the r and R Planes

The desired result of the inversion scheme is to obtain a match of $r(\omega_i)$ values to $R(\omega_i)$ values. To accomplish this, it is required that the phase, ϕ_g , be continuous along the $r(\omega)$ curve. The same is true for the phase ϕ_G along the $R(\omega)$ curve.

From equation (25) the $g(\omega)$ transformation variable may be written in terms of the interpolated variables, $\rho(\bar{\omega}_k)$, as:

$$\bar{g}_k = 1n \left| \frac{d\rho}{d\omega} \right|_k + j \underbrace{\left((\phi_g)_k \pm 2 (\bar{n}_g)_k \pi \right)}_{(\phi_g)_k} \quad (38)$$

and from equation (28) the expression for $G(\omega)$

$$\bar{G}_k + 1n \left| \frac{d\rho}{d\omega} \right|_k + j \underbrace{\left((\phi_G)_k \pm 2 (\bar{n}_G)_k \pi \right)}_{(\phi_G)_k} \quad (39)$$

To satisfy the matching requirement between the $r(\omega)$ and $R(\omega)$ curves the values of $(\bar{n}_g)_k$ and $(\bar{n}_G)_k$ must be chosen for each frequency, $\bar{\omega}_k$, so that the phases ϕ_g and ϕ_G are continuous along the $r(\omega)$ and $R(\omega)$ curves. To insure continuous changes in phase along the curves the differences: $|\phi_k - \phi_{k-1}|$ and $|\phi_{k+1} - \phi_k|$ must be small.

The smoothing requirement for the continuous phase along the $r(\omega)$ curve leaves the value of $(\bar{n}_g)_k$ arbitrary for one frequency, e.g. the lowest frequency, ω_1 , which is $\bar{\omega}_3$. The value of $(\bar{n}_g)_{k=3}$ is then chosen so as to cause the $r(\omega)$ curve to rotate in the desired sense, (i.e. clockwise or counter clockwise) to obtain a match to the $R(\omega)$ data curve. The values of the phases $(\phi_g)_k$ and $(\phi_G)_k$ at the point $k=3$ will be identified as ϕ_D (for data phase) and ϕ_C (for computed phase) respectively.

Figure 6 illustrates some possible cases for the phase relations between the data curve, $R(\omega)$, and two examples of computed curves, $[r(\omega)]_1$ and $[r(\omega)]_2$. The phase values are identified as the direction assumed by a vector

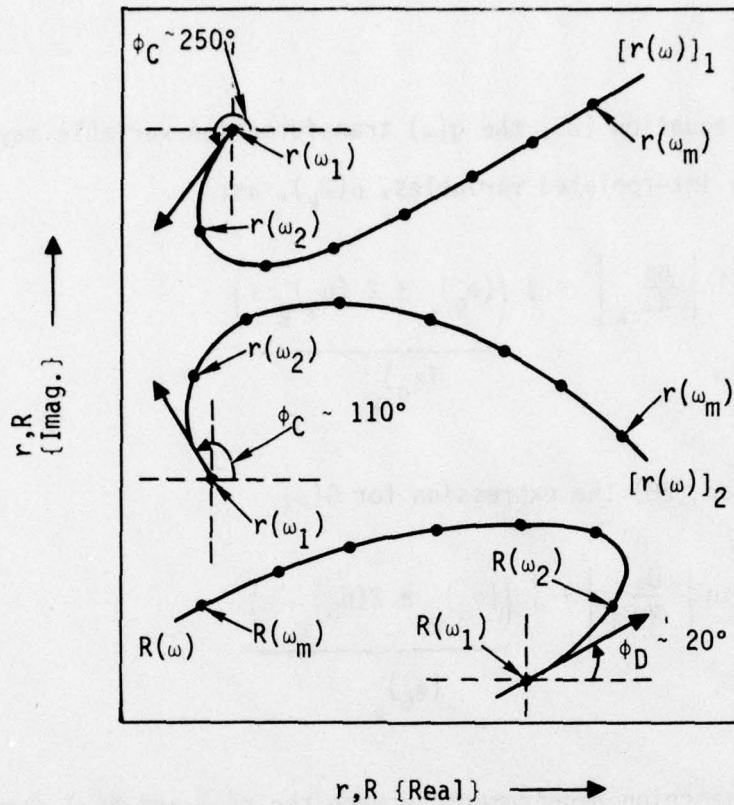
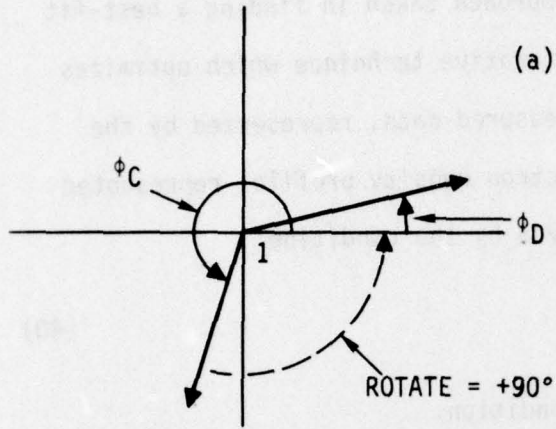
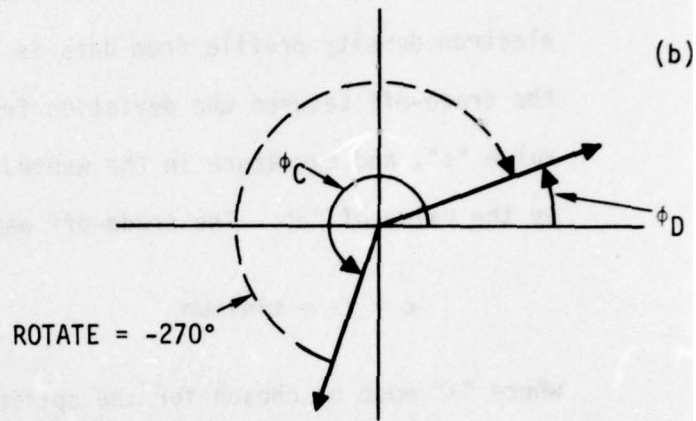


Figure 6. Examples of ϕ_D for Data Values, $R(\omega)$, and ϕ_C for Computed Values, $r(\omega)$.

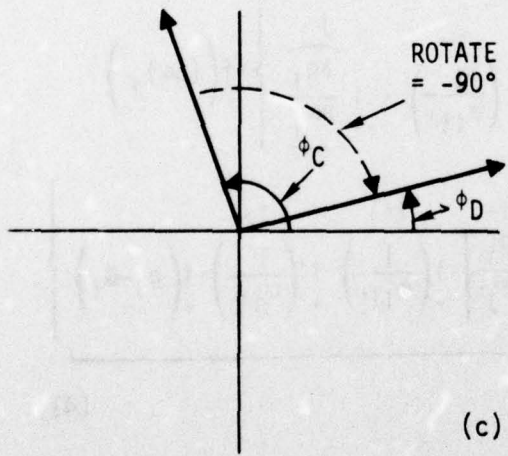
originating at the lowest frequency point, $r(\omega_1)$ (or $R(\omega_1)$) directed tangent to the $r(\omega)$ (or $R(\omega)$) curve as the curve extends toward the point $r(\omega_2)$ (or $R(\omega_2)$). There are two ways that the $[r(\omega)]_1$ (or $[r(\omega)]_2$) curve can be transformed to match the $R(\omega)$ curve. These are a clockwise rotation or a counter-clockwise rotation of the $r(\omega)$ curve about the point $r(\omega_1)$. This rotation is accomplished in the INVERT computer program by the variable ROTATE. A positive value assigned to ROTATE results in a counter-clockwise rotation while a negative value gives a clockwise rotation. Examples of the choices of values for ROTATE are presented in figure 7. The procedure to be used in executing the program is to try one direction of rotation and, if it does not prove successful, try the other direction.



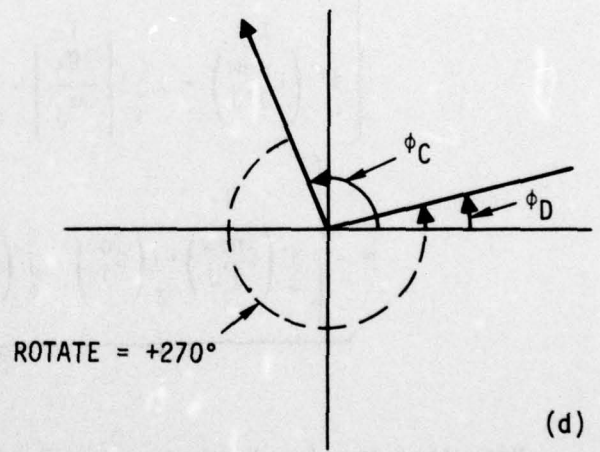
Counterclockwise rotation
for the curve $[r(\omega)]_1$
toward the curve $[R(\omega)]$.



Clockwise rotation
for the curve $[r(\omega)]_1$
toward the curve $[R(\omega)]$.



Clockwise rotation
for the curve $[r(\omega)]_2$
toward the curve $[R(\omega)]$.



Counterclockwise rotation
for the curve $[r(\omega)]_2$
toward the curve $[R(\omega)]$.

Figure 7. Examples of the Choices of Values for the ROTATE Variable to Cause the Curve $r(\omega)$ to Rotate Toward the $R(\omega)$ Curve.

J. The Inversion Equation

As stated in section II,D, the approach taken in finding a best-fit electron density profile from data is an iterative technique which optimizes the trade-off between the deviation from measured data, represented by the value "s", and curvature in the wanted electron density profile, represented by the value of "c". The trade-off was given by the condition:

$$c + \lambda s \rightarrow \text{minimum} \quad (40)$$

where "λ" must be chosen for the optimum condition.

The actual equation to be solved in order to obtain the desired electron density profile is derived in section II-C of reference 3 as:

coefficient matrix, \underline{K}

$$\left\{ \begin{matrix} j \rightarrow \\ \downarrow \end{matrix} \left(C_{j'j}^{\text{new}} \right) + \lambda \begin{matrix} j \rightarrow \\ \downarrow \end{matrix} \left[\frac{\partial g_j}{\partial \alpha_j} \right] \cdot \begin{matrix} i \rightarrow \\ \downarrow \end{matrix} \left(\frac{1}{\sigma_{ii'}} \right) \cdot \begin{matrix} i' \rightarrow \\ \downarrow \end{matrix} \left(\frac{1}{\sigma_{ii'}} \right) \cdot \begin{matrix} j \rightarrow \\ \downarrow \end{matrix} \left(\frac{\partial g_j}{\partial \alpha_j} \right) \right\} \cdot \begin{matrix} j \rightarrow \\ \downarrow \end{matrix} \left((\Delta \alpha)_j \right)$$

$$= - \left\{ \begin{matrix} j \rightarrow \\ \downarrow \end{matrix} \left(C_{j'j}^{\text{new}} \right) \cdot \begin{matrix} j \rightarrow \\ \downarrow \end{matrix} \left(\alpha_j^0 \right) - \begin{matrix} j \rightarrow \\ \downarrow \end{matrix} \left(V_j \right) + \lambda \begin{matrix} j \rightarrow \\ \downarrow \end{matrix} \left[\frac{\partial g_j}{\partial \alpha_j} \right] \cdot \begin{matrix} i \rightarrow \\ \downarrow \end{matrix} \left(\frac{1}{\sigma_{ii'}} \right) \cdot \begin{matrix} i' \rightarrow \\ \downarrow \end{matrix} \left(\frac{1}{\sigma_{ii'}} \right) \cdot \begin{matrix} i \rightarrow \\ \downarrow \end{matrix} \left(g_i - G_i \right) \right\}$$

T-vector

(41)

Note that the $(\Delta \alpha_j)$ matrix and the (α_j^0) matrix can also be denoted as the vectors $(\vec{\Delta \alpha}_j)$ and $(\vec{\alpha}_j^0)$ respectively. Eq. (41) may be written in the form of equation (14) as

$$\underline{K}(\lambda, g(\omega_i)) \cdot (\vec{\Delta \alpha}_j) = \underline{T}(\lambda, \alpha_j^0, g(\omega_i), G(\omega_i)) \quad (42)$$

with solutions

$$(\Delta \vec{\alpha}_j) = (\underline{K}^{-1}) \cdot \vec{T} \quad (43)$$

Note that the coefficient matrix, \underline{K} , must be non-singular for a solution, $(\Delta \vec{\alpha}_j)$, to be obtained.

The C_{jj}^{new} matrix of equation (41) is given by equation (22), the \vec{V} vector is given by equation (23), and the g_j and G_j terms are given by equations (24) and (28) respectively. The uncertainty matrix $(1/\sigma_{jj})$ is derived in appendix D of reference 3, where the resulting equations are:

$$i' \begin{pmatrix} i \rightarrow \\ \frac{1}{\sigma_{i'i}} \\ \downarrow \\ 0 \end{pmatrix}_{\text{new}} = i' \begin{pmatrix} i \rightarrow \\ \left(\frac{\partial g_{j'}}{\partial r_j} \right)_{r=k}^{-1} \\ \downarrow \\ 0 \end{pmatrix} \cdot i \begin{pmatrix} i \rightarrow \\ \frac{1}{\sigma_{ii}} \\ \downarrow \\ 0 \end{pmatrix} \quad (44)$$

$$i \begin{pmatrix} i' \rightarrow \\ \frac{1}{\sigma_{i'i}} \\ \downarrow \\ 0 \end{pmatrix}_{\text{new}} = i \begin{pmatrix} i \rightarrow \\ \frac{1}{\sigma_{i'i}} \\ \downarrow \\ 0 \end{pmatrix} \cdot i' \begin{pmatrix} i' \rightarrow \\ \left(\frac{\partial g_{j'}}{\partial r_j} \right)_{r=R}^{-1} \\ \downarrow \\ 0 \end{pmatrix} \quad (45)$$

where the matrix:

$$i \begin{pmatrix} i \rightarrow \\ \frac{1}{\sigma_{ii}} \\ \downarrow \\ 0 \end{pmatrix} \quad (46)$$

is a diagonal matrix with real constant elements and is composed of the expected uncertainties as assigned to the data values, $R(\omega_j)$. The expression "r=R" implies that, in the derivatives $(\partial g/\partial r)$, everywhere an r_j occurs replace it with R_j (i.e. a data value).

The derivative terms ($\partial g/\partial r$) are derived in appendix E of reference 3 as:

$$\left(\frac{\partial g_{i'}}{\partial r_i} \right) = \left\{ \frac{a_{i' i}}{\sum_{i''=1}^m a_{i' i''} r_{i''}} \right\} \quad (47)$$

$$i' = 2, \dots, m$$

$$i = 1, \dots, m$$

and the a_{ij} factors are defined in equation (35).

The derivative terms ($\partial g_i/\partial \alpha_j$) are derived in section III-I of reference 3 as

$$\left(\frac{\partial g_i}{\partial \alpha_j} \right) = \frac{\left(\sum_{i'}^n (a_{i i'}) \frac{\partial r_i}{\partial \alpha_j} \right)}{\left(\sum_{i'}^n a_{i i'} r_i \right)} \quad (48)$$

where the derivative ($\partial r_i/\partial \alpha_j$) is given in section IV-E of reference 3 as:

$$\frac{\partial r_i}{\partial \alpha_j} = 2 J \left\{ \frac{{}_1R_{\parallel} \frac{\partial {}_1R_{\perp}}{\partial \alpha_j} - {}_1R_{\perp} \frac{\partial {}_1R_{\parallel}}{\partial \alpha_j}}{(-{}_1R_{\perp} - J {}_1R_{\parallel})^2} \right\} \quad (49)$$

$$J = \sqrt{-1}$$

III. SEQUENCE OF PROGRAM EXECUTION

A more explicit description of the inversion technique used in the INVERT computer program is as follows.

First, the initial electron density profile is defined to have a constant value, $\alpha(z)$, for all heights from the bottom height through the top height. The electron density of this profile is computed for all z_j as:

$$\alpha(z)_j = \frac{\alpha(\text{Top stop}) - \alpha(\text{Bottom stop})}{2} \quad (50)$$

This set of $\alpha(z)_j$'s is defined as (α_j^0) initial. Note that the height increments chosen for the profile must satisfy the following relationship:

$$\frac{z(\text{Top stop}) - z(\text{Bottom stop})}{\text{DELTAH}} = n \cdot 2 \quad (51)$$

where n is an integer and DELTAH is an initial value input to the computer program.

In the matrix equation (41), the first solution desired is for " λ " = $\lambda_1 = 0$. This requires:

$$\begin{matrix} j \rightarrow \\ \downarrow \end{matrix} \left(C_{jj'}^{new} \right) \cdot \begin{matrix} j \rightarrow \\ \downarrow \end{matrix} \left(\Delta \alpha_j \right) + \begin{matrix} j \rightarrow \\ \downarrow \end{matrix} \left(C_{jj'}^{new} \right) \cdot \begin{matrix} j \rightarrow \\ \downarrow \end{matrix} \left(\alpha_j^0 \right)_{\text{Initial}} = \begin{matrix} j \rightarrow \\ \downarrow \end{matrix} \left(V_j \right) \quad (52)$$

where

$$\left(\Delta \vec{\alpha}_j \right) = \left(\vec{\alpha}_j \right)_{\text{new}} - \left(\vec{\alpha}_j^0 \right)_{\text{initial}} \quad (53)$$

and

$$j = 1, \dots, n$$

Substitution of equation (53) into equation (52) gives the solution:

$$\underset{\downarrow}{j} \left(\underset{\downarrow}{(\alpha_j)_{\text{new}}} \right) = \underset{\downarrow}{j}' \left(\underset{\downarrow}{C_{jj'}^{\text{new}}} \right) \cdot \underset{\downarrow}{j} \left(\underset{\downarrow}{V_j} \right) \quad (54)$$

This implies that a solution for $\underset{\downarrow}{(\alpha_j)_{\text{new}}}$ is possible as long as the $\underset{\sim}{C}^{\text{new}}$ matrix is non-singular. For the next step in the inversion, the above $\underset{\downarrow}{(\alpha_j)_{\text{new}}}$ becomes $\underset{\downarrow}{(\alpha_j)_1}$, which is identified as the profile, α^1 .

A full-wave computation, using the just defined set of profile values, $\underset{\downarrow}{(\alpha_j)_1}$, as the input profile will give the set of reflection coefficients, $r(\omega_{i_1})$ (or really $g(\omega_{i_1})$ and the derivatives $(\partial r_i / \partial \alpha_{j_1})$ (or $(\partial g_i / \partial \alpha_{j_1})$) needed to proceed with the inversion scheme.

The next step in the procedure is to substitute the values of $\underset{\downarrow}{(\alpha_j)_1}$ for $\underset{\downarrow}{(\alpha_j^0)}$ along with the values $g(\omega_{i_1})$, $(\partial g_i / \partial \alpha_{j_1})$, and " λ " (i.e. $\lambda = \lambda_2 = \lambda_1 = 0$) all into equation (41) and solve the equation for $\underset{\downarrow}{(\Delta \alpha_j)}$. The $\underset{\downarrow}{(\Delta \alpha_j)}$ solution will be identified as $\underset{\downarrow}{(\Delta \alpha_j)_2}$. This yields what will be called a "settle down" profile (i.e. $\underset{\downarrow}{(\alpha_j)_2} = \underset{\downarrow}{(\alpha_j)_1} + \underset{\downarrow}{(\Delta \alpha_j)_2}$). The use of the "settle down" profile in the inversion sequence will be discussed later.

Note that the variables, λ_{odd} , will be used to identify the increasing sequence of " λ " values used in the inversion procedure. Also, the terms, $\Delta \lambda_{\text{odd}}$, will be used to increment the λ_{odd} values. The variables λ_{even} will be used to identify the "settle down" steps in the inversion procedure. A relationship is that λ_{even} is equal to the previous λ_{odd} (i.e. $\lambda_2 = \lambda_1$, $\lambda_4 = \lambda_3$ etc.). This implies that $\lambda_2 = \lambda_1 = 0$. The term $\Delta \lambda_{\text{even}}$ will not be identified or used in the procedure. The term $\Delta \lambda_1$ is an initial value assigned to the computer program as the variable DELTAL.

substituted for the factor (α_j^0) . The only unknowns in the equations are the $(\Delta\alpha_j)$'s. Equation (41) is then solved for the vector, $(\Delta\alpha_j) \equiv (\Delta\alpha_j)_3$. For a solution to be possible, the coefficient matrix, K of equation (41) must be non-singular.

Note that, throughout the remainder of the discussion of the inversion procedure, a solution to equation (41) will be denoted as SOLEQS (or S.E.). This is the name of the subroutine used in the INVERT computer program to solve for the $(\Delta\alpha_j)$'s.

A new electron density profile is given by the vector equation:

$$(\alpha_j)_{\text{new}} = (\alpha_j)_{\text{previous}} + (\Delta\alpha_j)_{\text{new}} \quad (55)$$

This gives, at the point "c" in figure 8, the new profile $(\alpha_j)_3$. That is

$$(\alpha_j)_3 = (\alpha_j)_1 + (\Delta\alpha_j)_3 \quad (56)$$

where $(\alpha_j)_3$ is identified in figure 8 as the profile, α^3 .

A quantitative measure of the change in profile between α^1 and α^3 is given by:

$$D = \sqrt{\sum_{j=1}^n (\Delta\alpha_j)_q^2} \quad (57)$$

where in this instance $(\Delta\alpha_j)_q \equiv (\Delta\alpha_j)_3$.

At point "c" a full-wave computation, using the vector set $(\alpha_j)_3$ as the input profile, will give a new set of reflection coefficients, $r(\omega_i)_3$ (or $g(\omega_i)_3$) and the derivatives $(\partial r_i / \partial \alpha_j)_3$ (or $(\partial g_i / \partial \alpha_j)_3$) needed to proceed

with the inversion technique. If the profile α^3 lies close enough to the minimization curve of figure 8, then, the choice of value for λ_3 is acceptable and the inversion sequence may continue.

The requirement for acceptance of the α^3 profile is that the following equation must be true. That is:

$$\left| \bar{g}(\omega_i)_3 - \left\{ \bar{g}(\omega_i)_1 + \sum_{j=1}^{2n} \left(\frac{\partial \bar{g}_i}{\partial \alpha_j} \right)_1 (\Delta \alpha_j)_3 \right\} \right| \approx 0 \quad (58)$$

$$i = 1, \dots, m$$

Here the \bar{g} 's are real functions of the variables g_i (real) and g_i (imaginary), (thus the reason for the $2n$ instead of n on the summation). Also, the \bar{g} 's contain the uncertainty matrices, $(1/\sigma)$, of equations (44) and (45).

A more thorough explanation of the meaning of expression (58) is the following. First, let a variable $\bar{g}'(\omega_i)$ be defined as:

$$\bar{g}'(\omega_i)_3 = \bar{g}(\omega_i)_1 + \sum_{j=1}^{2m} \left\{ \left(\frac{\partial \bar{g}_i}{\partial \alpha_j} \right)_1 (\Delta \alpha_j)_3 \right\} \quad (59)$$

This is a linear first order relationship between the reflection coefficient function $\bar{g}(\omega_i)_1$ for profile α^1 and the reflection coefficient function $\bar{g}'(\omega_i)_3$ for profile α^3 . Now, the values computed for the $\bar{g}'(\omega_i)_3$ variables using the full-wave integration calculation will not necessarily be linearly related to the $\bar{g}(\omega_i)_1$ results. They will, in general, be described by second and higher order terms being added to the first order linear term, as are presented in

equation (59). The differences between the linear variables $\bar{g}'(\omega_i)$ and the full-wave variables $\bar{g}(\omega_i)$ are thus related to the magnitude of the higher order terms. That is:

$$\left\{ \begin{array}{l} \bar{g}(\omega_i)_3 - \bar{g}'(\omega_i)_3 \\ \text{full-wave equation (59)} \end{array} \right\} = \begin{array}{l} \text{2nd and higher order} \\ \text{terms are } (\Delta\alpha_j)_3 \end{array} \quad (60)$$

If the magnitudes of these higher order terms are unimportant, $\bar{g}(\omega_i)_3$ will be very close in value to $\bar{g}'(\omega_i)_3$ and the choice of λ_3 is probably not too large. On the other hand, if the difference between $\bar{g}(\omega_i)_3$ and $\bar{g}'(\omega_i)_3$ is large (see equation (58)) the choice of λ_3 is too large and will have to be reduced if the resulting α^3 profile is to remain close enough to the minimization curve.

It is important to point out at this time that small changes in the profile (i.e., $\Delta\alpha_j$) can result in very large changes in the reflection coefficients, as computed by the full-wave integration method, because of the addition of the higher order terms mentioned above.

To determine whether the requirement of equation (58) is met sufficiently, the following criteria is introduced and identified as a "SETTLE-DOWN" iteration. To carry out the "SETTLE-DOWN" procedure, the SOLEQS routine is executed, at the point "c" of figure 8, using the values $g(\omega_i)_3$, $(\partial g_i / \partial \alpha_j)_3$, $(\alpha_j)_3$ and $\lambda = \lambda_4 = \lambda_3$. The solution is the vector $(\Delta\alpha_j)_4$ which may be denoted as $(\Delta\alpha_j)_{SD}$. The subscript, SD, stands for "SETTLE-DOWN."

The $(\Delta\alpha_j)_{SD}$ may be thought of as those modifications which when added to the previous profile, α^3 , will give the "SETTLE-DOWN" profile, $(\alpha_j)_4 = \alpha^4$. That is

$$(\alpha_j)_4 = (\alpha_j)_3 + (\Delta\alpha_j)_{SD} \quad (60)$$

Note the location of the profile, α^4 , in figure 8. This is identified as point "d". The values obtained for the $(\Delta\alpha_j)_{SD}$ are used to determine a measure of the nonlinearity of the $g(\omega_i)$ reflection coefficients with respect to the changes in the profile, (α_j) .

Note here that all factors λ_{odd} , $\Delta\lambda_{odd}$, $(\alpha_j)_{odd}$, $(\Delta\alpha_j)_{odd}$ and α^{odd} refer to the full-wave related variables. The factors λ_{even} , $(\alpha_j)_{even}$, $(\Delta\alpha_j)_{even}$ and α^{even} refer to "SETTLE-DOWN" variables.

The purpose of the "SETTLE-DOWN" calculation is to define a quantitative measure of how acceptable the α^3 profile is for satisfying the minimization curve. Helpful functions for determining this requirement are:

$$(\bar{g}_i)_{SD} = \bar{g}(\omega_i)_{FW} + \sum_{j=1}^{2n} \left(\frac{\partial g_j}{\partial \alpha_j} \right)_{FW} (\Delta\alpha_j)_{SD} \quad (61)$$

$$i = 1, \dots, m$$

where the relation between \bar{g} and g is the same as defined for equation (58). The subscript, FW, stands for the previous full-wave solution for the g 's.

For the case of the profile α^3 equation (61) is:

$$(\bar{g}_i)_{SD} = \bar{g}(\omega_i)_3 + \sum_{j=1}^{2n} \left(\frac{\partial \bar{g}_i}{\partial \alpha_j} \right)_3 (\Delta \alpha_j)_4 \quad (62)$$

Equation (61) states that \bar{g}_{SD} is computed as a linear function of the \bar{g}_i 's.

A second function determined from the "SETTLE-DOWN" results is:

$$S' = \sum_{i=1}^m \left((\bar{g}_i)_{SD} - G_i \right)^2 \quad (63)$$

where G_i is a data value as given by equation (28). The expression S' will be used later in determining where the interactive procedure is to be stepped.

The variable used to give a quantitative measure to the "SETTLE-DOWN" criteria is given as

$$D \text{ PRIME} = \sqrt{\sum_{j=1}^n (\Delta \alpha_j)_{SD}^2} \quad (64)$$

If the optimum magnitude of the "SETTLE-DOWN" is not exceeded then the choice of λ^3 was not too large and the profile α^3 is close enough for the inversion sequence to continue. The variable DPRIME is indicated in figure 8 by D' .

In order to continue at an optimum rate of progress (i.e., incrementing " λ " so as to move along the minimization curve), or to back up (i.e., decreasing the value of " λ "), the following procedure is used by the INVERT program.

A "threshold" term is defined by the equation

$$\text{RATIO} = \frac{D \text{ PRIME}}{D \text{ OPT}} \quad (65)$$

The numerical results of equation (65) is compared to the set of iteration rules presented in table I. These rules state how " λ " is to be incremented (i.e., $\lambda_{\text{next}} = \lambda_{\text{previous}} + \Delta\lambda$) for the next execution of the SOLEQS routine in order for the inversion process to continue.

Assuming that the value of RATIO gives a favorable result as, determined by the rules of table I, an increment $\Delta\lambda_5$ is added to λ_3 to get to point "e" of figure 8 and the inversion procedure continues.

From the above discussion it is seen that the inversion scheme proceeds in a "boot strap" fashion of obtaining solutions $(\Delta\alpha_j)$ to equation (41) for a particular value of " λ ", computing a new profile $((\alpha_j)_{\text{new}} = (\alpha_j)_{\text{old}} + (\Delta\alpha_j))$, doing full-wave calculations on the $(\alpha_j)_{\text{new}}$ to get new values of the reflection coefficient functions $g(\omega_i)_{\text{new}}$ and $(\partial g_i / \partial \alpha_j)_{\text{new}}$, checking the linearity of $g(\omega_i)_{\text{new}}$ by doing "SETTLE-DOWN" calculations, and finally increasing the value of " λ ". The process then continues by following through with the same series of steps again.

The final solution (i.e., final inverted electron density profile) is obtained as α^{final} as shown in figure 8. The definition as to what determines that any given profile in the iterative sequence should be identified as the final profile will be that profile for which the variable S' of equation (63) reaches a turning point (or "elbow") in a plot of S' versus iterative step number.

TABLE I

RATIO THRESHOLDS

RATIO	COMMENTS	$\Delta\lambda$	λ	COMMENTS
0.0 "A"	Convergence is going too slowly, so want to speed up. Use $\Delta\lambda_{\text{new}}$ to get λ_{new}	$\Delta\lambda_{\text{new}} = \sqrt{2}\Delta\lambda_{\text{old}}$	$\lambda_{\text{new}} = \lambda_{\text{old}} + \Delta\lambda_{\text{new}}$	① Using λ_{new} solve SOLEQS to get $(\Delta\alpha_j)_{\text{new}}$. ② Compute new profile $(\alpha_j)_{\text{new}}$ ③ Do full-wave on $(\alpha_j)_{\text{new}}$
0.8 "B"	Convergence is going about right, so want to continue at same increment. Use $\Delta\lambda_{\text{new}}$ to get λ_{new}	$\Delta\lambda_{\text{new}} = \Delta\lambda_{\text{old}}$	$\lambda_{\text{new}} = \lambda_{\text{old}} + \Delta\lambda_{\text{new}}$	Same as "A" above.
1.2 "C"	Convergence is going too fast. Want to slow down. Use $\Delta\lambda_{\text{new}}$ to get λ_{new}	$\Delta\lambda_{\text{new}} = \Delta\lambda_{\text{old}}/2$	$\lambda_{\text{new}} = \lambda_{\text{old}} + \Delta\lambda_{\text{new}}$	Same as "A" above.
2.0 "D"	Results are too far off convergence curve. Will compute	$\Delta\lambda_{\text{new}} = \Delta\lambda_{\text{old}}/2$ "saved"	$\lambda_{\text{new}} = \lambda_{\text{old}}$	① Using λ_{old} to solve SOLEQS for $(\Delta\alpha_j)_{\text{new}}$ is the same as using $(\Delta\alpha_j)_{\text{new}}$

TABLE 1

CONTINUED

RATIO	COMMENTS	$\Delta\lambda$	λ	COMMENTS
4.0	$\Delta\lambda_{new}$, however it will not be used but will be saved for future steps.			from settle down solution of SOLEQS. ② Use $(\Delta\alpha_j)_{settle\ down}$ to get new profile. ③ Do full-wave on $(\alpha_j)_{new}$
4.0 "E"	Same as "D" above	$\Delta\lambda_{new} = \Delta\lambda_{old}/2$ "saved"	$\lambda_{new} = \lambda_{old} - \Delta\lambda_{old}$ $= \lambda_{previous}$	① Set $(\alpha_j)_{new} = (\alpha_j)_{prev.}$ ② Do full-wave on these (α_j) 's
∞				

The function S' is a linear approximation to the function S which is given by:

$$S = \overbrace{g_i - G_i}^{i \rightarrow} \cdot \underset{\downarrow}{i} \left(\frac{1}{\sigma_{ii'}} \right)^{i' \rightarrow} \cdot \underset{\downarrow}{i} \left(\frac{1}{\sigma_{i'i}} \right)^{i \rightarrow} \cdot \underset{\downarrow}{i} \left(g_i - G_i \right) \quad (66)$$

where the G_i are data values as given by equation (28) and contain some amount of error. Equation (66) is functionally related to equation (6) and replaces the "S" factor in equations (12) and (13).

Figure 9 illustrates the general characteristics of the S' and S functions as related to the number of the iterative step at any point in the inversion procedure. The simulated data used in figure 9 has error statistics as will be discussed in section V-A. That is the error is gaussian distributed and

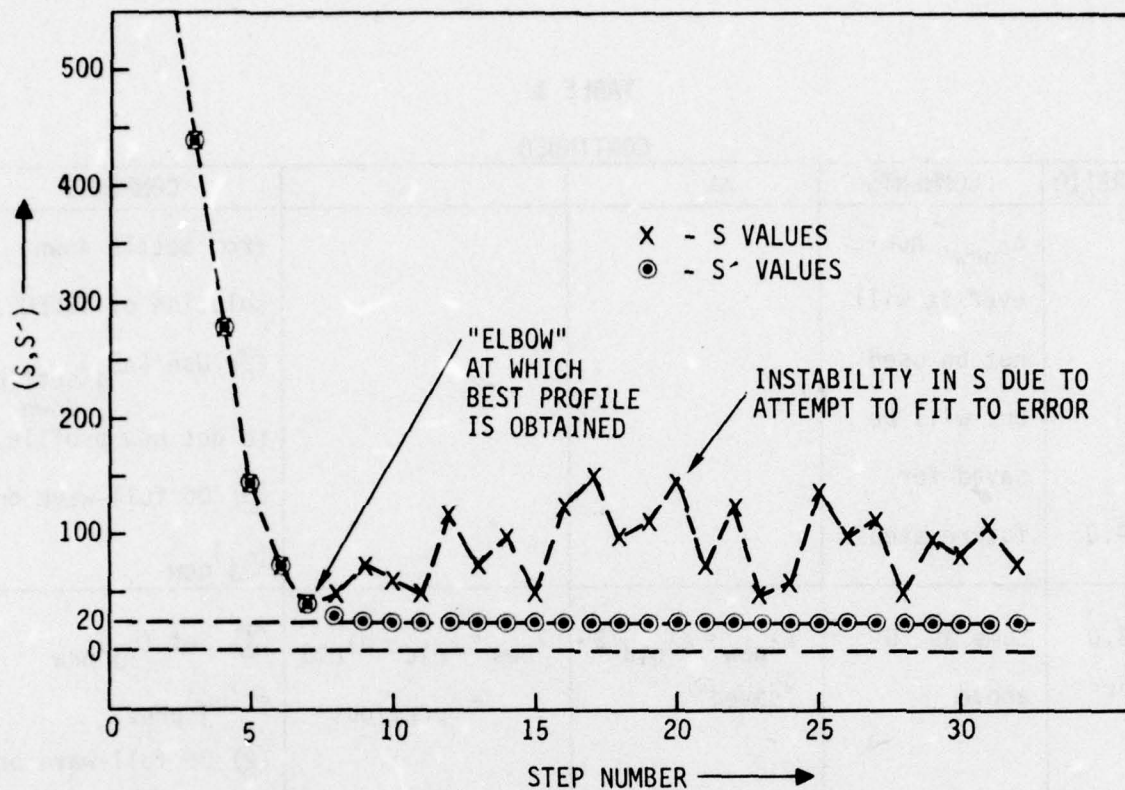


Figure 9. Comparison between S and S' as a Function of Iterative Step Number.

uncorrelated and also the expected error values, σ_i , (i.e., the standard deviation values) are known. For the results presented in figure 9, there are 20 data points used in the computation (i.e., $i = 1, \dots, m$ where $m = 20$). This implies that the lowest value of S (or S') to be expected is approximately 20. The approximate nature of this number is due to the number of data points being finite. That is, for larger sets of data $(S/m) \rightarrow 1$ as $m \rightarrow \infty$ if the error for each data parameter is independent of the error for any other data parameter. Continuation of the iterative process to obtain S (or S') values much less than 20 represents an unsuccessful attempt to fit the profile to the error in the data. The transition from fitting to data to fitting to the error is illustrated in figure 9 as an "elbow" in the S' curve. Beyond this point oscillations are introduced into the profile in an

attempt to fit to the error which in turn leads to instability in the values of S . The linear projections represented by the values of S' give an indication of the best fit that can be expected.

Note that the very definite transition point, represented by the "elbow" in figure 9, is the result of complete knowledge of the error statistics. That is, the expected errors (i.e., the standard deviation values, σ_i , which are used with gaussian distributed random numbers to simulate the errors) for this simulated example are known and the error for each data variable, $G(\omega_i)$, is independent of the error for every other data variable. Otherwise, the transition represented by the "elbow" would be much less well-defined and might even be undetectable if too little is known about the actual error statistics.

Experience gained in executing the INVERT program indicates that there is an optimum magnitude of the "SETTLE-DOWN" of the profile which can be tolerated for a valid sequence of profiles. Singularities of the coefficient matrix of equation (41) will lead, however, to ever increasing values of DPRIME as the value of " λ " corresponding to the singularity condition is approached. When this occurs in the inversion sequence, no further progress can be made. This case is illustrated in figure 10. The only possibility for progress is to re-execute the INVERT computer program from the beginning with a new initial value of the input variable, ROTATE, (i.e., the direction of rotation of the $r(\omega_i)$ curve is reversed). If this option is not successful in alleviating the problem, no solution is possible, for the given set of reflection coefficient data, when using the INVERT program.

IV. STRUCTURE AND DESCRIPTION OF THE INVERT COMPUTER PROGRAM

A. Program Structure

Figure 11 presents the general flow of the INVERT computer program showing the relationship between the various subroutines.

An important limitation built into the INVERT program is that execution is possible for an arbitrary choice of frequencies only as long as the following rule is applied:

$$\left\{ \frac{\text{Frequency Max.} - \text{Frequency Min.}}{\text{DELTA Frequency}} \right\} \leq 200 \quad (67)$$

B. Description of Computer Program Routines

The function of each program subroutine is listed below:

(1) MAIN ROUTINE:

This routine controls the input of inversion parameters through the Namelist, DATUM. It also provides for the input of reflection coefficient data and the input of full-wave parameters.

(2) SUBROUTINE DINPUT:

Input of data: reflection coefficients (R), uncertainties (σ), and propagation frequencies (ω).

(3) SUBROUTINE FWINPT:

Input of full-wave parameters through the Namelist, FULL WV. These parameters include the angle of incidence of the radio wave onto the ionosphere, "THETA", and the geomagnetic parameters of the earth: propagation azimuth, "AZIM", (i.e., degrees east of magnetic north), Codip angle, "CODIP", magnetic field

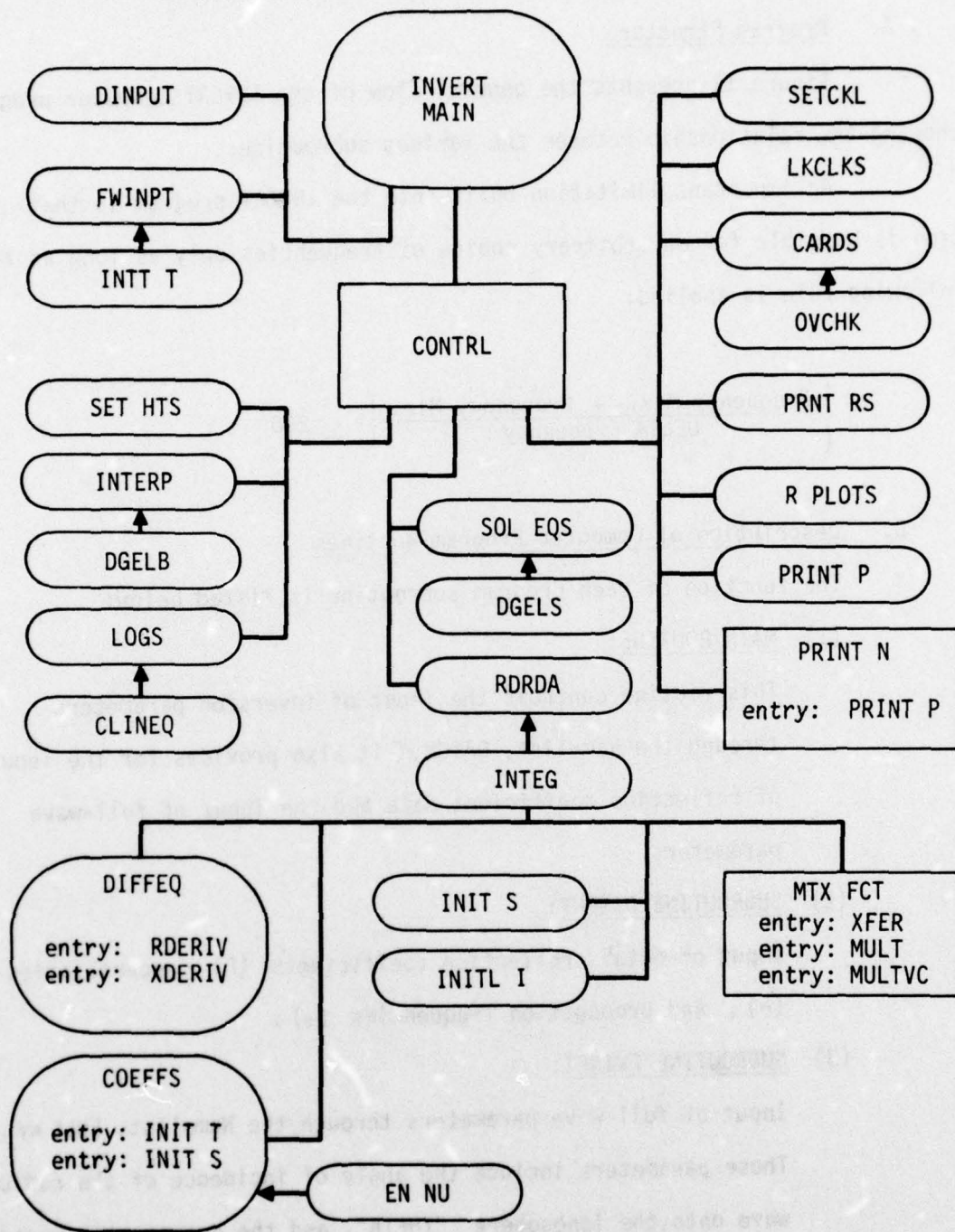


Figure 11. Structure of the INVERT Computer Program.

strength, "MAGFLD", (in Webers/square meter). Also input are the options for obtaining printer plots of electron density profiles, "IPROFP", and reflection coefficient plots, "IRPLOT".

(4) SUBROUTINE CONTRL:

This routine controls the running sequence of the remainder of the total program, especially the incrementing of the trade-off parameter " λ " which determines the convergence of the inversion scheme.

(5) SUBROUTINE SETHTS:

This routine sets up the initial electron density profile. Here, use is made of the input parameters: HTSTOP(1) and HTSTOP(2) (i.e., the profile heights where the maximum and minimum electron densities are assigned. Also of importance is that the value of (HTSTOP(1) - HTSTOP(2))/DELTAH must be a multiple of 2. DELTAH is the height increment between each profile point. See equation (51).

(6) SUBROUTINE INTERP:

This routine sets up coefficients used for interpolation between the given data points of reflection coefficients (R) versus propagation frequency (ω). These coefficients are the elements of the β -matrix of equation (34). This routine uses the subroutine DGELB.

(7) SUBROUTINE DGELB:

The routine solves a system of simultaneous linear equations with a coefficient matrix of band structure, (i.e., $\chi = C^{-1} \cdot \gamma$ where C^{-1} is of the form:

$$\zeta^{-1} = \begin{pmatrix} b & a & & & \\ & c & b & a & \\ & & c & b & a \\ & & & c & b \\ & & & & c & b \end{pmatrix}$$

(8) SUBROUTINE LOGS:

The routine sets up the parameters for making use of the G function of equation (28). That is for $G_i = \ln(dR/d\omega)_i$ where R_i are the input variables (i.e., functions of the reflection coefficients). The routine generates a continuous curve $R(\omega)$ (e.g., continuous in phase) by making use of the β -matrix of equation (34). Subsets (a_{ij}) of the β -matrix are used to compute the derivative term $(dR/d\omega)_i$ along the $R(\omega)$ curve. This derivative is given by equation (35). Also obtained are the derivatives $(dG/dR)_i$ of equation (47). The uncertainty matrix $(1/\sigma)_{\text{new}}$ of equations (44) and (45) are also computed. This routine uses the subroutine CLINEQ.

(9) SUBROUTINE CLINEQ:

This routine finds the solution, X , of the matrix equation $A * X = L * U * X = B$ where L and U are triangular matrices.

(10) SUBROUTINE RDRDA:

This routine sets up the parameters for making use of the "g" function of equation (24). It computes the following parameters: $(dr/d\omega)_i$ of equation (37), $(\partial r_i / \partial \alpha_j)$ of

equation (49) and $(\partial g_i / \partial \alpha_j)$ of equation (48) to be used in the subroutine SOLEQS. The continuity of phase for $(dr/d\omega)$ is also accomplished in this routine.

(11) SUBROUTINE INTEG:

This routine does the full-wave integration for the ionospheric reflection coefficients (i.e., R_{ν} of equation (1) and $(\partial R_{\nu} / \partial \alpha_j)$). The integration is accomplished by a Runge-Kutta procedure. INTEG makes use of the following subroutines: INITL I, DIFFEQ and COEFFS.

(12) SUBROUTINE INITL I:

This routine computes the ionospheric reflection coefficient matrix, R_{ν} , for a sharply bounded ionosphere. These results are used as starting values for the R_{ν} integration of subroutine INTEG.

(13) SUBROUTINE COEFFS:

This routine computes the coefficients for dR_{ν}/dz .

(14) SUBROUTINE INIT T:

This routine computes variables which are used in the full-wave solution and are functions only of the geomagnetic parameters, (AZIM, CODIP and MAGFLD) and the angle of incidence, THETA.

(15) SUBROUTINE INIT S:

This routine computes variables which are used in the full-wave solution and are functions of the variables of INIT T and also of the propagation frequency, ω .

(16) SUBROUTINE DIFFEQ:

This routine computed the derivatives (dR/dz) and $d/dz (\partial R/\partial \alpha(z_j))$ for use in the full-wave integration. R is the ionospheric reflection coefficient matrix of equation (1).

(17) SUBROUTINE EN NU:

This routine interpolates electron density and collision frequency profiles as a function of ionospheric height.

(18) SUBROUTINE SOL EQS:

This routine obtains the solution of equation (41) in terms of the change in profile ($\Delta\alpha_j$) from given values of derivatives ($\partial g_i/\partial \alpha_j$) and a given value of " λ ". This routine also computes the parameters, "error width" and "resolution". The solution ($\Delta\alpha_j$) is obtained through the use of the subroutine DGELS.

(19) SUBROUTINE DGELS:

Solves the matrix equation (41) for ($\Delta\alpha_j$).

(20) SUBROUTINE CARDS:

"Punched" card output for continuing inversion procedure in a subsequent computer run.

(21) SUBROUTINES SET CKL AND LK CLKS:

Systems routines for keeping track of elapsed time so that output to subroutine CARDS will be obtained before computer run is terminated by exceeding the time limit.

(22) SUBROUTINE R PLOTS:

This routine provides computer printer plots of ionospheric reflection coefficient functions, R (data) and r (computed).

(23) SUBROUTINES PRINT N AND PRINT P:

These routines print and plot the electron density profile of the ionosphere.

(24) SUBROUTINE PRINT RS:

This routine prints summary tables of computed parameters for each successive iteration in the inversion scheme. The printed values are: r_i (real, imaginary), magnitude of $(dr/d\omega)$, phase of $(dr/d\omega)$ in degrees, the transformation function g (real) and g (imag.), in radians.

(25) SUBROUTINE OVCHK:

This routine checks for overflows and underflows during computation.

(26) SUBROUTINE MTX FCT:

This subroutine provides for matrix manipulations.

C. Identification of NAMELIST Variables

(1) NAMELIST: & FULL WV

(a) 'THETA' -- Incident angle of the transmitted wave onto the ionosphere. (degrees)

(b) Geomagnetic parameters of the earth's magnetic field.

'AZIM' -- The clockwise angle between magnetic north and the horizontal propagation direction east of north. (degrees)

'CODIP' -- The magnetic co-dip angle measured from the vertical downward. (degrees)

The magnetic equator is specified by 'CODIP' = 90° while the north magnetic pole is identified as 0°.

"MAGFLD" -- The intensity of the earth's magnetic field (Webers/square meters).

- (c) "IPROFP" -- Option for obtaining printer plots of a simulated original electron density profile from which reflection coefficients have been previously computed and are to be used as input data to the program "INVERT". If 'IPROFP' = 1, then computer plots of this simulated profile will be superimposed upon plots which are automatically printed of electron densities obtained from successive iterations of the inversion scheme. Note that the profile heights of the simulated profile must be identical to the heights of the profiles obtained through the iterative steps. If not the program execution stops.

If 'IPROFP' = 0, only the iterative profiles are plotted. This is because if the input data values, $R(\omega_i)$, were obtained from actual measurements, there would be no simulated data profile and therefore no profile to plot.

- (d) 'RPLOTS' -- Option for obtaining printer plots of r (real) and r (imaginary) in the complex r -plane, 'RPLOTS' = 1 gives the plots, 'RPLOTS' = 0 results in no plots.

(2) NAMELIST: &DATUM

- (a) 'HTSTOP' -- The heights (km) of the electron density profile at which the profile "stops" are to be applied. "HTSTOP(1)" is the higher height corresponding to "STOPS(1)" while "HTSTOP(2)" is the lower height corresponding to "STOPS(2)".

- (b) 'STOPS' -- The limiting values (in electrons/cm³) of the electron densities that the iterated profile can take on at the heights. HTSTOP(1) and HTSTOP(2).
- (c) 'NUFLAG' -- This parameter allows for a choice of exponential collision frequency profiles to be used in the calculations. The exponential collision frequency is given by $\nu(z) = \text{COEFNU} * \text{EXPNU}(z)$ where "z" is the ionospheric height in km.
- (i) If "NUFLAG = 0", the values are:
- $$\text{COEFNU} = 1.816 \text{ E} + 11, \text{ coll/sec}$$
- $$\text{EXPNU} = -0.15 \text{ km}^{-1}$$
- (ii) If "NUFLAG = 1", the values are:
- $$\text{COEFNU} = 4.303 \text{ E} + 11, \text{ coll/sec}$$
- $$\text{EXPNU} = -0.1622 \text{ km}^{-1}$$
- (iii) If a different exponential is desired, set "NUFLAG = 0" and enter the new values of COEFNU and EXPNU through the NAMELIST, DATUM.
- (d) 'DELTA' -- The initial value of the parameter $\Delta\lambda$.
- (e) 'DOPT' -- A term used in the iterative procedure and defined by:

$$\text{RATIO} = \frac{\text{DPRIME}}{\text{DOPT}}$$

- (f) 'ALPHA' -- The values of the logs of electron densities at each height of the profile, (i.e., $\alpha(z)_j = \log_{10} N_e(z)$). These ALPHA values are obtained from a previous execution of the INVERT program and are output from the subroutine "CARDS".

- (g) 'NRSTPS(J,I)' -- The number of Runge-Kutta steps in a pair of height intervals between heights $(h_{(2j-1)})$ and $(h_{(2j+1)})$ for the i -th frequency, ω_i .
- (h) 'NR' -- The number of increments that have been made in " λ " in solving equation (41).
- (i) 'IPRNTE' -- The truncation error in Runge-Kutta analysis is printed when IPRNTE = 1.
- (j) 'IPRINT' -- Printout of Budden's reflection coefficients as a function of height. Printed when IPRINT = 1.
- (k) 'ROTATE' -- Determines the rotation of the computed $r(\omega)$ curve in the r -plane. The values used are that:

ROTATE	{	<ul style="list-style-type: none"> = "+", is counter-clockwise rotation. = "-", is clockwise rotation. = "0", the program chooses the rotation direction of smallest difference between the ϕ_D and the ϕ_C of section (II-I).
--------	---	---

The following tables give the initial values of the NAMELIST variables.

TABLE II
 NAMELIST VARIABLES AND INITIAL VALUES,
 NAMELIST: &FULL WV

NAME	VALUE	UNITS
AZIM	90.0	degrees
CODIP	12.0	degrees
MAGFLD	5.3 E - 5	Weber/square-meter
THETA	22	degrees
IRPLOT	1	none
IPROFP	1	none
IPRINT	0	none

TABLE III
 NAMELIST VARIABLES AND INITIAL VALUES,
 NAMELIST: &DATUM

NAME	VALUE	UNITS	NAME	VALUE	UNITS
ALAMDA	0.0	km	HTSTOP(1)	93	km
ROTATE	0.0	degrees	HTSTOP(2)	50	km
DELTAH	1.0	km	STOPS(1)	1.0 E 3	electrons/cm ³
DELTA L	1.0/64.0	km	STOPS(2)	3.0	electrons/cm ³
DELF	0.1	KHz	COEFNU	1.816 E 11	collisions/sec
DOPT	1.0	none	EXPNU	-0.15	km ⁻¹
IPRNTE	0	none	ALPHA(51)	none	none
NUFLAG	0	none	NRSTPS	none	none
NR	1.0	none			
MIN	10.0	minutes			

V. EXECUTION OF INVERT USING SIMULATED DATA

A. Errors

To illustrate the inversion procedure, the INVERT computer program may be executed using simulated reflection coefficient data as input.

The simulated data input is described by:

$$R(\omega_i)_\epsilon = R(\omega_i)_{F.W.} + \epsilon_i \quad (68)$$

$$i=1, \dots, m$$

where

ω_i -- are the propagation frequencies.

$R(\omega_i)_{F.W.}$ -- are the reflection coefficient variables as defined by equation (2) and are obtained from execution of the FULLWAVE computer program. These values may be thought of as the true values of the Reflection coefficients of the data when no error is present.

ϵ_i -- are the actual simulated errors assumed to be present in the data $R(\omega_i)_\epsilon$ at each frequency.

The above variables are computed using two auxiliary computer programs. These are identified as FULLWAVE and RANDOM. FORTRAN listings of these programs are included in appendices A and B of this report.

B. The Auxiliary Program, FULLWAVE

The FULLWAVE computer program gives simulated reflection coefficient values for the factor $R(\omega_i)_{F.W.}$ of equation (68). These values are obtained from the integration of equation (4) and the generation of equation (5) (or equation (2)). The input to the program is a simulated electron density profile (i.e. the "Data Profile"), assumed collision frequency profile, geomagnetic field parameters, the frequency range desired and a value to be assigned to the expected error, SIGMA.

Figure 12 illustrates the relationship of the various subroutines for the FULLWAVE program.

The following is a list of the subroutines used in the FULLWAVE computer program.

(1) MAIN ROUTINE:

Provides for input to the program, including input electron density and collision frequency profile. It also provides for NAMELIST input.

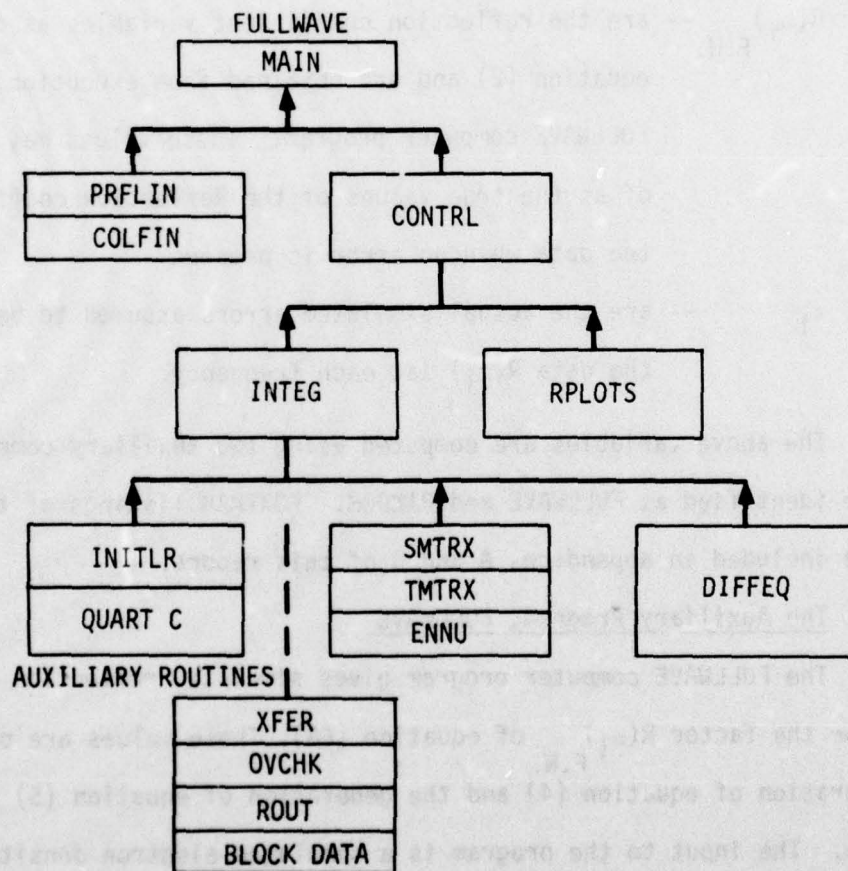


Figure 12. Program Flow for FULLWAVE.

(2) SUBROUTINE PRFLIN:

Provides for reading an electron density profile into the program. This profile is a table in terms of height and electron/cm³.

(3) SUBROUTINE COLFIN:

Provides for reading a collision frequency profile into the program. This profile consists of only the frequencies at the top and bottom heights of the electron density profile.

(4) SUBROUTINE CONTRL:

Controls the running sequence of the total program.

(5) SUBROUTINE INTEG:

Performs the integration of the differential equations for the ionospheric reflection matrix using Runge-Kutta integration formulas. Subroutines utilized for this operation are: INITLR, QUARTC, SMTRX, TMTRX, ENNU, DIFFEQ.

(6) SUBROUTINE RPLOTS:

Provides for printer plots of real and imaginary parts of the reflection coefficient function of equations (2) and (5).

(7) AUXILIARY ROUTINES:

BLOCK DATA, XFER, OVCHK and ROUT.

The following are NAMELIST, "DATUM" input variables:

(1) FLOW:

The lowest frequency (kHz) for which full-wave computations are to be made.

(2) FHI:

The highest frequency (kHz) for which full-wave computations are to be made.

- (3) FINC:
The increment in frequency (kHz).
- (4) THETA:
The angle of incidence on the ionosphere of the transmitted radio wave.
- (5) AZIM:
The azimuth of the radio wave direction measured in degrees east of magnetic north.
- (6) CODIP:
The codip of the magnetic field vector. CODIP = 90° at the earth's magnetic equator and = 0° at the earth's north magnetic pole.
- (7) MAGFLD:
The magnitude of the earth's magnetic field (webers/m²)
- (8) NUFLAG:
A flag for determining the collision frequency profile to be used in the "full wave" computations. The following criteria is used:

NUFLAG = 0, COEFNU = 1.816E11 collisions/sec.

EXPNU = -0.15 km⁻¹

NUFLAG = 1, COEFNU = 4.303E11 collisions/sec

EXPNU = -0.1622 km⁻¹

If a table of collision frequency values is to be read in through the subroutine COLFIN, then NUFLAG must also be set to zero.

(9) NPUNCH:

If NPUNCH = 0, no output cards are punched. If NPUNCH = 1, output cards are obtained with frequency (kHz), real and imaginary parts of the R variable of equation (2) and a choice for the expected error, SIGMA, on each card.

(10) IPROFP:

Provides for printer plot of electron density profile:

IPROFP = 1: plot is obtained

IPROFP = 0: no plot

(11) IRPLOT:

Provides for printer plot of R-parameter from equation (2).

IRPLOT = 1: plot is obtained

IRPLOT = 0: no plot

(12) SIGMA:

The value of the expected error, σ_{ij} , assigned to each data value, $R(\omega_i)$.

TABLE IV
 INITIAL VALUES OF FULLWAVE
 NAMELIST VARIABLES

NAME	INITIAL VALUE	UNITS
FLOW	0.5	kHz
FHI	25.0	kHz
FINC	0.5	kHz
THETA	0.0	degrees
AZIM	none	degrees
CODIP	none	degrees
MAGFLD	none	webers/meter ²
NUFLAG	0	none
COEFNU	1.816×10^{11}	collisions/sec.
EXPNU	-0.15	km ⁻¹
NPUNCH	1	none
IPROFP	1	none
IRPLOT	1	none
SIGMA	20(0.03)	none

C. The Auxiliary Program, RANDOM

The computer program RANDOM provides for gaussian random error values, ϵ_i , to be added to the FULLWAVE reflection coefficient values $R(\omega_i)$ F.W. The actual simulated error, ϵ_i , is computed by the expression:

$$\epsilon_i = u_{iq} \times \sigma_i \quad (69)$$

$i=1, \dots, m$

where

σ_i -- is the expected error in the data as chosen for each frequency, ω_i .

u_{iq} -- is obtained from a random number generator where "q" is the number identifying the particular set of random numbers used.

Define a random variable $u(x)$, which is gaussian distributed with standard deviation, $\sigma = 1$, as:

$$u(x) \approx \frac{1}{\sqrt{2\pi}} \exp(-x^2/2) \quad (70)$$

This distribution is shown in figure 13 where the distribution is truncated between -6 and +6.

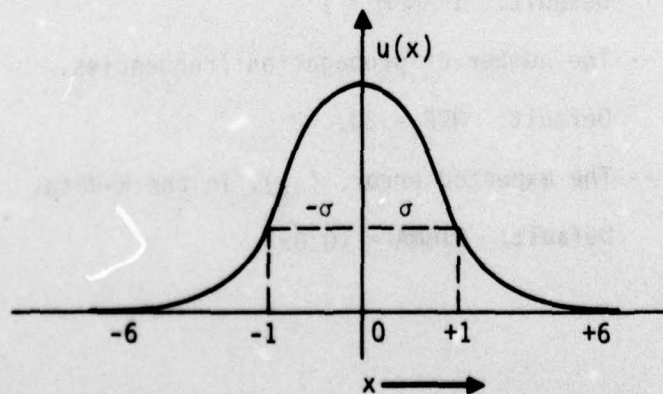


Figure 13. Gaussian Distribution.

Let different sets of random numbers be identified as $q = 1, 2, \dots$, where the value assigned to "q" is input to the program, RANDOM by the NAMELIST variable, IRANDM. A given q-th set of random numbers is given by the values, u_{iq} , $i=1, \dots$.

Note that the value chosen for σ_i represents the standard deviation of $\varepsilon_i = u_{iq} \times \sigma_i$ as a large number of sets, q, are considered.

In the computer program, RANDOM, the complex values of $R(\omega_i)_{F.W.}$ are divided into real and imaginary parts. The same value, σ_i , is then applied to each of the parts (i.e. real and imaginary) of a given $R(\omega_i)_{F.W.}$. Separate values of u_{iq} are applied, however, to the real part of $R(\omega_i)_{F.W.}$ and to the imaginary part of $R(\omega_i)_{F.W.}$. In the complex R plane the expected error, σ_i , is represented as a circle of radius σ_i about the data point $R(\omega_i)_\varepsilon$.

The above definitions are such that for a given frequency, ω_i , and given chosen value for the expected error, σ_i , then the true value, $R(\omega_i)_{F.W.}$, will lie within the error circle of radius σ_i , drawn around the value $R(\omega_i)_\varepsilon$, 68.3% of the time as a large number of error sets, q, are considered. This result is illustrated in figure 14.

The NAMELIST variables of the computer program RANDOM consist of the following:

IRANDM -- Determines a specific set of random numbers.

Default: IRANDM = 1.

NRF -- The number of propagation frequencies.

Default: NRF = 20.

SIGMA -- The expected error, (σ_i), in the R-data.

Default: SIGMA = (0.03).

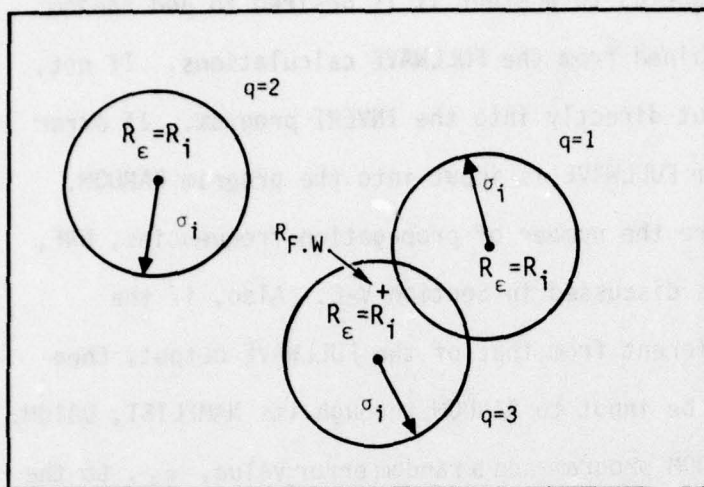


Figure 14. "Error Circles" for a given Frequency, ω_j and given expected error, σ_j , for three error sets, q .

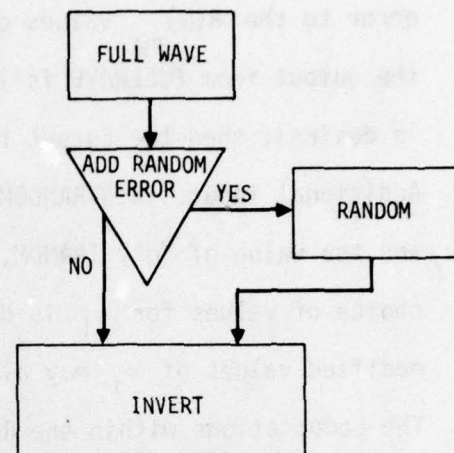


Figure 15. Program flow for using simulated data in INVERT program.

D. Examples of INVERT Execution for Simulated Data

The following sequence of computer runs must be carried through in order to complete an execution of the INVERT program using simulated data. Figure 15 shows the sequence of the required programs.

First, the set of reflection coefficient values, $R(\omega_j)_{F.W.}$ is obtained from the auxiliary program, FULLWAVE. The inputs to this program are the known electron density profile (i.e. simulated data profile), incidence angle, THETA, the geomagnetic field parameters and the range of propagation frequencies desired. The chosen expected error, SIGMA (σ_j), may also be input. There are, however, default values of σ_j incorporated into the program. The output consists of the set of values: frequency, ω_j , computed values of $R(\omega_j)_{F.W.}$ {real} and $R(\omega_j)_{F.W.}$ {imaginary} along with values of the expected error, SIGMA (i.e. σ_j). There is one set of all the above values for each frequency, ω_j .

Second, a choice is made as to whether it is desired to add random error to the $R(\omega)_{FW}$ values obtained from the FULLWAVE calculations. If not, the output from FULLWAVE is input directly into the INVERT program. If error is desired, then the output from FULLWAVE is input into the program RANDOM. Additional inputs into RANDOM are the number of propagation frequencies, NRF, and the value of "q", IRANDM, as discussed in Section V-C. Also, if the choice of values for σ_i is different from that of the FULLWAVE output, then modified values of σ_i may also be input to RANDOM through its NAMELIST, DATUM. The computations within the RANDOM program add a random error value, ϵ_i , to the FULLWAVE values, $R(\omega_i)_{FW}$, to give the data values, $R(\omega_i)_E$. The output from the RANDOM program consists of the set of values: frequency, ω_i , computed values of $R(\omega_i)$ {real} and $R(\omega_i)$ {imaginary} along with values of the expected error, $SIGMA$ (i.e., σ_i). There is one set of the above values for each frequency, ω_i .

Finally, the input to the INVERT program consists of the known electron density profile (i.e., simulated data profile), the output values from either FULLWAVE or RANDOM, the incidence angle (THETA), the geomagnetic field variables, and various constraints which are applied in the inversion procedure. The output from INVERT is the inverted electron density profile.

The following examples of input-output values illustrate the inversion sequence for simulated data.

Example I lists the input values to the FULLWAVE program including the profile of electron density per cm^3 at each height and the NAMELIST variables. Example II shows the printed output from FULLWAVE consisting of a table and corresponding plot of the input profile. Example III gives the printed output from FULLWAVE. This includes a table of the values of the output variables and a plot of the $R(\omega_i)_{FW}$ data values in the complex plane. Note that the table

values are also punched on cards to be used as input into either the RANDOM program or directly into INVERT if the no-error case is desired.

Example IV illustrates the input values into the RANDOM program. Note that IRANDM=2 for this case. Example V gives the printed input. Example VI presents the random numbers computed using IRANDM=2. These numbers are used to compute the actual simulated error, ϵ_i . The equation is:

$$\left\{ \begin{array}{l} R(\omega_i)_{\epsilon} \{\text{real}\} = R(\omega_i)_{FW} \{\text{real}\} + (U_{i,2}(\text{real}) \times \sigma_i) \\ R(\omega_i)_{\epsilon} \{\text{imag}\} = R(\omega_i)_{FW} \{\text{imag}\} + (U_{i,2}(\text{imag}) \times \sigma_i) \end{array} \right\} \quad (71)$$

Using the values for ω_1 of example V to illustrate the computation gives:

$$\left\{ \begin{array}{l} R(\omega_1)_{\epsilon} \{\text{real}\} = 0.290 + (-3.233 \times 0.03) \\ \qquad \qquad \qquad = 0.193 \\ R(\omega_1)_{\epsilon} \{\text{imag}\} = -0.332 + (1.687 \times 0.03) \\ \qquad \qquad \qquad = -0.281 \end{array} \right\} \quad (72)$$

The results of these computations are given in example VII. The printed and punched output showing final results of the RANDOM calculations are presented in example VIII. This punched output is to be used as input to INVERT.

In order to show a comparison between the results from two different choices of sets of random errors to be added to the FULLWAVE output, a computation for IRANDM=3 is also presented in examples IX, X and XI.

The series of input-output examples characteristic of INVERT execution are presented as follows. Example XII gives the input values to INVERT. These values are for the reflection coefficient data containing random error set number 2. The input values are for the variables: propagation frequency ω_i , the data values $R(\omega_i)$ {real}, $R(\omega_i)$ {imag} and the expected error, σ_i . Also included are the two sets of NAMELIST variables. These are identified as FULLWV and DATUM. Example XIII shows the printout of input data to INVERT.

Example XIV is the printout of initial steps in the INVERT execution. The following examples are to be compared closely with the discussion concerning figure 8 in section III of this report. Example XV is obtained from the INVERT subroutine PRINTN. Example XVI shows the computed results of $r(\omega_i)$, $|(dr/d\omega)_i|$, $\phi(dr/d\omega)_i$, $|g|$ and ϕ_g as given by equations (5), (25) and (37) as obtained from the full-wave integration of the profile in example XV. Example XVII is the plot of $R(\omega)_\epsilon$ and $r(\omega)$ values as presented in the complex R (or r) plane. The results of $R(\omega_i)$ are those listed under "data values" for "RE(R)" and "IM(R)" in example XIV. The inversion values, $r(\omega_i)$, as obtained from full-wave integration of the initial profile of example XV are listed as RE(R) and IM(R) in example XVI and identified as "computed values" in example XVII. Note the large mismatch between the $R(\omega_i)_\epsilon$ data values and the $r(\omega_i)$ computed values shown in example XVII.

As discussed in section III, the inversion procedure contains a checking criteria which aids in preventing divergence by limiting the size of the increments allowed for the variable " $\Delta\lambda$ ". This criteria is identified as the "SETTLE-DOWN". Example XVIII shows the "SETTLE-DOWN" printout including the variables: S' from equation (63), DPRIME from equation (64), RATIO from equation (65) along with a table and printer plot of the "SETTLE-DOWN" profile as obtained from the subroutine PRINTP.

Note in the actual computer printout two additional variables are also present. These are printed as "UNC" (i.e., the uncertainty) and "RES" (i.e., the resolution). These variables are presented and discussed in reference 5 pp. 25-28.

According to the iterative sequence described in section III, the inversion procedure continues as shown in example XIX for steps (a) through (m). That is a new $\Delta\lambda$ (i.e., $\Delta\lambda_3$) is computed giving a new value for " λ " (i.e., λ_3). Next a solution for a new profile increment $(\Delta\alpha)_j$, (i.e., $(\Delta\alpha_j)_5$) is obtained by solving equation (41) with the SOLEQS routine. The profile increment is added to the previous profile giving the new profile α_j^3 . Full-wave solutions of $r(\omega_i)\{\text{real}\}$ and $r(\omega_i)\{\text{imag}\}$ are obtained for the profile α_j^3 . These $r(\omega_i)$ values are then plotted in the complex r -plane, as in example XVII. The value of " S ", from equation (66), is also computed at this point. The SETTLE-DOWN criteria is then applied with results computed for S' , from equation (63), DPRIME, from equation (64) and RATIO, from equation (65). The iterative sequence then proceeds again in a similar manner through the steps (a) through (m) with the calculation of a new $\Delta\lambda$ (i.e., $\Delta\lambda_5$) and new " λ " (i.e., λ_5). The complete "boot-strap" procedure is repeated several times with computer plots of each iterative profile and of each match between $R(\omega_i)_\epsilon$ data and the resulting fullwave values of $r(\omega_i)$ presented at each step.

If the resulting values of S' , from equation (63), and of S , from equation (66) are manually plotted against iterative step number, the results presented in figure 16 are obtained. As discussed previously in section III, with regard to figure 9, an "elbow" is obtained for the S' curve. Beyond this step oscillations are introduced into the profile by an attempt to fit to the error in the $R(\omega_i)_\epsilon$ values. This in turn leads to instability in the values of S and this instability is shown in the figure.

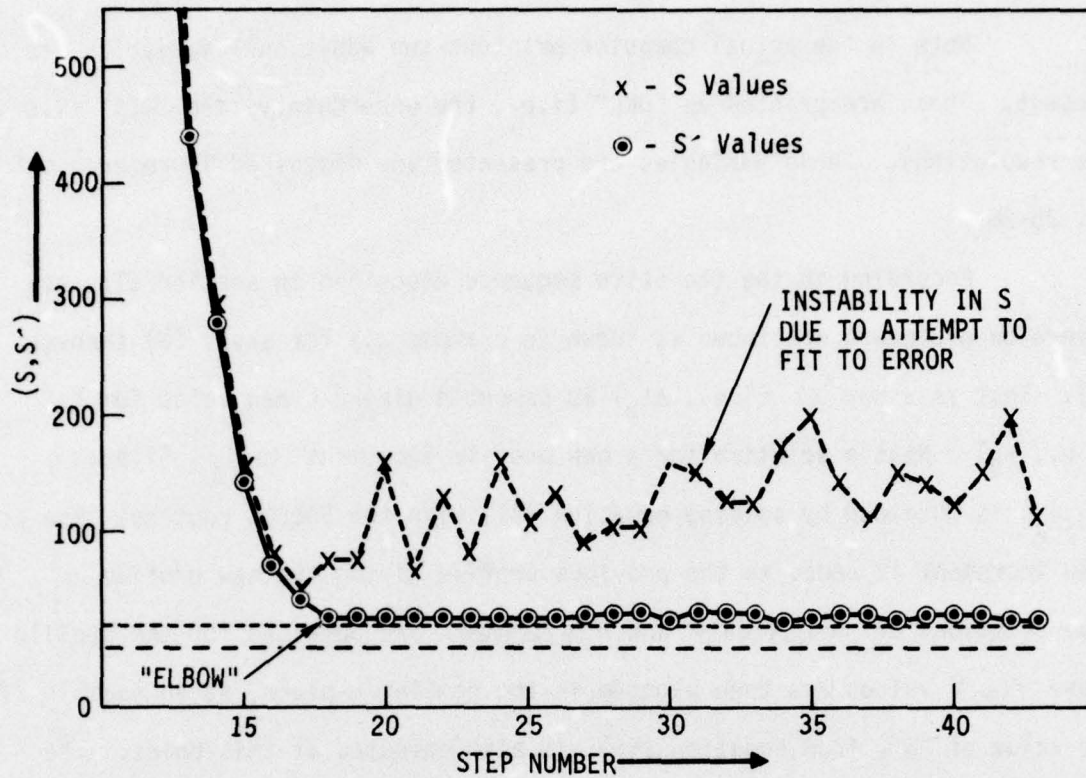


Figure 16. S and S' Curves for Random Error Set Number 2.

Figure 17 illustrates the inversion results in terms of the S and S' curves for the set of random errors obtained from examples IX, X and XI, (i.e., $IRANDM = 3$). Note that these results are qualitatively similar to those of figure 16 but quantitatively different because of the different sets of random errors present. As a matter of interest, figure 18 is presented to show the relationship between the S and S' curves when no random error is present.

The manual plotting of the S and S' results, as given by figure 16, allows for the selection of a particular iterative step (i.e., that step number at the "elbow" position) to be chosen as the one which gives the required inverted profile of electron density. For the case presented above, for the random error set number 2, the profile for step number 18 is shown in example XX. The plot compares the original simulated data profile with the chosen final

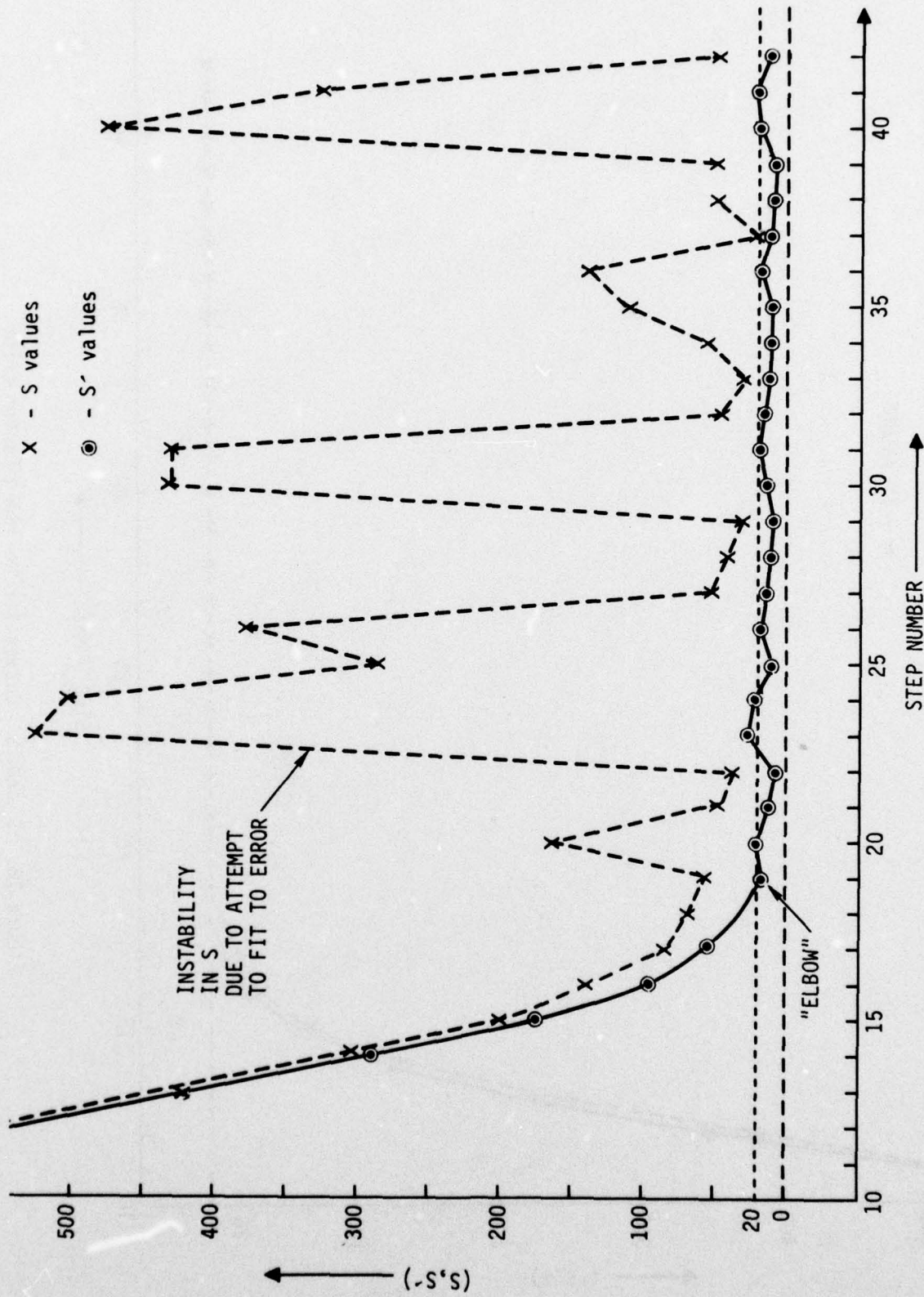


Figure 17. S and S' Curves for Random Error Set Number 3.

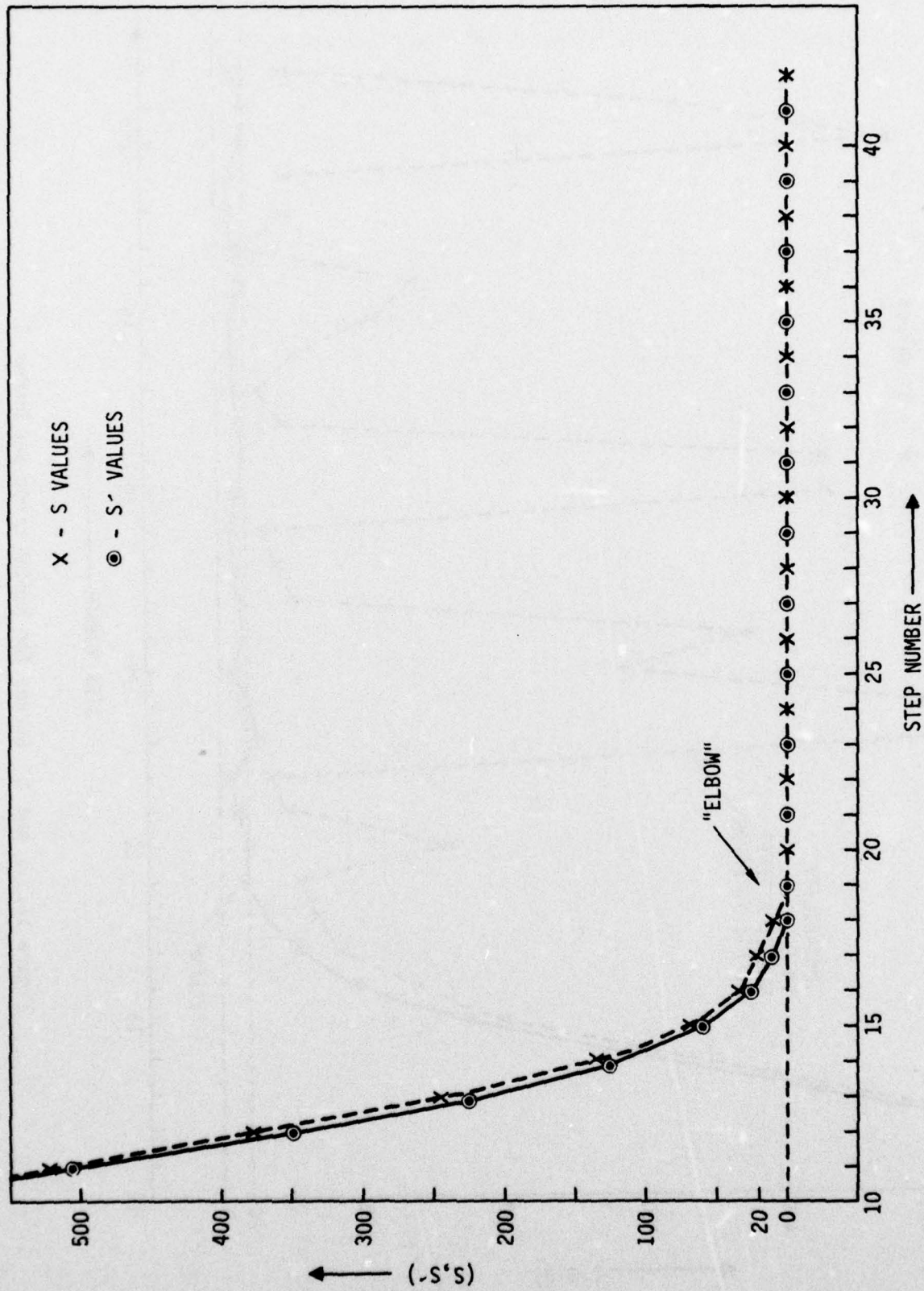


Figure 18. S and S' Curves for No Added Random Error.

inverted profile. Example XXI illustrates the R-plane comparison between the $R(\omega_i)$ data points and the full-wave computed values $r(\omega_i)$ using the above inverted profile. Also shown in this example are the printouts obtained from the INVERT calculations.

Section V-C states that for a given frequency, ω_i , and given chosen value for the expected error, σ_i , the true value of $R(\omega_i)$ will lie within the "error circle" of radius σ_i , drawn around the value $R(\omega_i)_{FW}$, 68.3% of the time for a large number of error sets. The results of example XXI are shown plotted, with these "error circles", in figure 19.

For the random error set number 3 (see examples IX, X and XI), the S and S' results were shown plotted in figure 17. The inverted profile for step number 19 is shown in example XXII. This plot compares the original simulated data profile with the chosen final inverted profile. Example XXIII illustrates the R-plane comparison between the $R(\omega_i)$ data points and the full-wave computed values, $r(\omega_i)$, using the inverted profile of example XXII as input. The results of example XXIII are also shown plotted with "error circles" in figure 20.

In figure 18, for the no error case, the scale used for the S and S' axis was such that the magnitudes of S and S' appear to be zero at step number 19 and beyond. Figure 21 shows the same plot as figure 18 except on a much expanded scale. In this case the existence of truncation error in the Runge-Kutta integration procedures is obvious. Examination of the results, shown plotted in figure 21, give the choice of integrated profile to be that represented by step number 28. This profile is presented in example XXIV as compared with the original simulated data profile. Example XXV then gives the R-plane comparison between the data points, $R(\omega_i)$ and the full-wave computed values, $r(\omega_i)_{FW}$ from the inverted profile of step number 28.

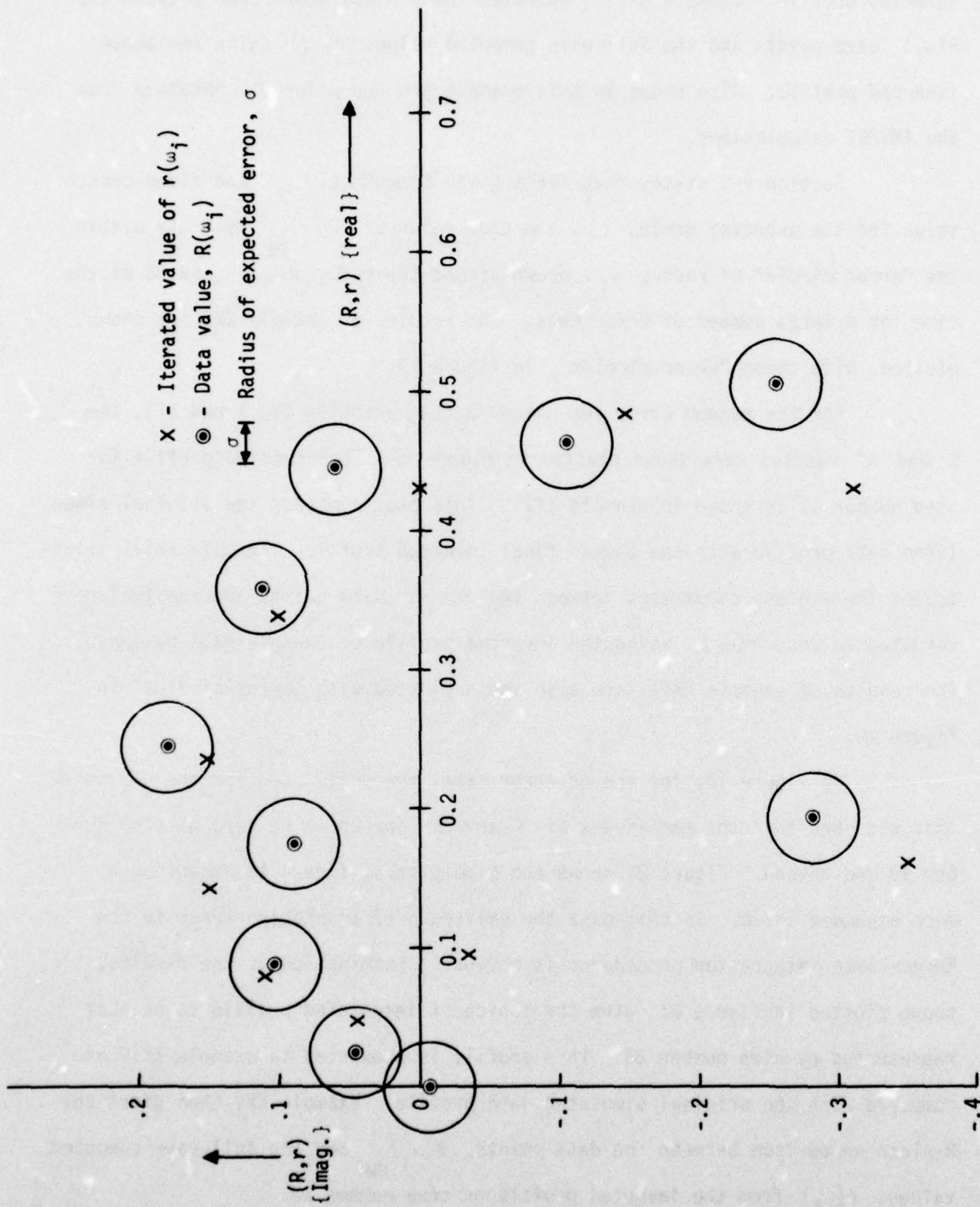


Figure 19. R, r -plane Plot for Random Error Set No. 2 - with Error Circles.

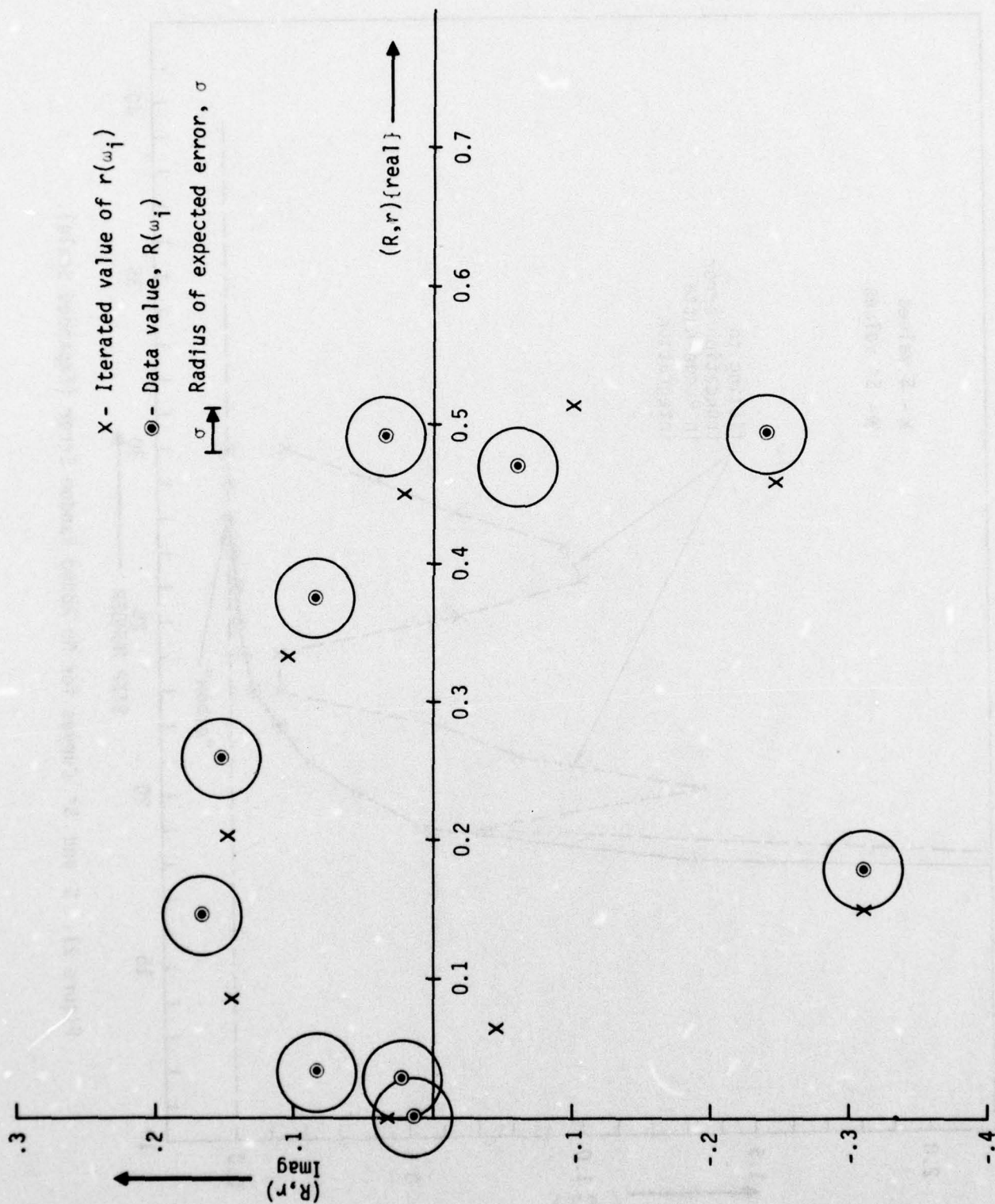


Figure 20. R, r-plane Plot for Random Error Set No. 3 - with Error Circles.

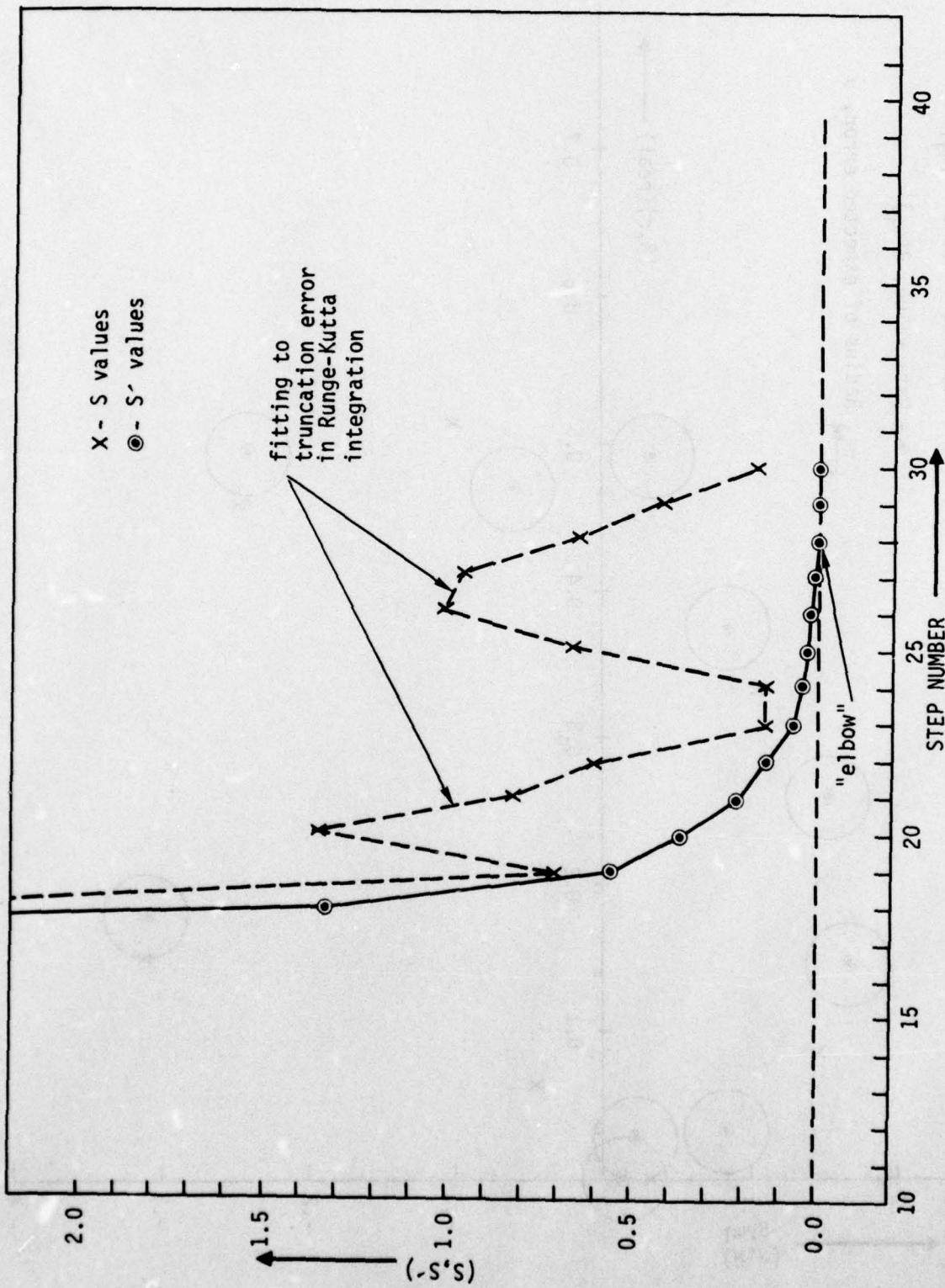


Figure 21. S and S' Curves for No Added Random Error (Expanded Scale).

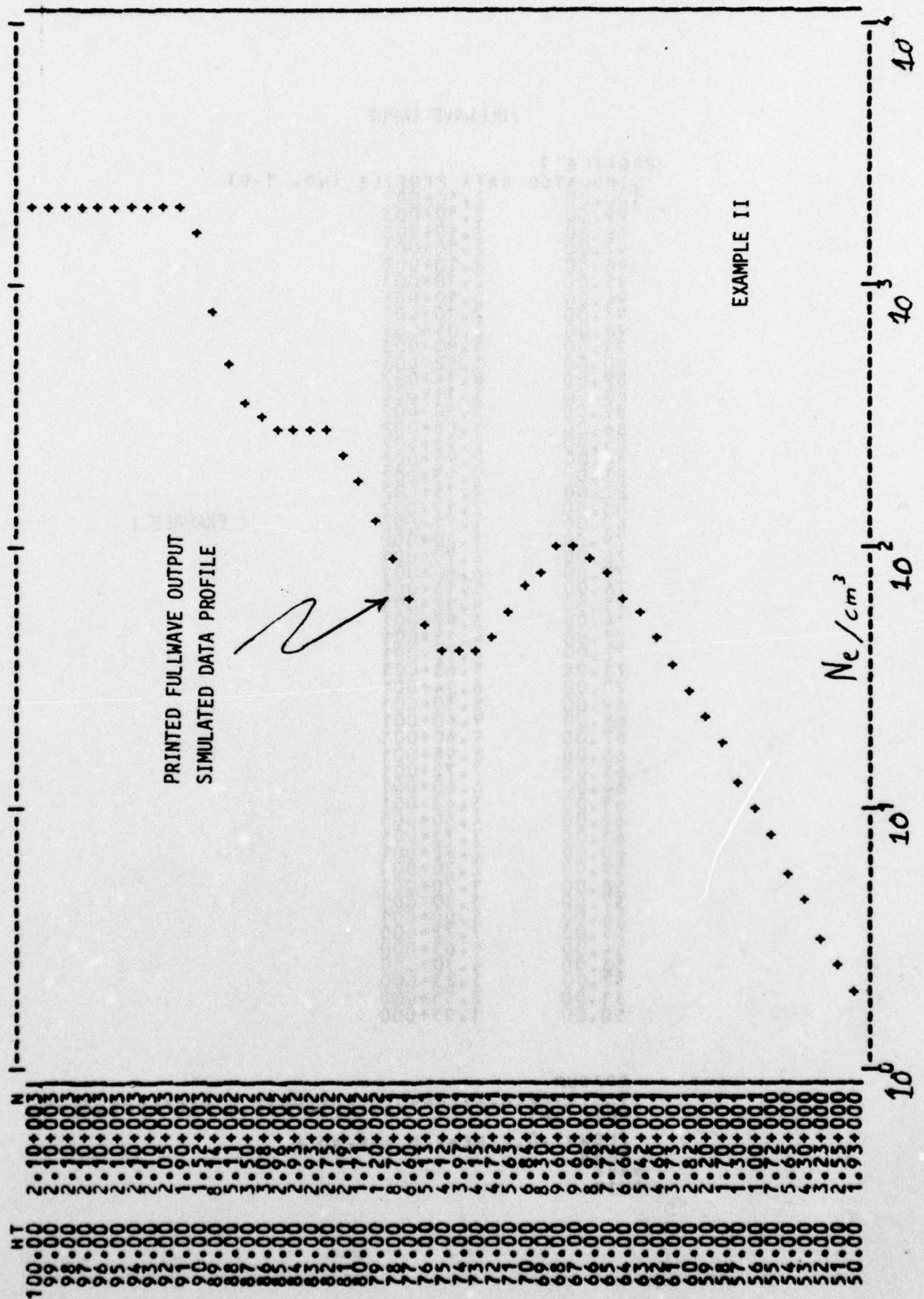
FULLWAVE INPUT

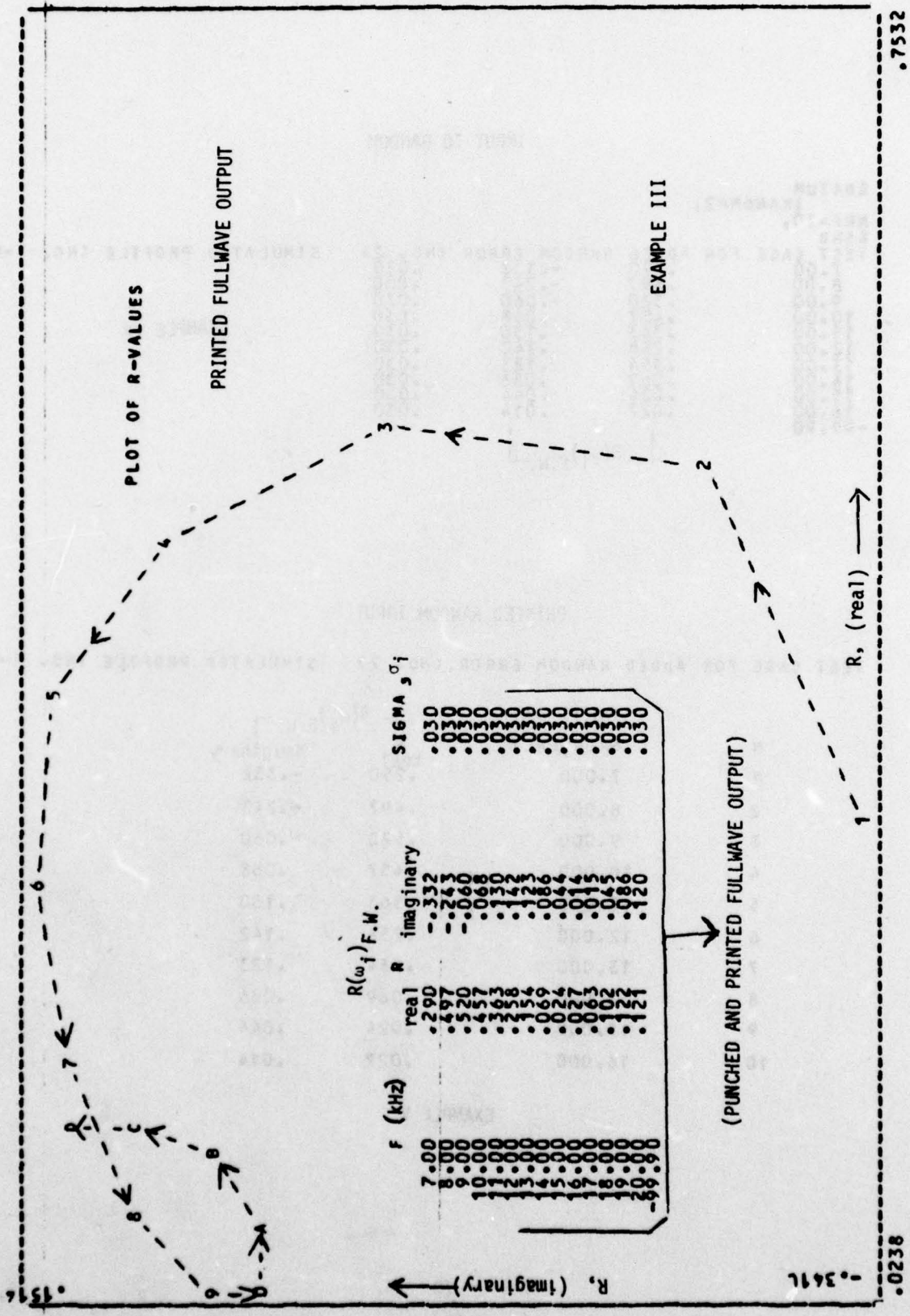
PROFILE 1
SIMULATED DATA PROFILE (NO. 1-B)

100.00	2.10+003
99.00	2.10+003
98.00	2.10+003
97.00	2.10+003
96.00	2.10+003
95.00	2.10+003
94.00	2.10+003
93.00	2.10+003
92.00	2.05+003
91.00	1.90+003
90.00	1.52+003
89.00	8.14+002
88.00	5.11+002
87.00	3.50+002
86.00	3.08+002
85.00	2.96+002
84.00	2.93+002
83.00	2.93+002
82.00	2.75+002
81.00	2.19+002
80.00	1.71+002
79.00	1.20+002
78.00	8.70+001
77.00	6.60+001
76.00	5.13+001
75.00	4.12+001
74.00	3.97+001
73.00	4.15+001
72.00	4.72+001
71.00	5.63+001
70.00	6.84+001
69.00	8.30+001
68.00	9.60+001
67.00	9.60+001
66.00	8.98+001
65.00	7.72+001
64.00	6.60+001
63.00	5.42+001
62.00	4.60+001
61.00	3.83+001
60.00	3.20+001
59.00	2.70+001
58.00	2.30+001
57.00	1.90+001
56.00	1.50+001
55.00	1.10+001
54.00	8.30+000
53.00	4.30+000
52.00	3.23+000
51.00	2.55+000
50.00	1.93+000

EXAMPLE I

SDATUM
THEIA=22.0,AZIM=90.0,CODIP=12.0,MAGFLD=5.3E-5,
NUFLAG=1,
FLOW=7.0,FINC=1.0,FHI=20.0,
SEND
WAVE
QUIT





INPUT TO RANDOM

SDATUM
 IRANDM=2,
 NRF=10,
 SEND

TEST CASE FOR ADDED RANDOM ERROR (NO. 2)	SIMULATED PROFILE (NO. 1-B)		
7.00	.290	-.332	.030
8.00	.497	-.243	.030
9.00	.520	-.060	.030
10.00	.457	.068	.030
11.00	.363	.130	.030
12.00	.258	.142	.030
13.00	.154	.123	.030
14.00	.069	.086	.030
15.00	.024	.044	.030
16.00	.027	.014	.030
-99.90			

EXAMPLE IV

$R(\omega_i)$ F.W.

PRINTED RANDOM INPUT

TEST CASE FOR ADDED RANDOM ERROR (NO. 2) SIMULATED PROFILE (NO. 1-B)

N	INPUT DATA FREQUENCY	$R(\omega_i)$ F.W.	
		R real	imaginary
1	7.000	.290	-.332
2	8.000	.497	-.243
3	9.000	.520	-.060
4	10.000	.457	.068
5	11.000	.363	.130
6	12.000	.258	.142
7	13.000	.154	.123
8	14.000	.069	.086
9	15.000	.024	.044
10	16.000	.027	.014

EXAMPLE V

PRINTED RANDOM OUTPUT

TEST CASE FOR ADDED RANDOM ERROR (NO. 2) SIMULATED PROFILE (NO. 1-B)

GAUSSIAN RANDOM X DATA

I	IRANDM	XREAL($U_{i,p}$)	IRANDM	XIMAG($U_{i,p}$)
1	1331975123	-3.233	1192245589	1.687
2	1582285849	.281	731406625	-1.123
3	875543089	-1.894	527354449	-1.523
4	1284630164	-.499	916623169	-.049
5	994699281	-.051	1634101777	-.549
6	1451175441	-.445	1752708625	1.271
7	118048273	.708	763979281	-1.057
8	830842385	.587	1744184849	.632
9	583476753	.045	1984505361	-.145
10	487400977	-.858	1913988625	-.770

EXAMPLE VI

PRINTED RANDOM OUTPUT

TEST CASE FOR ADDED RANDOM ERROR (NO. 2) SIMULATED PROFILE (NO. 1-B)

GAUSSIAN RANDOM R DATA

N	FREQUENCY	R _i R(ω_i) F.W.	RR ($R(\omega_i)$)	SIGMA
1	7.000	.290	.193	.030
2	8.000	.497	.503	.030
3	9.000	.520	-.277	.030
4	10.000	.457	-.106	.030
5	11.000	.363	.442	.030
6	12.000	.258	.365	.030
7	13.000	.154	.245	.030
8	14.000	.069	.175	.030
9	15.000	.024	.087	.030
10	16.000	.027	.025	.030
			.001	.009

EXAMPLE VII

PRINTED AND PUNCHED RANDOM OUTPUT

EXAMPLE VIII

SIMULATED DATA PROFILE (NO. 1-B) ((FOR THULE))
 *** ADDED RANDOM ERROR *** (SET NO. 2)

7.00	.193	-.281	.030
8.00	.503	-.277	.030
9.00	.463	-.106	.030
10.00	.442	.067	.030
11.00	.365	.114	.030
12.00	.245	.180	.030
13.00	.175	.091	.030
14.00	.087	.105	.030
15.00	.025	.048	.030
16.00	.001	-.009	.030

R(ω_i)_e

TEST CASE FOR ADDED RANDOM ERROR (NO. 3) SIMULATED PROFILE (NO. 1-B)

PRINTED RANDOM INPUT

EXAMPLE IX

```

RDATUM
IRANDM=3,
NRF=10,
READ
TEST CASE FOR ADDED RANDOM ERROR (NO. 3) SIMULATED PROFILE (NO. 1-B)

```

TEST CASE FOR ADDED RANDOM ERROR (NO. 3)	SIMULATED PROFILE (NO. 1-B)
0.0000	.297
0.0000	.497
0.0000	.520
11.0000	.457
11.0000	.363
12.0000	.258
12.0000	.154
12.0000	.769
15.0000	.024
15.0000	.027
15.0000	.014
15.0000	.014

$R(\omega_i)$ F.W.

PRINTED RANDOM OUTPUT

EXAMPLE X

TEST CASE FOR ADDED RANDOM ERROR (NO. 3) SIMULATED PROFILE (NO. 1-B)

GAUSSIAN RANDOM X DATA

I	IRANDM	XREAL	IRANDM	XIMAG
1	785370273	-3.722	1875951065	.657
2	1898370273	-1.041	408665303	.178
3	1898370273	-1.673	804026977	.027
4	1791550812550	1.240	404625425	-1.096
5	43957790812550	.388	980938257	-1.490
6	3505080812550	.024	629949669	.374
7	11305080812550	.470	370255377	.762
8	1235080812550	-1.198	519513617	-.752
9	1235080812550	.075	902637073	-.748
10	194323985	-1.051	1042621969	.031

PRINTED RANDOM OUTPUT

TEST CASE FOR ADDED RANDOM ERROR (NO. 3) SIMULATED PROFILE (NO. 1-B)

GAUSSIAN RANDOM R DATA

N	FREQUENCY	$R(\omega_i)$ F.W.	$RR - (R(\omega_i)_e)$	SIGMA
1	7.000	.290	-.332	.030
2	8.000	.497	-.243	.030
3	9.000	.520	-.060	.030
4	10.000	.457	.068	.030
5	11.000	.363	.130	.030
6	12.000	.256	.142	.030
7	13.000	.154	.123	.030
8	14.000	.069	.084	.030
9	15.000	.024	.044	.030
10	16.000	.27	.014	.030
			-.005	.015

PRINTED AND PUNCHED RANDOM OUTPUT

TEST CASE FOR ADDED RANDOM ERROR (NO. 3)	SIMULATED PROFILE (NO. 1-B)
7.00	.178
8.00	.496
9.00	.470
10.00	.494
11.00	.375
12.00	.259
13.00	.168
14.00	.033
15.00	.026
16.00	-.005
-99.90	

$R(\omega_i)_e$

EXAMPLE XI

SIMULATED DATA PROFILE (NO. 1-B) ((FOR THULE))
 *** ADDED RANDOM ERROR *** (SET NO. 2)

7.00	.193	-.281	.030
8.00	.505	-.277	.030
9.00	.463	-.106	.030
10.00	.442	.067	.030
11.00	.365	.114	.030
12.00	.245	.180	.030
13.00	.175	.091	.030
14.00	.087	.105	.030
15.00	.025	.048	.030
16.00	.001	-.009	.030
17.00	.000	.000	.000

FULL WAVE IN PUT $R(w_i)_e$

WAVE

&DATUM
 THETA = 22.000 DEGREES AZIM = 90.000 DEGREES
 IPRINT = 0 IPROFP = 1 IRPLOT = 1
 CODIP = 12.000 DEGREES MAGFLD = 5.300-005

&END

SIMULATED DATA PROFILE (NO. 1-B)

NAME

&DATUM

HTSTOP(1) = 100.000 HTSTOP(2) = 50.000

STOPS(1) = 7.000+002 STOPS(2) = 3.000+000

IPRNTA = 0 IPRNTE = 0 NUFLAG = 1

MIN = 9 DOPT = .400 DELTAH = 1.000
 ROTATE = 90.000 ALAMDA = .000 DELTAL = .0625
 NUFLAG = 1 COEFNU = 4.303+011 EXPNU = -.1622
 &END

PRINTOUT OF INVERT INPUT

EXAMPLE XIII

IN DGELB, IER = 0 B - values from equation (34)

-.25267	.31921	-.08406	.02214	-.00583	} etc
-.25267	.31921	-.08406	.02214	-.00583	
-.25107	.31560	-.08152	.02147	-.00562	
-.24629	.30476	-.07387	.01945	-.00512	
-.23831	.28670	-.06114	.01610	-.00424	
-.22713	.26141	-.04331	.01140	-.00300	
-.21277	.22890	-.02038	.00537	-.00141	
-.19521	.18916	.00764	-.00201	.00053	
-.17446	.14220	.04076	-.01073	.00283	
etc					

CLINEQ ERR = .000 Computed values of σ_{ij} from equation (45)

3.33+001	.00	.00	.00	.00	} etc	
.00	.00	.00	.00	.00		
3.33+001	6.58+000	3.10+000	-1.22+000	5.22+000		
-5.61-008	4.88+000	-9.45+000	4.28+000	-2.94+000		
3.33+001	7.68+000	7.64-001	-5.99-001	2.57+000		
2.87-008	5.84+000	-2.33+000	2.11+000	-1.45+000		
3.33+001	7.88+000	1.14-008	-1.52+000	5.22+000		
6.15-008	5.84+000	1.72-009	5.34+000	-2.94+000		
etc						

DATA VALUES--	$R(\omega_i)$		$\left \frac{dR_E}{d\omega} \right _i$	$\phi \left\{ \frac{dR}{d\omega} \right\}_i$	$\ln \left \frac{dR_E}{d\omega} \right _i$	$\phi \left\{ \frac{dR_E}{d\omega} \right\}_i$ radians
FREQ	RE(R)	IM(R)	MAG(DRDF)	A(DRDF)	RE(G)	IM(G)
7.00	.193	-.281	.193	-.281	-.19300	-.28100
8.00	.505	-.277	.029	36.552	-3.52534	-.63796
9.00	.463	-.106	.045	108.173	-3.11172	1.88797
10.00	.442	.067	.020	105.858	-3.91548	1.84757
11.00	.346	.114	.027	150.606	-3.61752	2.42853
12.00	.246	.180	.019	192.487	-3.94156	3.35053
13.00	.174	.091	.018	214.276	-4.02412	3.73982
14.00	.087	.105	.017	181.259	-4.08574	3.16357
15.00	.035	-.048	.016	221.435	-4.10440	4.21384
16.00	.001	-.009	.010	251.639	-4.56526	4.59493

END OF INITIALIZATION

EXAMPLE XIV

PRINTED INVERT OUTPUT

FULL-WAVE SOLUTIONS 1

7.00
8.00
9.00
10.00
11.00
13.00
15.00
16.00

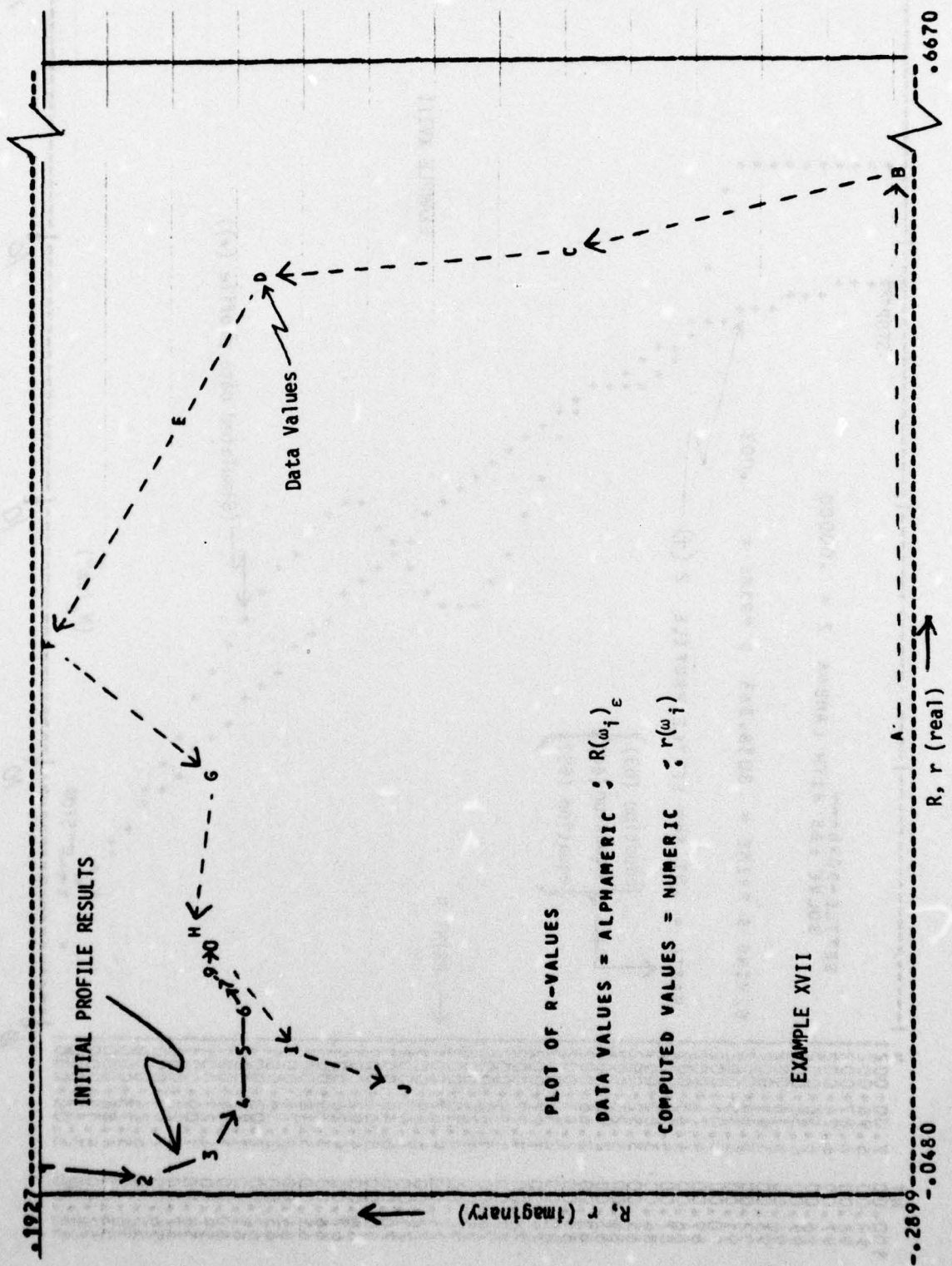
FREQ	RE(R)	IM(R)	MAG(DRDF)	A(DRDF)	$\phi' \left(\frac{dr}{d\omega} \right)$	$\ln \left \frac{dr}{d\omega} \right $	RE(G)	IM(G)	ERR	RK	RK STEPS
7.00	-.043	.184	-.043	.184	.184	-.04296	.18373	.001		50	
8.00	-.048	.130	.047	-.315	-.315	-.45769	-.18902	.001		50	
9.00	-.037	.072	.036	-.290	-.290	-.94299	-.89477	.001		50	
10.00	-.021	.080	.027	-.477	-.477	-.11101	-.10350	.001		50	
11.00	.044	.097	.021	8.002	8.002	.22012	.14793	.001		50	
12.00	.058	.097	.016	40.830	40.830	-.93103	-.71538	.002		50	
13.00	.077	.098	.006	-1.477	-1.477	-.67078	-.02578	.002		50	
15.00	.077	.097	.009	-10.799	-10.799	-.37040	-.18848	.003		50	

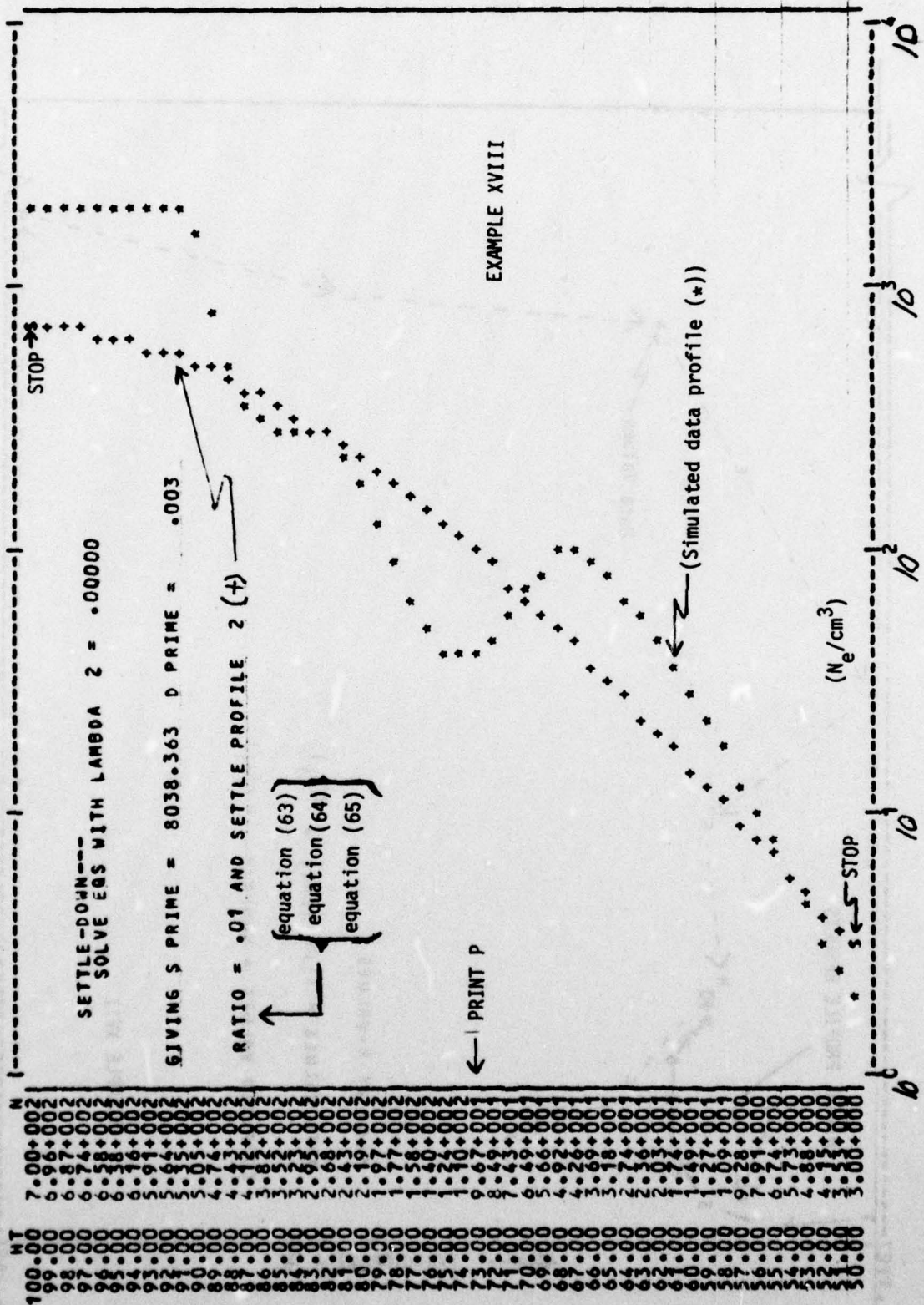
radians

$\phi \left\{ \frac{d}{d\omega} \right\}$

S 1 = 8032.817 USING FULL-WAVE VALUES (see equation (66))

EXAMPLE XVI





EXAMPLE XVIII

MT	N
100.00	7.00+002
99.00	9.96+002
98.00	6.87+002
97.00	6.74+002
96.00	6.58+002
95.00	6.41+002
94.00	5.91+002
93.00	5.64+002
92.00	5.70+002
91.00	4.43+002
90.00	4.12+002
89.00	3.82+002
88.00	3.52+002
87.00	3.23+002
86.00	2.93+002
85.00	2.63+002
84.00	2.33+002
83.00	2.03+002
82.00	1.73+002
81.00	1.43+002
80.00	1.13+002
79.00	8.77+001
78.00	1.58+002
77.00	1.20+002
76.00	1.10+002
75.00	1.10+002
74.00	9.69+001
73.00	8.49+001
72.00	7.43+001
71.00	6.62+001
70.00	5.92+001
69.00	4.26+001
68.00	3.61+001
67.00	3.18+001
66.00	2.74+001
65.00	2.33+001
64.00	2.03+001
63.00	1.74+001
62.00	1.47+001
61.00	1.27+001
60.00	1.09+001
59.00	9.28+000
58.00	8.73+000
57.00	8.15+000
56.00	7.54+000
55.00	6.92+000
54.00	6.30+000
53.00	5.68+000
52.00	5.06+000
51.00	4.44+000
50.00	3.82+000

NEW LAMBDA---

(a) DELTA LAMBDA 3 = .08839

(b) SOLVE EQS WITH LAMBDA 3 = .08839

(c) GIVING D = .241 LINEAR S = 6874.953 AND PROFILE 3

(d) FULL-WAVE SOLUTIONS 3

(e)

FREQ	RE(R)	IM(R)	MAG(DRDF)	A(DRDF)	RE(G)	IM(G)
7.00	-.056	.159	-.056	.159	-.05577	.15869
8.00	-.046	.108	.046	-69.578	-4.68869	-1.21437
9.00	-.027	.076	.036	-34.336	-4.94305	-.59928
10.00	.007	.067	.029	.094	-5.13483	-.00163
11.00	.034	.074	.026	27.527	-5.25919	.48044
12.00	.053	.088	.020	46.226	-5.53450	.80679
13.00	.067	.100	.012	55.184	-6.06836	.96515
14.00	.067	.106	.006	28.122	-6.81095	.49081
15.00	.073	.106	.008	-6.951	-6.40821	-.12132
16.00	.083	.105	.011	-6.682	-6.12248	-.11663

(f) S 3 = 6796.962 USING FULL-WAVE VALUES

(g) PLOT OF R-VALUES

(h) SETTLE-DOWN---
SOLVE EQS WITH LAMBDA 4 = .08839

(i) GIVING S PRIME = 6363.873 D PRIME = .207

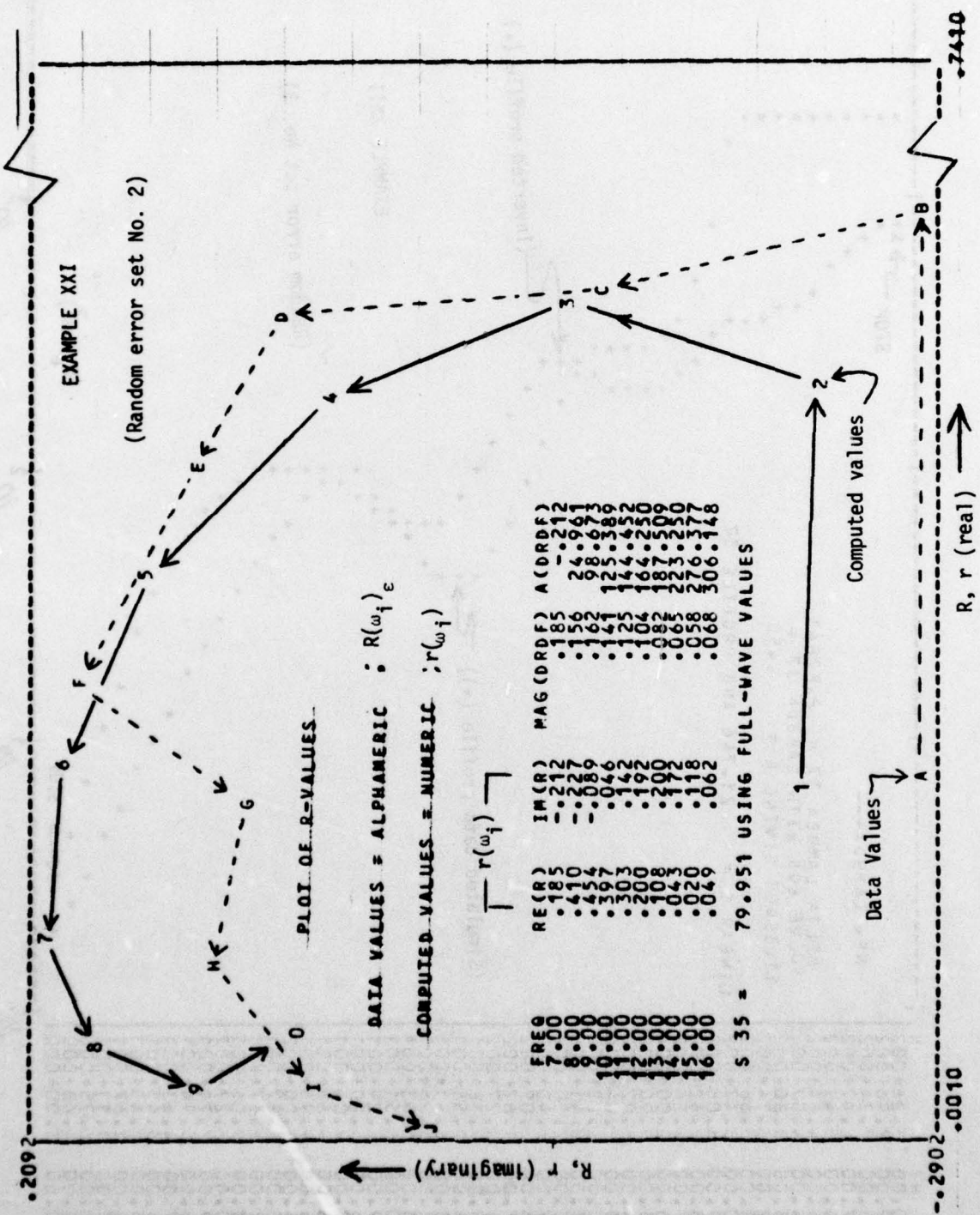
(j) RATIO = .52 AND SETTLE PROFILE 4

(k) NEW LAMBDA--- DELTA LAMBDA 5 = .12500

(l) SOLVE EQS WITH LAMBDA 5 = .21339

(m) GIVING D = .728 LINEAR S = 4239.557 AND PROFILE 5

EXAMPLE XIX



AD-A050 256

NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA
'INVERT', A COMPUTER PROGRAM FOR OBTAINING D-REGION ELECTRON DE--ETC(U)
NOV 77 D G MORFITT, C H SHELLMAN
DNA-MIPR-77-521
NL

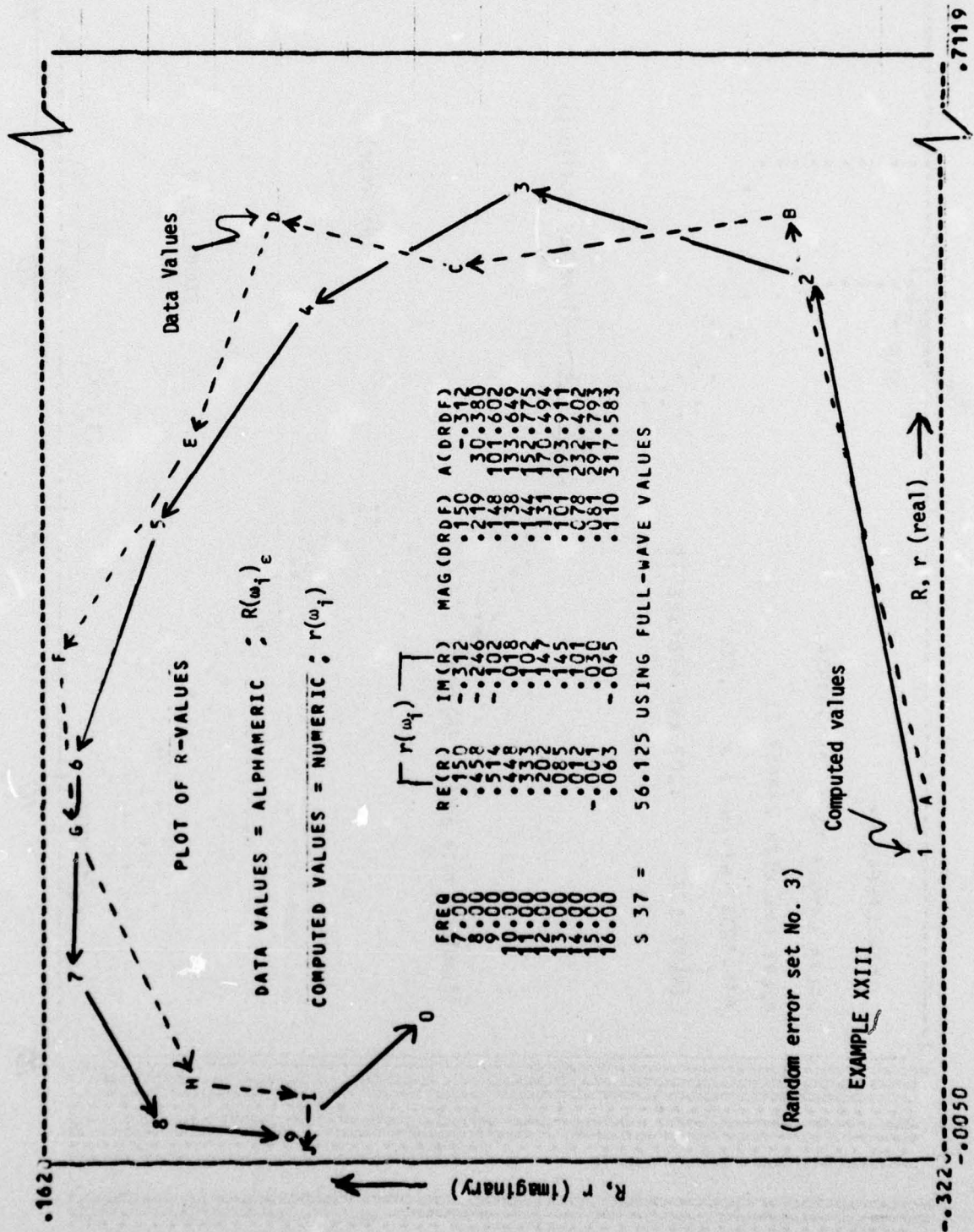
UNCLASSIFIED

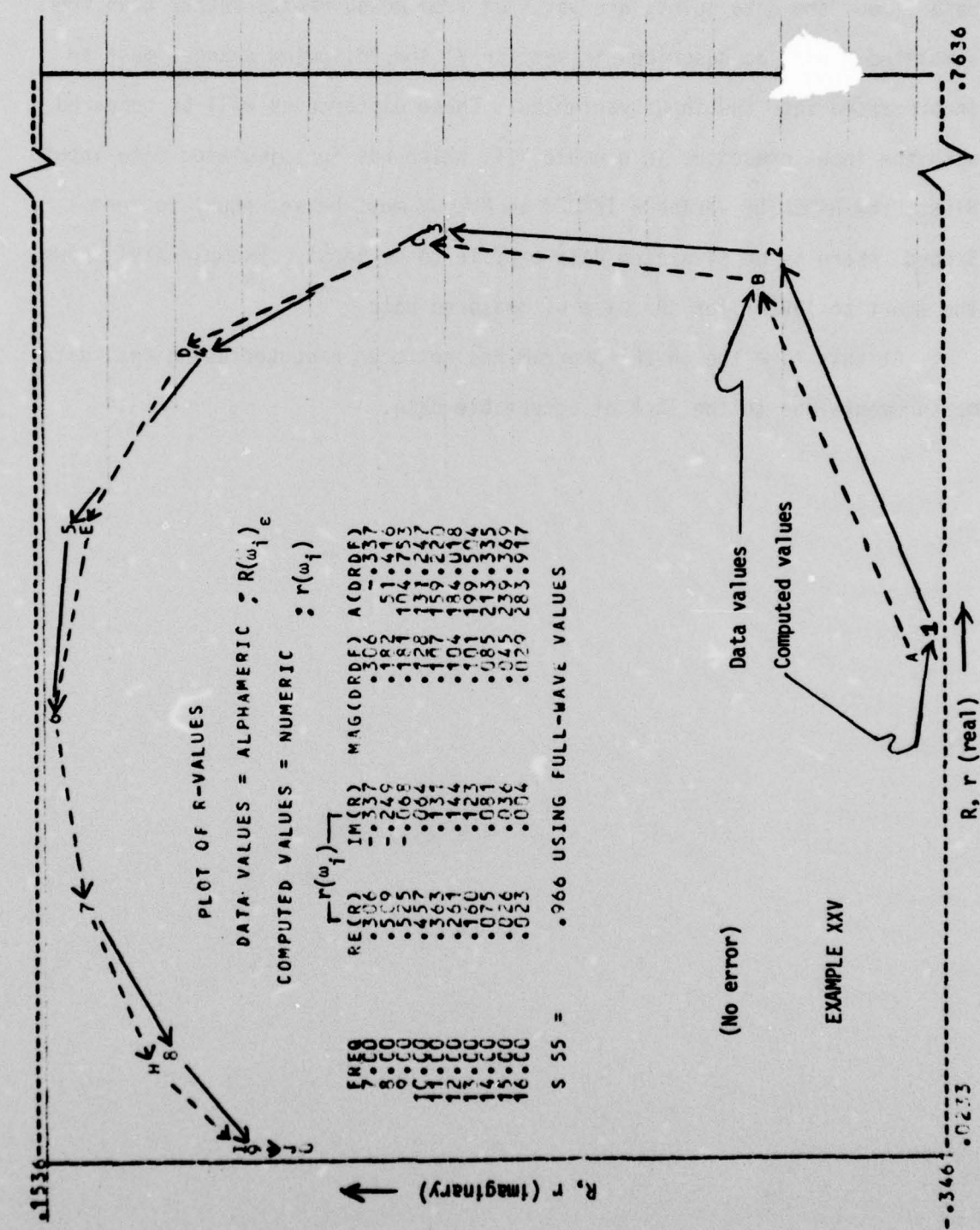
2 OF 2

AD-A050256



END
DATE
FILMED
3-78
DDC





PLOT OF R-VALUES
 DATA VALUES = ALPHAMERIC : $R(\omega_i) \epsilon$
 COMPUTED VALUES = NUMERIC : $r(\omega_i)$

$r(\omega_i)$		$R(\omega_i)$	
RE(R)	IM(R)	MAG(DRDF)	A(CDRDF)
7.000000	-.3337	.306	-.337
8.000000	-.249	.182	-.410
19.000000	-.0687	.128	107.7537
12.000000	.0971	.177	137.2220
13.000000	.1374	.104	184.018
14.000000	.123	.101	199.504
15.000000	.081	.085	213.335
16.000000	.036	.045	239.369
	.023	.029	283.917

S 55 = .966 USING FULL-WAVE VALUES

(No error)

EXAMPLE XXV

VI. COMMENTS ON EXECUTION OF INVERT FOR MEASURED DATA

For execution of the INVERT computer program using real experimental data (i.e., the data points are obtained from measurements rather than from simulated values as described in section V) the following changes must be incorporated into the input variables. These differences will be compared with the input presented in example XII, which was for simulated data input. First, the NAMELIST variable IPROFP in FULLWV must be set equal to zero. Second, there is no simulated data profile to be input. Example XXVI shows the input to INVERT for the case of measured data.

At this time the INVERT program has not been executed using real data measurements due to the lack of acceptable data.

VII. REFERENCES

1. Martin, J. N., and Hildebrand, V. E., Review of Steep-Incidence, VLF, Multiple-Frequency Sounding Techniques and Capabilities. Naval Weapons Center Technical Publication 772, 1968.
2. Budden, K. G., Radio Waves in the Ionosphere, Cambridge University Press, 1961.
3. Morfitt, D. G., and Shellman, C. H., A Technique for Obtaining D-Region Electron Density Profiles from VLF Reflection Coefficients, Defense Nuclear Agency Interim Report No. 781, 16 November 1977, prepared by Naval Ocean Systems Center.
4. Budden, K. F., "The Numerical Solution of Differential Equations Governing Reflection of Long Radio Waves from the Ionosphere," Royal Society of London, Proceedings. Series A: Mathematical and Physical Sciences, V. 227, p. 516-537, 1955.
5. Shellman, C. H., Determination of D-Region Electron Density Distributions from Radio Propagation Data, NELC TR 1856, 23 January 1973.

The following references are also of interest for inversion procedures.

6. Phillips, David L., "A Technique for the Numerical Solution of Certain Integral Equations of the First Kind." J. Ass. Comput. Mach., V. 9, p. 84-97, 1962.
7. Tomey, S., "On the Numerical Solution of Fredholm Integral Equations of the First Kind by Inversion of the Linear System Produced by Quadrature." J. Ass. Comput. Mach., V. 10, p. 97-101, 1963.
8. Lipa, Belinda, "Derivation of Directional Ocean-Wave Spectra by Integral Inversion of Second-Order Radar Echos." Radio Science, V. 12, No. 3, p. 425-434, 1977.

APPENDIX A

FORTRAN LISTING OF THE "FULLWAVE" COMPUTER PROGRAM

```

MORFITT*FULLWAVE(1).MAIN
  1      C      PROGRAM FULLWAVE
  2      C MAIN
  3      C FOR MS
  4      C
  5      COMMON/SIG COM/SIGMA(50)
  6      COMMON /R PNCH/NPUNCH
  7      COMMON/PLOT C/IPROFP,IRPLOT,ICALCP
  8      COMMON/RTOLC/RTOL
  9      COMMON/THETA C/THETA
 10      COMMON/FLD COM/AZIM,CODIP,MAG FLD
 11      COMMON/F COM/FLOW,FHI,FINC
 12      COMMON/EXPNU C/COEFNU,EXPNU
 13      COMMON/MPRNT/MPRINT
 14      COMMON/XPRNT/XPRINT
 15      COMMON/EXTRA9/IBUFFR(500)
 16      NAMELIST/DATUM/AZIM,CODIP,MAG FLD,
 17      $          THETA,RTOL,
 18      $          FLOW,FHI,FINC,
 19      $          COEFNU,EXPNU,NUFLAG,
 20      *          IPROFP,IRPLOT,ICALCP,NPUNCH,
 21      $          MPRINT,XPRINT,SIGMA
 22      INTEGER PROF,COLF,WAVE,ADAT,QUIT
 23      REAL MAG FLD
 24      INTEGER XPRINT
 25      DIMENSION IBCD(20)
 26      DATA PROF/'PROF'/,COLF/'COLF'/,IDEN/'ID '/,
 27      $          WAVE/'WAVE'/,ADAT/' &DA'/,IEND/' &EN'/,
 28      $          QUIT/'QUIT'/
 29      DATA BDATA/'TUM '/,END2/'D '/,NAME/'NAME'/
 30      C
 31      C
 32      NPUNCH=1
 33      THETA = 0.0
 34      FLOW = 0.5
 35      FHI = 25.0
 36      FINC = 0.5
 37      RTOL = 0.001
 38      MPRINT = 0
 39      XPRINT = 0
 40      IPROFP=1
 41      IRPLOT=1
 42      ICALCP=0
 43      C
 44      COEFNU = 1.816E11
 45      EXPNU = -0.15
 46      NUFLAG = 0
 47      C
 48      PRINT 67
 49      67 FORMAT('1')
 50      C
 51      10 READ 901,IBCD

```

```

52      PRINT 902,IBCD
53      IF (IBCD(1) .EQ. PROF) GO TO 11
54      IF (IBCD(1) .EQ. COLF) GO TO 12
55      IF (IBCD(1) .EQ. IDEN) GO TO 13
56      IF (IBCD(1) .EQ. WAVE) GO TO 40
57      IF (IBCD(1) .EQ. QUIT) GO TO 905
58      IF (IBCD(1) .EQ. NAME) GO TO 90
59      PRINT 900
60      900 FORMAT (' ', 'ERROR IN CONTROL CARD')
61      905 CALL OVCHK('MAIN ',905)
62      STOP
63
64      C
65      11 CALL PRFL IN
66      GO TO 10
67      12 CALL COLF IN
68      GO TO 10
69      13 READ 901,IBCD
70      PRINT 902,IBCD
71      PUNCH 901,IBCD
72      GO TO 10
73
74      C
75      90 READ(5,DATUM)
76      WRITE(6,550) ADAT,BDATA
77      550 FORMAT(' ',2A4)
78      WRITE(6,555) AZIM,CODIP,MAGFLD,THETA,RTOL
79      555 FORMAT(' ', AZIM = ',F8.3, ' CODIP = ',F8.3, ' MAGFLD = ',1PE10.3,
80      * ' THETA = ',OPF10.4, ' DEGREES', ' RTOL = ',OPF8.4)
81      WRITE(6,444) FLOW,FHI,FINC,NPUNCH
82      444 FORMAT(' ',FLOW = ',F10.4, ' KHZ', ' FHI = ',F10.4, ' KHZ',
83      * ' FINC = ',F10.4, ' KHZ', ' NPUNCH = ',I5)
84      IF (NUFLAG .NE. 1) WRITE(6,557) NUFLAG,COEFNU,EXPNU
85      557 FORMAT(' ', NUFLAG = ',I2, ' COEFNU = ',1PE11.4, ' EXPNU = ',
86      * OPF9.5)
87      WRITE(6,551) IEND,END2
88      551 FORMAT(' ',2A4,/)
89      IF (NUFLAG .EQ. 0) GO TO 10
90      COEFNU = 4.303E11
91      EXPNU = -0.1622
92      GO TO 10
93
94      C
95      40 CALL CONTRL
96      GO TO 10
97
98      C
99      901 FORMAT(20A4)
100     902 FORMAT(' ',20A4)
101
102     C
103     END

```

RT,S FULLWAVE.BDATA

```
MORFITT*FULLWAVE(1).BDATA
  1          BLOCK DATA
  2          COMMON/SIG COM/SIGMA(50)
  3          DATA SIGMA/50*0.03/
  4          END
```

```
@PRT,S FULLWAVE.PRFLIN
```

MORFITT*FULLWAVE(1).PRFLIN

```
1      SUBROUTINE PRFL IN
2      C FOR MS
3      C
4      COMMON/PLOT C/IPROFP,IRPLOT,ICALCP
5      COMMON/NR OF AS/NR A
6      COMMON/HTS COM/HTS(101)
7      COMMON/ALOGN C/ALOGEN(101)
8      COMMON/EXPNU C/COEFNU,EXPNU
9      DIMENSION IBCD(20),HTS NU(2),VAL NU(2)
10     DIMENSION Y(101)
11     DIMENSION LINE(81)
12     INTEGER V
13     C
14     DATA ISYMN/'+'/,IBLANK/' '/
15     DATA ALN TEN/2.302585/
16     DATA V/0174000000000/
17     FMESH=4.0*ALN TEN/80.0
18     C
19     C
20     READ 901,(IBCD(L),L=1,20)
21     PRINT 902,(IBCD(L),L=1,20)
22     J = 1
23     111 READ 904,HT,EN
24     904 FORMAT (F7.2,5X,E9.2)
25     IF(HT .LT. 0.0) GO TO 1000
26     HTS(J) = HT
27     Y(J) = EN
28     IF(J .NE. 1 .AND. HTS(J) .GE. HTS(J-1)) STOP
29     PRINT 906,HT,EN
30     906 FORMAT (' ',F7.2,5X,1PE9.2)
31     IF (EN .LT. 1.0E-20) EN = 1.0E-20
32     ALOGEN(J) = ALOG(EN)
33     NR A = J
34     J = J+1
35     GO TO 111
36     C
37     C
38     ENTRY COLF IN
39     J = 1
40     21 READ 905,HT,ENU
41     905 FORMAT(F7.2,4X,3(1X,1PE9.2))
42     PRINT 888,HT,ENU
43     888 FORMAT(' ',F7.2,4X,3(1X,1PE9.2))
44     IF (HT .LT. 0.0) GO TO 23
45     IF (J .GE. 3) GO TO 27
46     HTS NU(J) = HT
47     VAL NU(J) = ENU
48     J = J+1
49     GO TO 21
50     C
51     23 EXP NU = ALOG(VAL NU(1)/VAL NU(2))/(HTS NU(1)-HTS NU(2))
```

```

52         COEFNU = VAL NU(2)*EXP(-EXP NU*HTS NU(2))
53         RETURN
54     1000 CONTINUE
55     C
56     C     IF(ICALCP .EQ. 0) GO TO 75
57     C     CALL BGNPL(1)
58     C     CALL TITLE(IBC,40,'ELECTRONS',9,'ALTITUDE',8,5.,5.)
59     C     CALL XLOG(.1,1.,0.,20.)
60     C     CALL CURVE(Y,HTS,NRA,0)
61     C     CALL ENDPL(1)
62     C
63     C     75 IF(IPROFP .EQ. 0) GO TO 76
64     C     PRINT 60
65     C     60 FORMAT('1')
66     C     PRINT 100,V,V,V,V,V
67     C     100 FORMAT ('0',5X,'HT',9X,'N',18X, A1,19('-'), A1,19('-'),
68     C     * A1,19('-'), A1,19('-'), A1)
69     C     DO 11 J=1,NP A
70     C     DO 50 K = 1,81
71     C     50 LINE(K) = IBLANK
72     C     JPOST=ALOGEN(J)/FMESH+1.5
73     C     IF(JPOST .GE. 1 .AND. JPOST .LE. 81)
74     C     *LINE(JPOST)=ISYMN
75     C     11 PRINT 105,HTS(J),Y(J),V,LINE,V
76     C     105 FORMAT (' ',F7.2,1X,1PE9.2,17X, A1,81A1, A1)
77     C     PRINT 501,V,V,V,V,V
78     C     501 FORMAT(' ',35X, A1,19('-'), A1,19('-'), A1,19('-'), A1,
79     C     * 19('-'), A1 )
80     C     PRINT 502
81     C     502 FORMAT(' ',35X,'0',19X,'1',19X,'2',19X,'3',19X,'4')
82     C
83     C     76 CONTINUE
84     C     RETURN
85     C
86     C     27 PRINT 907
87     C     907 FORMAT ('0','TWO HTS ONLY ALLOWED IN NU PROFILE')
88     C     STOP
89     C
90     C     901 FORMAT (20A4)
91     C     902 FORMAT (' ',20A4)
92     C
93     C     END

```

SPRT,S FULLWAVE.INITLR

MORFITT*FULLWAVE(1).INITLR

```
1      SUBROUTINE INITL R
2  C FOR MS
3  C
4      COMMON/XPRNT/XPRINT
5      COMMON/M MTX/M11,M21,M31,M12,M22,M32,M13,M23,M33
6      COMMON/INTEGR/R11,R21,R12,R22
7      COMMON/C COM/C,S
8      COMPLEX
9      S      M11,M21,M31,M12,M22,M32,M13,M23,M33,
10     S      R11,R21,R12,R22,
11     S      QTEMP(4),EIP104,Q(2),
12     S      B4,B3,B2,B1,B0,
13     S      D11,D12,D13,D31,D32,D33,
14     S      P(2),T(2),
15     S      DEN,FACTOR
16     INTEGER XPRINT
17     DATA EIP104/(0.707107,0.707107)/
18  C
19  C
20     CSQ = C**2
21  C
22     B4 = 1.0+M33
23     B3 = S*(M13+M31)
24     B2 = -(CSQ+M33)*(1.0+M11)+M13*M31-(1.0+M33)*(CSQ+M22)+M23*M32
25     B1 = S*(M12+M23+M21*M32-(CSQ+M22)*(M13+M31))
26     B0 = (1.0+M11)*(CSQ+M22)*(CSQ+M33)+M12*M23*M31+M13*M21*M32
27     S      -M13*(CSQ+M22)*M31-(1.0+M11)*M23*M32-M12*M21*(CSQ+M33)
28  C
29     CALL QUARTC(B4,B3,B2,B1,B0,QTEMP)
30  C
31     K = 0
32     DO 21 KT=1,4
33     IF (REAL(EIP104*QTEMP(KT)) .LT. 0.0) GO TO 21
34     K = K+1
35     IF (K .GE. 3) GO TO 90
36     Q(K) = QTEMP(KT)
37 21 CONTINUE
38     IF (K .NE. 2) GO TO 90
39  C
40     IF(XPRINT .NE. 0) PRINT 901,(Q(J),J=1,2)
41 901 FORMAT('0',Q= ',2(E15.4,E13.4),/)
42  C
43     DO 31 J=1,2
44     D11 = 1.0+M11-Q(J)**2
45     D12 = M12
46     D13 = M13+Q(J)*S
47     D31 = M31+Q(J)*S
48     D32 = M32
49     D33 = CSQ+M33
50     DEN = D11*D33-D13*D31
51     P(J) = (-D12*D33+D13*D32)/DEN
```

```

52      31 T(J) = Q(J)*P(J)-S*(-D11*D32+D12*D31)/DEN
53      C
54      DEN = (T(1)*C+P(1))*(C+Q(2))-(T(2)*C+P(2))*(C+Q(1))
55      FACTOR = 2.0/DEN
56      R11 = (T(1)*(C+Q(2))-T(2)*(C+Q(1)))*FACTOR
57      R22 = ((T(1)*C+P(1))-(T(2)*C+P(2)))*FACTOR
58      R12 = -(T(1)*P(2)-T(2)*P(1))*FACTOR
59      R21 = -(Q(1)-Q(2))*FACTOR
60      RETURN
61      C
62      90 PRINT 900
63      900 FORMAT("0", "PROBLEM IN SORTING Q VALUES")
64      STOP
65      C
66      END

```

```

&PRT,S FULLWAVE.TMTRX

```

```

MORFITT*FULLWAVE(1).TMTRX
 1      SUBROUTINE T MTRX
 2      C FOR MS
 3      C
 4      COMMON/FRQ COM/FREQ
 5      COMMON/FLD COM/AZIM,CODIP,MAG FLD
 6      COMMON/WN/WAVE NR
 7      COMMON/EN COLL/HT,COLL,EN
 8      COMMON/M MTX/M11,M21,M31,M12,M22,M32,M13,M23,M33
 9      COMMON/STORE X/X11(001),X44(001),X12(001),X34(001),X14(001),
10      S      X31(001),X42(001),X32(001),X41(001)
11      COMPLEX
12      S      U,USQ,D,IUD,
13      S      USQD,TA,TB,
14      S      D TEMP,
15      S      M13D,M31D,M23D,M32D,
16      S      M2332D,M1331D,M2331D,M3213D,
17      S      M11,M21,M31,M12,M22,M32,M13,M23,M33,
18      S      X11,X44,X12,X34,X14,X31,X42,X32,X41
19      REAL MAG FLD,LY,MY,NY,IUD PRT,
20      S      LSQYSQ,MSQYSQ,NSQYSQ,LMYSQ,LNYSQ,MNYSQ
21      DIMENSION U PARTS(2),USQ PRT(2),D PARTS(2),IUD PRT(2)
22      EQUIVALENCE (U,U PARTS(1)),(USQ,USQ PRT(1)),
23      * (D,D PARTS(1)),(IUD,IUD PRT(1))
24      DATA PI/3.141592653/
25      DATA DTR/0.01745329252/
26      DATA COEFF X/3.182357E03/,COEFF Y/1.758796E11/
27      DATA VEL LT/2.997928E05/
28      C
29      C
30      CALL EN NU
31      C
32      X = COEF EN*EN
33      Z = COLL*OV OMGA
34      U PARTS(1) = 1.0
35      U PARTS(2) = -Z
36      USQ PRT(1) = 1.0-Z**2
37      USQ PRT(2) = -Z-Z
38      D = -X/(U+(USQ-YSQ))
39      IUD PRT(1) = Z*D PARTS(1)-D PARTS(2)
40      IUD PRT(2) = D PARTS(1)+Z*D PARTS(2)
41      USQD = USQ*D
42      C
43      M11 = USQD-LSQYSQ*D
44      M22 = USQD-MSQYSQ*D
45      M33 = USQD-NSQYSQ*D
46      TA = NY*IUD
47      TB = LMYSQ*D
48      M21 = +TA-TB
49      M12 = -TA-TB
50      TA = MY*IUD
51      TB = LNYSQ*D

```

```

52      M13 = +TA-TB
53      M31 = -TA-TB
54      TA = LY*IUO
55      TR = MNYSQ*D
56      M32 = +TA-TB
57      M23 = -TA-TB
58      C
59      D = 1.0/(1.0+M33)
60      M13D = M13*D
61      M31D = M31*D
62      M23D = M23*D
63      M32D = M32*D
64      M2332D = M32*M23D
65      M1331D = M31*M13D
66      M2331D = M31*M23D
67      M3213D = M32*M13D
68      C
69      L = 1
70      X11(L) = -M31D
71      X44(L) = -M13D
72      X12(L) = M32D
73      X34(L) = M23D
74      X14(L) = D
75      X31(L) = M2331D-M21
76      X42(L) = M3213D-M12
77      X32(L) = M22-M2332D
78      X41(L) = 1.0+M11-M1331D
79      RETURN
80      C
81      C
82      ENTRY INIT T
83      OMEGA = 2.0*PI*FREQ*1000.0
84      WAVE NR = OMEGA/VEL LT
85      COEF EN = COEFF X*1.0E06/OMEGA**2
86      OV OMGA = 1.0/OMEGA
87      SIN DIP = SIN(CODIP*DTR)
88      DIR CS L = SIN DIP*COS(AZIM*DTR)
89      DIR CS M = SIN DIP*SIN(AZIM*DTR)
90      DIR CS N = -COS(CO DIP*DTR)
91      Y = -COEFF Y*MAG FLD/OMEGA
92      YSQ = Y**2
93      LY = DIR CS L*Y
94      MY = DIR CS M*Y
95      NY = DIR CS N*Y
96      LSQYSQ = DIR CS L**2*YSQ
97      MSQYSQ = DIR CS M**2*YSQ
98      NSQYSQ = DIR CS N**2*YSQ
99      LMYSQ = DIR CS L*DIR CS M*YSQ
100     LNYSQ = DIR CS L*DIR CS N*YSQ
101     MNYSQ = DIR CS M*DIR CS N*YSQ
102     RETURN
103     C

```

104 END

6PRT,S FULLWAVE.ENUU

```

MORFITT*FULLWAVE(1).ENNU
  1      SUBROUTINE EN NU
  2      C FOR MS
  3      C
  4      COMMON/NR OF AS/NR A
  5      COMMON/JAY COM/J
  6      COMMON/HTS COM/HTS(101)
  7      COMMON/ALOGN C/ALOGEN(101)
  8      COMMON/EXPNU C/COEFNU,EXPNU
  9      COMMON/EN COLL/HT,ENU,EN
 10      C
 11      C
 12      F1 = (HT-HTS(J+1))*FACTR
 13      F2 = (HTS(J)-HT)*FACTR
 14      EN = EXP(ALOGEN(J)*F1+ALOGEN(J+1)*F2)
 15      ENU = COEFNU*EXP(EXPNU*HT)
 16      RETURN
 17      C
 18      C
 19      ENTRY SET FTR
 20      FACTR = 1.0/(HTS(J)-HTS(J+1))
 21      RETURN
 22      C
 23      END

```

```

@PRT,S FULLWAVE.INTEG

```

```

MORFITT*FULLWAVE(1).INTEG
  1      SUBROUTINE INTEG
  2  C FOR MS
  3  C
  4      COMMON/RTOLC/RTOL
  5      COMMON/NR OF AS/NR A
  6      COMMON/HTS COM/HTS(101)
  7      COMMON/XPRNT/XPRINT
  8      COMMON/INTEGR/R(8),DRDH(8)
  9      COMMON/X INTGR/X(8),DXDH(8)
 10      COMMON/EN COLL/HT
 11      COMMON/WN/WAVE NR
 12      COMMON/OVRFLO/IOVFLO
 13      COMMON/JAY COM/JAY
 14      DIMENSION
 15      S      RO(16),HDELRO(16),DELR1(16),DELR2(16),
 16      S      XO(8),HDELXC(8),DELX1(8),DELX2(8),
 17      S      R SAVE(16),
 18      S      RKHTS(401)
 19      INTEGER XPRINT
 20      DATA MAX NR/401/
 21      DATA DHMIN/0.01/
 22  C
 23  C
 24      N = 8
 25      IOVFLO = 0
 26      IF(XPRINT .NE. 0) CALL R COLS
 27      THIRD = 1.0/3.0
 28      CALL INIT T
 29      JAY = 1
 30      JRK = 1
 31      HT = HTS(1)
 32      CALL SET FTR
 33      CALL S MTRX
 34      CALL INITL R
 35  C
 36      CALL XFER (R,X,8)
 37      IF(XPRINT .NE. 0) PRINT 901
 38      901 FORMAT ('0')
 39      IF(XPRINT .NE. 0) CALL PRINT R
 40  C
 41      NRAM1 = NR A-1
 42      DO 79 J=1,NRAM1
 43      JAY = J
 44      CALL SET FTR
 45      30 CALL XFER(R,R SAVE,N)
 46      HT = RKHTS(JRK)
 47      CALL XFER (R,X,8)
 48      31 DELH = (RKHTS(JRK+1)-RKHTS(JRK))*0.5
 49      IF (ABS(DELH) .LT. DHMIN) GO TO 90
 50      DH = DELH*WAVE NR/2.0
 51      HDH = 0.5*DH

```

```

52      TDH = 2.0*DH
53      C
54      CALL R DERIV
55      IF (IOVFLO .NE. 0) GO TO 60
56      C
57      DO 32 I=1,N
58      RO(I) = R(I)
59      MDELRO(I) = DRDH(I)*HDH
60      32 R(I) = RO(I)+MDELRO(I)
61      C
62      DO 33 I=1,8
63      XO(I) = X(I)
64      MDELXO(I) = MDELRO(I)*2.0
65      33 X(I) = XO(I)+MDELXO(I)
66      C
67      HT = HT+0.5*DELM
68      CALL S MTRX
69      CALL R DERIV
70      IF(IOVFLO .NE. 0) GO TO 60
71      C
72      DO 34 I=1,N
73      DELR1(I) = DRDH(I)*DH
74      34 R(I) = RO(I)+0.5*DELR1(I)
75      C
76      CALL R DERIV
77      IF(IOVFLO .NE. 0) GO TO 60
78      C
79      DO 35 I=1,N
80      DELR2(I) = DRDH(I)*DH
81      35 R(I) = RO(I)+DELR2(I)
82      C
83      HT = HT+0.5*DELM
84      CALL S MTRX
85      CALL R DERIV
86      CALL X DERIV
87      IF(IOVFLO .NE. 0) GO TO 60
88      C
89      DO 36 I=1,N
90      MDELR3 = DRDH(I)*HDH
91      DELR4 = (MDELRO(I)+DELR1(I)+DELR2(I)+MDELR3)*THIRD
92      36 R(I) = RO(I)+DELR4
93      C
94      DO 37 I=1,8
95      DELX1(I) = DXDH(I)*TDH
96      37 X(I) = XO(I)+0.5*DELX1(I)
97      C
98      CALL R DERIV
99      CALL X DERIV
100     IF(IOVFLO .NE. 0) GO TO 60
101     C
102     DO 42 I=1,N
103     RO(I) = R(I)

```

```

104      HDELRO(I) = DRDH(I)*MDM
105      42 R(I) = RO(I)+HDELRO(I)
106      C
107          DO 43 I=1,8
108              DELX2(I) = DXDH(I)*TDM
109      43 X(I) = XO(I)+DELX2(I)
110      C
111          HT = HT+0.5*DELM
112          CALL S MTRX
113          CALL R DERIV
114          IF(IOVFLO .NE. 0) GO TO 60
115      C
116          DO 44 I=1,N
117              DELR1(I) = DRDH(I)*DM
118      44 R(I) = RO(I)+0.5*DELR1(I)
119      C
120          CALL R DERIV
121          IF(IOVFLO .NE. 0) GO TO 60
122      C
123          DO 45 I=1,N
124              DELR2(I) = DRDH(I)*DM
125      45 R(I) = RO(I)+DELR2(I)
126      C
127          HT = HT+0.5*DELM
128          CALL S MTRX
129          CALL R DERIV
130          CALL X DERIV
131          IF(IOVFLO .NE. 0) GO TO 60
132      C
133          DO 46 I=1,N
134              HDELR3 = DRDH(I)*MDM
135              DELR4 = (HDELRO(I)+DELR1(I)+DELR2(I)+HDELR3)*THIRD
136      46 R(I) = RO(I)+DELR4
137      C
138          DO 47 I=1,8
139              HDELX3 = DXDH(I)*MDM
140              DELX4 = (HDELX0(I)+DELX1(I)+DELX2(I)+HDELX3)*THIRD
141      47 X(I) = XO(I)+DELX4
142      C
143          IF(XPRINT .NE. 0) CALL PRINT R
144          DO 51 I=1,7,2
145              ERRSQ = (R(I)-X(I))**2+(R(I+1)-X(I+1))**2
146              RSQ = R(I)**2+R(I+1)**2
147              IF (RSQ .GT. 1.0) ERRSQ = ERRSQ/RSQ
148              IF (ERRSQ .GT. RTOL**2) GO TO 61
149      51 CONTINUE
150          GO TO 70
151      C
152      60 IF (XPRINT .NE. 0) CALL PRINT R
153      61 IF (XPRINT .NE. 0) PRINT 960
154      960 FORMAT (' ', 'OVRFLO')
155          IF (NR RK .GE. MAX NR) GO TO 95

```

```

156      IOVFLO = 0
157      CALL XFER(R SAVE,R,N)
158      CALL XFER (R SAVE,X,8)
159      HT = RKHTS(JRK)
160      CALL S MTRX
161      C
162      NRRKM1 = NR RK-1
163      DO 62 JCOUNT=JRK,NRRKM1
164      JJ = NRRKM1-JCOUNT+JRK
165      62 RKHTS(JJ+2) = RKHTS(JJ+1)
166      RKHTS(JRK+1) = (RKHTS(JRK)+RKHTS(JRK+2))*0.5
167      NR RK = NR RK+1
168      GO TO 31
169      C
170      70 JRK = JRK+1
171      IF (ABS(RKHTS(JRK)-HTS(J+1)) .GT. 0.001) GO TO 30
172      79 CONTINUE
173      RETURN
174      C
175      C
176      ENTRY SET RK
177      DO 81 J=1,NR A
178      81 RKHTS(J) = HTS(J)
179      NR RK = NR A
180      RETURN
181      C
182      90 PRINT 990
183      990 FORMAT ('0','STEP TOO SMALL IN INTEG')
184      STOP
185      C
186      95 PRINT 995
187      995 FORMAT ('0','TOO MANY STEPS IN INTEG')
188      STOP
189      C
190      END

```

@PRT,S FULLWAVE.ROUT

MORFITT+FULLWAVE(1).ROUT

```
1      SUBROUTINE R OUT
2  C FOR MS
3  C
4      COMMON/INTEGR/R(8)
5      COMMON/X INTGR/X(8)
6      COMMON/EN COLL/HT,EC OMIT(2)
7  C
8  C
9      ENTRY R COLS
10     PRINT 900
11     900 FORMAT('0',8X,'HT',10X,'11R11',16X,'11R1',16X,'1R11',17X,'1R1')
12     RETURN
13  C
14  C
15     ENTRY PRINT R
16     PRINT 901,HT,(X(I),I=1,8)
17     PRINT 901,HT,(R(I),I=1,8)
18     901 FORMAT(' ',F10.2,4(2X,F9.5,F9.5))
19     PRINT 902
20     902 FORMAT ('0')
21     RETURN
22  C
23     END
```

@PRT,S FULLWAVE.SMTRX

```

MORFITT*FULLWAVE(1).SMTRX
 1      SUBROUTINE S MTRX
 2 C    FOR MS
 3 C
 4      COMMON/THETA C/THETA
 5      COMMON/STORE X/X11(001),X44(001),X12(001),X34(001),X14(001),
 6 S      X31(001),X42(001),X32(001),X41(001)
 7      COMMON/C COM/C,S
 8      COMMON/S MTX/A11,A22,B11,B12,B22,C11,C21,C22,D11,D21,D12,D22
 9      COMPLEX
10 S      X11,X44,X12,X34,X14,X31,X42,X32,X41,
11 S      TEMP,TWOC,
12 S      A11,A22,B11,B12,B22,C11,C21,C22,D11,D21,D12,D22
13      DATA RTD/57.29578/
14 C
15 C
16      CALL T MTRX
17 C
18      L = 1
19      TEMP = X41(L)+X41(L)
20      A11 = TEMP+TEMP
21      A22 = 4.0
22      TEMP = S*X44(L)-C*X41(L)
23      B11 = TEMP+TEMP
24      B12 = -X42(L)-X42(L)
25      TWOC = C+C
26      B22 = -TWOC
27      TEMP = -S*X11(L)-C*X41(L)
28      C11 = TEMP+TEMP
29      C21 = X31(L)+X31(L)
30      C22 = -TWOC
31      D11 = CS*(X11(L)-X44(L))-(1.0-X14(L))+CSQ*(X41(L)-X14(L))
32      D21 = -C*X31(L)+S*X34(L)
33      D12 = S*X12(L)+C*X42(L)
34      D22 = -X32(L)
35      RETURN
36 C
37 C
38      ENTRY INIT S
39      C = COS(THETA/RTD)
40      CSQ = C**2
41      S = SIN(THETA/RTD)
42      CS = C*S
43      RETURN
44 C
45      END

```

@PRT,S FULLWAVE.DIFFEQ

MORFITT*FULLWAVE(1).DIFFEQ

```

1      SUBROUTINE DIFF EQ
2      C FOR MS
3      C
4      COMMON/INTEGR/R(8),DRDH(8)
5      COMMON/X INTGR/X(8),DXDH(8)
6      COMMON/S MTX/A11,A22,B11,B12,B22,C11,C21,C22,D11,D21,D12,D22
7      COMMON/EXTRA7/R11,R21,R12,R22,
8      $      DR11DH,DR21DH,DR12DH,DR22DH
9      COMMON/OVRFLO/IOVFLO
10     COMPLEX
11     $      R11,R21,R12,R22,
12     $      DR11DH,DR21DH,DR12DH,DR22DH,
13     $      A11,A22,B11,B12,B22,C11,C21,C22,D11,D21,D12,D22,
14     $      D11R11,D12R21,D21R12,D22R22,
15     $      R11R22,R12R21
16     DIMENSION R MTRX(16),DERIV(16)
17     EQUIVALENCE(R11,R MTRX),(DR11DH,DERIV)
18     C
19     C
20     ENTRY R DERIV
21     CALL XFER (R,R MTRX,8)
22     IXFLAG = 0
23     20 DO 21 I=1,8
24     IF(ABS(R MTRX(I)) .GT. 1.0E4) GO TO 90
25     21 CONTINUE
26     C
27     D11R11 = D11*R11
28     D12R21 = D12*R21
29     D21R12 = D21*R12
30     D22R22 = D22*R22
31     R11R22 = R11*R22
32     R12R21 = R12*R21
33     C
34     DR11DH = A11+(B11+C11+D11R11+D12R21+D21R12)*R11
35     $      +B12*R21+C21*R12+D22*R12R21
36     DR21DH = (B22+C11+D11R11+D12R21+D22R22)*R21
37     $      +C21*R22+D21*R11R22
38     DR12DH = (B11+C22+D11R11+D21R12+D22R22)*R12
39     $      +B12*R22+D12*R11R22
40     DR22DH = A22+(B22+C22+D12R21+D21R12+D22R22)*R22
41     $      +D11*R12R21
42     IF (IXFLAG .NE. 0) GO TO 40
43     C
44     DO 24 I=1,7,2
45     DRDH(I) = DERIV(I+1)
46     24 DRDH(I+1) = -DERIV(I)
47     RETURN
48     C
49     C
50     ENTRY X DERIV
51     CALL XFER (X,R MTRX,8)

```

```
52          IXFLAG = 1
53          GO TO 20
54 C
55          40 DO 42 I=1,7,2
56             DXDH(I) = DERIV(I+1)
57          42 DXDH(I+1) = -DERIV(I)
58             RETURN
59 C
60          90 IOVFLO = 1
61             RETURN
62 C
63          END
```

@PRT,S FULLWAVE.CONTRL

```

MORFITT*FULLWAVE(1).CONTRL
  1      SUBROUTINE CONTRL
  2      C
  3      COMMON/SIG COM/SIGMA(50)
  4      COMMON /R PNCH/NPUNCH
  5      COMMON/NRF COM/NPOINT
  6      COMMON/PLOT C/IPROFP,IRPLOT,ICALCP
  7      COMMON/RDATA C/X(50),Y(50),F(50)
  8      COMMON/F COM/FLOW,FHI,FINC
  9      COMMON/FRQ COM/FREQ
10      COMMON/INTEGR/R11,R21,R12,R22
11      COMMON/C COM/C
12      COMPLEX R11,R21,R12,R22,
13      S      CPLX I,R
14      DATA CPLX I/(0.0,1.0)/
15      ALAST=-99.9
16      C
17      C
18      PRINT 900
19      900 FORMAT ('1',6X,'F',13X,'R',16X,'SIGMA',/)
20      CALL INIT S
21      CALL SET RK
22      C
23      FREQ = FLOW
24      NPOINT = 0
25      11 CALL INTEG
26      NPOINT = NPOINT+1
27      F(NPOINT)=FREQ
28      R11 = C*R11-1.0
29      R21 = C*R21
30      R12 = C*R12
31      R22 = C*R22-1.0
32      R = (-R22+CPLX I*R12)/(-R22-CPLX I*R12)
33      X(NPOINT) = REAL(R)
34      Y(NPOINT) = AIMAG(R)
35      PRINT 910,FREQ,R,SIGMA(NPOINT)
36      910 FORMAT (' ',F7.2,3X,2F9.3,3X,F10.3)
37      IF(NPUNCH .GT. 0) PUNCH 103,FREQ,R,SIGMA(NPOINT)
38      103 FORMAT(F7.2,3X,2F10.3,F10.3)
39      C
40      FREQ = FREQ+FINC
41      IF(FREQ .LE. FHI) GO TO 11
42      PRINT 910,ALAST
43      IF(NPUNCH .GT. 0) PUNCH 103,ALAST
44      C
45      C      IF(ICALCP .EQ. 0) GO TO 75
46      C      CALL BGNPL(1)
47      C      CALL CROSS
48      C      CALL TITLE(' ',1,'REAL',4,'IMAG',4,8.,8.)
49      C      CALL GRAPH(-1.,.25,-1.,.25)
50      C      CALL MARKER(3)
51      C      CALL CURVE(X,Y,NPOINT,1)

```

```
52      C      CALL ENDPL(1)
53      C
54      75 IF(IRPLOT .EQ. 0) GO TO 76
55      PRINT 60
56      60 FORMAT('1')
57      CALL RPLOTS
58      C
59      76 CONTINUE
60      RETURN
61      C
62      END
```

@PRT,S FULLWAVE.QUARTC

MORFITT*FULLWAVE(1).QUARTC

```

1      SUBROUTINE QUARTC(B4,FOUR B3,SIX B2,FOUR B1,ONE B0,Q)
2  C FOR MS
3  C
4      COMPLEX FOUR B3,SIX B2,FOUR B1,ONE B0,Q,B4,
5      $      H,I,G,H PRIME,G PRIME,
6      $      SQ ROOT,P POS,P,LOG P,
7      $      CUB RT0,CUB RT1,CUB RT2,OMEGA1,OMEGA2,
8      $      ROOT P,ROOT Q,ROOT R,
9      $      TWLV B3,TWLV B2,F,DFDQ,DEL Q,FB4
10     COMPLEX*16 B3,B2,B1,B0,B3 SQ
11     REAL MAG POS,MAG NEG
12     DIMENSION Q(4),P RI(2)
13     EQUIVALENCE(P,P RI)
14     DATA OMEGA1/(-0.5,0.8660254038)/,OMEGA2/(-0.5,-0.8660254038)/
15     DATA PRECSN/1.0E-4/
16  C
17  C
18     B3 = FOUR B3/(4.0*B4)
19     B2 = SIX B2/(6.0*B4)
20     B1 = FOUR B1/(4.0*B4)
21     B0 = ONE B0/B4
22  C
23     B3 SQ = B3**2
24     H = B2-B3 SQ
25     I = B0-4.0*B3*B1+3.0*B2**2
26     G = B1+B3*(-3.0*B2+2.0*B3 SQ)
27     H PRIME = -I/12.0
28     G PRIME = -G**2/4.0-H*(H**2+3.0*H PRIME)
29  C
30     SQ ROOT = CSQRT(G PRIME**2+4.0*H PRIME**3)
31     P = (-G PRIME+SQ ROOT)*0.5
32     MAG POS = ABS(P RI(1))+ABS(P RI(2))
33     P POS = P
34     P = (-G PRIME-SQ ROOT)*0.5
35     MAG NEG = ABS(P RI(1))+ABS(P RI(2))
36     IF(MAG POS .GT. MAG NEG) P = P POS
37     LOG P = CLOG(P)
38     CUB RT0 = CEXP(LOG P/3.0)
39     CUB RT1 = OMEGA1*CUB RT0
40     CUB RT2 = OMEGA2*CUB RT0
41  C
42     ROOT P = CSQRT(CUB RT0-H PRIME/CUB RT0-H)
43     ROOT Q = CSQRT(CUB RT1-H PRIME/CUB RT1-H)
44     ROOT R = CSQRT(CUB RT2-H PRIME/CUB RT2-H)
45  C      IF(CABS(G) .LT. 1.0E-50) GO TO 21
46     IF(CABS(G) .LT. 1.0E-36) GO TO 21
47     SIGN = -ROOT P*ROOT Q*ROOT R*2.0/G
48     IF(SIGN .LT. 0.0) ROOT R = -ROOT R
49     21 Q(1) = +ROOT P+ROOT Q+ROOT R-B3
50     Q(2) = +ROOT P-ROOT Q-ROOT R-B3
51     Q(3) = -ROOT P+ROOT Q-ROOT R-B3

```

TJM
TJM

```

52      Q(4) = -ROOT P-ROOT Q+ROOT R-B3
53  C
54      FB4 = 4.0*B4
55      TWLV B3 = 3.0*FOUR B3
56      TWLV B2 = 2.0*SIX B2
57      DO 52 J=1,4
58      ITER = 0
59      51 F = (((B4*Q(J)+FOUR B3)*Q(J)+SIX B2)*Q(J)+FOUR B1)*Q(J)+ONE B0
60      DFDQ = ((FB4*Q(J)+TWLV B3)*Q(J)+TWLV B2)*Q(J)+FOUR B1
61      DEL Q = -F/DFDQ
62      Q(J) = Q(J)+DEL Q
63      IF(CABS(DEL Q/Q(J)) .LT. PRECSN) GO TO 52
64      ITER = ITER+1
65      IF(ITER .GE. 10) GO TO 90
66      GO TO 51
67      52 CONTINUE
68      RETURN
69  C
70      90 PRINT 900
71      900 FORMAT('Q', 'Q FAILS TO CONVERGE IN QUARTC')
72      STOP
73  C
74      END

```

@PRT,S FULLWAVE.XFER

```
MORFITT*FULLWAVE(1).XFER
  1      SUBROUTINE XFER (A,B,N)
  2      C FOR MS
  3      C
  4      DIMENSION A(1),B(1)
  5      C
  6      C
  7      DO 11 J=1,N
  8      11 B(J) = A(J)
  9      RETURN
 10      C
 11      END
```

```
@PRT,S FULLWAVE.RPLOTS
```

```

MORFITT*FULLWAVE(1).RPLOTS
1      SUBROUTINE RPLOTS
2      C
3      COMMON/RDATA C/X(50),Y(50),F(50)
4      COMMON/NRF COM/NR F
5      DIMENSION ISYMD(50),XD(50),YD(50),
6      * LINE(101)
7      INTEGER V
8      C
9      DATA ISYMD/'1','2','3','4','5','6','7','8','9','0',
10     * 'A','B','C','D','E','F','G','H','I','J',
11     * 'K','L','M','N','O','P','Q','R','S','T',
12     * 'U','V','W','X','Y','Z','1','2','3','4'/
13     DATA IBLANK/' '/
14     DATA V/0174000000000/
15     C
16     XMAX=-1.0E9
17     XMIN=+1.0E9
18     YMAX=-1.0E9
19     YMIN=+1.0E9
20     C
21     DO 10 L=1,NR F
22     XD(L)=X(L)
23     YD(L)=Y(L)
24     IF(XD(L) .LT. XMIN) XMIN=XD(L)
25     IF(XD(L) .GT. XMAX) XMAX=XD(L)
26     IF(YD(L) .LT. YMIN) YMIN=YD(L)
27     IF(YD(L) .GT. YMAX) YMAX=YD(L)
28     10 CONTINUE
29     C
30     IF((XMAX-XMIN)/10.0 .GT. (YMAX-YMIN)/6.5) GO TO 100
31     YMESH=(YMAX-YMIN)/52.0
32     XMESH=YMESH*0.8
33     XMAX=XMIN+(100.0*XMESH)
34     GO TO 200
35     100 XMESH=(XMAX-XMIN)/100.0
36     YMESH=XMESH/0.8
37     YMAX=YMIN+(52.0*YMESH)
38     200 CONTINUE
39     PRINT 30
40     30 FORMAT('1', ' PLOT OF R-VALUES')
41     YTOP=YMAX+YMESH
42     PRINT 31,YTOP
43     31 FORMAT(' ',F10.4,101(' - '))
44     DO 300 I=1,53
45     DO 350 J=1,101
46     350 LINE(J) = IBLANK
47     DO 400 L=1,NR F
48     ITEST = (YD(L)-YMIN)/YMESH+0.5
49     IF((ITEST +I) .NE. 53) GO TO 400
50     JPOST = (XD(L)-XMIN)/XMESH+1.5
51     LINE(JPOST) = ISYMD(L)

```

```
52 400 CONTINUE
53 C
54 300 PRINT 555,V,LINE,V
55 555 FORMAT(' ',9X, A1,101A1, A1)
56 YBOT=YMIN-YMESH
57 PRINT 32,YBOT
58 32 FORMAT(' ',F10.4,101('-') )
59 PRINT 33,XMIN,XMAX
60 33 FORMAT(' ',5X,F10.4,90X,F10.4)
61 C
62 RETURN
63 END
```

@PRT,S FULLWAVE.OVCHK

MORFITT*FULLWAVE(1).OVCHK

```

1      SUBROUTINE OVCHK(CALLER,STMT)                                DMP
2  C ***** THIS SUBROUTINE WILL CALL THE ROUTINE OVUNFL WHICH    DMP
3  C ***** CHECKS FOR OVERFLOW AND UNDERFLOW. THE ROUTINE RETURNS AN DMP
4  C ***** INTEGER VALUE WHICH TELLS WHETHER OVERFLOW AND/OR UNDERFLOW DMP
5  C ***** HAVE OCCURRED SINCE THE PROGRAM STARTED OR SINCE THE LAST TIME DMP
6  C ***** ONE OF THE OVERFLOW/UNDERFLOW CHECKING ROUTINES WAS CALLED, DMP
7  C ***** WHICHEVER HAS BEEN MORE RECENT. THE INTEGER VALUES RETURNED DMP
8  C ***** AND THEIR MEANINGS ARE AS FOLLOWS:                    DMP
9  C *****      1= OVERFLOW ONLY                                  DMP
10 C *****      2= NO OVERFLOW OR UNDERFLOW                     DMP
11 C *****      3= UNDERFLOW ONLY                                DMP
12 C *****      4= OVERFLOW AND UNDERFLOW                       DMP
13 C ***** THIS SUBROUTINE (OVCHK) WILL PRINT OUT A MESSAGE STATING DMP
14 C ***** WHETHER OVERFLOW HAS OR HAS NOT OCCURRED. FOR PURPOSES DMP
15 C ***** OF THIS CONVERSION, UNDERFLOW OCCURRENCES HAVE BEEN IGNORED DMP
16 C ***** BECAUSE THE 1110 TAKES THE SAME ACTION (I.E. SETS THE RESULT DMP
17 C ***** TO ZERO) IN THE CASE OF UNDERFLOW AS DOES THE 360.    DMP
18 C ***** THE SUBROUTINE ALSO PRINTS OUT THE NAME OF THE CALLING DMP
19 C ***** ROUTINE (VARIABLE 'CALLER') AND THE STATEMENT NUMBER FROM DMP
20 C ***** WHICH IT WAS CALLED (VARIABLE 'STMT')                 DMP
21      CHARACTER*6 CALLER                                         DMP
22      INTEGER STMT                                              DMP
23      CALL OVUNFL(IOVFL)                                         DMP
24      IF (IOVFL .EQ. 1 .OR. IOVFL .EQ. 4) GO TO 150             DMP
25      WRITE (6,100)                                              DMP
26      100 FORMAT ('NO OVERFLOW HAS NOT OCCURRED')              DMP
27      GO TO 250                                                  DMP
28      150 WRITE (6,200)                                          DMP
29      200 FORMAT ('NO OVERFLOW HAS OCCURRED')                  DMP
30      250 CONTINUE                                              DMP
31      WRITE (6,300) STMT,CALLER                                  DMP
32      300 FORMAT ('OVCHK WAS CALLED FROM STATEMENT ',I5,' IN SUBROUTINE ' DMP
33      1 ,A6)                                                    DMP
34      RETURN                                                    DMP
35      END                                                        DMP

```

2FIN

PROGRAM RANDOM (CONT)

```

56. 26 FORMAT('0',20X,'GAUSSIAN RANDOM X DATA')
57. PRINT 27
58. 27 FORMAT('0',4X,'I',2X,'IRANDM',10X,'XREAL',10X,'IRANDM',
59. *10X,'XIMAG')
60. DO 28 I=1,NRF
61. 28 PRINT 29,I,IRDMR(I),RDMR(I),IRDMI(I),RDMI(I)
62. 29 FORMAT(' ',15,1X,I10,2X,F10.3,3X,I10,5X,F10.3)
63. C
64. PRINT 16,IDENT
65. PRINT 21
66. 21 FORMAT('0',20X,'GAUSSIAN RANDOM R DATA')
67. PRINT 22
68. 22 FORMAT('0',5X,'N',5X,'FREQUENCY',10X,'R',20X,'RR',20X,'SIGMA')
69. DO 30 I=1,NRF
70. PRINT 31,I,FREQ(I),R(I),RR(I),SIGMA(I)
71. 31 FORMAT(' ',2X,I5,F13.3,2X,2F10.3,2X,2F10.3,2X,2F10.3)
72. PUNCH 32,FREQ(I),RR(I),SIGMA(I)
73. 32 FORMAT('F',2,3X,2F10.3,F10.3)
74. 30 CONTINUE
75. C
76. PUNCH 32,ALAST
77. 58 CONTINUE
78. STOP
79. END

```

```

1. SUBROUTINE RANDOM (IX,X)
2. THIS ROUTINE IS SPECIFIC TO SYSTEM 1110. IT GENERATES APPROXIMATELY
3. GAUSSIAN DISTRIBUTED RANDOM FLOATING NUMBERS WITH
4. UNIT STANDARD DEVIATION. (2**29)/12 NUMBERS WILL BE
5. GENERATED BEFORE SEQUENCE IS REPEATED.
6. INITIAL VALUE OF IX MAY BE ANY INTEGER OF NINE DIGITS OR LESS.
7. P IS A UNIFORMLY DISTRIBUTED RANDOM VARIABLE WITH VALUES
8. BETWEEN 0.0 AND 1.0.
9.
10. RANDOM GENERATES A RANDOM NUMBER FROM A SEED AND SIMULATES AN
11. IBM 360 32 BIT WORD GENERATION. THE CALLING SEQUENCE IS THE
12. SAME AS FOR THE IBM 360 RANDOM SUBROUTINE.
13. THE SEED 'IX' MUST BE A POSITIVE INTEGER.
14. DATA IONES /017/
15. IX=IX*2-1
16. X=-6.0
17. DO 11 L=1,12
18. IX = IX * 65539
19. C IF (IX) 5, 6, 6
20. 5 IF(BITS(IX,5,1) .EQ. 0) GO TO 6
21. C BITS (IX, 1, 4) = IONES
22. IX = IX + 2147483647 + 1
23. IX = IX + 2147483647
24. GO TO 7
25. 6 BITS (IX, 1, 4) = 0
26. 7 YFL = IX
27. YFL = YFL * .4656613E-9
28. 11 X = X + YFL
29. RETURN
30. END

```

APPENDIX C

FORTRAN LISTING OF THE "INVERT" COMPUTER PROGRAM

PROGRAM INVERT

MORFITT*INVERT(1).MAIN

```

1      C          MAIN
2      C          PROGRAM ""INVERT""
3      C          THIS VERSION OF THE IONOSPHERIC PROFILE INVERSION
4      C          PROGRAM WILL WORK FOR AN ARBITRARY CHOICE OF FREQUENCIES
5      C          AS LONG AS THE FOLLOWING RULE IS APPLIED.
6      C
7      C          ((FMAX-FMIN)/DELTA) .LE. 200.
8      C
9      C
10     C          COMMON/NR COM/NR
11     C          COMMON/PLOT C/IPROFP,IRPLOT
12     C          COMMON/ANS COM/ANSWER(51)
13     C          COMMON/NORG C/NORG
14     C          COMMON/MIN C/MIN
15     C          COMMON/DOPT C/DOPT
16     C          COMMON/ROTAT C/ROTATE
17     C          COMMON/DELTA C/DELTA
18     C          COMMON/DELTAH C/DELTAH
19     C          COMMON/HS COM/HTSTOP(2)
20     C          COMMON/STOPS C/STOPS(2)
21     C          COMMON/EXPNU C/COEFNU,EXPNU
22     C          COMMON/IPA COM/IPRNTA
23     C          COMMON/IPE COM/IPRNTI
24     C          COMMON/ALPHA C/ALPHA(51)
25     C          COMMON/NS COM/NR STPS(25,20)
26     C
27     C          COMMON/AL COM/ALAMDA
28     C          COMMON/DL COM/DELTAL
29     C          NAMELIST/DATUM/
30     C          $          MIN,
31     C          $          NR,
32     C          $          DOPT,
33     C          $          ROTATE,
34     C          $          DELTA,
35     C          $          DELTAH,
36     C          $          HTSTOP,
37     C          $          STOPS,
38     C          $          COEFNU,EXPNU,NUFLAG,
39     C          $          IPRNTA,
40     C          $          IPRNTI,
41     C          $          ALPHA,
42     C          $          NR STPS,
43     C          $
44     C          $          ALAMDA,
45     C          $          DELTAL
46     C          DIMENSION IBCD(20)
47     C          DATA NAME/'NAME',ADAT/' &DA'/,DAT2/'TUM'/,
48     C          $ IEND/' &EN'/,END2/'D' /
49     C
50     C
51     C          PRINT 100

```

```

52      100 FORMAT('1')
53      C
54      READ 901,IBCD
55      PRINT 902,IBCD
56      C
57      CALL D INPUT
58      CALL FW INPT
59      C
60      IF(IPROFP .NE. 1) GO TO 66
61      C
62      C      READ IN AN ORIGINAL PROFILE
63      C
64      READ 901,IBCD
65      PRINT 902,IBCD
66      J=1
67      60 READ 67,HTD,ELD
68      67 FORMAT(F7.2,5X,1PE9.2)
69      IF(HTD .LT. 0.0) GO TO 98
70      ANSWER(J) = ALOG(ELD)
71      J = J+1
72      GO TO 60
73      98 NORG=J-1
74      66 CONTINUE
75      C
76      MIN = 10
77      UOPT = 1.0
78      NR=1
79
80      DELF= 0.1
81      DELTAH = 1.0
82      HTSTOP(1) = 93.0
83      HTSTOP(2) = 50.0
84      STOPS(1) = 1.0E3
85      STOPS(2) = 3.0E0
86      COEFNU = 1.816E11
87      EXPNU = -0.15
88      NUFLAG = 0
89      IPRNTA = 0
90      IPRNTE = 0
91      ROTATE = 0.0
92      ALAMDA = 0.0
93      DELTAL = 1.0/64.0
94      C
95      READ 901,IBCD
96      PRINT 902,IBCD
97      C
98      IF(IBCD(1) .EQ. NAME) GO TO 90
99      IF(IBCD(1) .NE. NAME) PRINT 600
100     600 FORMAT('0', ' NO NAMELIST INPUT.----PROGRAM STOPS----')
101     GO TO 999
102     90 READ(5,DATUM)
103     PRINT 601,ADAT,DAT2

```

```

104      601 FORMAT(' ',2A4)
105      IF (NUFLAG .EQ. 0) GO TO 27
106      COEFNU = 4.303E11
107      EXPNU = -0.1622
108      27 CONTINUE
109      PRINT 602,HTSTOP(1),HTSTOP(2),STOPS(1),STOPS(2)
110      602 FORMAT(' ', HTSTOP(1) = ',F8.3, HTSTOP(2) = ',F8.3,
111      * ' STOPS(1) = ',1PE10.3, STOPS(2) = ',1PE10.3)
112      PRINT 201,MIN,DOPT, DELTAH
113      201 FORMAT(' ', MIN = ',I5, DOPT = ',F9.3,
114      $ ' DELTAH = ',F9.3)
115      PRINT 202, ROTATE ,ALAMDA,DELTAL,IPRNTA,IPRNTE,NUFLAG
116      202 FORMAT(' ', ROTATE = ',F9.3, ALAMDA = ',F9.3, DELTAL = ',
117      $ F9.4, IPRNTA = ',I2, IPRNTE = ',I2, NUFLAG = ',I2)
118      IF(NUFLAG .EQ. 1) PRINT 203,NUFLAG,COEFNU,EXPNU
119      203 FORMAT(' ', NUFLAG = ',I2, COEFNU = ',1PE10.3,
120      $ ' EXPNU = ',GPF8.4)
121      PRINT 204,IFND,END2
122      204 FORMAT(' ', 2A4,/)
123      C
124      CALL CONTRL
125      C
126      901 FORMAT (20A4)
127      902 FORMAT (' ',20A4)
128      C
129      999 STOP
130      C
131      END

```

@PRT,S INVERT.CONTRL

MORFITT*INVERT(1).CONTRL

```

1      SUBROUTINE CONTRL
2
3      COMMON/NR COM/NR
4      COMMON/PLOT C/IPROFP,IRPLOT
5      COMMON/MIN C/MIN
6      COMMON/D OPT C/D OPT
7      COMMON/NS COM/NR STPS(25,20)
8      COMMON/NRF COM/NR F
9      COMMON/NRA COM/NR A
10     COMMON/ALPHA C/ALPHA(51)
11     COMMON/DA COM/DA(51)
12     COMMON/ADIST C/ADIST
13     COMMON/OVRFLO/IOVFLO
14     COMMON/S COM/S
15     COMMON/AL COM/ALAMDA
16     COMMON/SN COM/SNEXT
17     COMMON/DL COM/DELTAL
18     DIMENSION ALPHA(51),THRESH(4),NS(25,20)
19     DATA THRESH/0.8,1.2,2.0,4.0/
20
21     C
22     C
23     NSEC = 60*MIN
24     FNSEC = NSEC
25     CALL SETCLK
26
27     C
28     IFIRST = 1
29     CALL SET HTS
30     CALL INTERP
31     CALL LOGS
32
33     C
34     PRINT 101
35     101 FORMAT('0','END OF INITIALIZATION')
36     PRINT 103
37     103 FORMAT('1','INITIAL PROFILE--')
38     IF(ALAMDA .EQ. 0.0) GO TO 51
39     PRINT 104,NR,ALAMDA
40     104 FORMAT(' ',3X,'CONTINUE WITH LAMBDA',I3,' = ',F12.5,1X,
41     ' (NOT USED UNTIL SETTLE-DOWN)')
42
43     C
44     15 CALL PRINT N
45     PRINT 105,NR
46     105 FORMAT(' ',3X,'FULL-WAVE SOLUTIONS',I3)
47     CALL R DRDA
48     IF (IOVFLO .NE. 0) GO TO 90
49     CALL PRNT RS
50     PRINT 108,NR,S
51     108 FORMAT('0',3X,'S',I3,' = ',F9.3,1X,'USING FULL-WAVE VALUES')
52
53     C
54     IF(IRPLOT .EQ. 1) CALL RPLOTS
55
56     C

```

```

52      20 PRINT 200
53      200 FORMAT('1','SETTLE-DOWN---')
54      NR = NR + 1
55      C
56      CALL SOL EQS (0)
57      C
58      RATIO = ADIST/D OPT
59      PRINT 202,NR,ALAMDA,SNEXT,ADIST,RATIO,NR
60      202 FORMAT(' ',3X,'SOLVE EQS WITH LAMBDA',I3,' = ',F12.5,1X,
61      $ 'GIVING S PRIME = ',F9.3,2X, 'D PRIME = ',F7.3,2X,'RATIO = ',
62      $ F8.2,1X,'AND SETTLE PROFILE',I3)
63      CALL PRINT P
64      C
65      IF (RATIO .LT. THRESH(1)) DELTAL = SQRT(2.0)*DELTAL
66      IF (RATIO .GT. THRESH(2)) DELTAL = DELTAL/2.0
67      C
68      IF (RATIO .LT. THRESH(4) .OR. IFIRST .NE. 0) GO TO 40
69      30 CONTINUE
70      DO 32 J=1,NR A
71      32 ALPHA(J) = ALPHA0(J)
72      ALAMDA = AL0
73      PRINT 300,NR,ALAMDA,NR,DELTAL
74      300 FORMAT ('1','BACKSTEP---',
75      $ 15X,'SAVED LAMBDA',I3,' = ',F12.5,' DELTA LAMBDA',I3,' = ',
76      $ F12.5,' (NOT USED)')
77      NR JHF = (NR A-1)/2
78      DO 39 L=1,NR F
79      DO 39 JHF=1,NR JHF
80      39 NR STPS(JHF,L) = NS0(JHF,L)
81      CALL LKCLKS(SKEY)
82      IF(FNSEC .LT. SKEY) GO TO 70
83      GO TO 15
84      C
85      40 IF (RATIO .LT. THRESH(3)) GO TO 50
86      NR = NR +1
87      PRINT 400,NR,ALAMDA,NR,DELTAL
88      400 FORMAT('1','SAME LAMBDA---',
89      $ 15X,'(SAME) LAMBDA',I3,' = ',F12.5,9X,
90      $ 'DELTA LAMBDA',I3,' = ',F12.5,' (NOT USED)'/,19X,
91      $ 'PREVIOUS SOLUTION FROM SETTLE DOWN IS USED')
92      CALL LKCLKS(SKEY)
93      IF(FNSEC .LT. SKEY) GO TO 70
94      GO TO 60
95      C
96      50 NR = NR + 1
97      PRINT 500,NR,DELTAL
98      500 FORMAT('1','NEW LAMBDA---',15X,'DELTA LAMBDA',I3,' = ',F12.5)
99      IFIRST=0
100     C
101     AL0 = ALAMDA
102     ALAMDA = ALAMDA+DELTAL
103     C

```

```

104      51 CALL SOL EQS (1)
105      PRINT 504,NR,ALAMDA,ADIST,SNEXT,NR
106      504 FORMAT(' ',3X,'SOLVE EQS WITH LAMBDA',I3,' = ',F12.5,1X,
107      $ 'GIVING D = ',F7.3,2X,'LINEAR S = ',F9.3,1X,'AND PROFILE',I3)
108      C
109      CALL LKCLKS(SKEY)
110      IF(FNSEC .LT. SKEY) GO TO 70
111      C
112      DO 52 J=1,NR A
113      52 ALPHA(J) = ALPHA(J)
114      NR JHF = (NR A-1)/2
115      DO 58 L=1,NR F
116      DO 58 JHF=1,NR JHF
117      58 NSU(JHF,L) = NR STPS(JHF,L)
118      C
119      60 DO 61 J=1,NR A
120      61 ALPHA(J) = ALPHA(J)+DA(J)
121      C
122      CALL PRINT N
123      PRINT 601,NR
124      601 FORMAT(' ',3X,'FULL-WAVE SOLUTIONS',I3)
125      CALL R DRDA
126      IF(IOVFLO .NE. 0) GO TO 90
127      CALL PRNT RS
128      PRINT 602,NR,S
129      602 FORMAT('0',3X,'S',I3,' = ',F9.3,1X,'USING FULL-WAVE VALUES')
130      C
131      IF(IRPLOT .EQ. 1) CALL RPLOTS
132      C
133      GO TO 20
134      C
135      70 ALAMDA = ALC
136      CALL CARDS
137      STOP
138      C
139      90 PRINT 990
140      990 FORMAT ('0','IOVFLO FLAG SET')
141      STOP
142      C
143      END

```

@PRT,S INVERT.SETHTS

MORFITT*INVERT(1).SETHTS

```
1      SUBROUTINE SET HTS
2  C
3      COMMON/DLTAH C/DELTAH
4      COMMON/HTS COM/HTS(51)
5      COMMON/HS COM/HTSTOP(2)
6      COMMON/STOPS C/STOPS(2)
7      COMMON/NRF COM/NR F
8      COMMON/NS COM/NR STPS(25,20)
9      COMMON/ALPHA C/ALPHA(51)
10     COMMON/AL COM/ALAMDA
11     COMMON/NRA COM/NR A
12     COMMON/ALIMS C/AMAX,AMIN
13  C
14     DATA EPS/0.001/
15  C
16  C
17     AMAX = ALOG(STOPS(1))
18     AMIN = ALOG(STOPS(2))
19  C
20     FLOTE = (HTSTOP(1)-HTSTOP(2))/(2.0*DELTAH)
21     NR JHF = FLOTE+0.01
22     SLOP = ABS(FLOTE-NR JHF)
23     IF (SLOP .GT. 0.02) GO TO 90
24  C
25     IF (ALAMDA .EQ. 0.0) ALPHA(1) = (AMAX+AMIN)/2.0
26     HTS(1) = HTSTOP(1)
27     J = 1
28     21 J = J+1
29     HTS(J) = HTS(J-1)-DELTAH
30     IF (ALAMDA .EQ. 0.0) ALPHA(J) = ALPHA(1)
31     IF(HTS(J)-0.01 .GT. HTSTOP(2) ) GO TO 21
32     NR A = J
33  C
34     IF (NR A-1 .NE. 2*NR JHF) GO TO 90
35     IF (ALAMDA .NE. 0.0) RETURN
36  C
37     DO 21 L=1,NR F
38     DO 21 JHF=1,NR JHF
39     21 NR STPS(JHF,L) = 2
40     RETURN
41  C
42     90 PRINT 900
43     900 FORMAT ('0', '(HTSTOP(1)-HTSTOP(2))/DELTAH MUST BE A MULTIPLE OF 2.
44     $0')
45     STOP
46  C
47     END
```

BPRT,S INVERT.CARDS

MORFITT*INVERT(1).CARDS

```

1      SUBROUTINE CARDS
2      C
3      COMMON/ALPHA C/ALPHA(51)
4      COMMON/NR COM/NR
5      COMMON/NS COM/NR STPS(25,20)
6      COMMON/ROTAT C/ROTATE
7      COMMON/AL COM/ALAMDA
8      COMMON/DL COM/DELTAL
9      COMMON/NRA COM/NR A
10     COMMON/NRF COM/NR F
11     C
12     C
13     PRINT 100
14     100 FORMAT ('1', 'OUTPUT CARDS--')
15     C
16     PRINT 201, (ALPHA(J), J=1, NR A)
17     201 FORMAT(' ', ' ALPHA = ', 10(F6.3, ', '), /, (' ', 10X, 10(F6.3, ', ')))
18     PUNCH 202, (ALPHA(J), J=1, NR A)
19     202 FORMAT(' ', ' ALPHA = ', 10(F6.3, ', '), /, (' ', 10X, 10(F6.3, ', ')))
20     C
21     NR JHF = (NR A-1)/2
22     DO 31 L=1, NR F
23     PRINT 301, L, (NR STPS(JHF, L), JHF =1, NR JHF)
24     301 FORMAT (' ', ' NRSTPS(1, ', I2, ') = ', 12(I4, ', '), /, (' ', 16X, 12(I4, ', '
25     $))
26     31 PUNCH 302, L, (NR STPS(JHF, L), JHF =1, NR JHF)
27     302 FORMAT (' ', ' NRSTPS(1, ', I2, ') = ', 12(I4, ', '), /, (' ', 16X, 12(I4, ', '
28     $))
29     C
30     PRINT 401, ROTATE
31     401 FORMAT (' ', ' ROTATE = ', F6.1, ', ')
32     PUNCH 402, ROTATE
33     402 FORMAT (' ROTATE = ', F6.1, ', ')
34     C
35     PRINT 501, ALAMDA
36     501 FORMAT (' ', ' ALAMDA = ', 1PE9.2, ', ')
37     PUNCH 502, ALAMDA
38     502 FORMAT (' ALAMDA = ', 1PE9.2, ', ')
39     C
40     PRINT 601, DELTAL
41     601 FORMAT (' ', ' DELTAL = ', F12.5, ', ')
42     PUNCH 602, DELTAL
43     602 FORMAT (' DELTAL = ', F12.5, ', ')
44     C
45     PRINT 701, NR
46     701 FORMAT(' ', ' NR = ', I3, ', ')
47     PUNCH 702, NR
48     702 FORMAT(' NR = ', I3, ', ')
49     C
50     C      ROUTINE ADDED TO CHECK FOR OVERFLOW
51     610 CALL OVCHK('CARDS ', 610)

```

52 RETURN
53 C
54 END

SPRT,S INVERT.SOLEQS

MORFITT*INVERT(1).SOLEQS

```
1      SUBROUTINE SOL EQS (LFLAG)
2      C
3      COMMON/DLTAH C/DELTAH
4      COMMON/IPA COM/IPRNTA
5      COMMON/ALIMS C/AMAX,AMIN
6      COMMON/ALPHA C/ALPHA(51)
7      COMMON/HTS COM/HTS(51)
8      COMMON/NRA COM/NR A
9      COMMON/NRF COM/NR F
10     COMMON/AL COM/ALAMDA
11     COMMON/RNORM C/R NORM(40)
12     COMMON/DRDA C/DRDA(40,51)
13     COMMON/S COM/S
14     COMMON/DA COM/DA(51)
15     COMMON/SN COM/SNEXT
16     COMMON/ADIST C/ADIST
17     COMMON/AU COM/ALP UNC(51)
18     COMMON/RSN COM/RESLTN(51)
19     COMMON/TEMP C/B(1326),A(2040)
20     DIMENSION PEN(2),CAP A(51)
21     REAL*8 B,A,V(51),AUX(51),DSUM
22     DATA EPS/1.0E-4/
23     DATA ALN TEN/2.302585/
24     C
25     C
26     NR M = 2*NR F
27     IF (ALAMDA .EQ. 0.0 .AND. LFLAG .NE. 0) GO TO 12
28     S = 0.0
29     DO 11 I=1,NR M
30     11 S = S+R NORM(I)**2
31     12 CONTINUE
32     C
33     FACTR1 = 1.0
34     IF (ALAMDA .NE. 0.0) FACTR1 = 1.0E5/DELTAH**3/ALAMDA**2
35     PENLTY = FACTR1*1.0E2
36     PEN(1) = 0.0
37     IF (ALPHA(1) .LE. AMAX) PEN(1) = PENLTY
38     PEN(2) = 0.0
39     IF (ALPHA(NR A) .GE. AMIN) PEN(2) = PENLTY
40     ICOUNT = 0
41     C
42     20 JJ J = 1
43     DO 23 J=1,NR A
44     DO 23 JJ=1,J
45     DSUM = 0.0
46     IF (ALAMDA .EQ. 0.0) GO TO 22
47     DO 21 I=1,NR M
48     21 DSUM = DSUM+DRDA(I,JJ)*DRDA(I,J)
49     22 B(JJ J) = DSUM
50     23 JJ J = JJ J+1
51     C
```

```

52      DO 25 J=1,NR A
53      USUM = 0.0
54      IF (ALAMDA .EQ. 0.0) GO TO 25
55      DO 24 I=1,NR M
56      24 DSUM = DSUM+DRDA(I,J)*R NORM(I)
57      25 V(J) = -DSUM
58
59      C
60      B(1) = B(1)+2.0*FACTR1
61      B(2) = B(2)-3.0*FACTR1
62      B(3) = B(3)+6.0*FACTR1
63      J J = 6
64      NRAM2 = NR A-2
65      DO 27 J=3,NRAM2
66      B(J J-2) = B(J J-2)+FACTR1
67      B(J J-1) = B(J J-1)-4.0*FACTR1
68      E(J J) = B(J J)+6.0*FACTR1
69      27 J J = J J+J+1
70      B(J J-2) = B(J J-2)+FACTR1
71      B(J J-1) = B(J J-1)-4.0*FACTR1
72      B(J J) = B(J J)+5.0*FACTR1
73      J J = J J+NR A
74      B(J J-2) = B(J J-2)+FACTR1
75      B(J J-1) = B(J J-1)-2.0*FACTR1
76      B(J J) = B(J J)+FACTR1
77
78      C
79      V(1) = V(1)-FACTR1*(2.0*ALPHA(1)-3.0*ALPHA(2)+ALPHA(3))
80      V(2) = V(2)-FACTR1*(-3.0*ALPHA(1)+6.0*ALPHA(2)-4.0*ALPHA(3)
81      $      +ALPHA(4))
82      NRAM2 = NR A-2
83      DO 28 J=3,NRAM2
84      28 V(J) = V(J)-FACTR1*(ALPHA(J-2)-4.0*ALPHA(J-1)+6.0*ALPHA(J)
85      $      -4.0*ALPHA(J+1)+ALPHA(J+2))
86      V(NR A-1) = V(NR A-1)-FACTR1*(ALPHA(NR A-3)-4.0*ALPHA(NR A-2)
87      $      +5.0*ALPHA(NR A-1)-2.0*ALPHA(NR A))
88      V(NR A) = V(NR A)-FACTR1*(ALPHA(NR A-2)-2.0*ALPHA(NR A-1)
89      $      +ALPHA(NR A))
90
91      C
92      B(1) = B(1)+PEN(1)
93      J J = (NR A*(NR A+1))/2
94      B(J J) = B(J J)+PEN(2)
95
96      C
97      V(1) = V(1)-PEN(1)*(ALPHA(1)-AMAX)
98      V(NR A) = V(NR A)-PEN(2)*(ALPHA(NR A)-AMIN)
99
100     C
101     CALL DGELS (V,B,NR A,1,EPS,IER,AUX)
102     IF (IER .LT. 0) PRINT 903,IER
103     903 FORMAT(' ', 'FOR DA SOLUTION, IER = ', I3)
104
105     C
106     DO 31 J=1,NR A
107     31 DA(J) = V(J)
108
109     C
110     IAGAIN = 0

```

```

104         IF (PEN(1) .EQ. 0.0 .OR. ALPHA(1)+DA(1) .LE. AMAX) GO TO 32
105         PEN(1) = 0.0
106         IAGAIN = 1
107         GO TO 33
108     32 IF (PEN(1) .NE. 0.0 .OR. ALPHA(1)+DA(1) .GT. AMAX) GO TO 33
109         PEN(1) = PENLTY
110         IAGAIN = 1
111     33 IF (PEN(2) .EQ. 0.0 .OR. ALPHA(NR A)+DA(NR A) .GE. AMIN) GO TO 34
112         PEN(2) = 0.0
113         IAGAIN = 1
114         GO TO 35
115     34 IF (PEN(2) .NE. 0.0 .OR. ALPHA(NR A)+DA(NR A) .LT. AMIN) GO TO 35
116         PEN(2) = PENLTY
117         IAGAIN = 1
118     35 IF (IAGAIN .EQ. 0) GO TO 40
119         ICOUNT = ICOUNT+1
120         IF (ICOUNT .LT. 5) GO TO 20
121         PRINT 305
122     305 FORMAT ('0', 'ICOUNT TOO LARGE')
123         STOP
124
125     C
126     40 SNEXT = 0.0
127         ADIST=0.0
128         IF (ALAMDA .EQ. 0.0 .AND. LFLAG .NE. 0) RETURN
129         DO 46 I=1, NR M
130             DSUM = 0.0
131             DO 45 J=1, NR A
132                 45 DSUM = DSUM+DRDA(I,J)*DA(J)
133                 RNEXT = R NORM(I)+DSUM
134                 46 SNEXT = SNEXT+RNEXT**2
135
136     C
137         SUM SQ = 0.0
138         DO 47 J=1, NR A
139             IF (ALPHA(J) .GT. AMIN .AND. ALPHA(J) .LT. AMAX)
140                 $ SUM SQ = SUM SQ+DA(J)**2
141     47 CONTINUE
142         ADIST = SQRT(SUM SQ*DELTAH)/ALN TEN
143
144     C
145         IF (LFLAG .NE. 0) RETURN
146         IF (ALAMDA .EQ. 0.0) GO TO 70
147         JJ J = 1
148         DO 52 J=1, NR A
149             DO 52 JJ=1, J
150                 DSUM = 0.0
151                 DO 51 I=1, NR M
152                     51 DSUM = DSUM+DRDA(I, JJ)+DRDA(I, J)
153                     B(JJ J) = DSUM
154                 52 JJ J = JJ J+1
155
156     C
157         B(1) = B(1)+1.0*FACTR1
158         B(2) = B(2)-2.0*FACTR1
159         B(3) = B(3)+5.0*FACTR1

```

```

156      J J = 0
157      NRAM2 = NR A-2
158      DO 53 J=3, NRAM2
159          B(J J-2) = B(J J-2)+FACTR1
160          B(J J-1) = B(J J-1)-4.0*FACTR1
161          B(J J) = B(J J)+6.0*FACTR1
162      53 J J = J J+J+1
163          B(J J-2) = B(J J-2)+FACTR1
164          B(J J-1) = B(J J-1)-4.0*FACTR1
165          B(J J) = B(J J)+5.0*FACTR1
166          J J = J J+NR A
167          B(J J-2) = B(J J-2)+FACTR1
168          B(J J-1) = B(J J-1)-2.0*FACTR1
169          B(J J) = B(J J)+FACTR1
170      C
171          J I = 1
172          DO 57 I=1, NR M
173          DO 57 J=1, NR A
174          A(J I) = DRDA(I, J)
175      57 J I = J I+1
176      C
177          CALL DGELS (A, B, NR A, NR M, EPS, IER, AUX)
178          IF (IER .LT. 0) PRINT 905, IER
179      905 FORMAT(' ', 'FOR A SOLUTION, IER = ', I3)
180      C
181          DO 62 J=1, NR A
182          DSUM = 0.0
183          J I = J
184          DO 61 I=1, NR M
185          DSUM = DSUM+A(J I)**2
186      61 J I = J I+NR A
187      62 ALP UNC(J) = DSQRT(DSUM)/ALN TEN
188      C
189          DO 66 J=1, NR A
190          DO 64 JJ=1, NR A
191          DSUM = 0.0
192          J I = J
193          DO 63 I=1, NR M
194          DSUM = DSUM+A(J I)*DRDA(I, JJ)
195      63 J I = J I+NR A
196      64 CAP A(JJ) = DSUM
197      C
198          IF (IPRNTA .NE. 0) PRINT 907, J, HTS(J), (CAP A(JJ), JJ=1, NR A)
199      907 FORMAT(' ', 'CAP A', I4, ' AT', F7.2, ' = ', /, (10X, 10(1PE11.2)))
200      C
201          SUM NUM = 0.0
202          SUM DEN = 0.0
203          DO 65 JJ=1, NR A
204          SUM NUM = SUM NUM+(HTS(J)-HTS(JJ))**2*ABS(CAP A(JJ))
205      65 SUM DEN = SUM DEN+ABS(CAP A(JJ))
206      66 RESLTN(J) = SQRT(SUM NUM/SUM DEN)
207          RETURN

```

```
208      C
209      70 DO 71 J=1,NR A
210          ALP UNC(J) = C.0
211      71 RESLTN(J) = 0.0
212          RETURN
213      C
214          END
```

```
@PRT,S INVERT.PRINTN
```

```

MORFITT*INVERT(1).PRINTN
 1      SUBROUTINE PRINT N
 2      C
 3      COMMON/NORG C/NORG
 4      COMMON/ANS COM/ANSWER(51)
 5      COMMON/PLOT C/IPROFP,IRPLOT
 6      COMMON/NRA COM/NR A
 7      COMMON/HTS COM/HTS(51)
 8      COMMON/ALPHA C/ALPHA(51)
 9      COMMON/AU COM/ALP UNC(51)
10      COMMON/RSN COM/RESLTN(51)
11      COMMON/DA COM/DA(51)
12      COMMON/ALIMS C/AMAX,AMIN
13      INTEGER V
14      DIMENSION LINE(81)
15      C
16      C
17      DATA ISYMO/'*'/,ISYMN/'+'/,IBLANK/' '/,ISYMS/'S'/
18      DATA ALN TEN/2.302585/
19      DATA V/017400000000/
20      C
21      FMESH=4.0*ALN TEN/80.0
22      C
23      IF(IPROFP .EQ. 1 .AND. NRA .NE. NORG) GO TO 988
24      PRINT 100,V,V,V,V,V
25      100 FORMAT ('0',5X,'HT',9X,'N',19X, A1,19('-'), A1,19('-'),
26      * A1,19('-'), A1,19('-'), A1)
27      DO 11 J=1,NR A
28      EN = EXP(ALPHA(J))
29      DO 50 K = 1,81
30      50 LINE(K) = IBLANK
31      IF(IPROFP .EQ. 0) GO TO 60
32      JPOST=ANSWER(J)/FMESH+1.5
33      IF(JPOST .GE. 1 .AND. JPOST .LE. 81)
34      *LINE(JPOST)=ISYMO
35      60 CONTINUE
36      JPOST=ALPHA(J)/FMESH+1.5
37      IF(JPOST .GE. 1 .AND. JPOST .LE. 81)
38      *LINE(JPOST)=ISYMN
39      IF(J .NE. 1) GO TO 67
40      JPOST=AMAX/FMESH+1.5
41      LINE(JPOST)=ISYMS
42      67 IF(J .NE. NR A) GO TO 11
43      JPOST=AMIN/FMESH+1.5
44      LINE(JPOST)=ISYMS
45      11 PRINT 105,HTS(J),EN,V,LINE,V
46      105 FORMAT (' ',F7.2,1X,1PE9.2,17X, A1,81A1, A1)
47      GO TO 500
48      C
49      C
50      ENTRY PRINT P
51      PRINT 200,V,V,V,V,V

```

```

52 200 FORMAT ('0',5X,'HT',9X,'N',3X,'UNC',4X,'RES',5X, A1,19('-'),
53 * A1,19('-'), A1,19('-'), A1,19('-'), A1)
54 DO 21 J=1,NR A
55 EN = EXP(ALPHA(J)+DA(J))
56 DO 70 K=1,81
57 70 LINE(K)=IRLANK
58 IF(IPROFP .EQ. 0) GO TO 61
59 JPOST = ANSWER(J)/FMESH+1.5
60 IF(JPOST .GE. 1 .AND. JPOST .LE. 81)
61 *LINE(JPOST) = ISYMO
62 61 CONTINUE
63 JPOST=(ALPHA(J)+DA(J))/FMESH+1.5
64 IF(JPOST .GE. 1 .AND. JPOST .LE. 81)
65 *LINE(JPOST)=ISYMN
66 IF(J .NE. 1) GO TO 68
67 JPOST=AMAX/FMESH+1.5
68 LINE(JPOST)=ISYMS
69 68 IF(J .NE. NR A) GO TO 21
70 JPOST=AMIN/FMESH+1.5
71 LINE(JPOST)=ISYMS
72 21 PRINT 205,HTS(J),EN,ALP UNC(J),RESLTN(J),V,LINE,V
73 205 FORMAT (' ',F7.2,1X,1PE9.2,0PF6.2,1X,0PF6.2,4X, A1,81A1, A1)
74 C
75 500 PRINT 501,V,V,V,V,V
76 501 FORMAT(' ',35X, A1,19('-'), A1,19('-'), A1,19('-'), A1,
77 * 19('-'), A1 )
78 PRINT 502
79 502 FORMAT(' ',35X,'0',19X,'1',19X,'2',19X,'3',19X,'4')
80 C
81 GO TO 800
82 C
83 988 PRINT 987,NR A,NORG
84 987 FORMAT('0',//,'PROGRAM STOPS***', ' NRA = ',I5,' AND NORG = ',I5)
85 STOP
86 800 CONTINUE
87 RETURN
88 C
89 END

```

@PRT,S INVERT.RPLOTS

MORFITT*INVERT(1).RPLOTS

```

1      SUBROUTINE RPLOTS
2      C
3      COMMON/NRF COM/NR F
4      COMMON/RDATA C/R DATA(2,20)
5      COMMON/DSPLA C/R DSPLA(6,20)
6      INTEGER V
7      DIMENSION ISYMD(20),ISYMC(20),XD(20),YD(20),XC(20),YC(20),
8      * LINE(101)
9      C
10     DATA ISYMD/'A','B','C','D','E','F','G','H','I','J',
11     S           'K','L','M','N','O','P','Q','R','S','T',
12     DATA ISYMC/'1','2','3','4','5','6','7','8','9','U',
13     S           '1','2','3','4','5','6','7','8','9','D',
14     DATA IBLANK/' '/
15     DATA V/017400000000C0/
16     C
17     XMAX=-1.0E9
18     XMIN=+1.0E9
19     YMAX=-1.0E9
20     YMIN=+1.0E9
21     C
22     DO 10 L=1,NR F
23     XD(L)=R DATA(1,L)
24     YD(L)=R DATA(2,L)
25     IF(XD(L) .LT. XMIN) XMIN=XD(L)
26     IF(XD(L) .GT. XMAX) XMAX=XD(L)
27     IF(YD(L) .LT. YMIN) YMIN=YD(L)
28     IF(YD(L) .GT. YMAX) YMAX=YD(L)
29     10 CONTINUE
30     C
31     DO 20 L=1,NR F
32     XC(L)=R DSPLA(1,L)
33     YC(L)=R DSPLA(2,L)
34     IF(XC(L) .LT. XMIN) XMIN=XC(L)
35     IF(XC(L) .GT. XMAX) XMAX=XC(L)
36     IF(YC(L) .LT. YMIN) YMIN=YC(L)
37     IF(YC(L) .GT. YMAX) YMAX=YC(L)
38     20 CONTINUE
39     C
40     IF((XMAX-XMIN)/10.0 .GT. (YMAX-YMIN)/6.5) GO TO 100
41     YMESH=(YMAX-YMIN)/52.0
42     XMESH=YMESH*0.8
43     XMAX=XMIN+(100.0*XMESH)
44     GO TO 200
45     100 XMESH=(XMAX-XMIN)/100.0
46     YMESH=XMESH/0.8
47     YMAX=YMIN+(52.0*YMESH)
48     200 CONTINUE
49     PRINT 30
50     30 FORMAT('1', ' PLOT OF R-VALUES',5X, ' DATA VALUES = ALPHAMERIC',
51     * 5X, ' COMPUTED VALUES = NUMERIC')

```

```

52      YTOP=YMAX+YMESH
53      PRINT 31,YTOP
54      31 FORMAT(' ',F10.4,101('-'))
55      DO 300 I=1,53
56      DO 350 J=1,101
57      350 LINE(J) = IBLANK
58      DO 400 L=1,NR F
59      ITEST = (YD(L)-YMIN)/YMESH+0.5
60      IF((ITEST+1) .NE. 53) GO TO 400
61      JPOST = (XD(L)-XMIN)/XMESH+1.5
62      LINE(JPOST) = ISYMD(L)
63      400 CONTINUE
64      C
65      DO 401 L=1,NRF
66      ITEST=(YC(L)-YMIN)/YMESH+0.5
67      IF((ITEST+1) .NE. 53) GO TO 401
68      JPOST=(XC(L)-XMIN)/XMESH+1.5
69      LINE(JPOST) = ISYMC(L)
70      401 CONTINUE
71      C
72      300 PRINT 555,V,LINE,V
73      555 FORMAT(' ',9X, A1,101A1, A1)
74      YBOT=YMIN-YMESH
75      PRINT 32,YBOT
76      32 FORMAT(' ',F10.4,101('-'))
77      PRINT 33,XMIN,XMAX
78      33 FORMAT(' ',5X,F10.4,90X,F10.4)
79      C
80      RETURN
81      END

```

@PRT,S INVERT.DINPUT

MORFITT*INVERT(1).DINPUT

```
1      SUBROUTINE D INPUT
2      C
3      COMMON/F COM/F(20)
4      COMMON/RDATA C/R DATA(20)
5      COMMON/SIGMA C/SIGMA(20)
6      COMMON/NRF COM/NR F
7      COMPLEX R DATA,R
8      DIMENSION ID(20)
9      C
10     C
11     READ 101,ID
12     101 FORMAT (20A4)
13     PRINT 102,ID
14     102 FORMAT (' ',20A4)
15     C
16     L = 1
17     11 READ 103,FREQ,R,SIG
18     103 FORMAT (F7.2,3X,2F10.3,F10.3)
19     PRINT 104,FREQ,R,SIG
20     104 FORMAT (' ',F7.2,3X,2F10.3,F10.3)
21     IF (FREQ .LT. 0.0) GO TO 15
22     F(L) = FREQ
23     R DATA(L) = R
24     SIG*A(L) = SIG
25     L = L+1
26     GO TO 11
27     15 NR F = L-1
28     RETURN
29     C
30     END
```

6PRT,S INVERT.INTERP

```

MORFITT*INVERT(1).INTERP
  1      SUBROUTINE INTERP
  2      C
  3      COMMON/NRF COM/NR F
  4      COMMON/DELF C/DELF
  5      COMMON/F COM/F(20)
  6      COMMON/IF COM/INDEXF(20)
  7      COMMON/DIDR C/DIDR(205,20)
  8      COMMON/TEMP C/A(1000),V(4000)
  9      REAL*8 A,V
 10      DATA MAXNR/205/
 11      C
 12      C
 13      INDEXF(1)=3
 14      DO 11 L=2,NR F
 15      11 INDEXF(L) = 3.5+(F(L)-F(1))/DELF
 16      NR EQ = INDEXF(NR F)+2
 17      IF(NR EQ .GT. MAXNR) GO TO 90
 18      C
 19      C
 20      A(1) = 1.0
 21      A(2) = -2.0
 22      A(3) = 1.0
 23      A(4) = -2.0
 24      A(5) = 5.0
 25      A(6) = -4.0
 26      A(7) = 1.0
 27      C
 28      A(8) = 0.0
 29      A(9) = 0.0
 30      A(10) = 1.0
 31      A(11) = 0.0
 32      A(12) = 0.0
 33      C
 34      K = 15
 35      DO 23 L=2,NR F
 36      21 IF(K/5+1 .EQ. INDEXF(L)) GO TO 22
 37      A(K-2) = 1.0
 38      A(K-1) = -4.0
 39      A(K) = 6.0
 40      A(K+1) = -4.0
 41      A(K+2) = 1.0
 42      K = K+5
 43      GO TO 21
 44      C
 45      22 A(K-2) = 0.0
 46      A(K-1) = 0.0
 47      A(K) = 1.0
 48      A(K+1) = 0.0
 49      A(K+2) = 0.0
 50      23 K = K+5
 51      C

```

```

52          A(K-2) = 1.0
53          A(K-1) = -4.0
54          A(K) = 5.0
55          A(K+1) = -2.0
56          K = K+4
57          A(K-2) = 1.0
58          A(K-1) = -2.0
59          A(K) = 1.0
60      C
61          KL=1
62          DO 31 L=1,NR F
63          DO 31 K=1,NR EG
64          V(KL)=0.0
65      31  KL=KL+1
66      C
67          DO 33 L=1,NR F
68          KL = (L-1)*NR EQ+INDEXF(L)
69      33  V(KL)=1.0
70      C
71          CALL DGELB (V,A,NR EQ,NR F,2,2,1.0E-4,IER)
72          PRINT 401,IER
73      401  FORMAT ('1', 'IN DGELB, IER = ',I3,/)
74      C
75          KLIMIT = NR EQ-1
76          DO 52 K=2,KLIMIT
77          KL=K
78          DO 51 L=1,NR F
79          DIDR(K,L)=V(KL+1)-V(KL-1)
80      51  KL=KL+NR EG
81      52  PRINT 501,(DIDR(K,L),L=1,NR F)
82      501  FORMAT (' ',10F10.5)
83          RETURN
84      C
85          90 PRINT 900
86      900  FORMAT('0', 'DELF TOO SMALL FOR FREQUENCY RANGE USED')
87          STOP
88      C
89          END

```

6PRT,S INVERT.LOGS

MORFITT*INVERT(1).LOGS

```

1      SUBROUTINE LOGS
2      C
3      COMMON/DPHAS C/DPHASE
4      COMMON/F COM/F(20)
5      COMMON/RDATA C/R DATA(20)
6      COMMON/SIGMA C/SIGMA(20)
7      COMMON/NRF COM/NR F
8      COMMON/DIDR C/DIDR(205,20)
9      COMMON/IF COM/INDEXF(20)
10     COMMON/RD COM/RD(20)
11     COMMON/OVSIG C/OVR SIG(20,20)
12     COMMON/TEMP C/A(20,20),X(20,20),V(20,20)
13     COMPLEX R DATA,RD,OVR SIG,
14     S      R SUM,PARTS,R DSPLA(20)
15     COMPLEX*16 A,X,V
16     DIMENSION PART(2),OS PART(2,20,20)
17     EQUIVALENCE (PARTS,PART),(OVR SIG,OS PART)
18     DATA RTD/57.29578/
19     DATA PI/3.141593/
20     DATA NOIM/20/
21     C
22     C
23     RD(1) = R DATA(1)
24     R DSPLA(1) = R DATA(1)
25     PREVP = 0.0
26     K=2
27     DO 28 LL=2,NR F
28     21 K=K+1
29     R SUM = 0.0
30     DO 22 L=1,NR F
31     22 R SUM = R SUM+DIDR(K,L)*R DATA(L)
32     PARTS = CLOG(R SUM)
33     PHASE = PART(2)
34     24 IF (PHASE-PREVP .LT. PI) GO TO 25
35     PHASE = PHASE-2.0*PI
36     GO TO 24
37     25 IF (PHASE-PREVP .GT. -PI) GO TO 26
38     PHASE = PHASE+2.0*PI
39     GO TO 25
40     26 PREVP = PHASE IF(K .EQ. 3) DPHASE = PREVP*RTD
41     IF(K .NE. INDEXF(LL)) GO TO 21
42     PART(2) = PHASE
43     RD(LL) = PARTS
44     PART(1) = EXP(PART(1))
45     PART(2) = PART(2)*RTD
46     28 R DSPLA(LL) = PARTS
47     C
48     A(1,1) = 1.0
49     DO 51 L=2,NR F
50     51 A(L,1) = 0.0
51     DO 53 LL=2,NR F

```

```

52      R SUM = 0.0
53      K = INDEXF(LL)
54      DO 52 L=1,NR F
55      52 R SUM = R SUM+DIDR(K,L)*R DATA(L)
56      DO 53 L=1,NR F
57      53 A(L,LL) = DIDR(K,L)/R SUM
58      C
59      DO 55 L=1,NR F
60      DO 55 LL=1,NR F
61      55 V(L,LL) = 0.0
62      DO 56 L=1,NR F
63      56 V(L,L) = 1.0/SIGMA(L)
64      C
65      CALL CLINEQ (A,V,X,NR F,NDIM,0,ERR)
66      PRINT 600,ERR
67      600 FORMAT ('0', 'CLINEQ ERR = ',F7.3, '//)
68      C
69      DO 58 L=2,NR F
70      CALL CLINEQ(A,V(1,L),X(1,L),NR F,NDIM,1,ERR)
71      58 PRINT 600,ERR
72      C
73      DO 61 L=1,NR F
74      DO 61 LL=1,NR F
75      61 OVR SIG(L,LL) = X(LL,L)
76      C
77      DO 63 L=1,NR F
78      PRINT 603,(OS PART(1,L,LL),LL=1,NR F)
79      PRINT 603,(OS PART(2,L,LL),LL=1,NR F)
80      603 FORMAT (' ',10(1PE12.2))
81      63 PRINT 604
82      604 FORMAT ('0')
83      C
84      PRINT 700
85      700 FORMAT ('1', 'DATA VALUES--')
86      PRINT 701
87      701 FORMAT ('0',3X,'FREQ',6X,'RE(R)',3X,'IM(R)',2X,'MAG(DRDF)',1X,'A(D
88      DRDF)',4X,'RE(G)',4X,'IM(G)')
89      DO 71 L=1,NR F
90      71 PRINT 703,F(L),R DATA(L),R DSPLA(L),RD(L)
91      703 FORMAT (' ',F7.2,3X,2F8.3,3X,2F8.3,F14.5,F9.5)
92      RETURN
93      C
94      END

```

PRT,S INVERT.RDRDA

```

MORFITT=INVERT(1),RDRDA
  1      SUBROUTINE R DRDA
  2
  3      C
  4      COMMON/DELF C/DELF
  5      COMMON/S COM/S
  6      COMMON/ROSTAT C/ROTATE
  7      COMMON/DPHAS C/DPHASE
  8      COMMON/ROSTAT C/ROTATE
  9      COMMON/DPHAS C/DPHASE
 10      COMMON/ROSTAT C/ROTATE
 11      COMMON/DPHAS C/DPHASE
 12      COMMON/ROSTAT C/ROTATE
 13      COMMON/DPHAS C/DPHASE
 14      COMMON/ROSTAT C/ROTATE
 15      COMMON/DPHAS C/DPHASE
 16      COMMON/ROSTAT C/ROTATE
 17      COMMON/DPHAS C/DPHASE
 18      COMMON/ROSTAT C/ROTATE
 19      COMMON/DPHAS C/DPHASE
 20      COMMON/ROSTAT C/ROTATE
 21      COMMON/DPHAS C/DPHASE
 22      COMMON/ROSTAT C/ROTATE
 23      COMMON/DPHAS C/DPHASE
 24      COMMON/ROSTAT C/ROTATE
 25      COMMON/DPHAS C/DPHASE
 26      DIMENSION PART(2),U LEVEL(25)
 27      COMPLEX
 28      $      RD,
 29      $      OVR SIG,
 30      $      R11,R21,R12,R22,
 31      $      DR LIST,ER LIST,
 32      $      R NORM,DRDA,
 33      $      R DSPLA,
 34      $      CPLX I,PARTS,
 35      $      RNUM,RDEN,R(20),
 36      $      URNUM,URDEN,UR CPLX,
 37      $      DPNUM,DRDEN,DRIV,
 38      $      R SUM,R NEW(20),DRDA SV(20)
 39      EQUIVALENCE (PARTS,PART)
 40      DATA UFRCTN/0.3/
 41      DATA CPLX I/(0.0,1.0)/
 42      DATA PI/3.141593/
 43      DATA RTD/57.29578/
 44      C
 45      C
 46      NR JHF = (NR A-1)/2
 47      C
 48      DO 12 J=1,NR A
 49      DO 12 LL=1,NR F
 50      12 DRDA(LL,J) = 0.0
 51      C

```

```

52      DO 49 L=1,NR F
53      FREQ = F(L)
54      21 PRINT 201,FREQ
55      201 FORMAT (' ',F7.2)
56      LFREQ = L
57      CALL INTEG
58      IF (IOVFLO .NE. 0) RETURN
59      C
60      RNUM = -R22+CPLX I*R12
61      RDEN = -R22-CPLX I*R12
62      R(L) = RNUM/RDEN
63      R DSPLA(1,L) = R(L)
64      C
65      SUM = 0.0
66      DO 31 J=1,NR JHF
67      URNUM = -ER LIST(4,J)+CPLX I*ER LIST(3,J)
68      URDEN = -ER LIST(4,J)-CPLX I*ER LIST(3,J)
69      UR CPLX = (URNUM-(RNUM/RDEN)*URDEN)/RDEN
70      PARTS = UR CPLX
71      U LEVEL(J) = PART(1)**2+PART(2)**2
72      31 SUM = SUM+U LEVEL(J)
73      USQ = SUM
74      UR(L) = SQRT(SUM)
75      C
76      USQ MAX = (UFRCTN*SIGMA(L))**2
77      IF (USQ .LT. USQ MAX) GO TO 40
78      C
79      U MAX = 1.0E9
80      U NEXT = 0.0
81      34 SUM = 0.0
82      DO 35 J=1,NR JHF
83      U TEMP = U LEVEL(J)
84      IF (U TEMP+1.01 .LT. U MAX .AND. U TEMP .GT. U NEXT)
85      $      U NEXT = U TEMP
86      IF (U TEMP .GT. U MAX) U TEMP = U MAX
87      35 SUM = SUM+U TEMP
88      IF (SUM .LT. USQ MAX) GO TO 36
89      U MAX = U NEXT
90      U NEXT = 0.0
91      GO TO 34
92      C
93      36 DO 37 J=1,NR JHF
94      IF (U LEVEL(J)*0.99 .GT. U MAX) NR STPS(J,L) = 2*NR STPS(J,L)
95      37 CONTINUE
96      C
97      GO TO 21
98      C
99      40 N STEPS(L) = 0
100     DO 41 J=1,NR JHF
101     41 N STEPS(L) = N STEPS(L)+NR STPS(J,L)
102     DO 42 J=1,NR A
103     DRNUM = -DR LIST(4,J)+CPLX I*DR LIST(3,J)

```

```

104          DRDEN = -DR LIST(4,J)-CPLX I*DR LIST(3,J)
105          DRIV = (DRNUM-(RNUM/RDEN)*DRDEN)/RDEN
106          IF (L .EQ. 1) DRDA(1,J) = DRIV
107          DO 42 LL=2,NR F
108          K = INDEXF(LL)
109          42 DRDA(LL,J) = DRDA(LL,J)+DIDR(K,L)*DRIV
110          49 CONTINUE
111          C
112          R DSPLA(2,1) = R(1)
113          R DSPLA(3,1) = R(1)
114          R NEW(1) = R(1)-RD(1)
115          PREVP = (DPHASE-ROTATE)/RTD
116          K=2
117          DO 69 LL=2,NR F
118          61 K = K+1
119          R SUM = 0.0
120          DO 62 L=1,NR F
121          62 R SUM = R SUM+DIDR(K,L)*R(L)
122          PARTS = CLOG(R SUM)
123          PHASE = PART(2)
124          64 IF (PHASE-PREVP .LT. PI) GO TO 65
125          PHASE = PHASE-2.0*PI
126          GO TO 64
127          65 IF (PHASE-PREVP .GT. -PI) GO TO 66
128          PHASE = PHASE+2.0*PI
129          GO TO 65
130          66 PREVP = PHASE
131          IF(K .EQ. 3) ROTATE = (DPHASE-PREVP)*RTD
132          IF(K .NE. INDEXF(LL)) GO TO 61
133          C
134          PART(2) = PHASE
135          R NEW(LL) = PARTS-RD(LL)
136          DO 68 J=1,NR A
137          68 DRDA(LL,J) = DRDA(LL,J)/R SUM
138          R DSPLA(3,LL) = PARTS
139          PART(1) = EXP(PART(1))/(2.0*DELTA)
140          PART(2) = PART(2)*RTD
141          R DSPLA(2,LL) = PARTS
142          69 CONTINUE
143          C
144          DO 72 L=1,NR F
145          R SUM = 0.0
146          DO 71 LL=1,NR F
147          71 R SUM = R SUM+OVR SIG(L,LL)*R NEW(LL)
148          72 R NORM(L) = R SUM
149          C
150          DO 75 J=1,NR A
151          DO 73 LL=1,NR F
152          73 DRDA SV(LL) = DRDA(LL,J)
153          DO 75 L=1,NR F
154          R SUM = 0.0
155          DO 74 LL=1,NR F

```

```
156          74 R SUM = R SUM+OVR SIG(L,LL)+DRDA SV(LL)
157          75 DRDA(L,J) = R SUM
158          C
159          S=0.0
160          DO 81 L =1,NR F
161          81 S = S+CABS(R NORM(L))**2
162          RETURN
163          C
164          END
```

6PRT,S INVERT.PRNTS

MORFITT*INVERT(1).PRNTRS

```
1      SUBROUTINE PRNT RS
2      C
3      COMMON/DSPLA C/R DSPLA(6,20)
4      COMMON/F COM/F(20)
5      COMMON/UR COM/UR(20),N STEPS(20)
6      COMMON/NRF COM/NR F
7      C
8      C
9      PRINT 100
10     100 FORMAT ('U',3X,'FREQ',6X,'RE(R)',3X,'IM(R)',2X,'MAG(DRDF)',1X,'A(D
11     1RDF)',9X,'RE(G)',4X,'IM(G)',12X,'RK ERR',3X,'RK STEPS')
12     DO 11 L=1,NR F
13     11 PRINT 101,F(L),(R DSPLA(K,L),K=1,6),UR(L),N STEPS(L)
14     101 FORMAT (' ',F7.2,3X,2F8.3,3X,2F8.3,F14.5,F9.5,10X,F8.3,7X,I4)
15     RETURN
16     C
17     END
```

@PRT,S INVERT.FWINPT

```

MORFITT=INVERT(1).FWINPT
 1      SUBROUTINE FW INPT
 2      C
 3      COMMON/PLOT C/IPROFP,IRPLOT
 4      COMMON/THETA C/THETA
 5      COMMON/FLD COM/AZIM,CODIP,MAGFLD
 6      COMMON/IPRNT C/IPRINT
 7      REAL MAGFLD
 8      C
 9      NAMEDLIST /FULLWV/THETA,AZIM,CODIP,MAGFLD,IPRINT,
10      * IPROFP,IRPLOT
11      DATA WAVE/'WAVE'/
12      DATA ADAT/' &DA'/,DAT2/'TUM '/,IEND/' &EN'/,END2/'D '/
13      DATA FUL1/'FULL'/,FUL2/' WAV'/,FUL3/'E IN'/,FUL4/' PUT'/
14      C
15      IPROFP=1
16      IRPLOT=1
17      THETA = 22.0
18      AZIM = 90.0
19      CODIP = 12.0
20      MAGFLD = 5.3E-5
21      IPRINT = 0
22      C
23      READ 12, CBD
24      12 FORMAT(1A4)
25      IF( CBD .NE. WAVE) PRINT 14
26      14 FORMAT('D', ' FULLWV NAMEDLIST IS ABSENT.--PROGRAM STOPS--.')
27      IF( CBD .NE. WAVE) GO TO 999
28      PRINT 10,FUL1,FUL2,FUL3,FUL4
29      10 FORMAT(' ',4A4/)
30      IF( CBD .EQ. WAVE) PRINT 13,WAVE
31      13 FORMAT(' ',A4)
32      READ(5,FULLWV)
33      PRINT 16,ADAT,DAT2
34      16 FORMAT(' ',2A4)
35      PRINT 11,THETA,AZIM,CODIP,MAGFLD,IPRINT,IPROFP,IRPLOT
36      11 FORMAT(' ', ' THETA = ',F8.3, ' DEGREES', ' AZIM = ',F8.3,
37      * ' DEGREES', ' CODIP = ',F8.3, ' DEGREES',
38      * ' MAGFLD = ',1PE10.3, ' IPRINT = ',I2, ' IPROFP = ',I2,
39      * ' IRPLOT = ',I2)
40      PRINT 17,IEND,END2
41      17 FORMAT(' ',2A4,/)
42      C
43      CALL INIT T
44      RETURN
45      C
46      999 STOP
47      C
48      END

```

```

MORFITT*INVERT(1).INTEG
  1      SUBROUTINE INTEG
  2      C
  3      COMMON/NS COM/NP STPS(25,20)
  4      COMMON/DLTAH C/DELTAH
  5      COMMON/HTS COM/HTS(51)
  6      COMMON/NRA COM/NR A
  7      COMMON/IPRNT C/IPRINT
  8      COMMON/FREQ NR/LFREQ
  9      COMMON/JAY COM/JAY
 10      COMMON/TEMP C/DR LIST(8,51),ER LIST(8,25),
 11      $      A STOR(32,49),
 12      $      RO(56),HDELRO(56),DELR1(56),DELR2(56),
 13      $      XO(8),HDELXO(8),DELX1(8),DELX2(8),
 14      $      R SAVE(56),
 15      $      ERROR(8),
 16      $      A TEMP(32),A PROD(32),
 17      $      TEMPV(8)
 18      COMMON/INTEGR/R(56)
 19      COMMON/DRDH C/DRDH(56)
 20      COMMON/X COM/X(8)
 21      COMMON/DXDH C/DXDH(8)
 22      COMMON/EN COLL/HT
 23      COMMON/WN COM/WAVE NR
 24      COMMON/OVRFLO/IOVFLO
 25      COMMON/IEW COM/IEW
 26      DATA DHMIN/0.01/
 27      DATA RTOL/0.03/
 28      C
 29      C
 30      IOVFLO = 0
 31      CALL INIT S
 32      JAY = 1
 33      HT = HTS(1)
 34      CALL COEFFS
 35      CALL INITLI
 36      CALL SET P DR
 37      C
 38      NR EQ = 56-20*IEW
 39      N6N8 = 8-2*IEW
 40      N25N41 = 41-16*IEW
 41      N18N32 = 32-14*IEW
 42      C
 43      IF (IPRINT .NE. 0) PRINT 900
 44      IF (IPRINT .NE. 0) PRINT 902,HT,(R(I),I=1,N6N8)
 45      C
 46      NRAM1 = NR A-1
 47      NR JHF = (NR A-1)/2
 48      DO 69 J=2,NRAM1,2
 49      JM1 = J-1
 50      JHF = J/2
 51      JAY = JM1

```

```

52      CALL XFER (R,R SAVE,NR EQ)
53      CALL XFER (R,X,N6N8)
54      HT = HTS(JM1)
55      31 NS = NR STPS(JMF,LFREQ)
56      DELH = -2.0*DELTAH/FLOAT(NS)
57      DH = DELH*WAVE NR/2.0
58      HDH = 0.5*DH
59      TDH = 2.0*DH
60      C
61      NSOVR2 = NS/2
62      NSOVR4 = NS/4
63      DO 49 K=1,NSOVR2
64      CALL R DERIV
65      CALL X DERIV
66      IF (IOVFLO .NE. 0) GO TO 60
67      C
68      DO 32 I=1,NR EQ
69      R0(I) = R(I)
70      HDELRO(I) = DRDH(I)*HDH
71      32 R(I) = R0(I)+HDELRO(I)
72      C
73      DO 33 I=1,N6N8
74      X0(I) = X(I)
75      HDELX0(I) = DXDH(I)*DH
76      33 X(I) = X0(I)+HDELX0(I)
77      C
78      HT = HT+0.5*DELH
79      CALL COEFFS
80      CALL R DERIV
81      IF (IOVFLO .NE. 0) GO TO 60
82      C
83      DO 34 I=1,NR EQ
84      DELR1(I) = DRDH(I)*DH
85      34 R(I) = R0(I)+0.5*DELR1(I)
86      C
87      CALL R DERIV
88      IF (IOVFLO .NE. 0) GO TO 60
89      C
90      DO 35 I=1,NR EQ
91      DELR2(I) = DRDH(I)*DH
92      35 R(I) = R0(I)+DELR2(I)
93      C
94      HT = HT+0.5*DELH
95      CALL COEFFS
96      CALL R DERIV
97      CALL X DERIV
98      IF (IOVFLO .NE. 0) GO TO 60
99      C
100     DO 36 I=1,NR EQ
101     HDELR3 = DRDH(I)*HDH
102     DELR4 = (HDELRO(I)+DELR1(I)+DELR2(I)+HDELR3)/3.0
103     36 R(I) = R0(I)+DELR4

```

```

104 C
105     DO 37 I=1,N6N8
106     DELX1(I) = DXDH(I)*TDM
107     37 X(I) = X0(I)+0.5*DELX1(I)
108 C
109     IF (NS .NE. 2) GO TO 40
110     IF (JM1 .GT. 1) CALL XFER (R(N6N8+1),A STOR(1,JM1-1),N13N32)
111     CALL XFER (R(N25N41),DR LIST(1,JM1),N6N8)
112     CALL SET P DR
113     JAY = J
114 C
115     40 CALL R DERIV
116     CALL X DERIV
117     IF (IOVFLO .NE. 0) GO TO 60
118 C
119     DO 42 I=1,NR EQ
120     RO(I) = R(I)
121     HDELRO(I) = DRDH(I)*HDM
122     42 R(I) = RO(I)+HDELRO(I)
123 C
124     DO 43 I=1,N6N8
125     DELX2(I) = DXDH(I)*TDM
126     43 X(I) = X0(I)+DELX2(I)
127 C
128     HT = HT+0.5*DELH
129     CALL COEFFS
130     CALL R DERIV
131     IF (IOVFLO .NE. 0) GO TO 60
132 C
133     DO 44 I=1,NR EQ
134     DELR1(I) = DRDH(I)*DH
135     44 R(I) = RO(I)+0.5*DELR1(I)
136 C
137     CALL R DERIV
138     IF (IOVFLO .NE. 0) GO TO 60
139 C
140     DO 45 I=1,NR EQ
141     DELR2(I) = DRDH(I)*DH
142     45 R(I) = RO(I)+DELR2(I)
143 C
144     HT = HT+0.5*DELH
145     CALL COEFFS
146     CALL R DERIV
147     CALL X DERIV
148     IF (IOVFLO .NE. 0) GO TO 60
149 C
150     DO 46 I=1,NR EQ
151     HDELR3 = DRDH(I)*HDM
152     DELR4 = (HDELRO(I)+DELR1(I)+DELR2(I)+HDELR3)/3.0
153     46 R(I) = RO(I)+DELR4
154 C
155     DO 47 I=1,N6N8

```

```

156      HDELX3 = DXDH(I)*DH
157      DELX4 = (HDELX0(I)+DELX1(I)+DELX2(I)+HDELX3)/3.0
158      47 X(I) = X0(I)+DELX4
159      C
160      IF (K .NE. NSOVR4) GO TO 49
161      IF (JM1 .GT. 1) CALL XFER (R(N6N8+1),A STOR(1,JM1-1),N18N32)
162      CALL XFER (R(N25N41),DR LIST(1,JM1),N6N8)
163      CALL SET P DR
164      JAY = J
165      49 CONTINUE
166      C
167      IF (IPRINT .NE. 0) PRINT 900
168      IF (IPRINT .NE. 0) PRINT 901,(X(I),I=1,N6N8)
169      IF (IPRINT .NE. 0) PRINT 902,HT,(R(I),I=1,N6N8)
170      C
171      DO 51 I=1,N6N8
172      ERROR(I) = R(I)-X(I)
173      IF (ABS(ERROR(I)) .GT. RTOL) GO TO 61
174      51 CONTINUE
175      GO TO 65
176      C
177      60 IF (IPRINT .NE. 0) PRINT 900
178      IF (IPRINT .NE. 0) PRINT 902,HT,(R(I),I=1,N6N8)
179      61 IF (IPRINT .NE. 0) PRINT 600
180      600 FORMAT (" ", "OVRFLO")
181      IF (ABS(DELH/2.0) .LT. DHMIN) RETURN
182      IOVFLO = 0
183      CALL XFER (R SAVE,R,NR EQ)
184      CALL XFER (R,X,N6N8)
185      NR STPS(JHF,LFREQ) = 2*NR STPS(JHF,LFREQ)
186      HT = HTS(JM1)
187      JAY = JM1
188      CALL COEFFS
189      GO TO 31
190      C
191      65 CALL XFER (ERROR,ER LIST(1,JHF),N6N8)
192      IF (J .GT. 1) CALL XFER (R(N6N8+1),A STOR(1,J-1),N18N32)
193      CALL XFER (R(N25N41),DR LIST(1,J),N6N8)
194      CALL SET P DR
195      69 CONTINUE
196      C
197      J = NRAM1
198      CALL XFER (R(N25N41),DR LIST(1,J+1),N6N8)
199      C
200      J = J-1
201      CALL XFER(A STOR(1,J),A PROD,N18N32)
202      JHF = NR JHF-1
203      C
204      72 CALL MULTVC(A PROD,DR LIST(1,J),TEMPV)
205      CALL XFER(TEMPV,DR LIST(1,J),N6N8)
206      IF (J .NE. JHF*2) GO TO 73
207      CALL MULTVC (A PROD,ER LIST(1,JHF),TEMPV)

```

```

208      CALL XFER (TEMPV,ER LIST(1,JHF),N6N8)
209      JHF = JHF-1
210      73 J = J-1
211      IF (J .LE. 0) GO TO 74
212      CALL MULT(A PROD,A STOR(1,J),A TEMP)
213      CALL XFER(A TEMP,A PROD,N18N32)
214      GO TO 72
215      74 CONTINUE
216      IF (IEW .EQ. 0) RETURN
217      C
218      IF (IEW .NE. 1) GO TO 85
219      DO 82 II=1,4
220      I = 9-II
221      R(I) = R(I-2)
222      C
223      DO 81 J=1,NR A
224      81 DR LIST(I,J) = DR LIST(I-2,J)
225      C
226      DO 82 J=1,NR JHF
227      82 ER LIST(I,J) = ER LIST(I-2,J)
228      RETURN
229      C
230      85 R(7) = -(R(1)+R(3)+1.0)
231      R(8) = -(R(2)+R(4))
232      R(5) = R(2)-R(4)
233      R(6) = R(3)-R(1)
234      C
235      DO 86 J=1,NR A
236      DR LIST(7,J) = -(DR LIST(1,J)+DR LIST(3,J))
237      DR LIST(8,J) = -(DR LIST(2,J)+DR LIST(4,J))
238      DR LIST(5,J) = DR LIST(2,J)-DR LIST(4,J)
239      86 DR LIST(6,J) = DR LIST(3,J)-DR LIST(1,J)
240      C
241      DO 87 J=1,NR JHF
242      ER LIST(7,J) = -(ER LIST(1,J)+ER LIST(3,J))
243      ER LIST(8,J) = -(ER LIST(2,J)+ER LIST(4,J))
244      ER LIST(5,J) = ER LIST(2,J)-ER LIST(4,J)
245      87 ER LIST(6,J) = ER LIST(3,J)-ER LIST(1,J)
246      RETURN
247      C
248      900 FORMAT ('0')
249      901 FORMAT (' ',10X,4(2X,2F9.5))
250      902 FORMAT (' ',F10.2,4(2X,2F9.5))
251      C
252      END

```

BPRT,S INVERT. ENNU

```

MORFITT*INVERT(1).ENNU
  1      SUBROUTINE EN NU
  2      C
  3      COMMON/DLTAH C/DELTAH
  4      COMMON/JAY COM/J
  5      COMMON/HTS COM/HTS(51)
  6      COMMON/ALPHA C/ALPHA(51)
  7      COMMON/EXPNU C/COEFNU,EXPNU
  8      COMMON/EN COLL/HT,EN,NU
  9      COMMON/W COM/WT,WB
 10      REAL NU
 11      C
 12      C
 13      WT = (HT-HTS(J+1))/DELTAH
 14      WB = (HTS(J)-HT)/DELTAH
 15      EN = EXP(WT*ALPHA(J)+WB*ALPHA(J+1))
 16      NU = COEFNU*EXP(EXPNU*HT)
 17      RETURN
 18      C
 19      END

```

```
@PRT,S INVERT.INITLI
```

MORFITT*INVERT(1).INITLI

```

1      SUBROUTINE INITLI
2      C
3      COMMON/IPRNT C/IPRINT
4      COMMON/CS COM/C,S,CSQ
5      COMMON/M COM/M11,M22,M33,M21,M31,M23
6      COMMON/INTEGR/R11,R12,R22,IR SKIP(24),DP11,DR12,DR22
7      COMPLEX
8      S      M11,M22,M33,M21,M31,M23,
9      S      R11,R12,R22,
10     S      DR11,DR12,DR22,
11     S      B4,B2,B0,HALF B2,ROOT,Q(2),
12     S      D11,D12,D13,D31,D32,D33,
13     S      DEN,P(2),T(2),FACTOR,
14     S      DB4,DR2,DB0,DQ(2),
15     S      DD11,DD12,DD13,DD31,DD32,DD33,
16     S      DDEN,DP(2),DT(2)
17     C
18     C
19     B4 = 1.0+M33
20     B2 = -(CSQ+M33)*(1.0+M11)-M31**2-(1.0+M33)*(CSQ+M22)+M23**2
21     B0 = (1.0+M11)*(CSQ+M22)*(CSQ+M33)-2.0*M21*M31*M23
22     S      -(1.0+M11)*M23**2+(CSQ+M22)*M31**2+(CSQ+M33)*M21**2
23     C
24     HALF B2 = B2/2.0
25     ROOT = CSQRT(HALF B2**2-B4*B0)
26     Q(1) = CSQRT((-HALF B2+ROOT)/B4)
27     Q(2) = CSQRT((-HALF B2-ROOT)/B4)
28     C
29     IF (IPRINT .NE. 0) PRINT 200,Q(1),Q(2)
30     200 FORMAT ('Q', 'Q = ',2(2X,2E13.4),/)
31     C
32     DO 31 K=1,2
33     D11 = 1.0+M11-Q(K)**2
34     D12 = -M21
35     D13 = -M31+Q(K)*S
36     D31 = M31+Q(K)*S
37     D32 = M23
38     D33 = CSQ+M33
39     DEN = D11*D33-D13*D31
40     P(K) = (-D12+D33+D13+D32)/DEN
41     31 T(K) = Q(K)*P(K)-S*(-D11+D32+D12+D31)/DEN
42     C
43     DB4 = M33
44     DB2 = -M33*(1.0+M11)-(CSQ+M33)*M11-2.0*M31**2
45     S      -M33*(CSQ+M22)-(1.0+M33)*M22+2.0*M23**2
46     DR0 = M11*(CSQ+M22)*(CSQ+M33)+(1.0+M11)*M22*(CSQ+M33)
47     S      +(1.0+M11)*(CSQ+M22)*M33-6.0*M21*M31*M23
48     S      -(2.0+3.0*M11)*M23**2
49     S      +(2.0*CSQ+3.0*M22)*M31**2
50     S      +(2.0*CSQ+3.0*M33)*M21**2
51     C

```

```

52      DO 41 K=1,2
53      DQ(K) = -((DB4+Q(K)**2+DB2)*Q(K)**2+DB0)
54      $      /((4.0*B4+Q(K)**2+2.0*B2)*Q(K))
55      DD11 = M11-2.0*Q(K)*DQ(K)
56      DD12 = -M21
57      DD13 = -M31+DQ(K)*S
58      DD31 = M31+DQ(K)*S
59      DD32 = M23
60      DD33 = M33
61      DDEN = DD11*D33+D11*DD33-DD13*D31-D13*DD31
62      DP(K) = -DDEN*P(K)/DEN
63      $      +(-DD12*D33-D12*DD33+DD13*D32+D13*DD32)/DEN
41 DT(K) = DQ(K)*P(K)+Q(K)*DP(K)
65      $      -S*(-DD11*D32-D11*DD32+DD12*D31+D12*DD31
66      $      -(-D11*D32+D12*D31)*DDEN/DEN)/DEN
67 C
68      DEN = (T(1)+C+P(1))*(C+Q(2))-(T(2)+C+P(2))*(C+Q(1))
69      FACTOR = 2.0/DEN
70      R11 = (C*(T(1)-T(2))+T(1)*Q(2)-T(2)*Q(1))*FACTOR
71      R22 = (C*(T(1)-T(2))+P(1)-P(2))*FACTOR
72      R12 = (Q(2)-Q(1))*FACTOR
73 C
74      DDEN = (DT(1)+C+DP(1))*(C+Q(2))+(T(1)+C+P(1))*DQ(2)
75      $      -(DT(2)+C+DP(2))*(C+Q(1))-(T(2)+C+P(2))*DQ(1)
76      DR11 = -R11*DDEN/DEN
77      $      +(C*(DT(1)-DT(2))
78      $      +DT(1)*Q(2)+T(1)*DQ(2)-DT(2)*Q(1)-T(2)*DQ(1))*FACTOR
79      DR22 = -R22*DDEN/DEN
80      $      +(C*(DT(1)-DT(2))+DP(1)-DP(2))*FACTOR
81      DR12 = -R12*DDEN/DEN
82      $      +(DQ(2)-DQ(1))*FACTOR
83 C
84      R11 = C*R11-1.0
85      R22 = C*R22-1.0
86      R12 = C*R12
87 C
88      DR11 = C*DR11
89      DR22 = C*DR22
90      DR12 = C*DR12
91      RETURN
92 C
93      END

```

@PRT,S INVERT.COEFFS

MORFITT*INVERT(1).COEFFS

```

1      SUBROUTINE COEFFS
2      C
3      COMMON/THETA C/THETA
4      COMMON/FLD COM/AZIM,CODIP,MAGFLD
5      COMMON/FRQ COM/FREQ
6      COMMON/EN COLL/HT,EN,COLL
7      COMMON/WN COM/WAVE NR
8      COMMON/CS COM/C,S,CSQ
9      COMMON/M COM/M11,M22,M33,M21,M31,M23
10     COMMON/SD COM/D11A,S11B,D11B,S21,D21,S22,D22
11     COMMON/DSD COM/DD11A,DS11B,DD11B,DS21,DD21,DS22,DD22
12     COMMON/IEW COM/IEW
13     COMPLEX
14     S      M11,M22,M33,M21,M31,M23,
15     S      D11A,S11B,D11B,S21,D21,S22,D22,
16     S      DD11A,DS11B,DD11B,DS21,DD21,DS22,DD22,
17     S      U,USQ,D,IUD,
18     S      M23D,M31D,
19     S      M2323D,M3131D,M2331D,
20     S      T14 OVC,T32 OVC,CT41,T11,T31,T34 OVC,
21     S      DT14OC,DT32OC,DCT41,DT11,DT31,DT34OC
22     REAL MAGFLD,
23     S      MY,NY,IUD PRT,
24     S      MSQYSQ,NSQYSQ,MNYSQ
25     DIMENSION U PARTS(2),USQ PRT(2),D PARTS(2),IUD PRT(2)
26     EQUIVALENCE (U,U PARTS),(USQ,USQ PRT),(D,D PARTS),(IUD,IUD PRT)
27     DATA PI/3.141593/
28     DATA RTD/57.29578/
29     DATA COEFF X/3.182357E3/
30     DATA COEFF Y/1.758796E11/
31     DATA VEL LT/2.997928E5/
32     C
33     C
34     CALL EN NU
35     C
36     X = COEF EN*EN
37     Z = COLL*OV OMGA
38     U PARTS(1) = 1.0
39     U PARTS(2) = -Z
40     USQ PRT(1) = 1.0-Z**2
41     USQ PRT(2) = -Z-Z
42     D = -X/(U*(USQ-YSQ))
43     IF(ABS(D PARTS(1)) .LT. 1.0E-30 .AND.
44     S      ABS(D PARTS(2)) .LT. 1.0E-30) D = 0.0
45     IUD PRT(1) = Z*D PARTS(1)-D PARTS(2)
46     IUD PRT(2) = D PARTS(1)+Z*D PARTS(2)
47     M11 = USQ*D
48     M22 = M11-MSQYSQ*D
49     M33 = M11-NSQYSQ*D
50     M21 = NY*IUD
51     M31 = -MY*IUD

```

```

52      M23 = -MNYSQ*D
53 C
54      D = 1.0/(1.0+M32)
55      M23D = M23*D
56      M31D = M31*D
57      T14 OVC = OVR C-SSQ OVC*D
58      M2323D = M23*M23D
59      T32 OVC = (CSQ+M22-M2323D)*OVR C
60      M3131D = M31*M31D
61      CT41 = C*(1.0+M11+M3131D)
62      T11 = -S*M31D
63      M2331D = M23*M31D
64      T31 = M2331D-M21
65      T34 OVC = S OVR C*M23D
66 C
67      D11A = T11+T11
68      S11P = T14 OVC+CT41
69      D11P = T14 OVC-CT41
70      S21 = T34 OVC+T31
71      D21 = T34 OVC-T31
72      S22 = C+T32 OVC
73      D22 = C-T32 OVC
74 C
75      DT140C = -SSQ OVC+D*(D-1.0)
76      DT320C = (M22-M2323D*(D+1.0))*OVR C
77      DCT41 = C*(M11+M3131D*(D+1.0))
78      DT11 = T11*D
79      DT31 = M2331D*(D+1.0)-M21
80      DT340C = T34 OVC*D
81 C
82      DD11A = DT11+DT11
83      DS11E = DT140C+DCT41
84      DD11E = DT140C-DCT41
85      DS21 = DT340C+DT31
86      DD21 = DT340C-DT31
87      DS22 = DT320C
88      DD22 = -DT320C
89      RETURN
90 C
91 C
92      ENTRY INIT T
93      IEW = 1
94      DIR CS M = SIN(CODIP/RTD)*SIN(AZIM/RTD)
95      DIR CS A = -COS(CODIP/RTD)
96      C = COS(THETA/RTD)
97      S = SIN(THETA/RTD)
98      OVR C = 1.0/C
99      CSQ = C**2
100     S OVR C = S/C
101     SSQ OVC = S**2/C
102     RETURN
103 C

```

```
104 C      ENTRY INIT S
105        OMEGA = 2.0*PI*FREQ*1000.0
106        WAVE NR = OMEGA/VEL LT
107        COEF EN = COEFF X*1.0E06/OMEGA**2
108        OV OMGA = 1.0/OMEGA
109        Y = -COEFF Y*MAGFLD/OMEGA
110        YSQ = Y**2
111        MY = DIR CS M*Y
112        NY = DIR CS N*Y
113        MSQYSQ = DIR CS M**2*YSQ
114        NSQYSQ = DIR CS N**2*YSQ
115        MNYSQ = DIR CS M*DIR CS N*YSQ
116        RETURN
117
118 C      END
119
```

```
@PRT,S INVERT.DIFFEQ
```

MORFITT*INVERT(1).DIFFEQ

```

1      SUBROUTINE DIFF EQ
2      C
3      COMMON/SD COM/D11A,S11B,D11B,S21,D21,S22,D22
4      COMMON/DSD COM/DD11A,DS11B,DD11B,DS21,DD21,DS22,DD22
5      COMMON/W COM/WT,WB
6      COMMON/INTEGR/R(36)
7      COMMON/DRDH C/DRDH(36)
8      COMMON/X COM/X(6)
9      COMMON/DXDH C/DXDH(6)
10     COMMON/ORDER C/R11,R12,R22,P(3,3),DR(6),
11     $      DER(3),PDER(3,3),DDER(6)
12     COMMON/OVRFLO/IOVFLO
13     COMPLEX
14     $      D11A,S11B,D11B,S21,D21,S22,D22,
15     $      DD11A,DS11B,DD11B,DS21,DD21,DS22,DD22,
16     $      R11,R12,R22,P,DR,
17     $      DER,PDER,DDER,
18     $      B11,B12,B22,R12R12,C12,
19     $      DB11,DB22,DB12,DC12,
20     $      X11,X12,X21,X22,
21     $      G11,G12,G21,G22,G23,G32,G33,
22     $      DTEMP(3)
23     DIMENSION R MTRX(36),DERIV(36)
24     EQUIVALENCE (R11,R MTRX),(DER,DERIV)
25     C
26     C
27     ENTRY R DERIV
28     CALL XFER (R,R MTRX,36)
29     IXFLAG = 0
30     C
31     20 DO 21 I=1,6
32     IF (ABS(R MTRX(I)) .GT. 9.0) GO TO 90
33     21 CONTINUE
34     C
35     B11 = R11*(D11A-D11B)
36     B12 = R12*D21
37     B22 = R22*D22
38     R12R12 = R12*R12
39     C12 = R12*S21
40     C
41     DER(1) = (B11+B12+B12-S11B-S11B)*R11+R12R12+D22+C12+C12-D11A-D11B
42     DER(2) = (B11+B12+B22 -S11B-S22)*R12+(R11+D21+S21)*(R22+1.0)
43     DER(3) = (B12+B12+B22-S22-S22)*R22+R12R12+(D11A-D11B)+B12+B12+D22
44     IF (IXFLAG .NE. 0) GO TO 60
45     C
46     X11 = -S11B+B11+B12
47     X12 = S21+R11+D21+R12+D22
48     X21 = D21+R12*(D11A-D11B)+R22+D21
49     X22 = -S22+B12+P22
50     C
51     G11 = X11+X11

```

```

52      G12 = X12+X12
53      G21 = X21
54      G22 = X11+X22
55      G23 = X12
56      G32 = X21+X21
57      G33 = X22+X22
58      C
59      DO 31 L=1,3
60      PDER(1,L) = G11*P(1,L)+G12*P(2,L)
61      PDER(2,L) = G21*P(1,L)+G22*P(2,L)+G23*P(3,L)
62      31 PDER(3,L) =          G32*P(2,L)+G33*P(3,L)
63      C
64      DB11 = R11*(DD11A-DD11B)
65      DB22 = R22*DD22
66      DB12 = R12*DD21
67      DC12 = R12*DS21
68      C
69      DTEMP(1) = (DB11+DB12+DB12-DS11A-DS11B)*R11
70      S          +R12R12*DD22+DC12+DC12-DD11A-DD11B
71      DTEMP(2) = (DB11+DB12+DB22-DS11B-DS22)*R12
72      S          +(R11*DD21+DS21)*(R22+1.0)
73      DTEMP(3) = (DB12+DB12+DB22-DS22-DS22)*R22
74      S          +R12R12*(DD11A-DD11B)+DB12+DB12+DD22
75      C
76      DDER(1) = DTEMP(1)*WT      +G11*DR( 1)+G12*DR( 2)
77      DDER(4) = DTEMP(1)*WB      +G11*DR( 4)+G12*DR( 5)
78      DDER(2) = DTEMP(2)*WT      +G21*DR( 1)+G22*DR( 2)+G23*DR( 3)
79      DDER(5) = DTEMP(2)*WB      +G21*DR( 4)+G22*DR( 5)+G23*DR( 6)
80      DDER(3) = DTEMP(3)*WT          +G32*DR( 2)+G33*DR( 3)
81      DDER(6) = DTEMP(3)*WB          +G32*DR( 5)+G33*DR( 6)
82      C
83      C          (A+I*B)/I = B-I*A
84      DO 41 I=1,3,2
85      DRDH(I) = DERIV(I+1)
86      41 DRDH(I+1) = -DERIV(I)
87      RETURN
88      C
89      C
90      ENTRY X DERIV
91      CALL XFER (X,R MTRX,6)
92      IXFLAG = 1
93      GO TO 20
94      C
95      60 DO 61 I=1,5,2
96      DXDH(I) = DERIV(I+1)
97      61 DXDH(I+1) = -DERIV(I)
98      RETURN
99      C
100     C
101     ENTRY SET P DR
102     CALL XFER (R,R MTRX,36)
103     DO 72 L=1,3

```

```
104      DO 71 I=1,3
105      71 P(I,L) = 0.0
106      72 P(L,L) = 1.0
107      C
108      DO 74 I=1,3
109      DR(I) = DR(I+3)
110      74 DR(I+3) = 0.0
111      CALL XFER (R MTRX,R,36)
112      WT = 1.0
113      WB = 0.0
114      RETURN
115      C
116      90 IOVFLO = 1
117      RETURN
118      C
119      END
120
121
122
```

6PRT,S INVERT.MTXFCT

```

MORFITT*INVERT(1).MTXFCT
  1      SUBROUTINE MTX FCT
  2      C
  3      DIMENSION A(1),B(1)
  4      COMPLEX C(3,3),D(3,3),E(3,3),
  5      S      F(3,3),G(3),H(3),SUM
  6      C
  7      C
  8      ENTRY XFER (A,B,N)
  9      DO 11 J=1,N
 10      11 B(J) = A(J)
 11      RETURN
 12      C
 13      C
 14      ENTRY MULT (C,D,E)
 15      DO 22 J=1,3
 16      DO 22 JJ=1,3
 17      SUM = 0.0
 18      DO 21 JJJ=1,3
 19      21 SUM = SUM+C(J,JJJ)*D(JJJ,JJ)
 20      22 E(J,JJ) = SUM
 21      RETURN
 22      C
 23      C
 24      ENTRY MULTVC (F,G,H)
 25      DO 32 J=1,3
 26      SUM = 0.0
 27      DO 31 JJ=1,3
 28      31 SUM = SUM+F(J,JJ)*G(JJ)
 29      32 H(J) = SUM
 30      RETURN
 31      C
 32      END

```

@PRT,S INVERT.DGELB

MORFITT*INVERT(1).DGELB

```

1      SUBROUTINE DGELB(R,A,M,N,MUD,MLD,EPS,IER)                                DELB
2      C                                                                                   DELB
3      C .....                                                                           DELB
4      C                                                                                   DELB
5      C      SUBROUTINE DGELB                                                         DELB
6      C                                                                                   DELB
7      C      PURPOSE                                                                    DELB
8      C      TO SOLVE A SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS WITH A              DELB
9      C      COEFFICIENT MATRIX OF BAND STRUCTURE.                                     DELB
10     C                                                                                   DELB
11     C      USAGE                                                                      DELB
12     C      CALL DGELB(R,A,M,N,MUD,MLD,EPS,IER)                                     DELB
13     C                                                                                   DELB
14     C      DESCRIPTION OF PARAMETERS                                                  DELB
15     C      R      - DOUBLE PRECISION M BY N RIGHT HAND SIDE MATRIX                 DELB
16     C              (DESTROYED). ON RETURN R CONTAINS THE SOLUTION                   DELB
17     C              OF THE EQUATIONS.                                                DELB
18     C      A      - DOUBLE PRECISION M BY M COEFFICIENT MATRIX WITH                 DELB
19     C              BAND STRUCTURE (DESTROYED).                                       DELB
20     C      M      - THE NUMBER OF EQUATIONS IN THE SYSTEM.                         DELB
21     C      N      - THE NUMBER OF RIGHT HAND SIDE VECTORS.                         DELB
22     C      MUD     - THE NUMBER OF UPPER CODIAGONALS (THAT MEANS                     DELB
23     C              CODIAGONALS ABOVE MAIN DIAGONAL).                               DELB
24     C      MLD     - THE NUMBER OF LOWER CODIAGONALS (THAT MEANS                     DELB
25     C              CODIAGONALS BELOW MAIN DIAGONAL).                               DELB
26     C      EPS     - SINGLE PRECISION INPUT CONSTANT WHICH IS USED AS              DELB
27     C              RELATIVE TOLERANCE FOR TEST ON LOSS OF                          DELB
28     C              SIGNIFICANCE.                                                    DELB
29     C      IER     - RESULTING ERROR PARAMETER CODED AS FOLLOWS                    DELB
30     C              IER=0 - NO ERROR,                                                DELB
31     C              IER=-1 - NO RESULT BECAUSE OF WRONG INPUT PARAME- DELB
32     C                      TERS M,MUD,MLD OR BECAUSE OF PIVOT ELEMENT DELB
33     C                      AT ANY ELIMINATION STEP EQUAL TO 0,                     DELB
34     C              IER=K - WARNING DUE TO POSSIBLE LOSS OF SIGNIFI- DELB
35     C                      CANCE INDICATED AT ELIMINATION STEP K+1, DELB
36     C                      WHERE PIVOT ELEMENT WAS LESS THAN OR DELB
37     C                      EQUAL TO THE INTERNAL TOLERANCE EPS TIMES DELB
38     C                      ABSOLUTELY GREATEST ELEMENT OF MATRIX A. DELB
39     C                                                                                   DELB
40     C      REMARKS                                                                      DELB
41     C      BAND MATRIX A IS ASSUMED TO BE STORED ROWWISE IN THE FIRST DELB
42     C      ME SUCCESSIVE STORAGE LOCATIONS OF TOTALLY NEEDED MA DELB
43     C      STORAGE LOCATIONS, WHERE DELB
44     C      MA=M*MC-ML*(ML+1)/2 AND ME=MA-MU*(MU+1)/2 WITH DELB
45     C      MC=MIN(M,1+MUD+MLD), ML=MC-1-MLD, MU=MC-1-MUD. DELB
46     C      RIGHT HAND SIDE MATRIX R IS ASSUMED TO BE STORED COLUMNWISE DELB
47     C      IN N*M SUCCESSIVE STORAGE LOCATIONS. ON RETURN SOLUTION DELB
48     C      MATRIX R IS STORED COLUMNWISE TOO. DELB
49     C      INPUT PARAMETERS M, MUD, MLD SHOULD SATISFY THE FOLLOWING DELB
50     C      RESTRICTIONS MUD NOT LESS THAN ZERO DELB
51     C      MLD NOT LESS THAN ZERO DELB

```

```

52 C                                MUD+MLD NOT GREATER THAN 2*M-2.                                DELB
53 C                                NO ACTION BESIDES ERROR MESSAGE IER=-1 TAKES PLACE IF THESE DELB
54 C                                RESTRICTIONS ARE NOT SATISFIED.                                DELB
55 C                                THE PROCEDURE GIVES RESULTS IF THE RESTRICTIONS ON INPUT DELB
56 C                                PARAMETERS ARE SATISFIED AND IF PIVOT ELEMENTS AT ALL DELB
57 C                                ELIMINATION STEPS ARE DIFFERENT FROM 0. HOWEVER WARNING DELB
58 C                                IER=K - IF GIVEN - INDICATES POSSIBLE LOSS OF SIGNIFICANCE. DELB
59 C                                IN CASE OF A WELL SCALED MATRIX A AND APPROPRIATE TOLERANCE DELB
60 C                                EPS, IER=K MAY BE INTERPRETED THAT MATRIX A HAS THE RANK K. DELB
61 C                                NO WARNING IS GIVEN IF MATRIX A HAS NO LOWER CODIAGONAL. DELB
62 C
63 C                                SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED DELB
64 C                                NONE DELB
65 C
66 C                                METHOD DELB
67 C                                SOLUTION IS DONE BY MEANS OF GAUSS ELIMINATION WITH DELB
68 C                                COLUMN PIVOTING ONLY, IN ORDER TO PRESERVE BAND STRUCTURE DELB
69 C                                IN REMAINING COEFFICIENT MATRICES. DELB
70 C
71 C                                ..... DELB
72 C                                ..... DELB
73 C                                ..... DELB
74 C
75 C                                DIMENSION R(1),A(1) DELB
76 C                                DOUBLE PRECISION F,A,PIV,TE,TOL DELB
77 C
78 C                                TEST ON WRONG INPUT PARAMETERS DELB
79 C                                IF(MLD)47,1,1 DELB
80 C                                1 IF(MUD)47,2,2 DELB
81 C                                2 MC=1+MLD+MUD DELB
82 C                                IF(MC+1-M-M)3,3,47 DELB
83 C
84 C                                PREPARE INTEGER PARAMETERS DELB
85 C                                MC=NUMBER OF COLUMNS IN MATRIX A DELB
86 C                                MU=NUMBER OF ZEROS TO BE INSERTED IN FIRST ROW OF MATRIX A DELB
87 C                                ML=NUMBER OF MISSING ELEMENTS IN LAST ROW OF MATRIX A DELB
88 C                                MR=INDEX OF LAST ROW IN MATRIX A WITH MC ELEMENTS DELB
89 C                                MZ=TOTAL NUMBER OF ZEROS TO BE INSERTED IN MATRIX A DELB
90 C                                MA=TOTAL NUMBER OF STOPAGE LOCATIONS NECESSARY FOR MATRIX A DELB
91 C                                NM=NUMBER OF ELEMENTS IN MATRIX R DELB
92 C                                3 IF(MC-M)5,5,4 DELB
93 C                                4 MC=M DELB
94 C                                5 MU=MC-MUD-1 DELB
95 C                                ML=MC-MLD-1 DELB
96 C                                MR=M-ML DELB
97 C                                MZ=(MU*(MU+1))/2 DELB
98 C                                MA=M*MC-(ML*(ML+1))/2 DELB
99 C                                NM=N*M DELB
100 C
101 C                                MOVE ELEMENTS BACKWARD AND SEARCH FOR ABSOLUTELY GREATEST ELEMENT DELB
102 C                                (NOT NECESSARY IN CASE OF A MATRIX WITHOUT LOWER CODIAGONALS) DELB
103 C                                IFR=0 DELB

```

104		PIV=U.DC	DELB
105		IF(MLD)14,14,6	DELB
106	6	JJ=MA	DELB
107		J=MA-MZ	DELB
108		KST=J	DELB
109		DO 9 K=1,KST	DELB
110		TB=A(J)	DELB
111		A(JJ)=TB	DELB
112		TB=DAES(TB)	DELB
113		IF(TB-PIV)6,8,7	DELB
114	7	PIV=TB	DELB
115	8	J=J-1	DELB
116	9	JJ=JJ-1	DELB
117	C		DELB
118	C	INSEKT ZEROS IN FIRST MU ROWS (NOT NECESSARY IN CASE MZ=0)	DELB
119		IF(MZ)14,14,10	DELB
120	10	JJ=1	DELB
121		J=1+MZ	DELB
122		IC=1+MUD	DELB
123		DO 13 I=1,MU	DELB
124		DO 14 K=1,MC	DELB
125		A(JJ)=C.DC	DELB
126		IF(K-IC)11,11,12	DELB
127	11	A(JJ)=A(J)	DELB
128		J=J+1	DELB
129	12	JJ=JJ+1	DELB
130	13	IC=IC+1	DELB
131	C		DELB
132	C	GENEATE TEST VALUE FOR SINGULARITY	DELB
133	14	TOL=EPS*PIV	DELB
134	C		DELB
135	C		DELB
136	C	START DECOMPOSITION LOOP	DELB
137		KST=1	DELB
138		IDST=MC	DELB
139		IC=MC-1	DELB
140		DO 18 K=1,M	DELB
141		IF(K-MR-1)16,16,15	DELB
142	15	IDST=IDST-1	DELB
143	16	ID=IDST	DELB
144		ILR=K+MLD	DELB
145		IF(ILR=M)18,18,17	DELB
146	17	ILR=M	DELB
147	18	II=KST	DELB
148	C		DELB
149	C	PIVOT SEARCH IN FIRST COLUMN (ROW INDEXES FROM I=K UP TO I=ILR)	DELB
150		PIV=U.DC	DELB
151		DO 20 I=K,ILR	DELB
152		TR=OABS(A(II))	DELB
153		IF(TB-PIV)20,20,19	DELB
154	19	PIV=TB	DELB
155		J=1	DELB

156		JJ=II	DELB
157	20	IF(I-MR)22,22,21	DELB
158	21	ID=ID-1	DELB
159	22	II=II+ID	DELB
160	C		DELB
161	C	TEST ON SINGULARITY	DELB
162		IF(PIV)47,47,23	DELB
163	23	IF(I-R)26,24,26	DELB
164	24	IF(PIV-TOL)25,25,26	DELB
165	25	IER=K-1	DELB
166	26	PIV=1.00/A(JJ)	DELB
167	C		DELB
168	C	PIVOT ROW REDUCTION AND ROW INTERCHANGE IN PIGHT HAND SIDE R	DELB
169		ID=J-K	DELB
170		DO 27 I=K,NM,M	DELB
171		II=I+ID	DELB
172		TB=PIV*R(II)	DELB
173		R(II)=R(I)	DELB
174	27	R(I)=TB	DELB
175	C		DELB
176	C	PIVOT ROW REDUCTION AND ROW INTERCHANGE IN COEFFICIENT MATRIX A	DELB
177		II=KST	DELB
178		J=JJ+IC	DELB
179		DO 28 I=JJ,J	DELB
180		TB=PIV*A(I)	DELB
181		A(I)=A(II)	DELB
182		A(II)=TB	DELB
183	28	II=II+1	DELB
184	C		DELB
185	C	ELEMENT REDUCTION	DELB
186		IF(K-ILR)29,34,34	DELB
187	29	ID=KST	DELB
188		II=K+1	DELB
189		MU=KST+1	DELB
190		MZ=KST+IC	DELB
191		DO 30 I=II,ILR	DELB
192	C		DELB
193	C	IN MATRIX A	DELB
194		ID=ID+MC	DELB
195		JJ=I-MR-1	DELB
196		IF(JJ)31,31,30	DELB
197	30	ID=ID-JJ	DELB
198	31	PIV=-A(ID)	DELB
199		J=ID+1	DELB
200		DO 32 JJ=MU,MZ	DELB
201		A(J-1)=A(J)+PIV*A(JJ)	DELB
202	32	J=J+1	DELB
203		A(J-1)=0.00	DELB
204	C		DELB
205	C	IN MATRIX R	DELB
206		J=K	DELB
207		DO 33 JJ=I,NM,M	DELB

208		R(JJ)=R(JJ)+PIV*R(J)	DELB
209	33	J=J+M	DELB
210	34	KST=KST+MC	DELB
211		IF(ILR-MR)36,35,35	DELB
212	35	IC=IC-1	DELB
213	36	ID=K-MR	DELB
214		IF(ID)39,38,37	DELB
215	37	KST=KST-ID	DELB
216	38	CONTINUE	DELB
217	C	END OF DECOMPOSITION LOOP	DELB
218	C		DELB
219	C		DELB
220	C	BACK SUBSTITUTION	DELB
221		IF(MC-1)46,46,39	DELB
222	39	IC=2	DELB
223		KST=MA+ML-MC+2	DELB
224		II=M	DELB
225		DO 45 I=2,M	DELB
226		KST=KST-MC	DELB
227		II=II-1	DELB
228		J=II-MR	DELB
229		IF(J)41,41,40	DELB
230	40	KST=KST+J	DELB
231	41	DO 43 J=II,NM,M	DELB
232		TB=R(J)	DELB
233		MZ=KST+IC-2	DELB
234		ID=J	DELB
235		DO 42 JJ=KST,MZ	DELB
236		ID=ID+1	DELB
237	42	TB=TB-A(JJ)*R(ID)	DELB
238	43	R(J)=TB	DELB
239		IF(IC-MC)44,45,45	DELB
240	44	IC=IC+1	DELB
241	45	CONTINUE	DELB
242	46	RETURN	DELB
243	C		DELB
244	C		DELB
245	C	ERRON RETURN	DELB
246	47	IER=-1	DELB
247		RETURN	DELB
248		END	DELB

@PRT,S INVERT.DGELS

```

MORFITT*INVERT(1).DGELS
1      SUBROUTINE DGELS(R,A,M,N,EPS,IER,AUX)                                DELS
2      C                                                                    DELS
3      C .....                                                                DELS
4      C                                                                    DELS
5      C      SUBROUTINE DGELS                                              DELS
6      C                                                                    DELS
7      C      PURPOSE                                                        DELS
8      C      TO SOLVE A SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS WITH      DELS
9      C      SYMMETRIC COEFFICIENT MATRIX UPPER TRIANGULAR PART OF WHICH  DELS
10     C      IS ASSUMED TO BE STORED COLUMNWISE.                          DELS
11     C                                                                    DELS
12     C      USAGE                                                           DELS
13     C      CALL DGELS(R,A,M,N,EPS,IER,AUX)                               DELS
14     C                                                                    DELS
15     C      DESCRIPTION OF PARAMETERS                                       DELS
16     C      R      - DOUBLE PRECISION M BY N RIGHT HAND SIDE MATRIX      DELS
17     C      (DESTROYED). ON RETURN R CONTAINS THE SOLUTION OF            DELS
18     C      THE EQUATIONS.                                                DELS
19     C      A      - UPPER TRIANGULAR PART OF THE SYMMETRIC DOUBLE        DELS
20     C      PRECISION M BY M COEFFICIENT MATRIX. (DESTROYED)           DELS
21     C      M      - THE NUMBER OF EQUATIONS IN THE SYSTEM.              DELS
22     C      N      - THE NUMBER OF RIGHT HAND SIDE VECTORS.              DELS
23     C      EPS    - SINGLE PRECISION INPUT CONSTANT WHICH IS USED AS    DELS
24     C      RELATIVE TOLERANCE FOR TEST ON LOSS OF                       DELS
25     C      SIGNIFICANCE.                                                DELS
26     C      IER    - RESULTING ERROR PARAMETER CODED AS FOLLOWS         DELS
27     C      IER=0 - NO ERROR,                                             DELS
28     C      IER=-1 - NO RESULT BECAUSE OF M LESS THAN 1 OR               DELS
29     C      PIVOT ELEMENT AT ANY ELIMINATION STEP                        DELS
30     C      EQUAL TO 0,                                                  DELS
31     C      IER=K - WARNING DUE TO POSSIBLE LOSS OF SIGNIFI-            DELS
32     C      CANCE INDICATED AT ELIMINATION STEP K+1,                     DELS
33     C      WHERE PIVOT ELEMENT WAS LESS THAN DP                          DELS
34     C      EQUAL TO THE INTERNAL TOLERANCE EPS TIMES                    DELS
35     C      ABSOLUTELY GREATEST MAIN DIAGONAL                            DELS
36     C      ELEMENT OF MATRIX A.                                          DELS
37     C      AUX    - DOUBLE PRECISION AUXILIARY STORAGE APRAY            DELS
38     C      WITH DIMENSION M-1.                                          DELS
39     C                                                                    DELS
40     C      REMARKS                                                         DELS
41     C      UPPER TRIANGULAR PART OF MATRIX A IS ASSUMED TO BE STORED    DELS
42     C      COLUMNWISE IN M*(M+1)/2 SUCCESSIVE STORAGE LOCATIONS, RIGHT  DELS
43     C      HAND SIDE MATRIX R COLUMNWISE IN N*M SUCCESSIVE STORAGE      DELS
44     C      LOCATIONS. ON RETURN SOLUTION MATRIX R IS STORED COLUMNWISE  DELS
45     C      TOO.                                                         DELS
46     C      THE PROCEDURE GIVES RESULTS IF THE NUMBER OF EQUATIONS M IS  DELS
47     C      GREATER THAN 0 AND PIVOT ELEMENTS AT ALL ELIMINATION STEPS    DELS
48     C      ARE DIFFERENT FROM 0. HOWEVER WARNING IER=K - IF GIVEN -     DELS
49     C      INDICATES POSSIBLE LOSS OF SIGNIFICANCE. IN CASE OF A WELL  DELS
50     C      SCALED MATRIX A AND APPROPRIATE TOLERANCE EPS, IER=K MAY BE  DELS
51     C      INTERPRETED THAT MATRIX A HAS THE RANK K. NO WARNING IS      DELS

```

```

52 C          GIVEN IN CASE M=1.                                DELS
53 C          ERROR PARAMETER IER=-1 DOES NOT NECESSARILY MEAN THAT DELS
54 C          MATRIX A IS SINGULAR, AS ONLY MAIN DIAGONAL ELEMENTS DELS
55 C          ARE USED AS PIVOT ELEMENTS. POSSIBLY SUBROUTINE DGELG (WHICH DELS
56 C          WORKS WITH TOTAL PIVOTING) WOULD BE ABLE TO FIND A SOLUTION. DELS
57 C
58 C          SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED DELS
59 C          NONE DELS
60 C
61 C          METHOD DELS
62 C          SOLUTION IS DONE BY MEANS OF GAUSS-ELIMINATION WITH DELS
63 C          PIVOTING IN MAIN DIAGONAL, IN ORDER TO PRESERVE DELS
64 C          SYMMETRY IN REMAINING COEFFICIENT MATRICES. DELS
65 C
66 C          ..... DELS
67 C
68 C
69 C
70          DIMENSION A(1),P(1),AUX(1) DELS
71          DOUBLE PRECISION R,A,AUX,PIV,TB,TOL,PIVI DELS
72          IF(M)24,24,1 DELS
73 C
74 C          SEARCH FOR GREATEST MAIN DIAGONAL ELEMENT DELS
75          1 IER=0 DELS
76          PIV=0.00 DELS
77          L=0 DELS
78          DO 3 K=1,M DELS
79          L=L+K DELS
80          TB=DABS(A(L)) DELS
81          IF(TB-PIV)3,3,2 DELS
82          2 PIV=TB DELS
83          I=L DELS
84          J=K DELS
85          3 CONTINUE DELS
86          TOL=EPS*PIV DELS
87 C          MAIN DIAGONAL ELEMENT A(I)=A(J,J) IS FIRST PIVOT ELEMENT. DELS
88 C          PIV CONTAINS THE ABSOLUTE VALUE OF A(I). DELS
89 C
90 C
91 C          START ELIMINATION LOOP DELS
92          LST=0 DELS
93          NM=N*M DELS
94          LEND=M-1 DELS
95          DO 18 K=1,M DELS
96 C
97 C          TEST ON USEFULNESS OF SYMMETRIC ALGORITHM DELS
98          IF(PIV)24,24,4 DELS
99          4 IF(IER)7,5,7 DELS
100         5 IF(PIV-TOL)6,6,7 DELS
101         6 IER=K-1 DELS
102         7 LT=J-K DELS
103         LST=LST+K DELS

```

104	C			DELS
105	C	PIVOT ROW REDUCTION AND ROW INTERCHANGE IN RIGHT HAND SIDE R		DELS
106		PIVI=1.00/A(I)		DELS
107		DO 8 L=K,NM,M		DELS
108		LL=L+LT		DELS
109		TB=PIVI*R(LL)		DELS
110		R(LL)=R(L)		DELS
111	8	R(L)=TB		DELS
112	C			DELS
113	C	IS ELIMINATION TERMINATED		DELS
114		IF(K-M)9,19,19		DELS
115	C			DELS
116	C	ROW AND COLUMN INTERCHANGE AND PIVOT ROW REDUCTION IN MATRIX A.		DELS
117	C	ELEMENTS OF PIVOT COLUMN ARE SAVED IN AUXILIARY VECTOR AUX.		DELS
118	9	LR=LST+(LT*(K+J-1))/2		DELS
119		LL=LR		DELS
120		L=LST		DELS
121		DO 14 II=K,LEND		DELS
122		L=L+II		DELS
123		LL=LL+1		DELS
124		IF(L-LR)12,10,11		DELS
125	10	A(LL)=A(LST)		DELS
126		TR=A		DELS
127		GO TO 13		DELS
128	11	LL=L+LT		DELS
129	12	TR=A(LL)		DELS
130		A(LL)=A(L)		DELS
131	13	AUX(II)=TB		DELS
132	14	A(L)=PIVI*TB		DELS
133	C			DELS
134	C	SAVE COLUMN INTERCHANGE INFORMATION		DELS
135		A(LST)=LT		DELS
136	C			DELS
137	C	ELEMENT REDUCTION AND SEARCH FOR NEXT PIVOT		DELS
138		PIV=0.00		DELS
139		LLST=LST		DELS
140		LT=0		DELS
141		DO 12 II=K,LEND		DELS
142		PIVI=-AUX(II)		DELS
143		LL=LLST		DELS
144		LT=LT+1		DELS
145		DO 15 LLD=II,LEND		DELS
146		LL=LL+LLD		DELS
147		L=LL+LT		DELS
148	15	A(L)=A(L)+PIVI*A(LL)		DELS
149		LLST=LLST+II		DELS
150		LR=LLST+LT		DELS
151		TB=DABS(A(LR))		DELS
152		IF(TB-PIV)17,17,16		DELS
153	16	PIV=TB		DELS
154		I=LR		DELS
155		J=II+1		DELS

156	17	DO 10 LR=K,NM,M	DELS
157		LL=LR+LT	DELS
158	18	R(LL)=R(LL)+PIVI*R(LR)	DELS
159		END OF ELIMINATION LOOP	DELS
160	C		DELS
161	C		DELS
162	C	BACK SUBSTITUTION AND BACK INTERCHANGE	DELS
163	19	IF(LEND)24,23,20	DELS
164	20	II=M	DELS
165		DO 22 I=2,M	DELS
166		LST=LST-II	DELS
167		II=II-1	DELS
168		L=A(LST)+.5DD	DELS
169		DO 22 J=II,NM,M	DELS
170		TB=R(J)	DELS
171		LL=J	DELS
172		K=LST	DELS
173		DO 21 LT=II,LEND	DELS
174		LL=LL+1	DELS
175		K=K+LT	DELS
176	21	TB=TB-A(K)*R(LL)	DELS
177		K=J+L	DELS
178		R(J)=R(K)	DELS
179	22	R(K)=TB	DELS
180	23	RETURN	DELS
181	C		DELS
182	C		DELS
183	C	ERROR RETURN	DELS
184	24	IER=-1	DELS
185		RETURN	DELS
186		END	DELS

&PRT,S INVERT.CLINQ

MORFITT*INVERT(1).CLINEG

1	SUBROUTINE CLIN EQ (A, B, X, N,	00000
2	S N DIM, IFLAG, ERR)	00000
3	C	00000
4	C CLIN EQ USES L-U DECOMPOSITION TO	00000
5	C FIND THE TRIANGULAR MATRICES L, U	00000
6	C SUCH THAT L * U = A. L AND U ARE	00000
7	C STORED IN A. THIS FORM IS USED WITH	00000
8	C BACK-SUBSTITUTION TO FIND THE SOLN	00000
9	C X OF A * X = L * U * X = B.	00000
10	C N IS THE NUMBER OF EQUATIONS AND	00001
11	C N DIM IS THE DIMENSION OF ALL ARRAYS	00001
12	C IN THE PARAMETER LIST.	00001
13	C	00001
14	C IF IFLAG = 0, L, U, AND X ARE	00001
15	C COMPUTED.	00001
16	C IF IFLAG IS NON-ZERO, IT IS ASSUMED	00001
17	C THAT L AND U HAVE BEEN COMPUTED IN	00001
18	C A PREVIOUS CALL AND ARE STILL STORED	00001
19	C IN A. THUS ONLY X IS COMPUTED.	00001
20	C ERR IS THE ESTIMATED RELATIVE	00002
21	C ERROR OF THE SOLUTION VECTOR.	00002
22	C	00002
23	COMPLEX*16 A, B, X, T	00002
24	C INTEGER*2 IROW	TJM02
25	INTEGER*4 IROW	TJM02
26	DIMENSION A(N DIM, N DIM),	00002
27	B(N DIM), X(N DIM)	00002
28	DIMENSION IROW(50), Q(50)	00002
29	DATA EPS /1.0E-15/	00002
30	C	00002
31	C	00003
32	IF (N.GT.50) GO TO 900	00003
33	IF (IFLAG.NE.0) GO TO 600	00003
34	DO 050 I = 1,N	00003
35	Q(I) = 0.0	00003
36	DO 040 J = 1,N	00003
37	QQ = CDABS (A(I,J))	00003
38	U40 IF (Q(I).LT.QQ) Q(I) = QQ	00003
39	IF (Q(I).EQ.C.0) GO TO 901	00003
40	U50 CONTINUE	00003
41	EPR = EPS	00004
42	PPIV = 0.0	00004
43	DO 100 I = 1,N	00004
44	10L IROW(I) = I	00004
45	C	00004
46	DO 500 L = 1,N	00004
47	PIVOT = 0.0	00004
48	K = L - 1	00004
49	DO 240 I = L,N	00004
50	IF (K.LT.1) GO TO 230	00004
51	DO 220 J = 1,K	00005

52	220	A(I,L) = A(I,L) - A(J,L) * A(I,J)	0000
53	230	F = CDABS (A(I,L)) / Q(I)	0000
54		IF (PIVOT.GT.F) GO TO 240	0000
55		PIVOT = F	0000
56		NPIVOT = I	0000
57	240	CONTINUE	0000
58		IF (PIVOT.EQ.0.0) GO TO 901	0000
59		IF (PPIV.LE.PIVOT) GO TO 250	0000
60		ERR = EPR * PPIV / PIVOT	0000
61		IF (ERR.GE.1.0) GO TO 901	0000
62	250	PPIV = PIVOT	0000
63		IF (NPIVOT.EQ.L) GO TO 280	0000
64		Q(NPIVOT) = Q(L)	0000
65		J = IROW(L)	0000
66		IROW(L) = IROW(NPIVOT)	0000
67		IROW(NPIVOT) = J	0000
68		DO 260 I = 1,N	0000
69		T = A(L,I)	0000
70		A(L,I) = A(NPIVOT,I)	0000
71		A(NPIVOT,I) = T	0000
72	260	CONTINUE	0000
73	280	IF (L.EQ.N) GO TO 500	0000
74		T = (1.000,0.000) / A(L,L)	0000
75		K = L + 1	0000
76		M = L - 1	0000
77		DO 450 I = K,N	0000
78		IF (M.LT.1) GO TO 400	0000
79		DO 350 J = 1,M	0000
80	350	A(L,I) = A(L,I) - A(L,J) * A(J,I)	0000
81	400	A(L,I) = T * A(L,I)	0000
82	450	CONTINUE	0000
83	500	CONTINUE	0000
84		IF (ERR.GT.1.0E-5) PRINT 998, ERR	0000
85	C		0000
86	C		0000
87	600	DO 620 I = 2,N	0000
88	620	X(I) = (0.000,0.000)	0000
89		J = IROW(1)	0000
90		X(1) = B(J) / A(1,1)	0000
91		DO 700 I = 2,N	0000
92		J = IROW(I)	0000
93		K = I - 1	0000
94		DO 650 L = 1,K	0000
95	650	X(I) = X(I) + A(I,L) * X(L)	0000
96		X(I) = (B(J) - X(I)) / A(I,I)	0000
97	700	CONTINUE	0000
98		K = N - 1	0000
99		DO 800 I = 1,K	0000
100		J = N - I	0000
101		M = J + 1	0001
102		DO 800 L = M,N	0001
103		X(J) = X(J) - X(L) * A(J,L)	0001

104	800	CONTINUE	0001
105		RETURN	0001
106	C		0001
107	900	PRINT 999	0001
108		ERR = 1.0	0001
109		RETURN	0001
110	901	PRINT 997	0001
111		ERR = 1.0	0001
112		RETURN	0001
113	997	FORMAT ('ERROR IN CLIN EQ, MATRIX IS SINGULAR')	0001
114	998	FORMAT (' CAUTION-',	0001
115	1	' CLIN EQ HAS DECOMPOSED AN ILL-CONDITIONED MATRIX.',/,	0001
116	1	' RESULTS WILL HAVE RELATIVE ERROR =',E11.2)	0001
117	999	FORMAT ('ERROR IN CLIN EQ, MATRIX SIZE GREATER THAN 50')	0001
118		END	0001

* @PRT,S INVERT.OVCHK

MORFITT*INVERT(1).OVCHK

```
1      SUBROUTINE OVCHK(CALLER,STMT)                                DMP
2  C ***** THIS SUBROUTINE WILL CALL THE ROUTINE OVUNFL WHICH    DMP
3  C ***** CHECKS FOR OVERFLOW AND UNDERFLOW. THE ROUTINE RETURNS AN  DMP
4  C ***** INTEGER VALUE WHICH TELLS WHETHER OVERFLOW AND/OR UNDERFLOW  DMP
5  C ***** HAVE OCCURRED SINCE THE PROGRAM STARTED OR SINCE THE LAST TIME  DMP
6  C ***** ONE OF THE OVERFLOW/UNDERFLOW CHECKING ROUTINES WAS CALLED,    DMP
7  C ***** WHICHEVER HAS BEEN MORE RECENT. THE INTEGER VALUES RETURNED  DMP
8  C ***** AND THEIR MEANINGS ARE AS FOLLOWS:                      DMP
9  C *****      1= OVERFLOW ONLY                                     DMP
10 C *****      2= NO OVERFLOW OR UNDERFLOW                         DMP
11 C *****      3= UNDERFLOW ONLY                                   DMP
12 C *****      4= OVERFLOW AND UNDERFLOW                           DMP
13 C ***** THIS SUBROUTINE (OVCHK) WILL PRINT OUT A MESSAGE STATING    DMP
14 C ***** WHETHER OVERFLOW HAS OR HAS NOT OCCURRED. FOR PURPOSES      DMP
15 C ***** OF THIS CONVERSION, UNDERFLOW OCCURRENCES HAVE BEEN IGNORED  DMP
16 C ***** BECAUSE THE 1110 TAKES THE SAME ACTION (I.E. SETS THE RESULT  DMP
17 C ***** TO ZERO) IN THE CASE OF UNDERFLOW AS DOES THE 360.          DMP
18 C ***** THE SUBROUTINE ALSO PRINTS OUT THE NAME OF THE CALLING      DMP
19 C ***** ROUTINE (VARIABLE "CALLER") AND THE STATEMENT NUMBER FROM    DMP
20 C ***** WHICH IT WAS CALLED (VARIABLE "STMT")                      DMP
21      CHARACTER*6 CALLER                                           DMP
22      INTEGER STMT                                                 DMP
23      CALL OVUNFL(IOVFL)                                           DMP
24      IF (IOVFL .EQ. 1 .OR. IOVFL .EQ. 4) GO TO 150                DMP
25      WRITE (6,100)                                                DMP
26      100 FORMAT ('NO OVERFLOW HAS NOT OCCURRED')                 DMP
27      GO TO 250                                                     DMP
28      150 WRITE (6,200)                                             DMP
29      200 FORMAT ('NO OVERFLOW HAS OCCURRED')                     DMP
30      250 CONTINUE                                                 DMP
31      WRITE (6,300) STMT,CALLER                                    DMP
32      300 FORMAT ('OVCHK WAS CALLED FROM STATEMENT ',I5,' IN SUBROUTINE  DMP
33      1 ,A6)                                                       DMP
34      RETURN                                                       DMP
35      END                                                           DMP
```

&FIN