A numerical technique for the calculation of dispersion relation
OCT 80 I W KAY, P J DRAPER, W WASYLKIWSKYJ
MDA903-79-C-0202
IDA/F-1491
A NUMERICAL TECHNIQUE FOR THE CALCULATION OF DISPERSION RELATIONS AND MODE FUNCTIONS FOR UPPER OCEAN INTERNAL WAVES

Irvin W. Kay
Patricia J. Draper
Wasyl Wasylkiwskyj

October 1980

Prepared for
Defense Advanced Research Projects Agency

INSTITUTE FOR DEFENSE ANALYSES
SCIENCE AND TECHNOLOGY DIVISION
A Numerical Technique for the Calculation of Dispersion Relations and Mode Functions for Upper Ocean Internal Waves

Irvin W. Kay, Patricia J. Draper, Wasyl Wasylikosyj

Institute for Defense Analyses
400 Army-Navy Drive
Arlington, Virginia 22202

Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

Approved for public release; distribution unlimited.

This paper is concerned with calculating the internal wave eigenfunctions and dispersion relations for an infinitely deep ocean with an arbitrary Vaisala frequency profile. The method involves numerically integrating the differential equation from a depth where the profile is essentially an exponential function, and therefore where the eigenfunctions are known explicitly, to the surface where each eigenfunction must vanish. The equation that is implied by the surface boundary condition determines...
20. (continued)

the dispersion relation, which is obtained numerically by solving the equation by means of the method of *regula falsi*. A computer program that performs the necessary calculations is described and listed.
A NUMERICAL TECHNIQUE FOR THE CALCULATION OF DISPERSION RELATIONS AND MODE FUNCTIONS FOR UPPER OCEAN INTERNAL WAVES

Irvin W. Kay
Patricia J. Draper
Wasyl Wasylkiwskyj

October 1980

INSTITUTE FOR DEFENSE ANALYSES
SCIENCE AND TECHNOLOGY DIVISION
400 Army-Navy Drive, Arlington, Virginia 22202
Contract MDA 903 79 C 0282
DARPA Assignment A-49
### CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. DESCRIPTION OF THE PROBLEM AND SOME GENERAL PROPERTIES OF INTEREST</td>
<td>5</td>
</tr>
<tr>
<td>III. THE NUMERICAL PROCEDURE TO DETERMINE DISPERSION RELATIONS</td>
<td>9</td>
</tr>
<tr>
<td>IV. DETAILS OF THE NUMERICAL PROCEDURE</td>
<td>15</td>
</tr>
<tr>
<td>A. SOLUTION OF THE DIFFERENTIAL EQUATION</td>
<td>15</td>
</tr>
<tr>
<td>B. SOLUTION OF THE EIGENVALUE EQUATION</td>
<td>16</td>
</tr>
<tr>
<td>C. SELECTING INCREMENTS OF $\mu$</td>
<td>16</td>
</tr>
<tr>
<td>D. ESTIMATING THE ERROR IN $K$</td>
<td>17</td>
</tr>
<tr>
<td>V. COMPUTER REALIZATION OF THE ALGORITHM</td>
<td>19</td>
</tr>
<tr>
<td>A. INPUTS</td>
<td>19</td>
</tr>
<tr>
<td>1. NAMELIST/PARAM/XO, B, STOPK, STOPMU, ND, EPS</td>
<td>19</td>
</tr>
<tr>
<td>2. Data Cards</td>
<td>20</td>
</tr>
<tr>
<td>B. OUTPUTS</td>
<td>20</td>
</tr>
<tr>
<td>1. Printout</td>
<td>20</td>
</tr>
<tr>
<td>2. Disk or Tape</td>
<td>20</td>
</tr>
<tr>
<td>3. Plot</td>
<td>21</td>
</tr>
<tr>
<td>C. EXTERNAL REFERENCES</td>
<td>21</td>
</tr>
<tr>
<td>1. INTMODE</td>
<td>21</td>
</tr>
<tr>
<td>2. IDALIB</td>
<td>22</td>
</tr>
<tr>
<td>3. BESSEL</td>
<td>22</td>
</tr>
<tr>
<td>D. ERROR MESSAGES</td>
<td>22</td>
</tr>
<tr>
<td>E. DETAILS OF JOB EXECUTION</td>
<td>23</td>
</tr>
</tbody>
</table>
I. INTRODUCTION

Generally, the numerical approach to finding internal wave eigenfunctions involves numerical integration of the differential equation over the entire ocean depth. For a deep ocean, particularly when the horizontal wavelengths of interest are much shorter than the ocean depth, numerical integration can be confined to the first few hundred meters of the ocean. This simplification arises from the fact that for a wide range of possible profiles the variation in depth dependence is confined to the upper few hundred meters below the surface, after which the profile may be approximated by a decaying exponential. Moreover, for horizontal wavelengths much shorter than the ocean depth the exponentially decaying profile model may be extended to infinity, i.e., the ocean may be assumed to have infinite depth. Since the solutions of the differential equation for a continuously decreasing exponential profile are known in closed form (they are Bessel functions of the first kind), numerical integration of the differential equation needs to be carried out only over the upper few hundred meters of depth, within which an essentially arbitrary Väisälä frequency profile may be prescribed, as derived from experimental data.

The numerical procedure to be presented here will provide any mode function and dispersion relation over any wavelength range for an arbitrary Väisälä frequency profile, as long as it decays exponentially at great depths. The practical limitation, of course, is the amount of computer time one is willing to expend for results covering an appropriate dynamic range. For ordinary ranges of physical interest, the required computing
time appears to be of the order of minutes for a dispersion curve and seconds for a mode function.

Dr. P. A. Selwyn, who was kind enough to review the first draft of this paper, has called attention to the fact that there already exists a computer program (Refs. 1 and 2) that performs calculations similar to those undertaken here, but for the case in which the ocean depth is finite. One would expect that it might have been possible to modify the earlier program so as to adapt it to the infinite depth case as well.

Although, in a sense, the question of that possibility was moot by the time this information had been disclosed, nevertheless, the option still exists for a potential user who, if convinced that the earlier program has sufficient advantages over the present one, might be willing to risk whatever numerical complications may be involved in such a task. Therefore, it seems proper to discuss briefly some of the major similarities and differences between the two programs. Unfortunately, a direct comparison of their computational accuracies and speeds is not feasible at present.*

Both programs rely on the same general method, which is to start with the appropriate boundary condition at the deep end of the Vaisala frequency profile and numerically integrate the differential equation over depth for selected values of the wave number and phase velocity. These parameters are varied according to some iteration procedure until the value of the solution of the differential equation at the ocean surface is sufficiently close to zero.

The earlier program solves for the phase velocity in terms of the wave number. Here the converse is true.

*If the adaptation of the earlier program were based on the same idea used here to account for the boundary condition at infinity, i.e., matching with the Bessel function solution for an exponential profile, the running time of the program would be increased considerably.
The earlier program uses a fourth-order Runge-Kutta method for solving the differential equation, so that the error is of fifth order in the incremental step size. Here the Numerov-Manning-Millman method is used, and the error is of sixth order in the incremental step size. This gain in efficiency occurs because the present method takes advantage of the fact that the differential equation does not contain a first derivative term.

The earlier program uses a Newton-Rapheson iteration scheme to find the eigenvalue, whereas here the \textit{regula falsi} scheme is used. Generally, the Newton-Rapheson iteration converges faster but may fail to converge if the trial value is not sufficiently close to the true value. However, once started properly the \textit{regula falsi} iteration is guaranteed to converge.\footnote{There are cases in which the convergence is extremely slow, however. When this happens it has been found expedient to switch to internal halving, which is less efficient, in general, but which converges at a predictable rate.}

Other differences between the two programs primarily have to do with how they handle the problem of mode jumping, which occurs in the calculation when dispersion curves for successive modes approach each other too closely. The details are rather involved and it is difficult to assess the relative merits of the two approaches without considerable further study. However, it may be worth noting that the nature of the \textit{regula falsi} method permits the use of certain eigenvalue properties, derived from the classical Sturm-Liouville theory, that aid in resolving the mode-jumping difficulties encountered by the program presented here.

The mathematical statement of the problem to be solved and some properties of the eigenfunctions and dispersion relations needed in the subsequent development are summarized in Chapter II. In Chapter III the general framework for the numerical
solution of the dispersion curves is presented. The detailed
description of the algorithm for the numerical solution of the
differential equation and the associated eigenvalues is given
in Chapter IV. The computer implementation of this algorithm
INTMODE is described in Chapters V and VI. A more detailed
description is given in the Appendix.

Some examples of numerical results obtained with INTMODE
are presented in Chapter VII.
II. DESCRIPTION OF THE PROBLEM AND SOME GENERAL PROPERTIES OF INTEREST

Interest here is confined to internal waves with horizontal wavelengths of at most a few kilometers so that the effects of the inertial frequency do not enter. Accordingly, the mode functions \( \psi_m(y) \) are determined by the differential equation

\[
\frac{d^2 \psi_m}{dy^2} + \kappa^2 \left[ \frac{N^2(y)}{\Omega_m^2} - 1 \right] \psi_m(y) = 0,
\]

along with the boundary conditions

\[
\psi_m(0) = 0, \lim_{y \to +\infty} \psi_m(y) = 0,
\]

where the depth \( y \) is measured from the surface \( y = 0 \). In (1) \( K \) is a given wave number, \( \Omega_m \) is the mode frequency, and \( N(y) \) is the Väisälä frequency profile associated with a vertical thermocline in the ocean.

For a given value of \( K \) the differential equation (1) subject to the boundary conditions (2) determines uniquely the infinite set of mode functions \( \psi_m(y) \). For each mode a dispersion relation is thereby determined; that is, each \( \Omega_m \) is a well defined function of \( K \),

\[
\Omega_m = \Omega_m(K), \; m = 1,2, \ldots
\]

over the interval

\[
0 < K < \infty.
\]
where the set of functions $K_m(y)$ are determined by the differential equation

$$\frac{d^2 y}{dy^2} + \frac{2N(y)}{y} y^m = K y^m$$

subject to the boundary conditions (4). From this point of view, for each value of $\mu$ that is chosen above the minimum value needed for the existence of the particular modes being considered, there are a finite set of real positive eigenvalues $K_m$. As $\mu$ increases, the number of these eigenvalues $K_m$ also increases, but for $\mu$ smaller than some critical value there is a single positive eigenvalue $K_{max}$. According to the general theory of eigenfunctions for the linear second order differential equation (Ref. 4), a set of eigenvalues $K_m$ is determined by the boundary value problem (5) with the boundary conditions (2). From this point of view, for each $\mu$ there is a finite set of real positive eigenvalues $K_m$. As $\mu$ increases, the number of these eigenvalues also increases, but for $\mu$ smaller than some critical value there is a single positive eigenvalue $K_{max}$. The dispersion relations (3) can be expressed parametrically in terms of a quantity $\mu$, which approaches zero as $\mu$ approaches zero, i.e.,

$$\mu = \frac{K_m\left(y\right)}{y^m}$$

The dispersion relations (3) approach the maximum value of $\nu^m$, the dispersion constant $\nu^m$ is equal to the maximum value of $\nu^m$, the dispersion relation curves defined by (3) approach a different finite positive value $\nu^m$. As $K$ approaches zero the slope of each of the dispersion relation curves defined by (3) approaches a different finite positive value $\nu^m$, i.e.,

$$\frac{d\nu^m}{dK} = \frac{K}{\nu^m}$$

Actually, the convention is to regard the quantities $K^2$ as the eigenvalues, which results in their being characterized as negative real.
may be no eigenvalues at all. It is also known from the general theory that each $K_m$ is an increasing function of $\mu$ (cf. Ref. 5, p. 357).

It is assumed that the Väisälä frequency profile $N(y)$ decays exponentially at great depths. This behavior may be characterized more conveniently by actually requiring that after some depth $y_0$, $N(y)$ becomes an exponential function; i.e.,

$$N(y) = \hat{N}_0 e^{-\frac{y}{b}}, \quad y \geq y_0. \quad (6)$$

As observed in Ref. 1 the solution to the boundary value problem associated with (1) and (2) for an exponential profile of the form (6) is proportional to a function $\phi(y)$ given by

$$\phi(y) = J_{K_b}(\mu b\hat{N}_0 e^{-\frac{y}{b}}). \quad (7)$$

Thus, the boundary conditions (2) can be restated as

$$\psi(0) = 0, \quad \psi(y) = A J_{K_b}(\mu b\hat{N}_0 e^{-\frac{y}{b}}), \quad y \geq y_0, \quad (8)$$

where $A$ is a constant that can be chosen arbitrarily or to satisfy some normalization condition.

According to Ref. 3, a W.K.B. solution of (1) can be used to determine the approximate $m^{th}$ mode dispersion relation; i.e., within the W.K.B. approximation the dispersion relation is determined by the equation

$$\int_{N > \Omega} p dy = \pi (m - \frac{1}{4}), \quad (9)$$

where

$$p(y) = \frac{K}{\Omega} \left[ N^2(y) - \Omega^2 \right]^{1/2} \quad (10)$$

$$= \sqrt{\mu^2 N^2(y) - K^2}.$$
III. THE NUMERICAL PROCEDURE TO DETERMINE DISPERSION RELATIONS

In this chapter the principles underlying the numerical procedure to be used for calculating the dispersion relations will be outlined. The details of the procedure, itself, will be presented further on.

Since $K$ is an increasing function of $\mu$, it follows from (9) and (10), by considering the limit as $K$ approaches zero, that the minimum permissible value $\mu_m$ for the $m^{th}$ mode is given approximately by

$$\mu_m \sim \frac{m - \frac{1}{4}}{\int_0^\infty N(y)dy}. \quad (11)$$

Since (11) is only an approximate formula, in order to guarantee that a solution $K_m$ exists for the eigenvalue problem, values of $\mu$ chosen to calculate the dispersion relation for a given mode should be somewhat larger; e.g., it would be prudent to confine the choice of $\mu$ to values such that

$$\mu \geq \frac{3\pi + (m-1)\pi}{\int_0^\infty N(y)dy}. \quad (12)$$

The smallest value of $\mu$ given by (12) ought to be large enough to guarantee the existence of a real $K_m$ but, ideally, small enough to exclude the existence of any higher mode eigenvalue $K_n$, $n > m$. It will be found that this is generally true but that there are some noteworthy exceptions.
For any choice of positive $K$ and $\mu$ it is possible to find a solution of the differential equation (5), subject to the second condition in (8) which determines the necessary initial values to be imposed for that purpose at the point $y_0$. The differential equation can be integrated numerically from $y_0$ down to zero, where a value for $\psi(0)$ will thus be acquired. If the maximum magnitude of the solution over the interval $0 \leq y \leq y_0$ is $|\psi_{\text{max}}|$ then a function $\Psi(\mu,K)$, determined by this process, may be defined by

$$\Psi(\mu,K) = \frac{\psi(0)}{|\psi_{\text{max}}|}.$$  \hfill (13)

As defined, $\Psi(\mu,K)$ is a function of $\mu$ and $K$ alone; i.e., it is independent of the normalization constant $A$.\(^*$

Because there are cases in which (12) does not lead to a satisfactory initial value for $\mu$ (some modes are skipped) a slower but safer procedure than relying on the W.K.B. approximation has been adopted here. The differential equation (5) is solved numerically, subject to the second condition in (8), with $K$ set equal to zero and $\mu$ set equal to a sequence of values $\int_0^N N(y) \, dy$. For each value of $\mu$ in the sequence the sign of $\psi(0)$ is observed. When a change in the sign occurs the corresponding value of $\mu$ is used as a trial value, and it is assumed that zero bounds $K$ from below. If it is found that modes are still skipped, the $\mu$ increment is decreased and the procedure repeated.

If $K^2$ happens to be an eigenvalue, then $\Psi(\mu,K)$ vanishes. Thus, the problem of calculating the dispersion relations is

\(^*$\text{Because it automatically relates error to a specified dynamic range, the normalization (13) is needed for stability of the numerical process used to solve (14).}
equivalent to finding the real (positive) roots \( K(\mu) \) of the transcendental equation

\[
\Psi(\mu, K) = 0
\]  (14)
as the parameter \( \mu \) varies. For sufficiently small \( \mu \) there are no roots, while for \( \mu \) in the interval

\[
\mu_1 < \mu < \mu_2
\]

there is just one root, and for \( \mu \) in the interval

\[
\mu_2 < \mu < \mu_3
\]

there are exactly two roots, etc.

To calculate the roots of (14), interval halving or, for more rapid convergence, the *regula falsi* (Ref. 6) method can be used. The numerical procedure given here does, in fact, rely upon the *regula falsi* method to solve (14), although the technique of interval halving is used in certain circumstances, to be described, in order to reduce computing time.

It is necessary to begin with two trial values for \( K(\mu) \), \( \tilde{K}_1 \) and \( \tilde{K}_2 > \tilde{K}_1 \), such that \( \Psi(\mu, \tilde{K}_1) \) and \( \Psi(\mu, \tilde{K}_2) \) differ in sign, to guarantee that the root \( K(\mu) \) lies between \( \tilde{K}_1 \) and \( \tilde{K}_2 \); i.e.,

\[
\tilde{K}_1 < K(\mu) < \tilde{K}_2 .
\]

Since \( K(\mu) \) must be positive, initially, the trial value \( \tilde{K}_1 \) can be zero. For the initial upper bound \( \tilde{K}_2 \), a quantity defined by

\[
\tilde{K}_2 = N_{max}^{*}
\]  (15)

where \( N_{max} \) is the largest value attained by \( N(y) \), will suffice.

*The fact that \( \tilde{K}_2 \) as defined by (15) is an upper bound can be seen by multiplying (5) by \( N \) and integrating from 0 to \( \infty \). Integration of the derivative term by parts shows that that term is negative.
For the higher modes it will again be necessary to begin with $\tilde{K}_1 = 0$. However, $\tilde{K}_2$ may be set equal to the previously calculated value of $K$ for the mode one step down at the same value of $\mu$. That is, since it is known that

$$K_{m-1}(\mu) > K_m(\mu),$$

in calculating $K_m(\mu)$ a value for $\tilde{K}_2$ given by

$$\tilde{K}_2 \leq K_{m-1}(\mu)$$

(16)

can be used.

As indicated, the trial value $\tilde{K}_2$ should be slightly less than $K_{m-1}(\mu)$ to avoid accidentally falling back onto the $m$-th mode dispersion curve because of normal errors to be expected in the calculation. A test should be included here to guarantee that the choice of $\tilde{K}_2$ is not too much less than $K_{m-1}(\mu)$: the function $\psi(\mu, K)$ must change sign in going from $\tilde{K}_1$ to $\tilde{K}_2$.

In order to obtain a trial value satisfying (16), it is necessary to have an estimate of $K_{m-1}(\mu)$ that is known to be too small. Since it is generally not the case that $K_{m-1}$ would have been calculated previously for exactly the value of $\mu$ now encountered in the mode $m$ calculations, the estimate of $K_{m-1}(\mu)$ must be determined by interpolation, e.g., between values $K_{m-1}(\mu_n)$ and $K_{m-1}(\mu_n + \Delta \mu)$, where

$$\mu_n < \mu < \mu_n + \Delta \mu.$$

However, if the $K$ versus $\mu$ curves are concave upward such an interpolation will produce an estimate that is too large; hence, the desired sign change in $\psi(\mu, K)$ would not occur. On the other hand, the *regula falsi* method may still work even if this requirement is not met, it is not actually guaranteed to converge unless the sign change rule is imposed.
hand, a cruder, one-sided, interpolation that does guarantee
the sign change can be used. That is, instead of interpolating
between \( K_{m-1}(u_n) \) and \( K_{m-1}(u_n + \Delta u) \) the trial value estimate
becomes
\[
\tilde{K}_2 = \frac{K_{m-1}(u_n)}{u_n} \mu. \tag{17}
\]

From the fact that \( \Omega_{m-1} = K_{m-1}(u)/u \) is a monotonic increasing
function it can be readily inferred that \( \tilde{K}_2 \) defined by (17)
will, in fact, be smaller than \( K_{m-1}(u) \).

Once \( K \) has been calculated for the initial choice of \( u \) for
a given mode, the value of \( u \) is increased by adding a small
increment \( \Delta u \). A corresponding increment for the lower bound
trial value \( \tilde{K}_1 \) can be obtained from an interpolation analogous
to (17).

That this can be done so that \( \tilde{K}_1 \) continues to be a lower
bound can be seen as follows. By definition,
\[
K = u \Omega.
\]

Therefore,
\[
\frac{dK}{du} = \Omega + u \frac{d\Omega}{du}. \tag{18}
\]

Although \( \Omega \) increases monotonically with \( u \), it is uniformly bounded
by the maximum Väisälä frequency; hence, the second term on the
right of (18) approaches zero as \( u \) becomes arbitrarily large.
This is evident in view of the fact that, since the derivative
of \( \log \mu \) is \( \frac{1}{u} \), \( \frac{d\Omega}{du} \) approaches zero faster than \( \frac{1}{u} \). Then, for
large enough \( u \), according to (18),
\[
\Delta K \sim \Omega \Delta u. \tag{19}
\]

If it were not, a case of anomalous dispersion would be implied
since, as already observed, \( K \) is an increasing function of \( u \).
Moreover, because $Q$ is a monotonic function of $\mu$, the estimate of $\Delta K$ given by (19) is always too small.

Thus, the new $\tilde{K}_1$ can be chosen in accordance with (19); i.e.,

$$\tilde{K}_1 = K_\mu + \Omega_\mu \Delta \mu = K_\mu + \Delta \mu \frac{K_\mu}{\mu},$$

(20)

where $K_\mu$ is the previously calculated value of $K$ corresponding to the value of $\mu$ before the increment $\Delta \mu$ is added. For modes higher than the first ($m = 1$) the use of (17) to obtain the upper bound $\tilde{K}_2$ continues each time $\mu$ is incremented, while the lower bound $K_1$ is obtained from (20). For the first mode, however, (15) is the only estimate immediately available for the upper bound $\tilde{K}_2$ as $\mu$ is incremented, although (20) can still be used to estimate the lower bound $\tilde{K}_1$. 
IV. DETAILS OF THE NUMERICAL PROCEDURE

A. SOLUTION OF THE DIFFERENTIAL EQUATION

The Numerov-Manning-Millman method (Ref. 6, pp. 204-205) is particularly convenient for solving (5) numerically for given values of \( \omega \) and \( K \). The method requires two starting values; for a step size \( h \), \( \psi(y_0) \) and \( \psi(y_0 + h) \) must be furnished initially. Then the differential equation can be integrated by means of a single recursion relation that involves only \( \psi \) and its second derivative, which is obtained from \( \psi \) and the relationship supplied by the differential equation, itself.

The starting values of \( \psi \) are obtained by recognizing that at \( y_0 \) and \( y_0 + h \) the profile \( N(y) \) is an exponential function of the form (6). Thus, in accordance with (7), at these points \( \psi(y) \) can be set equal to \( J_K(\omega y_0 e^{-\frac{y_0}{b}}) \).

When \( N(y) \) is prescribed numerically over an interval \((0, y_0)\) the resolution of \( N(y) \) implies a limit on how small the step size \( h \) may be taken. Conversely, a natural limitation on how large \( h \) may be is the requirement that it be small compared to the minimum wavelength \( \lambda \) to be considered. Since the wavelength is given by

\[
\lambda = \frac{2\pi}{K},
\]

this means that the size of \( h \) is governed by the largest value to be considered for the wave number \( K \).
B. SOLUTION OF THE EIGENVALUE EQUATION

The eigenvalues $K$ that determine the dispersion relation for each mode are found by solving (14) over an appropriate range of values for $\mu$. For this purpose the *regula falsi* method (Ref. 6, pp. 4-5) seems most effective.

In some cases, convergence of the *regula falsi* method is too slow. Therefore, if twenty iterations occur without satisfying the prescribed error criterion the computer program switches to interval halving with an error criterion applied to $K$ rather than $\psi$.

C. SELECTING INCREMENTS OF $\mu$

In accordance with (12), the increment $\delta\mu$ used to obtain the starting value of $\mu$ in going from the dispersion relation for one mode to that for the next is normally given by adding increments

$$\delta\mu = \frac{1}{\int_0^\infty N(y)\,dy}$$

until $\psi(\mu,0)$ changes sign. At the start of the mode, $K_1$ is then set equal to zero.

If the increment $\Delta\mu$ along a single mode is too large, a jump to the next mode may occur. This can be guarded against by calculating $\psi(\mu + \Delta\mu, K_u)$ which in that case would have a different sign than $\psi(\mu, K_1)$, where $K_1$ is a lower bound used in calculating $K_u$.

Evidently, as a practical matter $\Delta\mu$ must not be too large. It is also true, however, that $\Delta\mu$ must not be too small. While, theoretically, trial values are chosen so as to guarantee the necessary sign change in $\psi(\mu,K)$ for the *regula falsi* method, in practice it turns out that when $\Delta\mu$ is sufficiently small the
sign change may, nevertheless, fail to occur. This is due to
the residual calculation error in the $K$ that corresponds to the
value of $\mu$ before it is incremented. This error is sufficient
in some cases to overcome the theoretical inequality relied
upon in the derivation of the rule for selecting $K_2$.

A compromise rule for selecting the $\mu$ increment is to let
$\Delta \mu$ be about $\frac{1}{10} \delta \mu$. That is, a reasonable choice that seems
adequate in practice is given by

$$\Delta \mu = \frac{1}{2 \int_0^\infty N(y)dy}.$$  

(22)

D. ESTIMATING THE ERROR IN $K$

The test used to determine when to stop the *regula falsi*
iterations in calculating $K$ is the condition

$$|\psi(\mu,K)| < \epsilon.$$  

(23)

The value chosen for this purpose in current applications
is $10^{-7}$, which is intended to provide at least a 60 dB dynamic
range for the corresponding mode functions.

Therefore, the error in $K$ is not given directly; however,
it can be estimated by linear extrapolation. If $\psi_n$ is the value
of $\psi(\mu,K)$ that just meets the test (23) and $\psi_{n-1}$ is the value
of $\psi(\mu,K)$ in the iteration just before that one, then the quan-
tity

$$\frac{\Delta K}{\Delta \psi} = \frac{K_n - K_{n-1}}{\psi_n - \psi_{n-1}}$$  

(24)

where $K_n$ and $K_{n-1}$ are the corresponding estimates of $K$ in the
two iterations, is approximately the rate of change of $K$ with
respect to a change in $\psi(\mu, K)$. Then the error $\epsilon_K$ in $K$ corresponding to $\epsilon$ will be given approximately by

$$
\epsilon_K = \frac{\Delta K}{\Delta \psi} \epsilon.
$$

(25)

The error estimate $\epsilon_K$ can be used to prevent the anomaly mentioned earlier, that too small a choice of $\Delta \mu$ can result in a failure to obtain a sign change in $\psi(\mu, K)$ using the trial value $K_1$ obtained by means of (20). The idea is to make sure that the error in the calculated value of $K$ is always negative, i.e., that the calculated value of $K$ is too small. This can be done by subtracting $\epsilon_K$ after the iterations for $K$ are completed.
V. COMPUTER REALIZATION OF THE ALGORITHM

The computer program DISPER was designed to calculate the $K = K(\mu)$ relationship using the numerical techniques described earlier in this paper. This program will write the $(K,\mu)$ pairs as calculated along each mode to disk or tape and will plot a graph of the $(K,\Omega)$ curves, referred to as dispersion curves.

A. INPUTS

The inputs to the program are of two types: (1) parameters read in under a NAMELIST option, and (2) data points read in from punched data cards.

1. NAMELIST/PARAM/XO, B, STOPK, STOPMU, ND, EPS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XO</td>
<td>Real</td>
<td>The X-coordinate of the last data point of the numerically defined function N(X). For the STD data $X_0 = 220. , m$.</td>
</tr>
<tr>
<td>B</td>
<td>Real</td>
<td>Decay constant For the STD data $B = 1300$.</td>
</tr>
<tr>
<td>STOPK</td>
<td>Real</td>
<td>The maximum K value for which the user wants dispersion curves.</td>
</tr>
<tr>
<td>STOPMU</td>
<td>Integer</td>
<td>The number of dispersion curves to be calculated.</td>
</tr>
<tr>
<td>ND</td>
<td>Integer</td>
<td>The number of data points +1 to be read into the array N.</td>
</tr>
</tbody>
</table>
EPS

Real
The error criteria imposed on the numerical solutions to the differential equations. For the STD data EPS = 1.E-7.

2. Data Cards

ITITLE

Integer
Ten character title of the N(x) values.

N

Real array
Dimensioned 500, read in on punched cards under the format (8F10.5). N contains the equispaced data points that numerically define the function N(x). The points are spaced a distance of XO/(ND-2) apart.

B. OUTPUTS

1. Printout
   a. The parameters defined by the NAMELIST option are listed at the end of the program for verification purposes.
   b. The value of \( \int N(y) \, dy \) is printed next, followed by the values of N.
   c. At the end of the calculations for each mode the number of (K,\( \mu \)) pairs, the maximum estimated error for K, and the complete list of (K,\( \mu \)) pairs for that mode are printed.
   d. Occasionally the error in K cannot be estimated. When that occurs a message indicating this fact and the current values of K and \( \mu \) are printed.

2. Disk or Tape

The values of the NAMELIST/PARAM/B, XO, STOPK, EPS, ND, STOPMU are written on TAPE2 under the format (4E22.7, 2I5//). Next the values of N are written to TAPE2 under the format (8E10.5).
At the end of each mode the number of \((K, l)\) pairs calculated is written to TAPE2, format \((/I5)\). The \((K, u)\) pairs are then written to TAPE2, format \((2E22.7)\).

TAPE2 may be defined as a permanent file by using a catalog control card, or it may be defined as a magnetic tape by using a label control card.

3. **PLOT**

A 10" by 10" graph consisting of the STOPMU different curves of the \((K, \Omega)\) pairs is plotted at the end of the program.

C. **EXTERNAL REFERENCES**

DISPER references several external subroutines that must be provided by the user through control cards that attach the appropriate permanent files.

The necessary routines are listed below under the name of the permanent file on which they reside.

1. **INTMODE**

   **INITIAL**  
   Reads the data values of \(N(X)\), calculates the integral  
   \[
   \int_{0}^{\infty} N(x) \, dx 
   \]

   **GUESS**  
   Calculates "best" estimate of \(K\) given \(u\).

   **DIFF**  
   Numerically solves the differential equation using the Numerov-Manning-Millman method. Called by GUESS.

   **XMUØ**  
   Function to find \(MUØ\) for each mode.

   **OUTPUTK**  
   Writes the \((K, u)\) pairs to TAPE2.

   **PLOTER**  
   Sets up the calls to the CalComp plotting routines.
PScale scales the axes to the calculated data. Called by PLOTER.

Errpro processes detected errors through a call to ABRTJOB. Is called by all of the routines on this permanent file.

2. IDALIB

PLOTS CalComp plotting routines called PLOT by PLOTER.

Plot line daxis symbol number

ABRTJOB error processor that generates TRACEBACK, prints error messages, and terminates the job. Called by ERRPRO.

3. BESSEL

JBESS jairy gamln routines to calculate the BESSEL functions. Acquired from the Argonne National Laboratory.

D. ERROR MESSAGES

We have attempted to anticipate some of the errors a user might encounter when using DISPER under very general conditions. If one of these errors is detected by the program the error processor ERRPRO is called. ERRPRO does three things, (1) prints a brief message describing the error, (2) indicates in which routine the error occurred, and (3) terminates the job without a dump.

A table of error messages generated by DISPER and possible corrective actions that might resolve the problem is presented below.
TABLE 1. ERROR MESSAGES AND POSSIBLE CORRECTIVE ACTIONS

<table>
<thead>
<tr>
<th>Message</th>
<th>Significance</th>
<th>Action</th>
<th>Issued By</th>
</tr>
</thead>
<tbody>
<tr>
<td>END-OF-FILE,UNITS</td>
<td>NO-1 is greater than the number of data points provided for N.</td>
<td>Decrease ND or provide more data.</td>
<td>INITIAL</td>
</tr>
<tr>
<td>INDEFINITE OPERAND</td>
<td>Solution to differential equation is inconsistent.</td>
<td>Look for coding errors in the routine DIFF.</td>
<td>DIFF</td>
</tr>
<tr>
<td>DIVISION BY ZERO</td>
<td>Algorithm for solving the differential equation has broken down.</td>
<td>Data points may be too far apart. Introduce more data, perhaps through interpolation.</td>
<td>DIFF</td>
</tr>
<tr>
<td>FIRST MODE CANNOT BE CONSTRUCTED</td>
<td>Cannot find MU0 for this mode.</td>
<td>Reexamine data.</td>
<td>IMU0</td>
</tr>
<tr>
<td>TOO MANY POINTS</td>
<td>More than 500 (K,u) pairs are need to construct this mode.</td>
<td>Decrease STOPK, or revise program by redimensioning SAVEK, SAVEMU, TEMPK, and TEMPMU.</td>
<td>DISPER</td>
</tr>
<tr>
<td>KI and K0 ON SAME SIDE OF CURVE</td>
<td>Estimates for K are not upper and lower bounds.</td>
<td>(1) Reduce EPS. (2) Reduce STOPMU or Increase STOPK.</td>
<td>GUESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1) EPS too severe. (2) Extrapolating initial estimate of KI beyond K values of the preceding mode.</td>
<td></td>
</tr>
</tbody>
</table>

E. DETAILS OF JOB EXECUTION

The following is a sample card deck for executing DISPER.

1.

```
1PU,1,T1,DATE=1-378
1AHEL,PLOT=MANG,
1ADMC,FPLOT=DELX*X+Y
1AETJATTACH(RECXL,INPA)
1ATTLE(AIPT,INPA)
1ATTLE(INAEPK,MINA)
1AMAP(OFF,LF)
1ALOOKNT(DISPE,INSE,EL,INTPDE)
1EXE
789
1SPRAM N=111, 4=5, NO=117, STP=1, STSU=6, E=7, T=11
1PLSF
```

```
This will produce a listing of the \((K,\mu)\) pairs for each mode and a plot of the dispersion curves. It will not create a tape or permanent file of the \((K,\mu)\) pairs.

If the user wishes a permanent record of the \((K,\mu)\) pairs additional control cards must be included.

For a permanent file the deck would contain two additional cards. The sample deck below is an example.

```
IPUTOMT  NAPAPER.3374A
REQUEST(TAPE)*HF
LABEL(PILOTAP0, THIS IS A SAMPLE LABEL)  NODE0REPL0T, O=LO, X=U, Y=V, SHAPED
ATTACH(REPLOT, M=1, S=P0)
ATTACH(TAPE, N=CG)
ATTACH(NRDIM, ID=PR, MR=1)
MAP (OFF)
LOAD (DIMPATH)
LDSET (LCHAR INH/REFSEL/INTOFF)
EXECUTE
CATALOG (TAPE?UI5RCURVE=INPR)
EXECUTE
SHAPE N=1111, R=5, ND=1970, S=1, T=SUPW=27, F=7
PULSE
```

If the \((K,\mu)\) pairs are to be written to a magnetic tape the job stream might look like the example shown below:
Note the change in the job card as well as the additional control cards.

Should both a magnetic tape and a permanent file be desired the job stream would look like the next example.
VI. PROGRAM MODE

The computer program MCDE was designed to calculate and plot the normalized mode functions $\Psi(u,K)$. Given (1) the dispersion curves created by DISPER, (2) a set of consecutive mode numbers, and (3) a value for K, this program will use the dispersion curves and linear interpolation to find the corresponding $u$ values. It will then calculate and plot the mode function for each mode number.

A. INPUTS

The inputs to this program are of two types: (1) the outputs of DISPER, and (2) a NAMELIST option.

1. TAPE5

TAPE 5 is defined to be the disk file or magnetic tape produced by DISPER.

- **B, X0, STOPK, EPS, ND, STOPMU**
  The first record on TAPE5 is the defining parameters used by DISPER to create the dispersion curves. They are read in under the format (4F22.7, 2I5//). For definitions see inputs to DISPER.

- **N**
  Empirical data, format (8F10.5). See Inputs to DISPER. There will be ND-1 values of N.

- **NPTS**
  Integer
  The number of (K,u) pairs for the current mode.
  Format (/I5).
XI, MUL
Real
The K and M values of each mode, (2E22.7).

STOPX
Real
The depth to which the mode function is to be calculated. Must be greater than or equal to X0.

IFIRST
Integer
The first mode to be calculated.

LAST
Integer
The last mode to be calculated. All modes between IFIRST and LAST are calculated.

B. OUTPUTS

1. Printouts
   a. The parameters defined by TAPE5 and the NAMELIST option are listed at the end of the program for verification purposes.
   b. N is listed.
   c. Mode number is printed followed by a list of PSI values for that mode. Format is (4E22.7).

2. Plots
   10" by 10" graphs of the (X, PSI) values will be plotted, one plot for each mode.

C. EXTERNAL REFERENCES

MODE references several external subroutines that must be provided by the user through control cards that attach the appropriate permanent files.

The necessary routines are listed below under the name of the permanent file on which they reside.
1. INTMODE

INITIAL

Reads the data values of \( N(x) \), calculates the integral

\[
\int_0^\infty N(x) \, dx
\]

DIFF

Numerically solves the differential equation using the Numerov-Manning-Millman method.

PSCALE

Scales the axes to the calculated data. Called by PLOTMODE.

ERRPRO

Processes detected errors through a call to ABRTJOB. Is called by all of the routines on this permanent file.

INTNP

Calculates the integral

\[
\int_0^\infty N^2(x) \psi^2(x) \, dx
\]

FUNCT2

Calculates \( X \times J(v,X)^2 \). Called by INTNP.

FUNCT4

Calculates the alternative asymptotic approximation for \( X \times J(v,X)^2 \). Called by INTNP.

PLOTMOD

Sets up the calls to the CalComp plotting routines.

2. IDALIB

PLOTS

CalComp plotting routines called by PLOTMOD.

PLOT

LINE

DAXIS

SYMBOL

NUMBER
A3FTJOB Error processor that generates TRACEBACK, prints error messages, and terminates the job. Called by ERRPRO.

GAUSS Numerical integrating routine using Gaussian quadrature. Called by INTNP.

3. BESSEL

JBESS JAIRY GAMLN Routines to calculate the BESSEL functions. Acquired from the Argonne National Laboratory.

D. ERROR MESSAGES

A table of error messages generated by MODE and possible corrective actions that might resolve the problem is presented below.

TABLE 2. ERROR MESSAGES GENERATED BY MODE AND POSSIBLE CORRECTIVE ACTIONS

<table>
<thead>
<tr>
<th>Message</th>
<th>Significance</th>
<th>Action</th>
<th>Issued By</th>
</tr>
</thead>
<tbody>
<tr>
<td>More than 500 pairs were needed for this mode.</td>
<td>DISPER was altered to permit more than 500 pairs to be calculated.</td>
<td>Make similar changes in MODE.</td>
<td>MODE</td>
</tr>
<tr>
<td>The maximum K value on TAPES is less than the K of interest.</td>
<td>The dispersion curves were not calculated to this value of K.</td>
<td>Rerun DISPER with STOPK greater than K, or reduce value of K.</td>
<td>MODE</td>
</tr>
<tr>
<td>There are not enough modes on TAPES.</td>
<td>STOPMU is less than LAST.</td>
<td>Reduce LAST to less than STOPMU, or rerun DISPER with STOPMU greater than LAST.</td>
<td>MODE</td>
</tr>
<tr>
<td>Indefinite operand 0/0</td>
<td>Solution to differential equation is inconsistent.</td>
<td>Look for coding errors in the routine DIFF.</td>
<td>DIFF</td>
</tr>
<tr>
<td>Division by Zero</td>
<td>Algorithm for solving the differential equation has broken down.</td>
<td>Data points may be too far apart. Introduce more data, perhaps through interpolation.</td>
<td>DIFF</td>
</tr>
</tbody>
</table>

E. DETAILS OF JOB EXECUTION

The following is a sample card deck for executing MODE, when the dispersion curves are on a permanent file.
Next is a sample card deck for executing MODE when the dispersion curves are on a magnetic tape. Note the VSN number should be the one assigned at the time DISPER executed.
VII. EXAMPLES OF DISPERSION RELATIONS AND EIGENFUNCTIONS OBTAINED WITH INTMODE

The two Väisälä frequency profiles considered in the sample calculations employing INTMODE are shown in Fig. 1. The profile labeled "exponentially stratified ocean" corresponds to a deep ocean without a thermocline and \( N(y) = 0.00528 \exp(-y/1300) \) radian/sec, where \( y \) is in meters. This profile is identical to the one used by Garrett and Munk (Ref. 7). For the exponential Väisälä frequency profile, the mode functions are Bessel functions. Consequently, results of INTMODE for this profile can be compared with results based on analytical formulae, thus providing a check on the accuracy of the numerical technique.

The sharp thermocline, labeled "STD data set", is taken from (Ref. 8) and is based on measured towed thermistor chain data in the tropical Pacific Ocean. The data extends to a depth of 220 meters; at greater depths an exponential profile with a decay constant of 1300 meters is assumed.

The dispersion curves for the first 25 modes, corresponding to the STD data set, are plotted in Fig. 2. The angular frequency is in radians/sec. For the STD data set, plots of the first four internal wave modes are shown in Fig. 3 for \( K = 0.01 \) radians/meter (\( \lambda \approx 628m \)) and in Fig. 4 for \( K = 0.02 \) radians/meter (\( \lambda \approx 328m \)). The mode functions are all normalized in accordance with

\[
\int_0^\infty \psi^2(y) N^2(y) dy = 1.
\]

Since \( N(y) \) decays exponentially with depth, this normalization constraint leads to a progressive increase of the mode maximum with mode number and depth, a feature corroborated by the plots in Figs. 3 and 4.
INTMODE is capable of yielding mode functions of any order (subject to course to the resolution of the data with respect to depth. Figure 5 shows a plot of the 25th mode.

The dispersion relations for the exponential profile are shown in Fig. 6; the first four mode functions are shown in Fig. 7 and Fig. 8, for \( K = .01 \) and \( K = .02 \) radians/meter, respectively.

Examples of dispersion relations for other profiles are shown in Figs. 9 and 10. The corresponding profiles are, respectively, those referred to in Fig. 12 as STD data and NRL data. An additional example of dispersion relations, corresponding to a pulse shaped profile, is shown in Fig. 11. This profile is of the form

\[
N(y) = \begin{cases} 
0 & 0 < y < 65.45 \text{m}, \\
0.0105 \text{ rps} & 65.45 \text{m} < y < 111 \text{m} \\
0 & y > 111 \text{m}.
\end{cases}
\]

In the region \( y < 111 \text{m} \) \( N(y) \) was approximated by the rapidly decaying exponential \( \exp(-y/5) \).
FIGURE 1. Väisälä frequency profiles used in the calculation of internal wave mode functions and dispersion relations.
FIGURE 2. Internal wave mode dispersion curves for the Väisälä frequency profile corresponding to the "STD data set" in Fig. 1.
FIGURE 3. The first four internal wave modes for the STD data set (Fig. 1) ($\lambda \approx 628$ meters).
FIGURE 4. The first four internal wave modes for the STD data set (Fig. 1) ($\lambda \approx 314$ meters).
FIGURE 5. The 25th mode for the STD data set ($\lambda \approx 628$ meters).
Figure 6. Internal wave mode dispersion curves for an exponentially decreasing Väisälä frequency profile.
FIGURE 7. The first four internal wave modes for the exponentially decreasing Väisälä frequency profile ($\lambda \approx 628$ meters).
FIGURE 8. The first four internal wave modes for the exponentially decreasing Väisälä frequency profile ($\lambda \approx 314$ meters).
FIGURE 9. Internal wave dispersion relations -- STD data.
FIGURE 10. Internal wave dispersion relations -- NRL data.
FIGURE 11. Internal wave dispersion relations -- pulse.
FIGURE 12. Väisälä Frequency Profiles.
REFERENCES


# APPENDIX A

## INTMODE SUBROUTINES

### Contents

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPER</td>
<td>A-3</td>
</tr>
<tr>
<td>XMUØ</td>
<td>A-10</td>
</tr>
<tr>
<td>GUESS</td>
<td>A-11</td>
</tr>
<tr>
<td>MODE</td>
<td>A-14</td>
</tr>
<tr>
<td>DIFF</td>
<td>A-18</td>
</tr>
<tr>
<td>INTNP</td>
<td>A-19</td>
</tr>
<tr>
<td>FUNCT 2</td>
<td>A-20</td>
</tr>
<tr>
<td>FUNCT 4</td>
<td>A-21</td>
</tr>
<tr>
<td>INITIAL</td>
<td>A-22</td>
</tr>
<tr>
<td>OUTPUTK</td>
<td>A-23</td>
</tr>
<tr>
<td>ERRPRO</td>
<td>A-24</td>
</tr>
<tr>
<td>PLOTMOD</td>
<td>A-25</td>
</tr>
<tr>
<td>PLOTER</td>
<td>A-26</td>
</tr>
<tr>
<td>PSCALE</td>
<td>A-27</td>
</tr>
</tbody>
</table>

A-1
A MCC 6400 SUBROUTINE

PROGRAM DIFFERENTIAL OUTPUT,TAPE3,TAPE4

*CHECK DIFFERENTIAL OUTPUT,TAPE3,TAPE4

PROGRAM DIFFERENTIAL OUTPUT,TAPE3,TAPE4

10 INSTITUTE FOR DEFENSE ANALYSIS
WASHINGTON, DC

20 MARCH 5, 1960

25 DIFER CALCULATES AND GRAPHS DISPER-
SION CLUES ACCORDING TO NUMERICAL
TECHNIQUES DEVELOPED BY ONE IS HEAT
AND W. W. ASYLTIN OF IDA

30 PROBLEM: GIVEN A FUNCTION N(X) DEFINED
NUMERICALLY FROM X = 0 TO X = XA AND
BY THE RELATION

N(X) = NO * EXP(-X / XA)

FOR X ≠ XA AND A DIFFERENTIAL

35 EQUATION:

36

37 D02PST1/N(X) 0 =

38 (K*2 - MU*2 - N(X)*2) * N(X)

39 WITH PARAMETERS MU AND K THAT TAKE
ON VARIOUS REAL VALUES

40 IT IS KNOWN THAT IN THE REGION WHERE
X ≤ XA THAT ONE SOLUTION OF THE

45 DIFFERENTIAL EQUATION IS THE RESSEL
FUNCTION WITH TUNER K = B AND ARGU-
MENT X = B / B

50 BY VARYING THE PARAMETERS MU AND K
ONE CAN FIND DIFFERENTIAL EQUATIONS, ONE OF
WHOSE SOLUTIONS MATCH THE RESSEL
FUNCTION SOLUTION WHEN X ≥ XA AND
ALSO VANISH AT X = B

55 THIS PROGRAM DETERMINES THE RELATION
K = K(MU)
THAT PROVIDES JUST SUCH DIFFERENTIAL

60 EQUATIONS

IT THEN GRAPHS (K / MU) VS. K
DESCRIPTION OF ARGUMENTS

The following parameters are members of the NAMELIST GROUP:

**CALLER PARAM**

- **X0**: REAL
  - The x-coordinate of the last point of the numerical data.
- **B**: REAL
  - The decay constant.
- **NN**: INTEGER
  - The number of numerically defined points - 1.
- **STOPK**: REAL
  - The maximum k value to be calculated for each hour.
- **STOPH**: INTEGER
  - The number of modes to be calculated.
- **EPS**: REAL
  - Error criteria imposed on the solution to the differential equation.

The last inputs to the program are the numerical values that define N(k) from 0 to 20, they are read from cards. The values must be punched as to be read under the FORMAT(153, k).

**N**

REAL ARRAY

**NIMEROON 800**

COMMON/INPUTS/R,NO,XA,END,ENDW,ENDK,ENDS

COMMON PST(580),N(EDS),NOPT3X,INT1,TITLE

COMMON/OUTS/SAUK(500),SAFNU(500)

DIMENSION TEMP(500),TEMPW(EDS)

REAL NO,BA,MU,KNO,KMAX,NINS,K1

INTEGER STOPW

NAMELIST/PARAM/NO,NO,NINS,ENDW,ENDK,ENDS

H0DF = 1

FORMAX = 10

PT = ACOS(1.,)

TFID = 6
PROGRAM NISPEC

115 C READ PARAMETERS
C 5 READ PARAM.
C PRINT PARAM.
120 C READ NUMERICAL VALUES FOR N AND INTEGRATE N
C CALL INITIAL(SUM1)
C PRINT 7001, SUM1
125 C FORMAT(* SIMP. F22.7)
C PRINT 100, IN(1,1,1,1,1)
C WRITE NAMELIST AND N TO TAPE?
130 C CALL OUTPUT (1),
C FIND MAXIMUM VALUE FOR N
135 C KMAX=N(1),
C DO 10 i=2,10
10 KMAX=MAX(KMAX,N(1))
C CALCULATE MUI FOR THE FIRST BRANCH
C KY = 1/SUM1
C MUI = AMUI(0,1,1)
C CALCULATE INITIAL ESTIMATE FOR THE INCREMENT IN MU
C ALONG A MUI BRANCH
145 C TNEVT = .1/ SUM1
C TNEVT = TNEVT
C START THE PROCESS OF APPROXIMATING K
150 C K1 = 0
C MUI=1
C K1=0
200 CONTINUE
155 C CALCULATE THE UPPER ESTIMATE OF K1
C K1 = MUI * KMAX
160 C CALL SUBROUTINE GUESS TO IMPROVE THE ESTIMATE K1
C K1 IS UPPER ESTIMATE. K1 IS LOWER ESTIMATE. THE
C IMPROVED ESTIMATE IS RETURNED IN K1.
165 C CALL GUESS(KA,K1,MU1)
170 C CHECK FOR ERROR CODES RETURNED IN KT

A-5
PROGRAM DISPEN

C          NT = 9        INDICATES THE ALGORITHM COULD NOT PRO-
C            DUCIZE A K FOR WHO ON THE FINAL BRANCH.
C          K1 IS NOT ABOVE THE CURVE.

175
C
IF(NT NEQ. 9)CALL FRPRN(5)

180
C
THE MAXIMUM ESTIMATED ERROR FOR K IS
C
RETURNED BY QUERK IN THE VARIABLE K0

185
C
FPK = K0
FREMAX = AMAX1(FPK,FREMAX)

190
C
INCREASE KOUNT. IF 500 (MU) PAIRS HAVE
C
BEEN CALCULATED AND KOUNT NOT REACHED AN
C
ERROR MESSAGE IS PRINTED, A FIN ERROR 92
C
APPEARED. AND THE JOB ABORTED EITHER BE-
C
CAUSE STOPK OR INCREASE THE DIMENSION OF
C
SAVE(500),SAVEH(500),TEMPK(500),TEMPU(RUN).
C
THEN CHANGE THE TEST ON KOUNT
C
KOUNT = KOUNT + 1
195
IF(KOUNT GT 500)CALL FRPRN(IER)
SAVE(KOUNT) = U1
SAVE(1,KOUNT) = MU
C
CALCULATE THE LOWER K ESTIMATE FOR THE MAX
C
MU VALUE
C
INCREASE MU
C
K1 = K1 + TDELT 1 K1 / MU
200
IF(K1 GT STOPK)GO TO 700
MU = MU + TDELT
205
96 FORMAT(1X,T3.2,2X,F2.15)
TO 200
C
WE ARE FINISHED WITH THIS BRANCH

210
C
CONTINUE
C
KOUNT
C
SAVE CURRENT (MU,K) PAIRS TO USE IN ESTIMATING
C
K1 ALONG THE NEXT MORE
C
NV 500 IM=KOUNT
TPPK(1)=SAVEK(1)
215
GO TO 700
C
TPPMU(I)=SAVEH(I)
C
CALL PUTPMU(KMNT)
C
PRINT THE MAXIMUM ERROR ESTIMATED FOR K FOR

A-6
PROGRAM DISPFF

C THIS MODE
C PRINT 9A, MODE, FKCK
9A FORMAT(* ESTIMATED MAXIMUM ERROR IN K FOR MODE *), IA, IB, IC, 
1F9.2, 7)
C BEGIN THE NEXT BRANCH
MODE = MODE + 1
IF (MODE < 1, STOP) GO TO A90
100 CONTINUE
LT = 0
C
210 CONTINUE
C FIRST APPROXIMATION OF K FOR THIS BRANCH
C BEGIN BY FINDING THE K FOR THE LAST BRANCH
250 IF (LT = 0) NPTS = TEMP/111 + 1, MU = MU + 10
260 CONTINUE
I = NPTS + 1
30 IF (I = 1) TEMP = TEMP/I/TEMP/111/1
C CALCULATE THE UPPER ESTIMATE OF K, K1
C
280 K1 = SLOPE * MU
C CALL SUBROUTINE GUESS TO IMPROVE THE ESTIMATE K
C K1 IS UPPER ESTIMATE, K2 IS LOWER ESTIMATE, THE
C IMPROVED ESTIMATE IS RETURNED IN K1.
C CALL GUESS, KA, K1, MUS
C CHECK FOR ERROR MONE RETURNED IN NT
C
270 NT = 0
A BRANCH WE ANTICIPATED BUT NOT 
APPROVED. HERE, A RESULT OF OUR ESTI-
MATION PROCEDURE FOR NU.
275 PRINT 6, EQ, 910C TO 1A00
C 221 CONTINUE
C
C THE MAXIMUM ESTIMATED ERROR FOR K IS
C RETURNED BY NUSE IN THE VARIABLE KA
C
A-7
PROGRAM NISPEN

C
FORM = KO
FORMA = AMAXJ(FPSK+ERRMAX)

200
C
C INCREMENT KOUNT, IF 500 (MU, K) PAIRS HAVE
C BEEN CALCULATED AND STORR NOT REACHED AS
C ERROR MESSAGE TO PRINTING A FIN ERROR 59
C CACERATED AND THE JNA HUNTID, EITHER RE-
C NUCE STORR OR INCREASE THE DIMENSION OF
C SAVEK(59).SAVEA(59).TERPK(59).TERNUI.8UN.
C KOUNTKOUNT + 1
C IF(KOUNT <E, R, R)AIL FPRPK(IPI)
C SAVEK(KOUNT) = KI
C SAVEA(KOUNT) = WI
C C CALCULATE THE LOWER X ESTIMATE FOR THE NEXT
C MU VALUE
C C INCREMENT MU
C KI = KI * TNEKT * KI / WI
C IF(KI <E, TSTUP) NO TO 300
C WI = WI * TNEKT
C NO TO 401
C C
C 400 CONTINUE
C LWL = SAVEMU(KOUNT)
C ENDFILE 2
C FITIND 2
C CALL PLOTR(INTR,1,KMAX)
C 330
C C POSITION THE FILE TO THE BEGINNING OF
C THE FIRST LINE
C READ(2,95,R,IC,STOPK,FAKE,NN,STOPM)
C 95 FORMAT(A18,7,F12.1/
C PRINT PARAM
C IF(M = NN = 1) PRINT(2194) (W(1) / 1. INTP)
C 94 FORMAT(4F12.1)
C 325
C C WE HAD TO CALCULATE OMEGA FOR EACH MU AND
C PLOT OMEGA VS K
C C WE ARAY SAVE311 TO HOLD THE OMEGA VALUES
C C
C READ(2) 99, INTR
C 99 FORMAT(1050 I = 1; 105)
C 100 FORMAT(2E7.1)
C 100 FORMAT(2E7.1)
C 360
A-8
PROGRAM ISPF(F)

101 FORMAT(IX,2PE22.7)
400 CONTINUE
   CALL PLOT(FR(NPTE.2(1),KMAX))
400 CONTINUE
   CALL PLOT(FR(NPTE.3(1),FORMAX))
   CALL ENDFRNT
C
C   JOB IS COMPLETE
C
C     STOP
C     END
FUNCTION XMUL

1
FUNCTION XMUL(Y, NY)
COMMON PST((500), X(504))
REAL MU
MU = Y + NY
CALL DIFF(MU, 0, PSTMAX)
KPLNT = 0
1000 CONTINUE
TEST = PST(1)
KPLNT = KPLNT + 1
IF (KPLNT .LT. 10) CALL FORPART(7)
MU = MU + NY
CALL DIFF(MU, 0, PSTMAX)
IF (TEST .EQ. PST(11), AT. 0.) GO TO 1000
KPLNT = MU
15 RETURN
END
SUBROUTINE GUES

*CHECK GUES
COMMON INDIVIDE, NMAX,MATCH,STOPI,STORP,STORP,STW
COMMON PSI, K00, N(K00) W0DIK1, N0DIK1, N0DIK1
REAL N0DIK1, N0DIK1, K1, K2
ERROR = 2
THE MAXIMUM ESTIMATE FOR RUN RUN K 15
RETURNED BY GUES IN THE VARIATION K0
IF(K0 = EQ, 0, 1) EPSK = 2

SOLVE DIFF. EQUATION WITH N0DIK1, K0
IF K0 CLOSE ENOUGH, RETURN WITH K1 = K0

ATTEMPT TO FIND AN UPPER BOUND THAT IS
CLOSE TO K0. THIS IS MORE EFFICIENT FOR
THOSE CASES WHERE K0 IS A MUCH CLOSER ESIT-
IMATE THAN K1.

K0UT = N
CALL DIFF(N0DIK1, K0, PRIMAX)
NP(EPSK, PSI(1), PRMAX, IF, EPSK) GO TO 10
K1 = K0

NC ESTIMATE OF THE ERROR IN K POSSIBLE

PRINT 98, K0, N0DIK1
9A FORMAT(NO ERROR ESTIMATE IN K POSSIBLE HERE) / * K0 = 0; F22.7*
1 512 * MU = 0; E22.7*
K0 = EPSK
RETURN
10 CONTINUE

TERMX = AR(SI(1) / PRIMAX)
TERMX = K0
TERM = PSI(1)
STEP1 = (K1 = K1) / 10
K1 = K0
KP = K0 = 1. 101
K1 = K1 + STEP1
CALL DIFF(N0DIK1, K1, PRMAX)
TERM = AR(SI(1) / PRMAX)
IF(TERM, NF, EPSK) NO GO 20
TERM = AR(K1) + TERMX + TERMX = TERMX + EPSK
IF(TERM, NF, EPSK) TERMX = TERMX
K0 = EPSK
K1 = K1 - TERMX
RETURN
20 CONTINUE
TERMX = K1
TERMX = TERMX

A-12
SUBROUTINE GUESST

IF (PSI(1)) * TEST .LT. 0.) GO TO 201

200 CONTINUE

C C

IF (EQA, NO SIGN CHANGE
PRINT 999, M1, K0, PSI(1)
C)
99 FORMAT (1X, 1PE15.7)

90 CONTINUE

IF (K0 .NE. 0.) GO TO 250
AT = 0
RETURN

C C

NOW USE THE REGULA FALSI METHOD

75 CONTINUE

K0 = 0
NCHECO = 0
NCHECI = 0

200 K0 = K0 + TEST * (K1 - K2) / (TEST - PSI(1))

K0 = K0 + 1
IF (K0 .GE. 200) GO TO 240
SAVE = PSI(1)
CALL DIFF(M1, K0, PSI(MAX))

TERM = ABS(Psi1(1)) / PSI(MAX)

IF (TERM .LT. EPS) GO TO 21A
TERM = ABS(K0 - TEMPK) / (TERM - TEMBMAX) * EPS
TERM = TERM * 20. FPAK = TERM
K0 = EPSK
K1 = K2 = TERM
RETURN

21A CONTINUE

TPPPK = K0
TPPPMAX = TERM
IF (TEST = PSI(1)) 210, 220, 220

220 CONTINUE

226 CONTINUE

K0 = K2
TEST = PSI(1)
PSI(1) = SAVE
GO TO 200

230 CONTINUE

236 CONTINUE

K2 = K0
GO TO 200

240 CONTINUE

SAVE = PSI(1)

C C

IF (TERVAL = HAVLINA

110 CONTINUE

K0 = 0
285 CONTINUE

K0 = K0 + 1
IF (K0 .GE. 200) GO TO 400
SUBROUTINE GUESS

115          TEPK = K2
            TEPMK = TERM
            K2 = (K1 + K9)/3.
            CALL DIFF(4)(K9, DBTMAX)
            TERM = ABS(TEP1(1)/PSI MAX)

120          IF(TERM .LT. EPSILON TO 100
            TERM) = ABS(K2 - TEPMK) / K2
            IF(TERM1 .LT. 1.E-6) ON TO 380
            IF(TEST .LT. PSI1(1) .LT. 0) ON TO 260

125          C
            K9 AND K2 ARE ON THE SAME STRES
            C
            K9 = K2
            PSF = PSI1(1)
            ON TO 260

130          CONTINUE

135          C
            K9 AND K2 ARE ON OPPOSITE STRES
            C
            K1 = K2
            SAVE = PSI1(1)
            ON TO 260

290          CONTINUE

300          FINISH PROCEEDING

140          C
            KT = 99   INDICATES OVER 50 ITERATIONS
            HAVE OCCURRED BUT NO SATISFACTORY
            ESTIMATE OF K REACHED

145          C
            KT = 99
            RETURN
            250 CONTINUE

150          C
            NO SIGN CHANGE HAS OCCURRED. EITHER
            K9 IS TOO LARGE OR K1 IS TOO SMALL.
            THEY LIE ON THE SAME SIDE OF THE
            CURVE

155          C
            CALL ERAPP(4)

300          CONTINUE

310          NORMAL RETURN

160          C
            TEP1 = ARS((K2 = TEPK) / TEP1 - TEPMK) . EPS
            TEP1(4) = TEP1(4) . EPSK = TEP1
            K9 = EPSK
            K1 = K2 - TEP1

165          RETURN
            END
A FORTRAN PROGRAM

PROGRAMER
P. I. NAPRED
INSTITUTE FOR DEFENSE ANALYSIS
ARMS CONTROL, VA
MARCH 26, 1970

MORE CALCULATES AND GRADES MODE
FUNCTIONS FOR DISPERSION CURVES
CREATED BY COMPATRON PROGRAM

DIFFERENTIAL EQUATION

GIVEN

THE DIFFERENTIAL EQUATION

\[ \frac{d^2 \psi}{dx^2} + \left( \frac{r}{x} \right) \psi = 0 \]

FOR \( x \geq 0 \)

(2) NUMERICAL DATA

FOR \( x \leq 0 \)

(3) BOUNDARY CONDITION

(4) \( \psi(0) = A \)

(5) M transform the solution to

THE DIFFERENTIAL EQUATION

C

INPUTS

NO INTERFER NUMBER OF DATA POINTS
R REAL DECAY CONSTANT
Xa REAL LAST X CUMMUTATE OF
C FOR REAL PROGRAM CRITERIA FOR
C K REAL THE VALUE OF K OF INTEREST
C STUDY REAL HOW FAR WILL THE K -
C ARRAY THAT HOLDS
C (6) DIMENSIONED DATA
C FIRST INTERFER FIRST INDEX OF INTEREST
C LAST INTERFER LAST INDEX OF INTEREST

A-14
COMMON/INPUT, N, N!, X, EPS, STORM, STOP, STUDY

REAL K, MIN, X, X!, EPS, FIRST, LAST, K, STOP

READ (5,9A) R, X, STEM, EPS, N, STORM

READ INPUT

K, STOP, FIRST, LAST

READ INPUT

PRINT NAMLIST, INPUT, FOR VERIFICATION THEN PRINT N(X)

PRINT INPUT

IF (K.GT. STEM) CALL ERPHON, TEMP
IF (LAST .GT. STORM) CALL ERPHON (FRA)

CALL INITIAL TO READ THE VALUES
OF N(X) OFF TAPE 5
GET N(X) DATA AND CALCULATE DELTAX

CALL INITIAL (SW1)

PRINT 101, N(1), N(2) AND
PRINT 99

TOP = N = 1

DELTAU = DT / SW1
IF (FIRST) THEN 150 TO 505

ADVANCE POINTED TO THE
FIRST MODE OF INTEREST, IF FIRST

TOP = FIRST = 1

READ 1 = 1, TO TOP
PROGRAM NODE

115  DEFAN(5,99)NPTS
120  CALL PLOT(100,10)
125  FOR EACH OF THE NODES
130  DEFAN(5,99)NPTS
135  DEFAN(EACH (K, MEGA) PAIR TO FIND K INTERVALS
140  DO LINEAR INTERPOLATION
145  SOLVE DIFFERENTIAL EQUATION
150  CALL DIFF(MUI, K, PMAX)
155  CALL INTPN(K* MUI, SW3)
160  ERROR CONDITION
165  EQU IN THE REST OF PSI
170  NORMALIZE

A-16
PROGRAM HIDE

C E7A L = I, NUM
C E7A (L) = E7A (L) / SQRT(E7A (M))
C E7A CONTINUE

175 C
C E7A NO PLOTTING AND OUTPUTTING WHERE
C E7A PRINT 48, I
C E7A PRINT 101, (PSI(L) + TI - 1 + NUM)
C E7A CALL PLOTHOPE (NUM - 1, X, X / MUL, & USRN )
C E7A START = J + 1
C E7A INVANCE POINTER TO NEXT MUNF

185 C
C E7A NC 68 J = START, NOTS
C E7A READ(S* 10010)UNUMY
C E7A CONTINUE
C E7A 516 CONTINUE
C E7A CALL ENPLOT
C E7A 517 FORMAT (X, 4F27.7)
C E7A 518 FORMAT (2E27.7)
C E7A 519 FORMAT (//, 15)
C E7A 915 FORMAT (1H1, * MUNF NUMBER, *13)
C E7A 916 FORMAT (4E15.7, 2, 15/)
C E7A END

A-17
SUBROUTINE NIFF

*CHECK NIFF
COMMON/INIT/),NDM,IN,map,STOPM,STOPK,STUP

COMMON PSI(RA),N(NN+1),NDP,NDTAK,NDTITL
REAL N,JM,JK,KN

CC

THE IS THE NUMEROU - MANNUM = MILLMAN PRINH1U
FOR SOLVING SECOND ORDER LINEAR DIFFERENTIAL
ECATIONS

CC
F1(x)x=0 = Nlua = V=A
PSI(1) = FIX
Y=0
STOP = NO

CC
DEFINE FIRST TWO VALUES OF PSI

TEMP1=Ker
TEMP2=MUS

CALL JUSE (TEM01,TEM02,XPRES-METAK/RT)*PSI(1)
CALL JUSE (TEM01,TEM02,PSI(2))
TEAMDELTAX/2+i
PEXMA = ARG(PSI(2))

CC
SOLVE THE EQUATION BACKWARD

DO JON I=2,ISTOP

YK1M = PSI(1) = PSI(I-1)

XYLI = NUM*PSI(I-1)/YK1M + (NN-I+1)+YK1M(I-1)*STEM

XYLI = NUM*PSI/I)/YK1M + (NN-I+1)+YK1M(I-1)*STEM

PEXMA = TEMP*PSI(I-1)

TEMP = EPS/PSI(I)

IF (TEMNEF .GT. EPS0) GO TO 900

100 PEXMA = EPS/ARG(PSI(1)),PEXMA!
STOP = ISTOP / 3

CC
RECDEL PSI

DO JON I = 1, JSTOP

INDE=1STPP+1

TEMP=PSI(I)

PE=PSI(INDE)

200 PE(INDE)=TFMP

RETURN

900 CONTINUE

IF (TEMNEF .LT. EPS0) THEN

CALL EMNPON(TEM0 + 33)

END
PROGRAM INTHP

*CHECK INTHP
COMMON/INTHP,K,N,N,X,Z,PSI,STOPH,STOPK,STOP
COMMON PSI(50),N(50),A,PSI,N(50)
COMMON /FUNCTION/ ANA
REAL N,PSI(N),K
EXTERNAL FUNCTION, FUNCTION
ARG = K

10 SUM3 = 0.5 * (1.0 + PSI(1))**2 + (N(0)*PSI(1))**2
       K = 0
       IF (K /= N) GO TO 10

10 SUM3 = SUM3 + PSI(1)**2

15 TEST = TEST / 12.0 + 1.0

20 F = GAUSS(4, 0.0, UPPER, FUNCTION)
    Z = Z / MOLUTE
    SUM3 = SUM3 + F
       K = K + 1

25 END
FUNCTION FUNCTION2(X)
COMMON/INPUTS/NO, N, X, EPS, STOPM, STOPK, STUPK
COMMON PST(RA0), N, IG0, NELIX, NTITLF
COMMON /FUNC/K
CALL NORM
CALL JBESE(X, EP1, X, Y)
FUNCTION2=BEES*2*Y
RETURN
END

A-20
FUNCTION FUNCTA

COMMON INPUTS/H, N0, X, EDN, TMAX, TPWH/ STOPK, STOPA
COMMON PS@I(500), N(EOD), N0, NMAX, TMAX, TITL
COMMON /FINCT/ K
REAL NU, N, K

ASYMPTOTIC APPROXIMATION FOR J1(KR, X) = 2*A

C C C
10 C C C
P1 = ACOS(X/4)
TERM1 = P1
TERM2 = P1

ARCX = (X < TERM2 + .25) * PI
TERM3 = (X < TERM2 + .25) * PI

FunctA = TERM1 * COS(ARCX)*TERM3 * SIN(ARCX) * 1000
RETURN
END
SUBROUTINE INITIAL

1 SUBROUTINE INITIAL (SUM)
COMMON/INPUTS/R,ND,XA,EB,*EPS,ATOMH,STARK,STUD
COMMON PSI(500),N(500),B,NDELTA,N,TITLE
REAL N,NU

5 CALL
REAL NU

CALCULATE DELTA FROM X=AND NO

10 NDELTA = X / (ND = 1)
CALL NUMERICAL VALUE

15 RETURN = ND = 1
RETURN(3*45) TITLE
RETURN

450 FORMAT(450)
READ(5,460) (N(I),I = 1,ISTOP)

20 CALL CALCULATE X(N) VALUE OF N

C C N = N(ND = 1)
KHN = ND * EXP(-DELTA/H)

25 CALL CALCULATE THE INTEGRAL OF N

C C SUMI = 50*ND*SUM(ND)
I = ND
GO TO I = 2 + ISTOP

300 SUM = SUM1 = N/I
SUM = SUM1 * NDELTA
SUM1 = B * NO * SUM1
RETURN

C C ERROR PROCESSING

35 CALL EMPPON(1)
RETURN
RETURN
RETURN
FORMAT(450)
END
SUBROUTINE OUTPUTK

*CHECK OUTPUT

SUBROUTINE OUTPUTK (COUNT)
COMMON INPUTS, RAND, X, Y, FOS, STOPK, STOPV
COMMON PSI, RSP, N, (N0), (NN), (H), (F), X, NT, ITLRF
COMMON STORMU

deal nnn

COMMON /OUT/ X (N0), Y (N0)
IF (COUNT .EQ. 1) GO TO 30

C

WRITE IDENTIFYING PARAMETERS TO TAPE 2
FOLLOWED BY THE VALUES OF NIX)

C

WRITE (2, 94, 1), X, N, STOPV, ENO, NN, STORMV
STOP = NN - 1
WRITE (2, 95) (N(I), INT, INTU)
RETURN

C

CONTINUE

C

PRINT NUMBER OF PAIRS AND LIST
THE PAIRS TO OUTPUT

C

PRINT THE NUMBER OF PAIRS AND
THE PAIRS TO TAPE 2

C

PRINT 98, COUNT
WRITE (2, 96) COUNT
IF (COUNT .NE. 1) COUNT
|
WRITE (2, 100) X(I), Y(I)
|
PRINT 101, X(I), Y(I)
|
RETURN

C

PRIM(1, INT, INT)
|
FORMAT (2(E5.3, 7P+8))
|
FORMAT (8F10.6)
|
FORMAT (1X, 2E2+9)
|
FORMAT (2E2+9)
|
END
SUBROUTINE ERROR

This error processing subroutine is designed for an IDA type environment with CUC 6400 equipment and NO/RK operating system.

The external reference ABRTJOB is a routine written in common to generate error messages, initialize tape hack, and abort the job with out a hind.

1000 CALL ABRTJOB(136) L NO MORE THAN 500 PARS WERE NEEDED FOR THIS BRANCH.
1005 CALL ABRTJOB(46) L THE MAXIMUM K VALUE ON TAPES IS LESS THAN THE K VALUE.
1006 CALL ABRTJOB(46) L THE MAXIMUM K VALUE ON TAPES IS LESS THAN THE K VALUE.
1007 CALL ABRTJOB(3A) L THERE ARE NOT ENOUGH MUNES ON TAPER.
1008 CALL ABRTJOB(2P) L END OF FILE ON UNIT F.
1009 CALL ABRTJOB(16) L KOUNT TOO LARGE.
1010 CALL ABRTJOB(2A) L INFINITY OPERAND N / A.
1012 CALL ABRTJOB(16) L TOO MANY POINTS.
1013 CONTINUE
1004 CALL ABRTJOB(331) K AND K1 ON SAME SIDE OF CURVE.

END
SUBROUTINE PLOT

1 SUBROUTINE PLOT(X,Y,NODE,XX,XN,STEP)
2 COMMON PLOT,XX,XN,STEP
3 COMMON PST(490),X(11),Y(11)
4 COMMON NSEL(4),XX(11),YSEL(11),YSEL(11)
5 COMMON PLOT(490),X(11),Y(11)
6 COMMON PST(490)
7 COMMON NSEL(4),XX(11),YSEL(11)
8 COMMON PLOT(490)
9 COMMON PST(490)
10 COMMON NSEL(4),XX(11),YSEL(11)
11 COMMON PLOT(490)
12 COMMON PST(490)
13 COMMON NSEL(4),XX(11),YSEL(11)
14 COMMON PLOT(490)
15 COMMON PST(490)
16 COMMON NSEL(4),XX(11),YSEL(11)
17 COMMON PLOT(490)
18 COMMON PST(490)
19 COMMON NSEL(4),XX(11),YSEL(11)
20 COMMON PLOT(490)
21 COMMON PST(490)
22 COMMON NSEL(4),XX(11),YSEL(11)
23 COMMON PLOT(490)
24 COMMON PST(490)
25 COMMON NSEL(4),XX(11),YSEL(11)
26 COMMON PLOT(490)
27 COMMON PST(490)
28 COMMON NSEL(4),XX(11),YSEL(11)
29 COMMON PLOT(490)
30 COMMON PST(490)

RETURN

FAC
SUBROUTINE PLOTER

*CHECK PLOT

SUBROUTINE PLOTER(NPLOT, NM, KMAX)
COMMON/OUT/XPLOT(500), YPLOT(500)

CALL PLOTEC(NPLOT, NM, KMAX)

DIMENSION NEL(4), XX(I), YY(I)
REAL KMAX

GO TO (10, 70, 90)

10 CONTINUE

NEL(1) = X
NEL(2) = YNPK / 100
NEL(3) = Y
NEL(4) = KMAX

CALL PSCAL(E(NEL(4))

IF (NEL(1) = 1.0 .LT. 1) THEN
XX(I) = NEL(2)
YY(I) = NEL(3)

100 CONTINUE

CALL PLOT(100.0)

CALL DAXIS(0.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0)

CALL SYM(10.0, 10.0, 10.0, 10.0, 10.0)

RETURN

20 CALL LINE(1, 0, 0, 0, DFL)

CALL PLOT(10.0, 0.0, 1.0)

30 CONTINUE

CALL SYM(40.0, 40.0, 40.0, 40.0, 40.0, 40.0, 40.0)

RETURN

35 FORMAT(1X, 20, 1X)

END

A-26
SUBROUTINE PSCALE

1        SUBROUTINE PSCALE SCAI F SCALE
     TFM = 1
     IF SCALE .NE. 1.0 NO TO 2
     1     J = SCALE*TFM
             IF (J.GT.1.0) GO TO 1A
             TFM = TFM * J
             GO TO 1
     2     J = SCALE/TFM
             IF (J.LE.1.0) GO TO 20
             TFM = TFM/J
             GO TO 2
     1A    SCALE = (J+.5)/(1.0*TFM)
     RETURN
     20    SCALE = (J+.5)*TFM/10.
     RETURN
     END