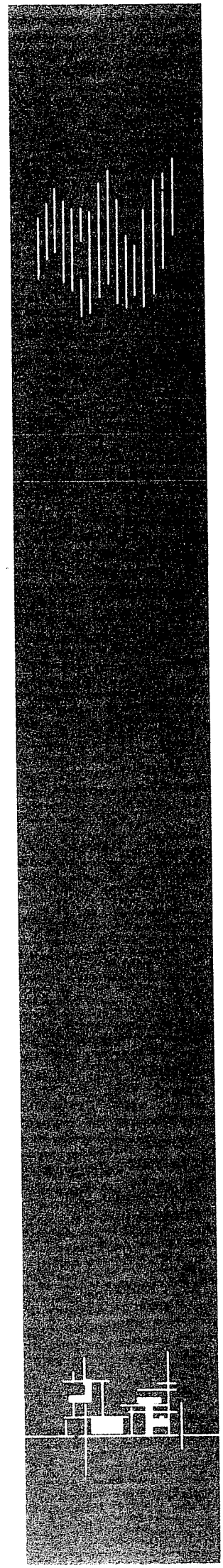

WHAMP-
WAVES IN HOMOGENEOUS, ANISOTROPIC MULTICOMPONENT PLASMAS

Kjell Rönnmark

KGI REPORT NO. 179
JUNE 1982

KIRUNA GEOPHYSICAL INSTITUTE
KIRUNA SWEDEN

WHAMP-
Waves in Homogeneous, Anisotropic, Multicomponent Plasmas

by

Kjell Rönnmark
Kiruna Geophysical Institute
University of Umeå
S-901 87 UMEÅ, Sweden

KGI Report No. 179
June 1982

Printed in Sweden
Kiruna Geophysical Institute
Kiruna, 1982
ISSN 0347-6405

Sir, In your otherwise beautiful poem (The Vision of Sin) there is a verse which reads

"Every moment dies a man,
every moment one is born."

Obviously, this cannot be true and I suggest that in the next edition you have it read

"Every moment dies a man
every moment $1\frac{1}{16}$ is born."

Even this value is slightly in error but should be sufficiently accurate for poetry.

CHARLES BABBAGE
(in a letter to lord TENNYSON)

List of contents

I. Introduction

In this report, a computer program which solves the dispersion relation of waves in a magnetized plasma is described. The dielectric tensor $\underline{\underline{\varepsilon}}$ is derived using the kinetic theory of homogeneous plasmas with Maxwellian velocity distributions. Up to six different plasma components can be included in this version of the program, and each component is specified by its density, temperature, particle mass, anisotropy and drift velocity along the magnetic field. The program is thus applicable to a very wide class of plasmas, and the method should in general be useful whenever a homogeneous magnetized plasma can be approximated by a linear combination of Maxwellian components.

The general theory underlying the program is outlined in the next section. It is shown that by introducing a Padé approximant for the plasma dispersion function Z, the infinite sums of modified Bessel functions which appear in the dielectric tensor $\underline{\underline{\varepsilon}}(\omega, \underline{k})$ may be reduced to a summable form. The resulting expression for $\underline{\underline{\varepsilon}}(\omega, \underline{k})$ is valid for all real $\underline{k}$ and very well suited for numerical evalution. The Padé approximant is derived in section III, where the accuracy of the approximation is also discussed.

In the following sections the subroutines making up the program are described, starting at the bottom level in section IV and reaching the main program in section VII. Finally the results are discussed in the last section.

II. Theory

Let us consider a homogeneous, magnetized multicomponent plasma. Each component consists of particles with charge q_j and mass m_j, and their phase-space density is given by the distribution function $f_j(\underline{v}, \underline{r}, t)$. The gyrofrequency of the particles is $\Omega_j=q_jBm_j^{-1}$ in a magnetic field with strength B. In the absence of external sources, the electric field $\underline{E}(\omega, \underline{k})$ of a wave with frequency ω and wave vector $\underline{k}$ satisfies a wave equation

$$\underline{\underline{D}}(\omega, \underline{k})\cdot\underline{E}=0 \qquad \text{(II-1)}$$

where $\underline{\underline{D}}$ may be expressed in terms of the dielectric tensor $\underline{\underline{\varepsilon}}(\omega, \underline{k})$ as

$$\underline{\underline{D}}(\omega, \underline{k})=(\underline{\underline{I}}k^2-\underline{k}\underline{k})\frac{c^2}{\omega^2} - \underline{\underline{\varepsilon}}(\omega, \underline{k}). \qquad \text{(II-2)}$$

Here $\underline{\underline{I}}$ is the unit tensor and c is the speed of light. The wave equation (1) has non-trivial solutions only if

$$D(\omega, \underline{k})=\det \underline{\underline{D}}(\omega, \underline{k}) = 0, \qquad \text{(II-3)}$$

and this is the dispersion relation we aim to solve.

The first, and most difficult step when solving the dispersion relation is the evaluation of the dielectric tensor. Using linearized kinetic theory, the standard derivation leads to

$$\underline{\underline{\varepsilon}}(\omega, \underline{k}) = \underline{\underline{I}} - \frac{\omega_p^2}{\omega^2}\left\{\underline{\underline{I}} - \sum_j \sum_{n=-\infty}^{\infty} \int d\underline{v}\, \underline{\underline{\Pi}} \frac{\frac{n\Omega_j}{v_\perp}\frac{\partial}{\partial v_\perp} + k_\parallel \frac{\partial}{\partial v_\parallel}}{\omega - k_\parallel v_\parallel + n\Omega_j} f_j^o\right\} \qquad \text{(II-4)}$$

Introducing a coordinate system spanned by the orthogonal unit vectors $\underline{e}_1$, $\underline{e}_2$, $\underline{e}_3$ oriented so that $\underline{B}=B\underline{e}_3$ and $\underline{k}=k_\perp \underline{e}_1+k_\parallel \underline{e}_3$, the matrix $\underline{\underline{\Pi}}$ is given by (see e.g. Ichimaru, 1973, Akhiezer et al. 1975, Clemmow and Dougherty, 1969)

$$\underline{\underline{\Pi}} = \begin{bmatrix} \left(\frac{n\Omega_j}{k_\perp} J_n\right)^2 & i\frac{n\Omega_j}{k_\perp} v_\perp J_n J_n' & \frac{n\Omega_j}{k_\perp} v_\parallel J_n^2 \\ -i\frac{n\Omega_j}{k_\perp} v_\perp J_n J_n' & (v_\perp J_n')^2 & -iv_\perp v_\parallel J_n J_n' \\ \frac{n\Omega_j}{k_\perp} v_\parallel J_n^2 & iv_\perp v_\parallel J_n J_n' & (v_\parallel J_n)^2 \end{bmatrix} \qquad \text{(II-5)}$$

The argument of the Bessel function J_n is $k_\perp v_\perp / \Omega_j$. Since the densities n_j are included in the plasma frequency

$$\omega_p = \left(\sum_j \omega_{pj}^2\right)^{1/2} = \left(\sum_j \frac{n_j q_j^2}{\varepsilon_o m_j}\right)^{1/2} \qquad \text{(II-6)}$$

the unperturbed distribution f^o is normalized to unity

$$\int f^o(\underline{v}) d\underline{v} = \int \sum_j f_j^o(\underline{v})\, d\underline{v} = 1. \qquad \text{(II-7)}$$

The most general distribution function considered in this report has the form

$$f^o(v_\perp, v_\parallel) = \sum_{j=1}^{6} (\pi^{1/2} V_j)^{-3} \exp\left(-\left(\frac{v_\parallel}{V_j} - V_{dj}\right)^2\right) \cdot \qquad \text{(II-8)}$$

$$\left\{ \frac{\Delta j}{\alpha_{1j}} \exp\left(-\frac{v_\perp^2}{\alpha_{1j} V_j^2}\right) + \frac{1 - \Delta j}{\alpha_{1j} - \alpha_{2j}} \left[\exp\left(-\frac{v_\perp^2}{\alpha_{1j} V_j^2}\right) - \exp\left(-\frac{v_\perp^2}{\alpha_{2j} V_j^2}\right) \right] \right\}$$

Here, V_j is the thermal velocity of a component with temperature $T_j = 1/2 m_j V_j^2$, and V_{dj} is a normalized drift velocity along the magnetic field. The parameters Δ_j, α_{1j}, and α_{2j} determine the depth and size of the loss-cone and the temperature anisotrophy.

The integral over velocity space in Equation (4) is evaluated by means of the relations

$$\int_0^\infty J_n^2 \left(\frac{k_\perp v_\perp}{\Omega_j}\right) \exp\left(-\frac{v_\perp^2}{V_j^2}\right) \frac{2v_\perp}{V_j^2}\, dv_\perp = \Lambda_n(\lambda_j) \qquad \text{(II-9)}$$

and

$$\pi^{-1/2} \int_{-\infty}^{\infty} \frac{\exp\left(-\frac{v_\parallel^2}{V_j^2}\right) dv_\parallel}{v_\parallel - (\omega - n\Omega_j) k_\parallel^{-1}} = Z\left(\frac{\omega - n\Omega_j}{k_\parallel V_j}\right), \qquad \text{(II-10)}$$

where $\Lambda_n(\lambda_j) = e^{-\lambda_j} I_n(\lambda_j)$, $\lambda_j = 1/2 (k_\perp V_j/\Omega_j)^2$, I_n is a modified Bessel function of order n, and Z is the plasma dispersion function (Fried and Conte, 1961). The dielectric tensor of a plasma with a particle distribution described by Equation (8) can then be written as

$$\underline{\underline{\varepsilon}}(\omega, \underline{k}) = \left(1 - \frac{\omega_p^2}{\omega^2}\right) \underline{\underline{I}} + \sum_{j=1}^{6} \frac{\omega_{pj}^2}{\omega^2 \alpha_{1j} (\alpha_{1j} - \alpha_{2j})} \qquad \text{(II-11)}$$

$$\left[(\alpha_{1j} - \Delta_j \alpha_{2j}) \underline{\underline{\chi}}^{\alpha_{1j}} + \alpha_{1j} (\Delta_j - 1) \underline{\underline{\chi}}^{\alpha_{2j}} \right]$$

To simplify the notation, we omit the index j, introduce the dimensionless quantities

$$p = \frac{k_\perp V}{\Omega} , \quad z = \frac{k_\parallel V}{\Omega} , \quad x = \frac{\omega}{\Omega} , \quad s_n = \frac{x - zV_d - n}{z} \tag{II-12}$$

and the function

$$\zeta_n^\alpha(x, z) = (\alpha s_n + n/z)\, Z(s_n) \tag{II-13}$$

The components of the suceptibility tensor $\underline{\underline{\chi}}^\alpha$ are then given by

$$\begin{aligned}
\chi_{11}^\alpha &= \alpha^2 + \sum_{n=-\infty}^{\infty} \lambda^{-1} n^2 \Lambda_n(\alpha\lambda)\, \zeta_n^\alpha \\
\chi_{12}^\alpha &= i\alpha \sum_{n=-\infty}^{\infty} n\Lambda_n'(\alpha\lambda)\, \zeta_n^\alpha \\
\chi_{13}^\alpha &= \alpha(1-\alpha)p/z + (2/\lambda)^{1/2} \sum_{n=-\infty}^{\infty} n\Lambda_n(\alpha\lambda)\frac{x-n}{z}\, \zeta_n^\alpha \\
\chi_{22}^\alpha &= \chi_{11}^\alpha - 2\alpha^2\lambda \sum_{n=-\infty}^{\infty} \Lambda_n'(\alpha\lambda)\, \zeta_n^\alpha \\
\chi_{23}^\alpha &= i\alpha p \sum_{n=-\infty}^{\infty} \Lambda_n'(\alpha\lambda)\, \frac{x-n}{z}\, \zeta_n^\alpha \\
\chi_{33}^\alpha &= \alpha\left[1 - (1-\alpha)p^2/z^2 + 2x^2/z^2\right] + 2\sum_{n=-\infty}^{\infty} \Lambda_n(\alpha\lambda)\left(\frac{x-n}{z}\right)^2 \zeta_n^\alpha
\end{aligned} \tag{II-14}$$

The evaluation of these expressions can be greatly simplified by introducing an approximation for the plasma dispersion function in the form

$$Z(s) \approx \sum_{l=1}^{L} b_l (s - c_l)^{-1} \tag{II-15}$$

Approximations of this form can be derived by a modified Pade´ method as discussed in section III, where the relations

$$\sum_{l=1}^{L} b_l = -1$$

$$\sum_{l=1}^{L} c_l b_l = 0$$

$$\sum_{l=1}^{L} c_l^2 b_l = -1/2$$

are also derived (Eq. III-10). Introducing $y = x - z(c_1 + V_d)$ the components of the suceptibility tensor may now be reduced to the form

$$\begin{aligned}
\chi_{11} &= \alpha + \alpha \sum_{l=1}^{L} b_l y\, \psi_l \\
\chi_{12} &= i\alpha \sum_{l=1}^{L} b_l\, \psi_l' \\
\chi_{13} &= \alpha p \sum_{l=1}^{L} (c_l + V_d)\, b_l\, \psi_l \\
\chi_{22} &= \chi_{11} - 2\alpha^2 \lambda \sum_{l=1}^{L} b_l\, y^{-1}\, \psi_l' \\
\chi_{23} &= i\alpha p \sum_{l=1}^{L} b_l\, (c_l + V_d)\, y^{-1}\, \psi_l' \\
\chi_{33} &= \alpha + 2\alpha \left\{ V_d^2 + \sum_{l=1}^{L} (c_l + V_d)^2 b_l\, y^{-1}\, [x - zV_d + \lambda \psi_l] \right\}
\end{aligned} \qquad \text{(II-16)}$$

Here we have introduced the notation

$$\begin{aligned}
\psi_l &= (1 + \alpha z c_l y^{-1})\, R(y, \alpha\lambda) \\
\psi_l' &= (1 + \alpha z\, c_l y^{-1})\, R'(y, \alpha\lambda)
\end{aligned} \qquad \text{(II-17)}$$

The function R is defined by

$$R(y, \lambda) = \sum_{n=-\infty}^{\infty} \frac{n^2}{\lambda} \frac{\Lambda_n(\lambda)}{y - n} \qquad \text{(II-18)}$$

and R' is not quite the derivative with respect to λ but rather

$$R'(y,\lambda) = \frac{\partial}{\partial\lambda}(\lambda R(y,\lambda)) = \sum_{n=-\infty}^{\infty} \frac{n^2 \Lambda_n'(\lambda)}{y-n} = y^2 \sum_{n=-\infty}^{\infty} \frac{\Lambda_n'(\lambda)}{y-n} \quad \text{(II-19)}$$

Some analytial properties of R and the algorithms for computing its value are discussed in section IV. In general, $R(y,\lambda)$ can be evaluated in about the same time as a single $\Lambda_n(\lambda)$.

Comparing Equation (14) to Equation (16), we note the following points:

1° Using Equation (14) we have roughly $\chi^\alpha \propto \sum_{n=-N}^{N} \Lambda_n \zeta_n$ while (16) gives $\chi^\alpha \propto \sum_{l=1}^{L} b_l \psi_l$.

Assuming that $\Lambda_n \zeta_n$ can be computed in the same time as ψ_l, the evaluation of (14) would be roughly 2N/L times as laborious as the evaluation of (16). Choosing L=8 as in the program we see that the efficiency of Equation (14) is comparable to that of (16) for $N \leq 4$.

2° When λ is large, the sum over n in Equation (14) is slowly convergent. Late terms decrease as $\exp(-n^2/2\lambda)$ and $N \gg \sqrt{2\lambda}$ is thus needed for convergence. This makes a direct evaluation of (14) very inefficient for large λ. As $\lambda \to \infty$ in Equation (16), all components of $\underline{\underline{\chi}}$ behave continuously and they reduce to their correct values in the limit.

3° The component χ_{33} in Equation (14) contains terms proportional to z^{-2}. When z is small these terms become large and they cancel the way they should only if the relation

$$\sum_{n=-\infty}^{\infty} \Lambda_n(\lambda) = 1$$

is fulfilled exactly.

This implies that N must be large when z is small, and truncation errors will become serious for any N if z is sufficiently small. Similar arguments apply to the terms proportional to z^{-1} in χ_{13} and χ_{23}. As $z \to 0$ in Equation (16), all components of $\underline{\underline{\chi}}$ behave continuously and they reduce to their correct values for z=0.

4° The components χ^{α}_{11} and χ^{α}_{13} in (14) are proportional to λ^{-1} and $\lambda^{-1/2}$ respectively. This may lead to serious truncation errors when λ is small, and the limit $\lambda \to 0$ can not be taken numerically. As $\lambda \to 0$ in Equation (16) all components of $\underline{\underline{\chi}}^{\alpha}$ behave continuously and they reduce to their correct values for $\lambda = 0$.

The dielectric tensor $\underline{\underline{\varepsilon}}(\omega, \underline{k})$ is computed by inserting the components of $\underline{\underline{\chi}}^{\alpha}$ given by Equation (16) in Equation (11). Introducing the refractive index $\underline{\mu} = \mu_1 \underline{e}_1 + \mu_3 \underline{e}_3$ and expanding the determinant of $\underline{\underline{D}}(\omega, \underline{k})$ as given by (2) we find

$$D(\omega, \underline{k}) = A\,(\mu^2 - \varepsilon_{22}) - B + C \qquad \text{(II-20)}$$

where

$$\begin{aligned} A &= \mu_1^2 \varepsilon_{11} + 2\mu_1 \mu_3 \varepsilon_{13} + \mu_3^2 \varepsilon_{33} \\ B &= (\mu_3 \varepsilon_{23} - \mu_1 \varepsilon_{12})^2 + \mu^2 (\varepsilon_{11} \varepsilon_{33} - \varepsilon_{13}^2) \\ C &= (\varepsilon_{11} \varepsilon_{33} - \varepsilon_{13}^2) \varepsilon_{22} + (\varepsilon_{11} \varepsilon_{23} + \varepsilon_{12} \varepsilon_{13}) \varepsilon_{23} + \\ &\quad + (\varepsilon_{33} \varepsilon_{12} + \varepsilon_{23} \varepsilon_{13}) \varepsilon_{12} \end{aligned} \qquad \text{(II-21)}$$

Formulas for the derivatives of D with respect to ω and $\underline{k}$ are given in Section VI. In the main program, Newton's iteration method is used to find a complex ω which satisfies the dispersion relation (3).

III. Approximation of the Z-funciton

The Pade´ method was first used to approximate the plasma dispersion function Z(s) by Martín and Gonzáles (1979), and their results were generalized by Martín et al. (1980) and Németh et al. (1981). The basic theory of Padé approximants can be found in the book by Baker (1975).

Following Martín et al. (1980), we consider approximations $Z_A(s)$ for the plasma dispersion function in the form

$$Z_A(s) = \frac{P^{L-1}(s)}{Q^L(s)} = \sum_{l=1}^{L} \frac{b_l}{s-c_l} \qquad \text{(III-1)}$$

where $P^{L-1}(s) = \sum_{l=0}^{L-1} p_l s^l$ and $Q^L(s) = 1 + \sum_{l=1}^{L} q_l s^l$

Inserting the convergent power series,

$$Z(s) = i\sqrt{\pi} - 2s - i\sqrt{\pi}\, s^2 + \frac{4}{3} s^3 + \frac{i}{2}\sqrt{\pi}\, s^4 - \frac{8}{15} s^5 + \ldots \qquad \text{(III-2)}$$

in the equation

$$Z(s)Q^L(s) = P^{L-1}(s) \qquad \text{(III-3)}$$

and identifying coefficients of equal powers of s we obtain a set of equations

$$\begin{cases} i\sqrt{\pi} = p_o \\ -2 + i\sqrt{\pi}\, q_1 = p_1 \\ -i\sqrt{\pi} - 2q_1 + i\sqrt{\pi}\, q_2 = p_2 \\ 4/3 - i\sqrt{\pi}\, q_1 - 2q_2 + i\sqrt{\pi} q_3 = p_3 \\ \cdots \end{cases} \qquad \text{(III-5)}$$

Here, we take $p_l = 0$ if $l > L-1$ and $q_l = 0$ if $l > L$.

An alternative set of equations is obtained by inserting the asymptotic series

$$Z(s) \simeq -s^{-1} - \frac{1}{2} s^{-3} - \frac{3}{4} s^{-5} - \frac{15}{8} s^{-7} - \ldots \qquad \text{(III-6)}$$

in Equation (3):

$$\begin{cases} -q_L = p_{L-1} \\ -q_{L-1} = p_{L-2} \\ -q_{L-2} - 1/2 q_L = p_{L-3} \\ -q_{L-3} - 1/2 q_{L-1} = p_{L-4} \\ \cdots \end{cases} \qquad \text{(III-7)}$$

Since we need 2L equations to determine all the p:s and q:s, we let J+K=2L and choose J equations from (5) and K equations from (7). The resulting approximant will satisfy

$$|Z_A(s) - Z(s)| = \begin{cases} O(s^J), & s \to 0 \\ O(s^{-K}) & s \to \infty \end{cases}$$

Alternatively, we could have started from the second form of (1) and expanded

$$Z_A(s) = \sum_{l=1}^{L} b_l \begin{cases} -\frac{1}{a_l} - \frac{s}{a_l^2} - \frac{s^2}{a_l^3} - \frac{s^3}{a_l^4} + \cdots & s \to 0 \\ s^{-1} + c_l s^{-2} + c_l^2 s^{-3} + c_l^3 s^{-4} + \cdots & s \to \infty \end{cases} \qquad \text{(III-8)}$$

Comparison with (2) and (6) leads to the equations

$$\begin{cases} \sum_{l=1}^{L} \frac{b_l}{c_l} = -i\sqrt{\pi} \\ \sum_{l=1}^{L} \frac{b_l}{c_l^2} = 2 \\ \sum_{l=1}^{L} \frac{b_l}{c_l^3} = i\sqrt{\pi} \\ \cdots \end{cases} \qquad \text{(III-9)}$$

and

$$\begin{cases} \sum_{l=1}^{L} b_l = -1 \\ \sum_{l=1}^{L} b_l c_l = 0 \\ \sum_{l=1}^{L} b_l c_l^2 = -1/2 \\ \sum_{l=1}^{L} b_l c_l^3 = 0 \\ \ldots \end{cases} \qquad \text{(III-10)}$$

In practice, the most convenient way to derive the partial fraction expansion of the approximant is to eliminate the p:s from L of the Equations (5) and (7) and determine the q:s from these. The equation $Q^L(s)=0$ is then solved for the poles c_1, c_2, ... c_L, and L non-trivial equations are chosen from (9) and (10) to determine the b:s.

Following this procedure, an eight pole approximant was derived using ten equations from (5) and six equations from (7). The values of the coefficients are given in Table 1.

Table 1

$c_1 = 2.237\ 687\ 789\ 201\ 900 - i\ 1.625\ 940\ 856\ 173\ 727$
$c_2 = -c_1^*$
$c_3 = 1.465\ 234\ 126\ 106\ 004 - i\ 1.789\ 620\ 129\ 162\ 444$
$c_4 = -c_3^*$
$c_5 = .8392\ 539\ 817\ 232\ 638 - i\ 1.891\ 995\ 045\ 765\ 206$
$c_6 = -c_5^*$
$c_7 = .2739\ 362\ 226\ 285\ 564 - i\ 1.941\ 786\ 875\ 844\ 713$
$c_8 = -c_7^*$

$b_1 = -.017\ 340\ 124\ 574\ 718\ 26 - i\ .046\ 306\ 392\ 916\ 803\ 22$
$b_2 = b_1^*$
$b_3 = -.739\ 916\ 992\ 322\ 5014 + i\ .839\ 517\ 997\ 809\ 9844$
$b_4 = b_3^*$
$b_5 = 5.840\ 628\ 642\ 184\ 073 + i\ .953\ 600\ 905\ 764\ 3667$
$b_6 = b_5^*$
$b_7 = -5.583\ 371\ 525\ 286\ 853 - i\ 11.2\ 085\ 431\ 912\ 6599$
$b_8 = b_7^*$

In the upper half of the s-plane the accuracy of this approximant should be sufficient for all purposes. However, for Im $s<0$ the errors increase as s approaches the poles c_1, and when s is large the omitted exponential term $-i2\pi^{1/2}\exp(-s^2)$ in the asymptotic series for Im $s<0$ may become important. Figures 1 and 2 show the relative errors $(\mathrm{Re}(Z_A(s) - Z(s))/\mathrm{Re}\ Z(s) \cdot 100\%$ and $\mathrm{Im}(Z_A(s) - Z(s))/\mathrm{Im}\ Z(s) \cdot 100\%$ for Im $s = -1/2$ Re s. We see that the relative error in Re Z_A stays less than 2% and the error in Im Z_A is less than 3%. The errors are largest for s between 2-i and 3-i 1.5, and they may thus be traced to the influence of the pole c_1.

An obvious problem with rational approximations for the Z-function is that the exponential behavior of Im $Z(s) = \pi^{1/2}\exp(-s^2)$ for real s can not be accurately approximated by a rational function of finite order. Approximants of the type discussed here have the rather disturbing property that Im $Z_A(s) < 0$ for some real values of s (see also Martín et al. (1980)). The absolute error in Im $Z_A(s)$ for real s is shown in Fig. 3, where the important point is that the small error for $5<s<10$ makes Im $Z_A(s)$ negative in this interval. Physically this means that a very weakly damped wave may appear as weakly unstable because Z is replaced by Z_A when solving the dispersion relation. Keeping this possibility in mind, it should however not be difficult to trace and disregard these "numerical instabilities" if they appear. The larger absolute errors appearing for $s\sim3$ should normally be insignificant since Im $Z_A > 0$ and the relative error is less than 2%.

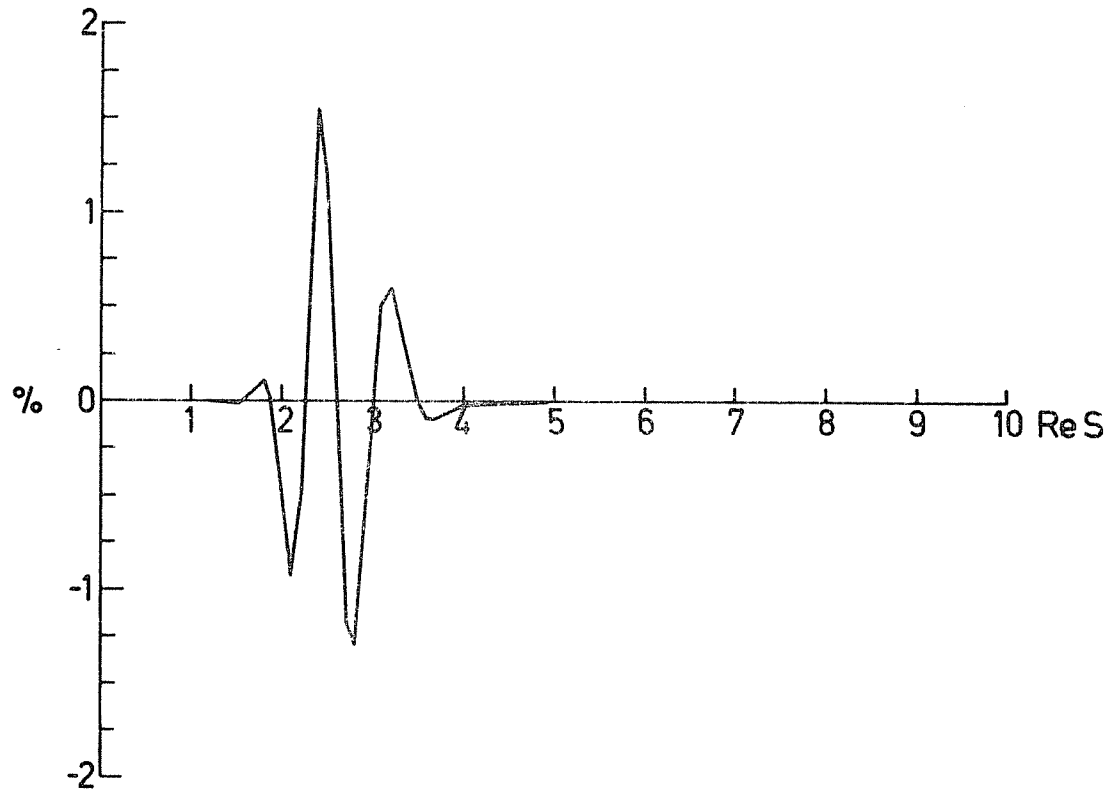

Figure 1 The relative error in Re $Z_A(s)$ versus Re s for Im s = - 1/2 Re s.

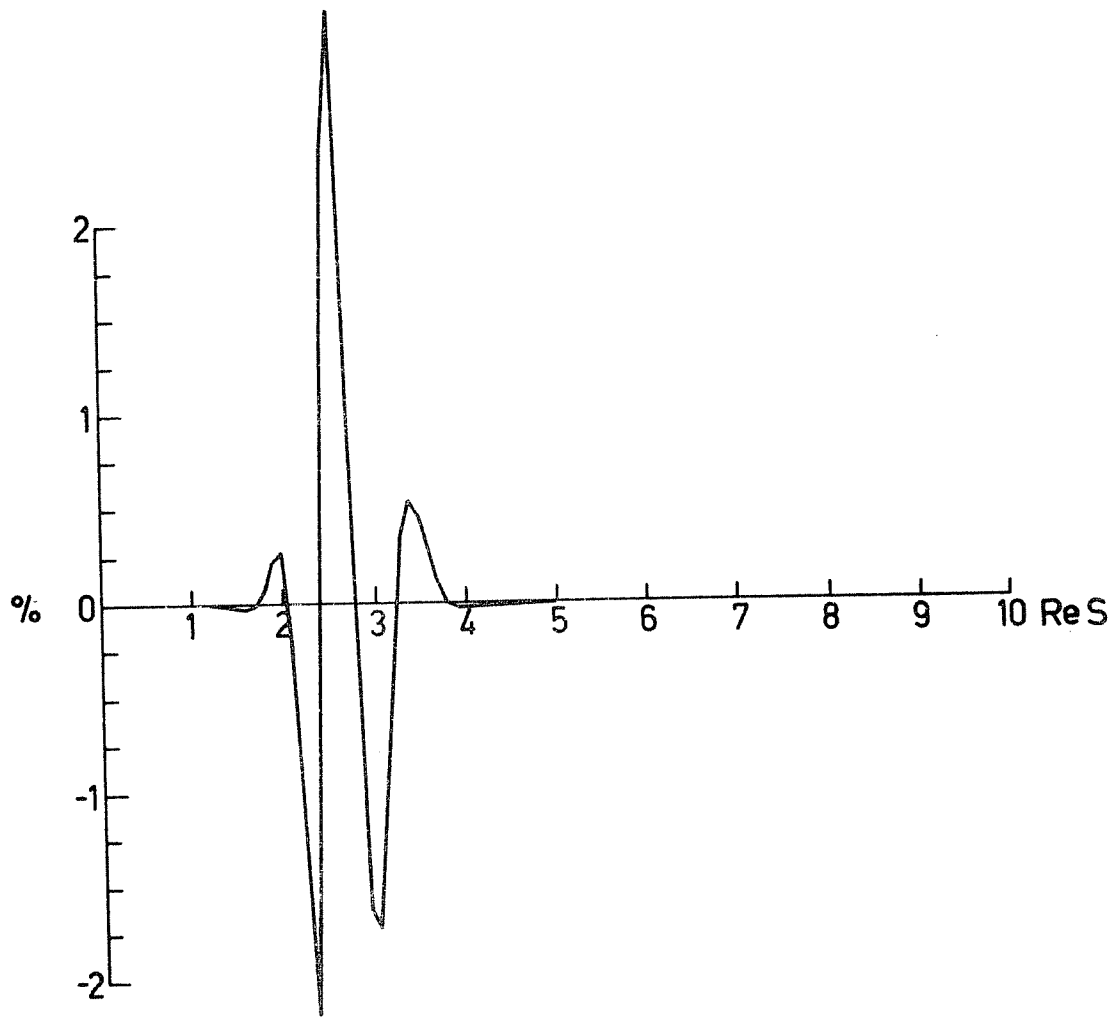

Figure 2 The relative error in Im Z_A (s) versus Re s for Im s = - 1/2 Re s.

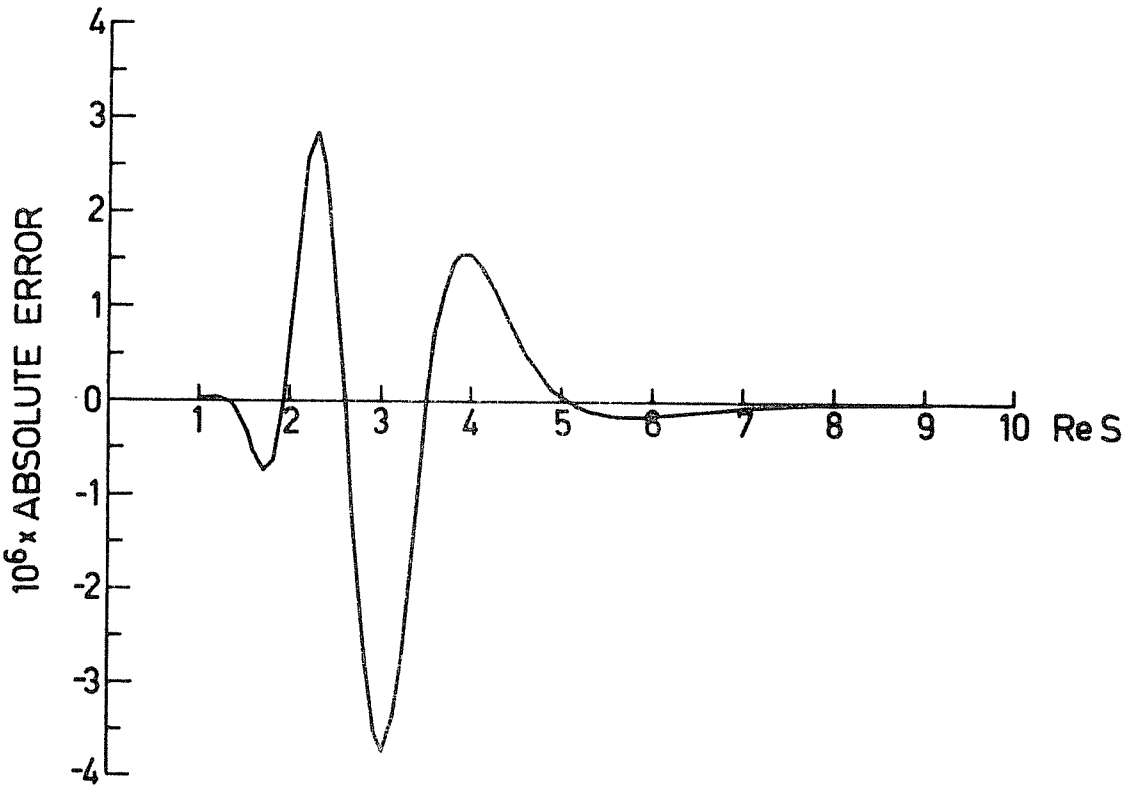

Figure 3 The absolute error in Im Z_A (s) versus s for Im s = 0.

IV. The function R(y, λ)

SUBROUTINE RTAY

The function R, introduced in the theory section, is defined by

$$R(y,\lambda) = \sum_{n=-\infty}^{\infty} \frac{n^2 \Lambda_n(\lambda)}{\lambda(y-n)} \qquad \text{(IV-1)}$$

with $\Lambda_n(\lambda) = e^{-\lambda} I_n(\lambda)$. Closely related functions have been examined by e g Aamodt (1967), Fredricks (1968), and Karpman et al. (1973), in their studies of waves with $k_{\parallel} \approx 0$.

From the corresponding relations for the modified Bessel functions it is easy to show that R satisfies the recurrence relations

$$\frac{\partial R(y,\lambda)}{\partial \lambda} = \frac{1}{2}\,[R(y-1,\lambda) + R(y+1,\lambda)] - R(y,\lambda) \qquad \text{(IV-2)}$$

and

$$yR(y,\lambda) = \frac{1}{2}\,\lambda[R(y-1,\lambda) - R(y+1,\lambda)] + 1 \qquad \text{(IV-3)}$$

Differentiating R twice with respect to λ using (2) and (3) we obtain the differential equation

$$\lambda^2 \frac{\partial^2 R}{\partial \lambda^2} + (3+2\lambda)\,\lambda \frac{\partial R}{\partial \lambda} + (1-y^2+3\lambda)R + y = 0 \qquad \text{(IV-4)}$$

Substituting a power series for R, we find the solution

$$R(y,\lambda) = \frac{y}{y^2-1} + y\frac{1\cdot 3}{(y^2-1)(y^2-4)}\,\lambda + y\frac{1\cdot 3\cdot 5}{(y^2-1)(y^2-4)(y^2-9)}\,\lambda^2 + \ldots \qquad \text{(IV-5)}$$

$$= \frac{\pi y^2}{\lambda \sin \pi y} \sum_{n=1}^{\infty} \frac{(2n)!}{(n+y)!\,(n-y)!} \frac{(-\frac{1}{2}\lambda)^n}{n!}$$

This series is easily computed and rapidly convergent for small λ. It is maybe less obvious that it is useful also for large λ if $y >> \lambda$. In this case the early terms ($n << y$) will have the form $y(2n+1)!!\ (\lambda y^{-2})^n$, and their magnitude will decrease rapidly. Later terms will remain small and insignificant unless y is extremely close to an integer.

SUBROUTINE RASY

When λ is large, an asymptotic series for $R(y,\lambda)$ is very useful. Since $e^{-\lambda} I_n(\lambda) \to (2\pi\lambda)^{-1/2}$ as $\lambda \to \infty$, we see from the definition (1) that $R(y,\lambda) \sim \lambda^{-1}\left[(2\lambda/\pi)^{-1/2}\, y^2 \cot \pi y - y\right]$ for large λ. Introducing $\eta = \lambda^{-1}$ and $P = \eta^{-1} R$, the differential equation (4) is transformed to

$$\eta^3 \frac{\partial^2 P}{\partial \eta^2} + (\eta - 2)\eta \frac{\partial P}{\partial \eta} + (1 - y^2\eta)P = -y \qquad \text{(IV-6)}$$

A particular solution of this equation is

$$G(\eta) = -y + y^3\eta + \frac{y^3(1-y^2)}{1\cdot 3}\eta^2 + \frac{y^3(1-y^2)(4-y^2)}{1\cdot 3\cdot 5}\eta^3 \ldots \qquad \text{(IV-7)}$$

The complementary function $H(\eta) = \eta^{-1/2} P(\eta)$ thus satisfies the homogeneous equation

$$\eta^2 \frac{\partial^2 H}{\partial \eta^2} + 2(\eta - 1)\frac{\partial H}{\partial \eta} + \left(\frac{1}{4} - y^2\right)H = 0 \qquad \text{(IV-8)}$$

with the boundary condition $H(0) = y^2(\pi/2)^{1/2} \cot \pi y$, and the solution is found to be

$$H(\eta) = y^2 \left(\frac{\pi}{2}\right)^{1/2} \cot \pi y \left\{ 1 + \frac{\frac{1}{4} - y^2}{2\ 1!}\eta + \frac{\left(\frac{1}{4}-y^2\right)\left(\frac{9}{4}-y^2\right)}{2^2 \cdot 2!}\eta^2 + \frac{\left(\frac{1}{4}-y^2\right)\left(\frac{9}{4}-y^2\right)\left(\frac{16}{4}-y^2\right)}{2^3 \cdot 3!}\eta^3 + \ldots \right\} \qquad \text{(IV-9)}$$

Transforming back to the original variables, we may write the asymptotic series in the convenient form

$$R(y,\lambda) \simeq \sum_{n=0}^{\infty} R_n \qquad (\lambda >> 1) \qquad \text{(IV-10)}$$

where R_n is defined by the recursion formula

$$R_{n+2} = \frac{n^2 - 4y^2}{(n+1)\ 4\lambda} R_n \qquad \text{(IV-11)}$$

and

$$R_o = -y/\lambda \quad \text{and} \quad R_1 = y^2 \left(\frac{\pi}{2\lambda^3}\right)^{1/2} \cot \pi y$$

This series gives very accurate results for $\lambda > 10$ if y is not to large, However, from the recursion relations (11) we see that if $y^2 > \lambda >> 1$, the early terms will increase in magnitude roughly as $y^2/(n\lambda)$. The convergence will obviously be slower in this case, and accuracy can easily be lost due to truncation errors.

SUBROUTINE RINT

An integral representation of the function $R(y,\lambda)$ can be derived by inserting

$$\frac{1}{y^2-n^2} = \frac{(-1)^n}{\sin \pi y} \int_o^{\pi} \cos n\varphi \cos y\varphi \, d\varphi \qquad \text{(IV-12)}$$

in the definition (1) and using the generating function for $I_n(\lambda)$

$$e^{\frac{\lambda}{2}(t+t^{-1})} = \sum_{n=-\infty}^{\infty} t^n I_n(\lambda) \qquad \text{(IV-13)}$$

This leads to the representation

$$R(y,\lambda) = - \frac{y}{\sin \pi y} \int_o^{\pi} e^{-\lambda(1+\cos\varphi)} \sin y\varphi \sin\varphi \, d\varphi \qquad \text{(IV-14)}$$

For large values of y, the factor sin yφ will oscillate rapidly, and a direct numerical integration becomes difficult. As described by Fredricks (1968), these oscillations can be removed by evaluating the integral along a properly chosen contour in the complex φ-plane.

We first partially integrate (14) to find

$$R(y,\lambda) = (y^2 I(y,\lambda) - y)/\lambda \qquad \text{(IV-15)}$$

where

$$I(y,\lambda) = \frac{1}{\sin\pi y}\int_0^{\pi} e^{-\lambda(1+\cos\varphi)}\cos\, y\varphi\, d\varphi =$$

$$= \frac{1}{\sin\pi y}\int_0^{\pi} e^{\lambda(\cos\varphi-1)}\cos\, y(\varphi-\pi)\, d\varphi = \qquad \text{(IV-16)}$$

$$= \frac{1}{2}(I_+ + I_-)\cot\pi y + \frac{1}{2i}(I_+ - I_-)$$

Here, we have defined

$$I_{\pm}(y,\lambda) = \int_0^{\pi} e^{\lambda(\cos\varphi-1) \pm iy\varphi} d\varphi \qquad \text{(IV-17)}$$

Letting $\varphi = \alpha + i\beta$ and $y = \omega + i\gamma$ we find that the imaginary part of the exponent is $-\lambda\sin\alpha\sinh\beta \pm \omega\alpha \mp \gamma\beta$. If y were real, as in the case considered by Fredricks (1968), the contours C ± shown in Figure 4 would be paths of stationary phase. The pase is zero on the imaginary axis and along the curves starting at $\pm B_o = \sinh^{-1}\omega/\lambda$. In terms of $\psi(\alpha) = y\alpha/\lambda\sin\alpha$ these curves are defined by $\beta = \pm\sinh^{-1}\psi$. The phase shift

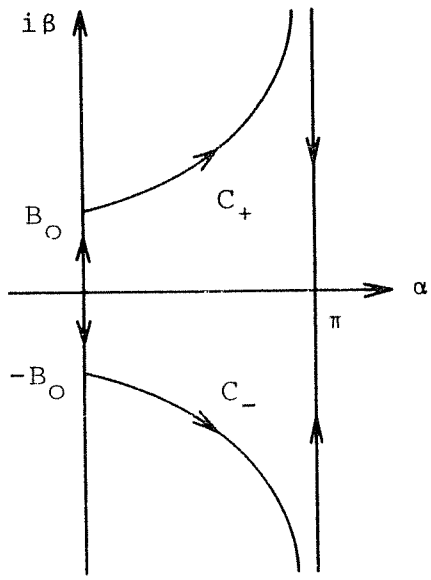

Figure 4 Contours for the integrals $I^{\pm}$ in the complex φ-plane.

takes place at $\alpha = \pi, \beta = \pm\infty$ where the integrands are exponentially small, and along the line $\alpha = \pi$ the phase is $\pm\omega\pi$. When y is complex, the contours of stationary phase can no longer be explicitly defined. However, as long as $\gamma << \omega$ we would expect the oscillations to be heavily damped along the contours C±. Integrating along these paths, we find

$$I_{\pm}(y,\lambda) = \pm i \int_o^{B_o} e^{\lambda(\cosh\beta-1)-y\beta} d\beta +$$

$$+ \int_o^{\pi} e^{\lambda[(1+\psi^2)^{1/2}\cos\alpha-1]-y\sinh^{-1}\psi \pm \alpha\gamma[1\pm i\frac{\partial \sinh^{-1}\psi}{\partial\alpha}]} d\alpha + \qquad \text{(IV-18)}$$

$$\pm i \int_{\infty}^{o} e^{\lambda(\cosh\beta-1)-y\beta \pm iy\pi} d\beta$$

Using Equation (16), we find after some manipulation that the contributions from the line $\alpha = \pi$ cancel, and the remaining contributions may be written

$$I(y,\lambda) = \int_o^{B_o} e^{\lambda(\cosh x-1)-yx} dx + \qquad \text{(IV-19)}$$

$$+ \int_o^{\pi} e^{T(x)-yB(x)} [F(x)C(x) + H(x)S(x)]\, dx$$

where $T(x) = \lambda[(1+\psi^2)^{1/2}\cos x-1]$, $B(x) = \sinh^{-1}\psi$, $C(x) = \cosh \gamma x$, $S(x) = i \sinh \gamma x$, $F(x) = \cot \pi y + G(x)$, $H(x) = 1-G(x)\cot\pi y$, and $G(x) = \partial B/\partial x$. This notation is chosen to conform with the variable names used in the code.

The derivatives of $I_{\pm}$ are from Equation (17) found to be

$$\frac{\partial I_{\pm}}{\partial y} = \pm i \int_{o}^{\pi} e^{\lambda(\cos\varphi-1) \pm iy\varphi} \varphi d\varphi$$

$$\frac{\partial I_{\pm}}{\partial \lambda} = \int_{o}^{\pi} e^{\lambda(\cos\varphi-1) \pm iy\varphi} (\cos\varphi-1)\, d\varphi \qquad \text{(IV-20)}$$

$$\frac{\partial^2 I_{\pm}}{\partial y \partial \lambda} = \pm i \int_{o}^{\pi} e^{\lambda(\cos\varphi-1) \pm iy\varphi} (\cos\varphi-1)\, \varphi d\varphi$$

Transforming these integrals to the contours $C\pm$ and combining them as prescribed by the derivatives of Equation (16) is tedious but strightforward. The derivatives of $I(y,\lambda)$ are in this way found to be

$$y\frac{\partial I}{\partial y} = -y\cot\pi y \int_{o}^{B_o} e^{\lambda(\cosh x-1)-yx} x dx$$

$$-y\int_{o}^{\pi} e^{T-yB} [F\cdot O - H\cdot P + D(C-G\cdot S)]dx$$

$$\lambda\frac{\partial I}{\partial \lambda} = \lambda \int_{o}^{B_o} e^{\lambda(\cosh x-1)-yx} \cosh x\, dx \qquad \text{(IV-21)}$$

$$+\int_{o}^{\pi} e^{T-yB} [(F\cdot T - H x\omega)C + (H\cdot T + F x\omega)S]dx$$

$$y\lambda\frac{\partial^2 I}{\partial y \partial \lambda} = -y\cot\pi y \int_{o}^{B_o} e^{\lambda(\cosh x-1)-yx} (\cosh x-1) x dx -$$

$$-y\int_{o}^{\pi} e^{T-yB} \left\{ F(T\cdot O - P\omega x) - H(T\cdot P + O\omega x) + [(T+G\omega x)C - (GT-\omega x)S]D \right\} dx$$

where we have introduced $D = \pi(1+\cot^2\pi y)$, $O(x) = B(x)\,C(x) + xS(x)$ and $P(x) = xC(x) - B(x)\,S(x)$. Finally, from Equation (15) we have

$$y\frac{\partial R}{\partial y} = 2R + y/\lambda(y^2\frac{\partial I}{\partial y} + 1)$$

$$R' = \frac{\partial}{\partial \lambda}(\lambda R) = y^2\frac{\partial I}{\partial \lambda} \qquad \text{(IV-22)}$$

$$y\frac{\partial R'}{\partial y} = 2R' + y^3\frac{\partial^2 I}{\partial y \partial \lambda}$$

The integrals (19) and (21) are evaluated by Gauss' quadrature formula, using 16 points. The abscissas A(I) and weights W(I) are taken from Abramowitz and Stegun (1965). When y is large, the integrand in the second integral becomes very small as x approaches π. The semi-empirical formula $UL = \pi - 2.8y(36+y)^{-1}$ gives $T(UL)-yB(UL) \sim -100$ and we can thus safely integrate only up to UL, gaining increased accuracy in the remaining interval.

SUBROUTINE RYLA

The subroutine RYLA returns the (2,2)-array RC,

$$RC = \begin{Bmatrix} R(y,\lambda) & y\frac{\partial R(y,\lambda)}{\partial y} \\ R'(y,\lambda) & y\frac{\partial R'(y,\lambda)}{\partial y} \end{Bmatrix} \qquad \text{(IV-23)}$$

The value of RC is calculated by one of the three methods described above. For given values of y and λ, the method chosen can be found from Figure 5. Comparing the different methods, we find that they agree to at least six decimal places along the borderlines shown in Figure 5, and the accuracy should in most cases be even better in the interior of each region.

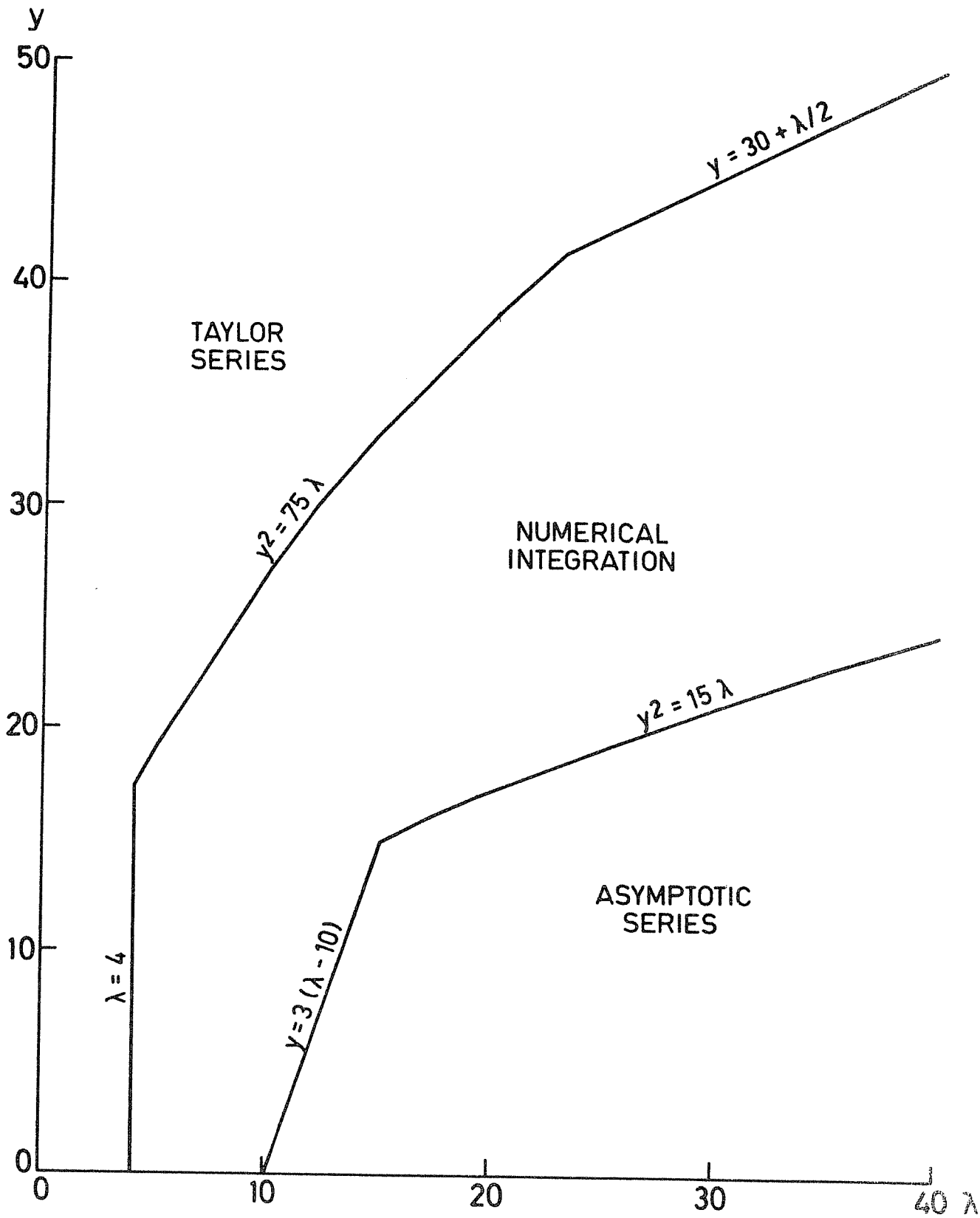

Figure 5 The different methods used to compute $R(y, \lambda)$ are indicated on this plot of the λ y-plane.

```
C     TIME= 82/04/29 - 10.29.29

      SUBROUTINE RYLA(Y,AL,RC)
      COMPLEX Y, RC(2,2)
C
C        **** CHOOSE METHOD OF EVALUATION  ****
      IF(AL.LT.4) GOTO 1
      AY=CABS(Y)
      IF(AY**2.GT.75.*AL) GOTO 1
      IF(AY.GT.30.+AL/2.) GOTO 1
C
      IF(AY**2.LT.15.*AL.AND.AL.GE.15.)     GOTO 2
      IF(AY.LE.3.*(AL-10.).AND.AL.LE.15.)) GOTO 2
C
C          ****  NUMERICAL INTEGRATION  ****
      CALL RINT(Y,AL,RC)
      RETURN
C
C          **** TAYLOR SERIES  ****
    1 CALL RTAY(Y,AL,RC)
      RETURN
C
C          **** ASYMPTOTIC SERIES  ****
    2 CALL RASY(Y,AL,RC)
      RETURN
      END
```

```
C       TIME= 82/04/29 - 10.31.47


        SUBROUTINE RTAY(Y,AL,RC)
        COMPLEX Y,Y2,RC(2,2),PN,PYN,COT
C                ******** TAYLOR SERIES ********
        Y2=Y*Y
   10   PN=Y/(Y2-1.)
        PYN=-Y*(Y2+1.)/(Y2-1.)**2
        RC(1,1)=PN
        RC(1,2)=PN
        RC(2,1)=PYN
        RC(2,2)=PYN
C
        DO 1 I=2,100
        COT=(2*I-1)/(Y2-I**2)*AL
        PYN=COT*(PYN-2.*Y2/(Y2-I**2)*PN)
        PN=COT*PN
        RC(1,1)=RC(1,1)+PN
        RC(2,1)=RC(2,1)+PYN
        RC(1,2)=RC(1,2)+I*PN
        RC(2,2)=RC(2,2)+I*PYN
        T=CABS(PN)*1.E8
        IF(T.LT.CABS(RC(1,1))) GOTO 2
    1   CONTINUE
    2   CONTINUE
        END
```

```
C     TIME= 82/04/29 - 10.31.47


      SUBROUTINE RASY(Y,AL,RC)
      COMPLEX Y,Y2,COT,P,PY,PP,PPY,PN,PYN,QN,QYN,RC(2,2)
C           ******** ASYMPTOTIC SERIES ********
      PI=3.14159265358979
      Y2=Y*Y
      COT=CCOS(PI*Y)/CSIN(PI*Y)
      C=1.E99
      PN=-Y/AL
      PYN=PN
      A=1./(AL*SQRT(2.*PI*AL))
      QN=PI*Y2*COT*A
      QYN=QN*(2.-Y*PI*COT)-Y*PI**2*Y2*A
C
      P=PN+QN
      PY=PYN+QYN
      PP=-PN-1.5*QN
      PPY=-PYN-1.5*QYN
      AY=CABS(Y)+2.
C
      DO 4 N=1,100
      M=N-1
      PYN=(PYN*(M*M-Y2)-2.*Y2*PN)/((2*M+1)*AL)
      PN =PN*(M*M-Y2)/((2*M+1)*AL)
      QYN=(QYN*((M+.5)**2-Y2)-2.*Y2*QN)/(2.*N*AL)
      QN =QN*((M+.5)**2-Y2)/(2.*N*AL)
      IF(M.LT.AY) GOTO 3
      C=N*(CABS(PN)+CABS(QN))
      IF(C.LE.1.E-7*CABS(PP)) GOTO 5
      IF(C.GE.T) GOTO 5
    3 P  =P  + PN  + QN
      PY =PY + PYN + QYN
      PP =PP -(N + 1.)*PN -(N +1.5)*QN
      PPY=PPY-(N + 1.)*PYN-(N +1.5)*QYN
    4 T=C
C
    5 RC(1,1)=P + PN + QN
      RC(2,1)=PY+ PYN+ QYN
      RC(1,2)=PP+P
      RC(2,2)=PPY+PY
      RETURN
      END
```

```
C      TIME= 82/04/29 - 10.31.47


       SUBROUTINE RINT(Y,AL,RC)
C             ******** NUMERICAL INTEGRATION ********
       COMPLEX RC(2,2),Y,COT,D,EXF,F,H,O,P,R,RY,RP,RPY,S
       DIMENSION A(16), W(16)
C        ABSKISSAS FOR GAUSSIAN INTEGRATION
       DATA A/                  -.98940 09349 91649,-.94457 50230 73232,
      D -.86563 12023 87831, -.75540 44083 55003,-.61787 62444 02643,
      D -.45801 67776 57227, -.28160 35507 79258,-.09501 25098 37637,
      D                       .98940 09349 91649, .94457 50230 73232,
      D  .86563 12023 87831,  .75540 44083 55003, .61787 62444 02643,
      D  .45801 67776 57227,  .28160 35507 79258, .09501 25098 37637/,
C      WEIGHT FACTORS
      D   W /                 .02715 24594 11754, .06225 35239 38647,
      D  .09515 85116 82492,  .12462 89712 55533, .14959 59888 16576,
      D  .16915 65193 95002,  .18260 34150 44923, .18945 06104 55068,
      D                       .02715 24594 11754, .06225 35239 38647,
      D  .09515 85116 82492,  .12462 89712 55533, .14959 59888 16576,
      D  .16915 65193 95002,  .18260 34150 44923, .18945 06104 55068/,
      D PI/3.14159265358979/
C
       DO 1 I=1,4
     1 RC(I)=(0.,0.)
       YA=AIMAG(Y)
       YR=REAL(Y)
       UL=PI-2.8*Y/(36.+Y)
       COT=CCOS(PI*Y)/CSIN(PI*Y)
       D=PI*(1.+COT**2)
       C=YR/AL
       XO=ALOG(C+SQRT(1.+C**2))
C
       DO 10 I=1,16
       X=UL/2.*(1.+A(I))
       Z=SIN(X)
       C=COS(X)
       G=YR/AL*X/Z
       T=SQRT(1.+G**2)
       B=ALOG(G+T)
       G=(1./X-C/Z)*G/T
       T=AL*(T*C-1.)
       Z=EXP(X*YA)
       C=.5*(Z+1./Z)
       S=(0.,.5)*(Z-1./Z)
       F=COT+G
       H=1.-G*COT
       EXF=CEXP(T-Y*B)
       O=B*C+X*S
       P=X*C-B*S
       XY=X*YR
       R=(F*C+H*S)*EXF
       RY=(F*O-H*P+D*(C-G*S))*EXF
       RP=((F*T-H*XY)*C+(H*T+F*XY)*S)*EXF
       RPY=(F*(T*O-XY*P)-H*(T*P+XY*O)+((T+XY*G)*C-(G*T-XY)*S)*D)*EXF
C
       X=XO/2.*(1.+A(I))
       Z=EXP(X)
       C=(Z+1./Z)/2.-1.
       P=CEXP(AL*C-Y*X)
       RC(1,1)=RC(1,1)+W(I)*(UL*R+XO*P)
       RC(2,1)=RC(2,1)-W(I)*(UL*RY+XO*X*P)
```

```
      RC(1,2)=RC(1,2)+W(I)*(UL*RP+XO*AL*C*P)
      RC(2,2)=RC(2,2)-W(I)*(UL*RPY+XO*AL*X*C*P)
   10 CONTINUE
C
      O=Y/AL
      P=Y**2/2.
      RC(1,1)=O*(Y*RC(1,1)/2.-1.)
      RC(2,1)=2.*RC(1,1)+O*(P*RC(2,1)+1.)
      RC(1,2)=Y*O*RC(1,2)/2.
      RC(2,2)=2.*RC(1,2)+O*P*RC(2,2)
      END
```

V. The suceptibility tensor $\underline{\underline{\chi}}^{\alpha}$; SUBROUTINE CHI

The suceptibility tensor $\underline{\underline{\chi}}^{\alpha}$, defined by Equation (II-16) is evaluated in the subroutine CHI, and the result is returned in the (6, 4)-array XSI.

The parameters for the approximation of the Z-function are stored in the arrays B (residues) and C (poles). If the argument of Z(s) is to far down in the complex s-plane, the error condition IERR=1 is returned to the calling routine.

For each term in the partial fractions expansion, the value of $R(y, \alpha\lambda)$, where $y = x - z(c_1 + V_d)$, is obtained from SUBROUTINE RYLA.

The functions

$$\psi = (1 + \alpha z c_1 y^{-1}) R(y,\alpha\lambda) \tag{V-1}$$

and

$$\psi' = (1 + \alpha z c_1 y^{-1}) R'(y,\alpha\lambda) \tag{V-2}$$

are stored in the variables PS and PSP respectively, and the components of $\underline{\underline{\chi}}^{\alpha}$ are formed according to Equation (II-16).

When KOL$\geq$2, the x-derivative of $\underline{\underline{\chi}}^{\alpha}$ is also evaluated. We first form

$$PSY = (1 + \alpha z c_1 y^{-1}) y \frac{\partial R}{\partial y} \tag{V-3}$$

and

$$PPY = (1 + \alpha z c_1 y^{-1}) y \frac{\partial R}{\partial y}' \tag{V-4}$$

and then obtain

$$DP = x \frac{\partial \psi}{\partial x} = \frac{x}{y} (PSY - \alpha z c_1 y^{-1} R) \tag{V-5}$$

and

$$DPP = x \frac{\partial \psi}{\partial x}' = \frac{x}{y} (PPY - \alpha z c_1 y^{-1} R') \tag{V-6}$$

Defining $D_1 = c_1 + V_d$, the components of $x\,\partial\underline{\underline{\chi}}^{\alpha}/\partial x$ may be written as

$$x\frac{\partial\chi^{\alpha}_{11}}{\partial x} = \alpha\sum_{l=1}^{8} B_l[x\psi + y\ DP]$$

$$x\frac{\partial\chi_{12}}{\partial x} = i\alpha\sum_{l=1}^{8} B_l\ DPP$$

$$x\frac{\partial\chi_{13}}{\partial x} = \alpha P\sum_{l=1}^{8} B_l D_l DP \qquad \text{(V-7)}$$

$$x\frac{\partial\chi_{22}}{\partial x} = x\frac{\partial\chi_{11}}{\partial x} - 2\alpha^2\lambda\sum_{l=1}^{8} B_l y^{-1}[DPP - xy^{-1}\psi']$$

$$x\frac{\partial\chi_{23}}{\partial x} = i\alpha p\sum_{l=1}^{8} B_l D_l y^{-1}[DPP - xy^{-1}\psi']$$

$$x\frac{\partial\chi_{33}}{\partial x} = 2\alpha\sum_{l=1}^{8} B_l D_l^2 y^{-1}[x + \lambda DP - xy^{-1}(x - zV_d + \lambda\psi)]$$

The z-derivative of $\underline{\underline{\chi}}^{\alpha}$ is evaluated when KOL$\geq$3. Noticing that $\partial y/\partial z = -D_1$ we obtain

$$DP = z\frac{\partial\psi}{\partial z} = \frac{z}{y}[\alpha c_1 xy^{-1}R - D_1 PSY] \qquad \text{(V-8)}$$

and

$$DPP = z\frac{\partial\psi}{\partial z} = \frac{z}{y}[\alpha c_1 xy^{-1}R' - D_1 PPY] \qquad \text{(V-9)}$$

and the expressions for $z\,\partial\underline{\underline{\chi}}^{\alpha}/\partial z$ are easily written down in terms of these functions.

When KOL$\geq$4, the p-derivatives are also evaluated. SUBROUTINE RYLA is first called to evaluate $R(y-1, \alpha\lambda)$, and then

$$DP = p \frac{\partial \psi}{\partial p} = 2(\psi' - \psi) \qquad (V-10)$$

and

$$DPP = p \frac{\partial \psi'}{\partial p} = 2 \left[\frac{y^2}{y^2-1} \alpha\lambda R'(y-1, \alpha\lambda) - \alpha\lambda\psi'\right] - yDP \qquad (V-11)$$

are formed. The formulas for $P \partial \underline{\underline{\chi}}^{\alpha} / \partial p$ are then trivially derived.

```
C     TIME= 82/04/29 - 10.31.47

      SUBROUTINE CHI(XSI,J,IB,KOL,IERR)
C                  ARGUMENTS: XSI     CONTAINS THE SUCEPTIBILITY TENSOR
C                                     ON RETURN.
C                             J       COMPONENT NUMBER.
C                             IB      INDEX FOR AA (ALPHA)
C                             KOL     DETERMINES WHETHER DERIVATIVES
C                                     SHOULD BE EVALUATED.
C                             IERR    ERROR FLAG, IS SET =1 IF DAMPING
C                                     IS TO STRONG.
      COMPLEX X,XX,XY,AI,BL,CL,DL,BLY,RC(2,2),XSI(6,4),
     2  B(8),C(8),PS,PSP,PSY,PPY,DP,DPP,Y,ZY
      COMMON /XPZ/ XX(6),PP(6),ZZ(6),AA(6,2),DD(6),ASS(6),VD(6)
C          **** RESIDUES FOR PADE APPROXIMANT ****
      DATA B/(-1.7340124574718 26E-2,-4.630639291680322E-2),
     B        (-1.7340124574718 26E-2, 4.630639291680322E-2),
     B        (-7.399169923225014E-1, 8.395179978099844E-1),
     B        (-7.399169923225014E-1,-8.395179978099844E-1),
     B        ( 5.840628642184073   , 9.536009057643667E-1),
     B        ( 5.840628642184073   ,-9.536009057643667E-1),
     B        (-5.583371525286853   ,-1.120854319126599 E1),
     B        (-5.583371525286853   , 1.120854319126599 E1)/,
C         **** POLES OF PADE APPROXIMANT ****
     C     C/ ( 2.237687789201900,  -1.625940856173727),
     C        (-2.237687789201900,  -1.625940856173727),
     C        ( 1.465234126106004,  -1.789620129162444),
     C        (-1.465234126106004,  -1.789620129162444),
     C        ( .8392539817232638,  -1.891995045765206),
     C        (-.8392539817232638,  -1.891995045765206),
     C        ( .2739362226285564,  -1.941786875844713),
     C        (-.2739362226285564,  -1.941786875844713)/
C
      X=XX(J)
      Z=ZZ(J)
      P=PP(J)
      A=AA(J,IB)
      VZ=VD(J)*Z
      AI=(0.,1.)*A
      IF(ASS(J).EQ.0.) AI=-AI
      AL=.5*P*P
      ALA=A*AL
C
      DO 1 I=1,24
    1 XSI(I)=(0.,0.)
      XI=AIMAG(X)
      IF(XI.GE.0.) GOTO 3
      NX=X
      RX=X-NX
C                      TEST FOR STRONG DAMPING
      IF(RX.GT..5) RX=1.-RX
      IF(XI.GE.-.6*RX.OR.XI.GE.-Z) GOTO 3
      IERR=1
      RETURN
    3 XSI(1,1)=A
      XSI(6,1)=A*(1.+2.*VD(J)**2)
      DO 4 L=1,8
      BL=B(L)
      CL=C(L)
      DL=CL+VD(J)
      Y=X-DL*Z
```

```
      BLY=BL/Y
C         ****** EVALUATE THE R-FUNCTION ******
      CALL RYLA(Y,ALA,RC)
      XY=1.+A*Z*CL/Y
      PS=XY*RC(1,1)
      PSP=XY*RC(1,2)
C               ****** FORM SUCEPTIBILITY TENSOR ******
      XSI(1,1)=XSI(1,1)+A*BL*Y*PS
      XSI(2,1)=XSI(2,1)+AI*BL*PSP
      XSI(3,1)=XSI(3,1)+A*P*BL*DL*PS
      XSI(4,1)=XSI(4,1)+BLY*PSP
      XSI(5,1)=XSI(5,1)+AI*P*BLY*DL*PSP
      XSI(6,1)=XSI(6,1)+2.*A*BLY*DL**2*(X-VZ+AL*PS)
C
      IF(KOL.LE.1) GOTO 4
C             ****** FORM X-DERIVATIVES OF XSI ******
      PSY=XY*RC(2,1)
      PPY=XY*RC(2,2)
      XY=X/Y
      DP =XY*(PSY-A*Z*CL/Y*RC(1,1))
      DPP=XY*(PPY-A*Z*CL/Y*RC(1,2))
      XSI(1,2)=XSI(1,2)+A*BL*(Y*DP+X*PS)
      XSI(2,2)=XSI(2,2)+AI*BL*DPP
      XSI(3,2)=XSI(3,2)+A*P*BL*DL*DP
      XSI(4,2)=XSI(4,2)+BLY*(DPP-XY*PSP)
      XSI(5,2)=XSI(5,2)+AI*P*BLY*DL*(DPP-XY*PSP)
      XSI(6,2)=XSI(6,2)+2.*A*BLY*DL**2*(X+AL*DP-XY*(X-VZ+AL*PS))
C
      IF(KOL.LE.2) GOTO 4
C             ****** FORM Z-DERIVATIVES OF XSI ******
      ZY=Z/Y
      DP =ZY*(A*CL*XY*RC(1,1)-DL*PSY)
      DPP=ZY*(A*CL*XY*RC(1,2)-DL*PPY)
      ZY=DL*ZY
      XSI(1,3)=XSI(1,3)+A*BL*Y*(DP-ZY*PS)
      XSI(2,3)=XSI(2,3)+AI*BL*DPP
      XSI(3,3)=XSI(3,3)+A*P*BL*DL*DP
      XSI(4,3)=XSI(4,3)+BLY*(DPP+ZY*PSP)
      XSI(5,3)=XSI(5,3)+AI*P*BLY*DL*(DPP+ZY*PSP)
      XSI(6,3)=XSI(6,3)+2.*A*BLY*DL**2*(AL*DP-VZ+ZY*(X-VZ+AL*PS))
C
      IF(KOL.LE.3) GOTO 4
C             ****** FORM P-DERIVATIVES OF XSI ******
      CALL RYLA(Y-1.,ALA,RC)
      DP=2.*(PSP-PS)
      DPP=2.*AL*((Y/(Y-1.))**2*RC(1,2)-PSP)-Y*DP
      XSI(1,4)=XSI(1,4)+A*BL*Y*DP
      XSI(2,4)=XSI(2,4)+AI*BL*DPP
      XSI(3,4)=XSI(3,4)+A*P*BL*DL*(DP+PS)
      XSI(4,4)=XSI(4,4)+BLY*(2.*PSP+DPP)
      XSI(5,4)=XSI(5,4)+AI*P*BLY*DL*(PSP+DPP)
      XSI(6,4)=XSI(6,4)+4.*A*BLY*DL**2*AL*PSP
    4 CONTINUE
C      **** COMPLETE XSI(4, ) ****
      XSI(4,1)=XSI(1,1)-2.*A**2*AL*XSI(4,1)
      XSI(4,2)=XSI(1,2)-2.*A**2*AL*XSI(4,2)
      XSI(4,3)=XSI(1,3)-2.*A**2*AL*XSI(4,3)
      XSI(4,4)=XSI(1,4)-2.*A**2*AL*XSI(4,4)
      END
```

VI. The dispersion functon $D(\omega, \underline{k})$; SUBROUTINE DIFU

In this routine, the dielectric tensor $\underline{\underline{\varepsilon}}(\omega, \underline{k})$ is first formed according to Equation (II-11), using the values of $\underline{\underline{\chi}}^{\alpha}(\omega, \underline{k})$ obtained from SUBROUTINE CHI. The components of $\underline{\underline{\varepsilon}}$ and the derivatives $\underline{\underline{\varepsilon}}_x = x\,\partial\underline{\underline{\varepsilon}}/\partial x$, $\underline{\underline{\varepsilon}}_z = z\,\partial\underline{\underline{\varepsilon}}/\partial z$, and $\underline{\underline{\varepsilon}}_p = p\,\partial\underline{\underline{\varepsilon}}/\partial p$ are stored in the (6, 4) array E. After evaluating the refractive index, the value of the dispersion function $D(\omega, \underline{k})$ is calculated as described by Equation (II-20). The corresponding formulas for the derivatives of D are:

$$\Omega_1 \frac{\partial D}{\partial \omega} = \frac{\Omega_1}{\omega}\left\{(\mu^2-\varepsilon_{22})A_x-(2\mu^2+\varepsilon_{22,x})A - B_x + C_x\right\} \qquad \text{(VI-1)}$$

where

$$A_x = \omega\frac{\partial A}{\partial \omega} = (\varepsilon_{11,x}-2\varepsilon_{11})\mu_1^2 + 2(\varepsilon_{13,x}-2\varepsilon_{13})\mu_1\mu_3 + (\varepsilon_{33,x}-2\varepsilon_{33})\mu_3^2$$

$$B_x = \omega\frac{\partial B}{\partial \omega} = 2(\mu_3\varepsilon_{23}-\mu_1\varepsilon_{12})\,[\mu_3(\varepsilon_{23,x}-\varepsilon_{23}) - \mu_1(\varepsilon_{12,x}-\varepsilon_{12})] + \mu^2\,[(\varepsilon_{11,x}-\varepsilon_{11})\varepsilon_{33}-2(\varepsilon_{13,x}-\varepsilon_{13})\varepsilon_{13}+(\varepsilon_{33,x}-\varepsilon_{33})\varepsilon_{11}] \qquad \text{(VI-2)}$$

$$C_x = \omega\frac{\partial C}{\partial \omega} = (\varepsilon_{11,x}\varepsilon_{33}+\varepsilon_{11}\varepsilon_{33,x}-2\varepsilon_{13}\varepsilon_{13,x})\varepsilon_{22} + (\varepsilon_{11}\varepsilon_{33}-\varepsilon_{13}^2)\varepsilon_{22,x}+\varepsilon_{23}(\varepsilon_{11,x}\varepsilon_{23}+2\varepsilon_{11}\varepsilon_{23,x}+\varepsilon_{12}\varepsilon_{13,x}+ 2\varepsilon_{12,x}\varepsilon_{13}) + \varepsilon_{12}(\varepsilon_{33,x}\varepsilon_{12}+2\varepsilon_{33}\varepsilon_{12,x}+\varepsilon_{23}\varepsilon_{13,x} + 2\varepsilon_{23,x}\varepsilon_{13})$$

$$D_z = \frac{\Omega_1}{k_{\parallel}V_1}\left\{(\mu^2-\varepsilon_{22})A_z + (2\mu_3^2 - \varepsilon_{22,z})\,A - B_z + C_z\right\} \qquad \text{(VI-3)}$$

where

$$A_z = k_{\parallel} \frac{\partial A}{\partial k_{\parallel}} = \mu_1^2 \varepsilon_{11,z} + 2\mu_1\mu_3(\varepsilon_{13,z}+\varepsilon_{13}) + \mu_3^2(\varepsilon_{33,z}+2\varepsilon_{33})$$

$$B_z = k_{\parallel} \frac{\partial B}{\partial k_{\parallel}} = 2(\mu_3\varepsilon_{23}-\mu_1\varepsilon_{12})[\mu_3(\varepsilon_{23,z}+\varepsilon_{23}) - \mu_1\varepsilon_{12}] +$$

$$+ 2\mu_3^2 (\varepsilon_{11}\varepsilon_{33}-\varepsilon_{13}^2)+ \mu^2(\varepsilon_{11,z}\varepsilon_{33}+\varepsilon_{11}\varepsilon_{33,z}-2\varepsilon_{13}\varepsilon_{13,z})$$

$$C_z = k_{\parallel} \frac{\partial C}{\partial k_{\parallel}} = (\varepsilon_{11,z}\varepsilon_{33}+\varepsilon_{11}\varepsilon_{33,z}-2\varepsilon_{13}\varepsilon_{13,z})\varepsilon_{22} + \quad \text{(VI-4)}$$

$$+ (\varepsilon_{11}\varepsilon_{33}-\varepsilon_{13}^2)\varepsilon_{22,z} + \varepsilon_{23}(\varepsilon_{11,z}\varepsilon_{23}+2\varepsilon_{11}\varepsilon_{23,z}+$$

$$+ \varepsilon_{12}\varepsilon_{13,z}+2\varepsilon_{12,z}\varepsilon_{13}) + \varepsilon_{12}(\varepsilon_{33,z}\varepsilon_{12}+2\varepsilon_{33}\varepsilon_{12,z} +$$

$$+ \varepsilon_{23}\varepsilon_{13,z}+2\varepsilon_{23,z}\varepsilon_{13})$$

and

$$D_P = \frac{\Omega_1}{V_1} \frac{\partial D}{\partial k_{\perp}} = \frac{\Omega_1}{k_{\perp}V_1} \left\{(\mu^2-\varepsilon_{22})A_P + (2\mu_1^2-\varepsilon_{22,P})A-B_P+C_P\right\} \quad \text{(VI-5)}$$

where

$$A_P = k_{\perp} \frac{\partial A}{\partial k_{\perp}} = \mu_1^2(\varepsilon_{11,P}+2\varepsilon_{11}) + 2\mu_1\mu_3(\varepsilon_{13,P}+\varepsilon_{13}) + \mu_3^2\varepsilon_{33,P}$$

$$B_P = k_{\perp} \frac{\partial B}{\partial k_{\perp}} = 2(\mu_3\varepsilon_{23}-\mu_1\varepsilon_{12})[\mu_3\varepsilon_{23,P}-\mu_1(\varepsilon_{12,P}+\varepsilon_{12})] +$$

$$+ 2\mu_1^2(\varepsilon_{11}\varepsilon_{33}-\varepsilon_{13}^2) + \mu^2(\varepsilon_{11,P}\varepsilon_{33}+\varepsilon_{11}\varepsilon_{33,P}-2\varepsilon_{13}\varepsilon_{13,P})$$

$$C_p = k_{\perp} \frac{\partial C}{\partial k_{\perp}} = (\varepsilon_{11,P}\varepsilon_{33}+\varepsilon_{11}\varepsilon_{33,P}-2\varepsilon_{13}\varepsilon_{13,P})\varepsilon_{22} + \quad \text{(VI-6)}$$

$$+ (\varepsilon_{11}\varepsilon_{33}-\varepsilon_{13}^2)\varepsilon_{22,P} + \varepsilon_{23}[\varepsilon_{11,P}\varepsilon_{23}+2\varepsilon_{11}\varepsilon_{23,P} +$$

$$+ \varepsilon_{12}\varepsilon_{13,P}+2\varepsilon_{12,P}\varepsilon_{13}] + \varepsilon_{12}[\varepsilon_{33,P}\varepsilon_{12}+2\varepsilon_{33}\varepsilon_{12,P} +$$

$$+ \varepsilon_{23}\varepsilon_{13,P}+2\varepsilon_{23,P}\varepsilon_{13}]$$

In order to compute the electric and magnetic field components we arbitarily specify $E_1 = 1$ and solve the wave equation (II-1)

$$\underline{\underline{D}}(\omega, \underline{k}) \cdot \underline{E} = 0$$

for the remaining field components. The solution is

$$\begin{cases} E_2 = \dfrac{(\mu_3^2 - \varepsilon_{xx})\varepsilon_{xy} - (\mu_1\mu_3 + \varepsilon_{xz})\varepsilon_{xy}}{\varepsilon_{xy}\varepsilon_{yz} + (\mu_1\mu_3 + \varepsilon_{xz})(\mu^2 - \varepsilon_{yy})} \\ E_3 = -\dfrac{(\mu_3^2 - \varepsilon_{xx})\varepsilon_{yz} - (\mu_1\mu_3 + \varepsilon_{xz})\varepsilon_{xy}}{(\mu_1^2 - \varepsilon_{zz})\varepsilon_{xy} - (\mu_1\mu_3 + \varepsilon_{xz})\varepsilon_{yz}} \end{cases} \qquad \text{(VI-7)}$$

In special cases such as $k_{\|} = 0$ or $k_{\perp} = 0$ there are waves with $E_1 = 0$ and the above solution will not be valid. For parallel propagation we have either an electrostatic wave with $E_1 = E_2 = 0$, $E_3 = 1$ or circularly polarized waves with $E_2 = \pm iE_1$, $E_3 = 0$. Waves propagating perpendicular to the magnetic field are either ordinary with $E_1 = E_2 = 0$, $E_3 = 1$ or extraordinary with $E_3 = 0$, $E_2 = -\varepsilon_{11}/\varepsilon_{12}E_1$. In all cases, the electrtic field amplitude is normalized so that $|\underline{E}|^2 = 1$ and the magnetic field is calculated from Faraday's law

$$\underline{B} = (\underline{k} \times \underline{E})/\omega \, .$$

```
C     TIME= 82/04/29 - 10.31.47


      SUBROUTINE DIFU(KOL,JMA,IERR)
C                     ARGUMENTS: KOL      =2 D AND ITS X-DERIVATIVE COMPUTED.
C                                         =4 ALL DERIVATIVES AND WAVE FIELDS.
C                                JMA      =1 - 6, NUMBER OF COMPONENTS.
C                                IERR        ERROR FLAG.
      COMPLEX X,XX,E(6,4),XSI(6,4),XP,DF,U1,U2,U3,U12,U13,U32,
     2 A,B,C,DA,DB,DC,D,DX,DZ,DP,EFL(3),BFL(3)
      COMMON /XPZ/ XX(6),PP(6),ZZ(6),AA(6,2),DD(6),ASS(6),VD(6),
     C DN(6),TA(6),XP(6),CV
      COMMON /COUT/X,P,Z,EFL,BFL,D,DX,DZ,DP,E
C
C                ********** FORM DIELECTRIC TENSOR **********
      DO 1 K=1,4
      DO 1 I=1,6
    1 E(I,K)=(0.,0.)
      E(1,1)=1.
      E(4,1)=1.
      E(6,1)=1.
C
      DO 5 J=1,JMA
      E(1,1)=E(1,1)-XP(J)
      E(4,1)=E(4,1)-XP(J)
      E(6,1)=E(6,1)-XP(J)
      IF(AA(J,1).NE.AA(J,2)) GOTO 2
      AA(J,2)=0.
      DD(J)=1.
    2 IB=1
      DF=XP(J)/(AA(J,1)*(AA(J,1)-AA(J,2)))
      Q=AA(J,1)-DD(J)*AA(J,2)
    3 CALL CHI(XSI,J,IB,KOL,IERR)
      IF(IERR.NE.0) RETURN
      DO 4 K=1,KOL
      DO 4 I=1,6
    4 E(I,K)=E(I,K)+DF*Q*XSI(I,K)
C
      IF(IB.EQ.2) GOTO 5
      Q=(DD(J)-1.)*AA(J,1)
      IF(Q.EQ.0.) GOTO 5
      IB=2
      GOTO 3
    5 CONTINUE
C                  *** DIELECTRIC TENSOR COMPUTED ***
C
C       ******** FORM REFRACTIVE INDEX, CV=SPEED OF LIGHT/THERM. SPEED. ****
      U1=PP(1)*CV/XX(1)
      U3=ZZ(1)*CV/XX(1)
      U12=U1**2
      U32=U3**2
      U2=U12+U32
      U13=2.*U1*U3
C
C                  ********* FORM DISPERSION FUNCTION ********
      A=U12*E(1,1)+U13*E(3,1)+U32*E(6,1)
      B=U2*(E(1,1)*E(6,1)-E(3,1)**2)+(U3*E(5,1)+U1*E(2,1))**2
      C=(E(1,1)*E(6,1)-E(3,1)**2)*E(4,1)+E(6,1)*E(2,1)**2
      C=C+(E(1,1)*E(5,1)+2.*E(2,1)*E(3,1))*E(5,1)
C
      D=(U2-E(4,1))*A-B+C
      IF(KOL.LE.1) RETURN
```

```
C        ****** COMPLETE X-DERIVATIVE OF DIELECTRIC TENSOR ******
      E(1,2)=E(1,2)-2.*(E(1,1)-1.)
      E(2,2)=E(2,2)-2.* E(2,1)
      E(3,2)=E(3,2)-2.* E(3,1)
      E(4,2)=E(4,2)-2.*(E(4,1)-1.)
      E(5,2)=E(5,2)-2.* E(5,1)
      E(6,2)=E(6,2)-2.*(E(6,1)-1.)
C          ******* X-DERIVATIVE OF DISPERSION FUNCTION *******
      DA=(E(1,2)-2.*E(1,1))*U12+U13*(E(3,2)-2.*E(3,1))+
     +    (E(6,2)-2.*E(6,1))*U32
      DB=2.*(U3*E(5,1)-U1*E(2,1))*(U3*(E(5,2)-E(5,1))-U1*(E(2,2)-E(2,1)
     +))+U2*((E(1,2)-E(1,1))*E(6,1)-2.*(E(3,2)-E(3,1))*E(3,1)+
     +(E(6,2)-E(6,1))*E(1,1))
      DC=(E(1,2)*E(6,1)+E(1,1)*E(6,2)-2.*E(3,1)*E(3,2))*E(4,1)
      DC=DC+(E(1,1)*E(6,1)-E(3,1)**2)*E(4,2)
      DC=DC+E(5,1)*(E(1,2)*E(5,1)+2.*E(1,1)*E(5,2)+
     +E(2,1)*E(3,2)+2.*E(2,2)*E(3,1))
      DC=DC+E(2,1)*(E(6,2)*E(2,1)+2.*E(6,1)*E(2,2)+
     +E(5,1)*E(3,2)+2.*E(5,2)*E(3,1))
C
      DX=((U2-E(4,1))*DA-(2.*U2+E(4,2))*A-DB+DC)/XX(1)
      IF(KOL.LE.2) RETURN
      DZ=(0.,0.)
      IF(ZZ(1).EQ.0.) GOTO 6
C          ******* Z-DERIVATIVE OF DISPERSION FUNCTION ******
      DA=U12*E(1,3)+U13*(E(3,3)+E(3,1))+U32*(E(6,3)+2.*E(6,1))
      DB=2.*(U3*E(5,1)-U1*E(3,1))*(U3*(E(5,3)+E(5,1))-U1*E(2,1))+
     +2.*U32*(E(1,1)*E(6,1)-E(3,1)**2)+
     +U2*(E(1,3)*E(6,1)+E(1,1)*E(6,3)-2.*E(3,1)*E(3,3))
      DC=(E(1,3)*E(6,1)+E(1,1)*E(6,3)-2.*E(3,1)*E(3,3))*E(4,1)
      DC=DC+(E(1,1)*E(6,1)-E(3,1)**2)*E(4,3)
      DC=DC+E(5,1)*(E(1,3)*E(5,1)+2.*E(1,1)*E(5,3)+
     +E(2,1)*E(3,3)+2.*E(2,3)*E(3,1))
      DC=DC+E(2,1)*(E(6,3)*E(2,1)+2.*E(6,1)*E(2,3)+
     +E(5,1)*E(3,3)+2.*E(5,3)*E(3,1))
C
      DZ=((U2-E(4,1))*DA+(2.*U32-E(4,3))*A-DB+DC)/ZZ(1)
    6 IF(KOL.LE.3) RETURN
C          ******* P-DERIVATIVE OF DISPERSION FUNCTION ******
      DP=(0.,0.)
      IF(PP(1).EQ.0.) GOTO 7
C
      DA=U12*(E(1,4)+2.*E(1,1))+U13*(E(3,4)+E(3,1))+U32*E(6,4)
      DB=2.*(U3*E(5,1)-U1*E(2,1))*(U3*E(5,4)-U1*(E(2,4)+E(2,1)))+
     +2.*U12*(E(1,1)*E(6,1)-E(3,1)**2)+
     +U2*(E(1,4)*E(6,1)+E(1,1)*E(6,4)-2.*E(3,1)*E(3,4))
      DC=(E(1,4)*E(6,1)+E(1,1)*E(6,4)-2.*E(3,1)*E(3,4))*E(4,1)
      DC=DC+(E(1,1)*E(6,1)-E(3,1)**2)*E(4,4)
      DC=DC+E(5,1)*(E(1,4)*E(5,1)+2.*E(1,1)*E(5,4)+
     +E(2,1)*E(3,4)+2.*E(2,4)*E(3,1))
      DC=DC+E(2,1)*(E(6,4)*E(2,1)+2.*E(6,1)*E(2,4)+
     +E(5,1)*E(3,4)+2.*E(5,4)*E(3,1))
C
      DP=((U2-E(4,1))*DA+(2.*U12-E(4,4))*A-DB+DC)/PP(1)
C                ******** COMPUTE ELECTRIC FIELD ********
    7 U13=U13/2.
      IF(U13.EQ.0.) GOTO 8
      A=(U32-E(1,1))*E(5,1)-(U13+E(3,1))*E(2,1)
      B=A/((U13+E(3,1))*(U2-E(4,1))+E(5,1)*E(2,1))
      C=A/((U13+E(3,1))*E(5,1)-(U12-E(6,1))*E(2,1))
      A=1.
      GOTO 10
```

```
    8 IF(CABS(U2-E(6,1)).LT.1.E-3) GOTO 9
      A=1.
      B=(0.,1.)
      C=CABS(E(1,1)-U32+B*E(2,1))
      IF(CABS(E(1,1)-U32-B*E(2,1)).LT.C) B=(0.,-1.)
      IF(U3.EQ.0.) B=-E(1,1)/E(2,1)
      C=0.
      GOTO 10
    9 A=0.
      B=0.
      C=1.
   10 Q=A+B*CONJG(B)+C*CONJG(C)
      Q=SQRT(Q)
      EFL(1)=A/Q
      EFL(2)=B/Q
      EFL(3)=C/Q
      V=1./299.7925
C       THE ELECTRIC FIELD IS 1 MV/M. THE MAGNETIC FIELD WILL BE IN GAMMA.
      BFL(1)=-V*U3*EFL(2)
      BFL(2)=V*(U3*EFL(1)-U1*EFL(3))
      BFL(3)=V*U1*EFL(2)
      END
```

VII. The main program and the input/output

PROGRAM WHAMP

The main program described here is mainly intended to be used in general surveys of the dispersion relation for various $\underline{k}$-vectors and plasma parameters. The description is rather brief, leaving out a lot of technical details and rather emphasizing the points which are essential for any program calling SUBROUTINE DIFU. In particular, it should be useful as an example or starting point when writing specialized programs for particular applications.

The interaction between the program and the user is channeled through the input/output routines described in the next paragraph. In the input routine, TYPIN, the plasma parameters may be changed. An initial value XOI for the frequency must be specified there, as well as the wave numbers for which the dispersion relation is to be solved. The wave vector components are specified by the arrays PM and ZM, and in general a rectangle PM(1) $\leq k_\perp V_1/\Omega_1 \leq$ PM(2) and ZM(1) $\leq k_\parallel V_1/\Omega_1 \leq$ ZM(2) in the $\underline{k}$-plane is covered by a net with mesh size PM(3) by ZM(3). If the value of PZL is set to one in SUBROUTINE TYPIN, PM and ZM are interpreted as the logarithms of the wave numbers, that is PM(1) $\leq \log(k_\perp V/\Omega_1) \leq$ PM(2) etc. The size of the meshes will then vary logarithmically.

In the first section of code, a number of variables are given their properly normalized values and the plasma parameters are printed. The number of the last component with non-zero density is stored in the variable JMA, which appears as an argument in the call to DIFU. Most of the communication between the main program and SUBROUTINE DIFU takes place through the common areas /XPZ/ and /COUT/. The variables in /XPZ/ are:

XX(J) = ω/Ω_J, normalized frequency of component J(COMPLEX).

PP(J) = $k_\perp V_J/\Omega_J$, normalized perpendicular wave number.

ZZ(J) = $k_\parallel V_J/\Omega_J$, normalized parallel wave number.

A(J) = α_{1J} (see Eq II-8).

B(J) = α_{2J} (see Eq II-8).

D(J) = Δ_J (see Eq II-8).

ASS(J) = particle mass in units of the proton mass. The electron mass is set to zero.

VD(J) = V_{dJ}, normalized drift velocity (see Eq II-8).

DN(J) = number density in particles/m^3.

TA(J) = temperature in keV.

XP(J) = ω_{PJ}^2/ω^2 (see Eq II-6) (COMPLEX).

CV = c/V_1, speed of light/thermal speed of first component.

These variables must all be given their proper values for J $\leq$ JMA before SUBROUTINE DIFU is called. The remaining variables are not used by DIFU, but they are needed to communicate with TYPIN. XC is the electron gyrofrequency in kHz, and PM, ZM, XOI, and PZL were discussed above.

The results from SUBROUTINE DIFU are returned through the common area /COUT/, which also is used to transfer the results to the output routine OUTPT. The variables in /COUT/ are:

X = ω/Ω_1 (COMPLEX)

P = $k_\perp V_1/\Omega_1$

Z = $k_\parallel V_1/\Omega_1$

EFL = Electric field components (Eq VI-7). |EFL| = 1 mV/m. The circularly polarized part of the field rotates in the right handed sense if Im EFL(2) > 0 (COMPLEX).

BFL = Magnetic field components in γ (COMPLEX)

DIR = Value of the dispersion function $D(\omega, \underline{k})$ (COMPLEX).

DIX = $\Omega_1 \partial D/\partial\omega$ (COMPLEX)

DIZ = $\Omega_1/V_1 \partial D/\partial k_\parallel$ (COMPLEX)

DIP = $\Omega_1/V_1 \partial D/\partial k_\perp$ (COMPLEX)

EPS = a (6, 4)-array containing $\underline{\underline{\varepsilon}}(\omega, \underline{k})$ and its derivatives x $\partial\underline{\underline{\varepsilon}}/\partial$ x, z $\partial\underline{\underline{\varepsilon}}/\partial$ z, and p $\partial\underline{\underline{\varepsilon}}/\partial$ p (COMPLEX)

VG(I), I=1: Perpendicular group veclociy/V_1.

I=2: Parallel group velocity/V_1.

SG(I) = Im x/VG(I). Can be used to estimate the spatial growth rate.

RI = kc/ω, refractive index (COMPLEX).

When DIFU is called with the first argument equal to two, the values of DIR and DIX are returned. Using Newton's iteration method, a correction.

$$CX = DIR/DIX \qquad \text{(VII-1)}$$

is calculated, and an improved approximation to the solution of the dispersion relation is obtained by replacing X by X-CX. This process is repeated until $|CX| \leq 10^{-6}|X|$.

Setting the first argument to four, we make one more call to SUBROUTINE DIFU, which returns the values of DIR, DIX, DIZ, and DIP as well as EFL and BFL. A last correction to the frequency is made, and the values of VG, SG and RI are computed. The results are printed in SUBROUTINE OUTPT.

When a solution to the disperison relation is found at one value of $\underline{k}$, the wave vector is incremented by $\Delta\underline{k}$ as prescribed by the KFS parameter in TYPIN. If KFS = 1, PM(3) is added to P, while ZM(3) is added to Z if KFS = 2. A new start approximation for the frequency is calculated as $\omega - \Delta\underline{k} \cdot \partial\omega/\partial\underline{k}$, and the iteration starts again. This is repeated until the dispersion relation has been solved at all the points specified by PM and ZM. Control is then returned to the user by a call to SUBROUTINE TYPIN.

```
C      TIME= 82/06/14 - 14.30.49

       PROGRAM WHAMP
       COMPLEX X,XO,XVO,XX(6),XP(6),DX,OME,FPX,DIR,DIX,DIZ,DIP,
      * EPS(6,4),DOX,DOZ,DOP,CX,EFL(3),BFL(3),RI
       DIMENSION REN(6),T(6),ST(6),ISP(6),ITID(7)
       CHARACTER SPE*3 (5)
       COMMON /XPZ/ XX,PP(6),ZZ(6),A(6),B(6),D(6),ASS(6),VD(6),
      C    DN(6),TA(6),XP,CV,PM(3),ZM(3),XOI,XC,PZL
       COMMON /COUT/ X,P,Z,EFL,BFL,DIR,DIX,DIZ,DIP,EPS,VG(2),SG(2),RI
       DATA DN/6.0E6,0.0E6,0.E6,0.,0.,0./,
      #  TA/.001,.001,.001,.001,.001,.001/,
      #   D/1.0,1.0,1.0,1.0,1.0,1.0/,
      #   A/1.,1.,1.,1.,1.,1./,
      #   B/0.10,0.10,0.10,0.10,0.10,0.10/,
      #  ASS/0.,0.,0.,0.,0.,0./,
      #  VD/0.,0.,0.0,0.,0.,0./,
      #  XC/3.30/
       DATA SPE/'E-  ','H+  ','HE+ ','O+  ','    '/

       NPL=0
     1 DEN=0.
       RED=0.
       DO 2 J=1,6
       REN(J)=1836.1*ASS(J)
       IF(REN(J).EQ.0.) REN(J)=1.
       T(J)=TA(J)/TA(1)
       ISP(J)=SQRT(ASS(J))
       IF(ISP(J).LT.4) ISP(J)=ISP(J)+1
       IF(DN(J).LE.0.) GOTO 2
       JMA=J
       RED=RED+DN(J)/REN(J)
       IF(ISP(J).EQ.1) DEN=DEN+DN(J)
     2 CONTINUE

       RN=REN(1)
C             ****  NORMALIZED TEMPERATURES AND VELOCITIES.  ****
       DO 3 J=1,JMA
       REN(J)=REN(J)/RN
       T(J)=T(J)*REN(J)
     3 ST(J)=SQRT(T(J))

       DEK=12405.
       PFQ=RED/DEK
       PX=SQRT(PFQ)
       XA=XC/RN
       TR=TA(1)/RN
       CV=TR*(1022.+TR)/(511.+TR)**2
       CV=1./SQRT(CV)
       DEK=DEK*RN
C             ****  PRINT PLASMA PARAMETERS.  ****
       CALL CLOCK(ITID)
       PRINT 100,(ITID(8-I),I=1,6)
   100 FORMAT(* DATE *,I4,*-*,I2,*-*,I2,*  TIME:*,I2,*.*,I2,*.*,I2,/)
       PRINT 101,PX,XC,DEN
   101 FORMAT(* PLASMA FREQ.:*F8.3*KHZ GYRO FREQ.:*F8.3*KHZ    *
      #  *ELECTRON DENSITY:*E9.3*M-3*/)
       DO 4 J=1,JMA
   102 FORMAT(* *A3*  DN=*E9.3*  T=*F7.4*  D=*F4.2
      **  A=*F4.2*  B=*F4.2* VD=*F6.2/)
     4 PRINT 102,SPE(ISP(J)),DN(J),TA(J),D(J),A(J),B(J),VD(J)
```

```
C
      IF(NPL.EQ.1) GOTO 6
C              ****  ASK FOR INPUT!  ****
    5 CALL TYPIN(NPL,KFS)
      IF(NPL.EQ.1) GOTO 1
    6 NPL=0
      KV=1
      PLG=PM(1)
      IF(PM(3).LT.0.) PLG=PM(2)
      P=PLG
      ZLG=ZM(1)
      IF(ZM(3).LT.0.) ZLG=ZM(2)
      Z=ZLG
      IF(PZL.NE.1.) GOTO 7
      P=10.**PLG
      Z=10.**ZLG
    7 X=XOI
   10 OME=(X*XA)**2
      FPX=PFQ/OME
      DO 11 J=1,JMA
      XX(J)=X*REN(J)
      PP(J)=P*ST(J)
      ZZ(J)=Z*ST(J)
   11 XP(J)=DN(J)/DEK/REN(J)/OME

      CALL DIFU(2,JMA,IERR)
      IF(IERR.NE.0) GOTO 50
C             ****  START OF ITERATION.  ****
      DO 20 I=1,20
      ADIR=CABS(DIR)
      IRK=0
      CX=DIR/DIX
   15 X=X-CX
      OME=(X*XA)**2
      FPX=PFQ/OME
      DO 16 J=1,JMA
      XP(J)=DN(J)/DEK/REN(J)/OME
   16 XX(J)=X*REN(J)
      IF(CABS(CX).LE.1.E-6*CABS(X)) GOTO 30
      CALL DIFU(2,JMA,IERR)
      IF(IERR.NE.0) GOTO 50
      IF(CABS(DIR).LT.ADIR) GOTO 20
      X=X+CX
      CX=CX/2.
      IRK=IRK+1
      IF(IRK.GT.20) GOTO 25
      GOTO 15
   20 CONTINUE
C
   25 PRINT 105,P,Z,X,I,IRK
  105 FORMAT(2X,*NO CONVERGENCE!*/*  KP=*,F6.3,*  KZ=*,
     +  F6.4,*  X=*,E12.2,E12.2/*  I=*,I3,*  IRK=*,I3/)
      GOTO 55
C             ****  CONVERGENCE!  ****
   30 CALL DIFU(4,JMA,IERR)
      IF(IERR.NE.0) GOTO 50
      X=X-DIR/DIX
C
      XI=AIMAG(X)
      VG(1)=-DIP/DIX
      VG(2)=-DIZ/DIX
      RI=SQRT(P**2+Z**2)*CV/X
```

```
      IF(VG(1).NE.0.) SG(1)=XI/VG(1)
      IF(VG(2).NE.0.) SG(2)=XI/VG(2)
C          ****  PRINT THE RESULTS.  ****
   34 CALL OUTPT
      P0=P
      Z0=Z
      X0=X
      IF(KV.EQ.0) GOTO 35
      XV0=X
      ZV0=Z
      ZL0=ZLG
      PV0=P
      PL0=PLG
      D0X=DIX
      D0Z=DIZ
      D0P=DIP
      KV=0
   35 GOTO(36,38) KFS
   36 PLG=PLG+PM(3)
C             **** UPDATE P AND Z.  ****
      IF(PLG.GE.PM(1).AND.PLG.LE.PM(2)) GOTO 39
      ZLG=ZLG+ZM(3)
      IF(ZLG.LT.ZM(1).OR.ZLG.GT.ZM(2)) GOTO 5
      KV=1
      PLG=PL0
      P=PV0
   37 Z=ZLG+PZL*(10.**ZLG-ZLG)
      GOTO 40
C
   38 ZLG=ZLG+ZM(3)
      IF(ZLG.GE.ZM(1).AND.ZLG.LE.ZM(2)) GOTO 37
      PLG=PLG+PM(3)
      IF(PLG.LT.PM(1).OR.PLG.GT.PM(2)) GOTO 5
      KV=1
      ZLG=ZL0
      Z=ZV0
   39 P=PLG+PZL*(10.**PLG-PLG)
C               ****  NEW START FREQUENCY.  ****
   40 IF(KV.EQ.0) GOTO 41
      DKP=P-PV0
      DKZ=Z-ZV0
      DX=(DKP*D0P+DKZ*D0Z)/D0X
      X=XV0-DX
      GOTO 10
   41 DKP=P-P0
      DKZ=Z-Z0
      DX=(DKP*DIP+DKZ*DIZ)/DIX
      X=X0-DX
      GOTO 10
   50 PRINT*,' TO HEAVILY DAMPED!'
      PRINT*,'    '
      IERR=0
   55 IF(KFS.EQ.1) PLG=1.E99
      IF(KFS.EQ.2) ZLG=1.E99
      GOTO 35
   60 CONTINUE
      END
```

INPUT/OUTPUT

The idea behind the input/output routines is to provide a maximum of flexibility with a minimum of typing. This version of the code is written for a NORD computer, and since some details are machine dependent, modifications may be required if the program is moved to another computer. The input/output arrangement described here has however been found very useful, and implementation of similar routines can be recomended.

When the subroutine TYPIN is called, the user is asked to give input to the program by the prompt "INPUT:". By responding with "H" (for HELP!) he will make the following printout appear:

```
AN INPUT LINE MAY CONSIST OF UP TO 80 CHARACTERS.
THE FORMAT IS:
NAME1=V11,V12,V13,...NAME2=V21,V22,...NAME
THE NAMES ARE CHOSEN FROM THE LIST:

NAME                PARAMETER
A(I)                THE ALPHA1 PARAMETER IN THE DISTRIBUTION.
                    (I) IS THE COMPONENT NUMBER, I=1 - 6.
B(I)                THE ALPHA2 PARAMETER IN THE DISTRIBUTION.
C                   THE ELECTRON CYCLOTRON FREQ. IN KHZ.
D(I)                THE DELTA PARAMETER IN THE DISTRIBUTION
F                   FREQUENCY, START VALUE FOR ITERATION.
L           L=1     THE P AND Z PARAMETERS ARE INTERPRETED
                    AS LOGARITHMS OF THE WAVE NUMBERS. THIS
                    OPTION ALLOWS FOR LOGARITHMIC STEPS.
            L=0     DEFAULT VALUE. LINEAR STEPS.
M(I)                MASS IN UNITS OF PROTON MASS.
N(I)                NUMBER DENSITY IN PART./CUBIC METER
P(I)                PERPENDICULAR WAVE VECTOR COMPONENTS.
                    P(1) IS THE SMALLEST VALUE, P(2) THE
                    LARGEST VALUE, AND P(3) THE INCREMENT.
S                   STOP! TERMINATES THE PROGRAM.
T(I)                TEMPERATURE IN KEV
V(I)                DRIFT VELOCITY / THERMAL VELOCITY.
Z(I)                Z-COMPONENT OF WAVE VECTOR. I HAS THE
                    SAME MEANING AS FOR P(I).
A NAME WITHOUT INDEX REFERS TO THE FIRST ELEMENT, "A" IS
THUS EQUIVALENT TO "A(1)". THE VALUES V11,V12,.. MAY BE
SPECIFIED IN I-, F-, OR E-FORMAT, SEPARATED BY COMMA(,).
THE "=" IS OPTIONAL, BUT MAKES THE INPUT MORE READABLE.
EXAMPLE: INPUT:A1.,2. B(3).5,P=.1,.2,1.E-2
THIS SETS A(1)=1., A(2)=2., B(3)=.5, P(1)=.1, P(2)=.2,
AND P(3)=.01. IF THE INCREMENT P(3)/Z(3) IS NEGATIVE, P/Z
WILL FIRST BE SET TO P(2)/Z(2) AND THEN STEPPED DOWN TO P(1)/Z(1)
THE LAST SPECIFIED OF P AND Z WILL VARY FIRST.
IF THE LETTER "O" (WITHOUT VALUE) IS INCLUDED, YOU WILL
 BE ASKED TO SPECIFY A NEW OUTPUT FORMAT.

INPUT:
```

Variables not specified in the input line generally retain their old values. Notice however that the P and Z variables (corresponding to PM and ZM in the main program) are somewhat different in the sense that specifying P(1) or Z(1) may change the values of P(2), P(3) or Z(2), Z(3). The effect of a specification like "P = P1, P2, P3" is conveniently thought of as "DO P = P1, P2, P3" with the conventions that

a) It only P1 is specified, the loop is executed once with P = P1.
b) If P1 and P2 are specified but not P3, the loop is executed twice with P = P1 and P = P2.

The same interpretation is given to the Z parameter.

The first time TYPIN is executed, or if the letter O appears in the input line, a call is made to the entry point INOUT in SUBROUTINE OUTPT. By printing "OUTPUT:", this routine asks the user to specify the output format. Answering "H" helps again, by making the following printout appear:

```
 OUTPUT:H

THE OUTPUT IS DETERMINED BY A STRING OF LETTERS:

A      ALL AVAILABLE OUTPUT.
B      WAVE MAGNETIC FIELD COMPONENTS.
D      DISPERSION FUNCTION AND DERIVATIVES.
E      WAVE ELECTRIC FIELD COMPONENTS.
F      FREQUENCY.
G      GROUP VELOCITY COMPONENTS.
P      PERPENDICULAR COMPONENT OF WAVE VECTOR.
R      REFRACTIVE INDEX.
S      SPATIAL GROWTH-RATES.
T      DIELECTRIC TENSOR AND DERIVATIVES.
Z      Z-COMPONENT OF WAVE VECTOR.

THE RESULTS ARE NORMALLY PRINTED ON ONE LINE IN THE ORDER THEY ARE
SPECIFIED. A NEW LINE IS OBTAINED BY INSERTING A "/" IN THE STRING.
EXAMPLE: OUTPUT: PZF/E
THE WAVE NUMBERS AND THE FREQUENCY ARE PRINTED ON ONE LINE,
AND THE ELECTRIC FIELD COMPONENTS ON THE NEXT.

 OUTPUT:
```

This should allow the user to choose a suitable output format.

```
C      TIME= 82/06/14 - 14.30.49


       SUBROUTINE TYPIN(NPL,KFS)
C              ARGUMENTS: NPL     NEW PLASMA. WHEN A PLASMA PARAMETER IS
C                                 CHANGED, NPL IS SET TO 1.
C                         KFS     =1  P SPECIFIED LAST.
C                                 =2  Z SPECIFIED LAST.
       CHARACTER INP*80,IC*1
       COMMON /XPZ/ ARRAY(88)
       DIMENSION TV(2)
       DATA IOUT/2/,KP,KZ/1,1/,ARRAY(88)/0./
C **** CORRESPONDENCE BETWEEN ARRAY AND NAMES IN WHAMP:  ****
C  ARRAY(IOF+ 0 12 18 24 30 36 42  48 54 60 66 78 79 82 85  86  87
C             XX PP ZZ A  B  D  ASS VD DN TA XP CV PM ZM XOI XC  PZL
C
     1 IV=1000
       PRINT 2
     2 FORMAT(/**$INPUT: *)
       READ 3,INP
     3 FORMAT(A80)
       NC=0
     4 NC=NC+1
       IF(NC.GT.80) GOTO 6
       IC=INP(NC:NC)
       IF(IC.EQ.' ') GOTO 4
       IS=ICHAR(IC).AND.177B
C*** "IS" IS AN INTEGER CONTAINING THE VALUE OF THE ASCII CHARACTER IC ***
       IF(IS.LT.47B) GOTO 4
       IF(IV.GT.88) GOTO 7
       IF(IS.GE.101B.AND.IS.LE.132B) GOTO 6
       IF(IS.LE.57B) GOTO 5
       IF(IS.GE.72B.AND.IS.LE.100B) GOTO 5
       IF(IS.GE.133B) GOTO 5
C
       TV(IE)=TV(IE)*DEK+(IS-60B)*DEC
       DEC=DEC*DEK/10.
       NV=1
       GOTO 4
C
     5 IF(IC.NE.',') GOTO 10
     6 IF(NV.NE.1) GOTO 7
       IF(IOF.GT.6) GOTO 25
       IF(IOF.GT.3.AND.IV.GT.66) GOTO 25
       IF(IOF.GT.1.AND.IV.GT.85) GOTO 25
       IF(IC.EQ.'E') GOTO 9
       ARRAY(IV+IOF)=TV(1)*10.**TV(2)
       IOF=IOF+1
       IF(IV.EQ.79) KP=IOF
       IF(IV.EQ.82) KZ=IOF
C
     7 IF(NC.GT.80) GOTO 50
       DEK=10.
       DEC=1.
       TV(1)=0.
       TV(2)=0.
       NV=0
       IE=1
       IF(IC.EQ.',') GOTO 4
       IV=1000
       IOF=1
       IF(IC.EQ.'A') IV=24
```

```
      IF(IC.EQ.'B') IV=30
      IF(IC.EQ.'C') IV=86
      IF(IC.EQ.'D') IV=36
      IF(IC.EQ.'F') IV=85
      IF(IC.EQ.'H') GOTO 30
      IF(IC.EQ.'L') IV=87
      IF(IC.EQ.'M') IV=42
      IF(IC.EQ.'N') IV=54
      IF(IC.EQ.'O') GOTO 8
      IF(IC.EQ.'P') IV=79
      IF(IC.EQ.'S') STOP
      IF(IC.EQ.'T') IV=60
      IF(IC.EQ.'V') IV=48
      IF(IC.EQ.'Z') IV=82
C
      IF(IV.GE.1000) GOTO 27
      IF(IV.LT.66.OR.IV.EQ.86) NPL=1
      IF(IV.EQ.79) KP=1
      IF(IV.EQ.79) KFS=1
      IF(IV.EQ.82) KZ=1
      IF(IV.EQ.82) KFS=2
      GOTO 4
C
    8 IOUT=0
      GOTO 4
C
    9 IE=2
      DEK=10.
      DEC=1.
      GOTO 4
C
   10 IF(IC.NE.'-') GOTO 11
      IF(TV(IE).NE.0.) GOTO 20
      DEC=-DEC
      GOTO 4
C
   11 IF(IC.NE.'.') GOTO 12
      IF(ABS(DEC).NE.1.) GOTO 20
      DEK=DEK/10.
      DEC=DEC/10.
      GOTO 4
C
   12 IF(IC.NE.')') GOTO 4
      IF(DEC.NE.1.) GOTO 20
      IF(IE.NE.1.OR.TV(1).LE.0.) GOTO 20
      IOF=TV(1)+.1
      TV(1)=0.
      NV=0
      GOTO 4
C
   20 PRINT 21,IC
   21 FORMAT(* AMBIGUITY CAUSED BY THE CHARACTER "*,A1,*"*)
   23 PRINT 24
   24 FORMAT(* THE REST OF THE LINE IS IGNORED. PLEASE TRY AGAIN!*)
      GOTO 1
C
   25 TV(1)=TV(1)*10.**TV(2)
      PRINT 26,TV(1)
   26 FORMAT(* THE VALUE*,E11.3,* WILL NOT FIT IN THE VAIRIABLE FIELD*)
      GOTO 23
C
   27 PRINT*,'$ HELP, YES OR NO?'
```

```
   READ *,IC
   IF(IC.EQ.'N') GOTO 1
30 PRINT 31
31 FORMAT(* AN INPUT LINE MAY CONSIST OF UP TO 80 CHARACTERS.*/
  F  *  THE FORMAT IS:*/* NAME1=V11,V12,V13,....NAME2=V21,V22,...NAME*/
  F  * THE NAMES ARE CHOSEN FROM THE LIST:*//
  F  * NAME                PARAMETER*/
  F  * A(I)                THE ALPHA1 PARAMETER IN THE DISTRIBUTION.*/
  F  *                     (I) IS THE COMPONENT NUMBER, I=1 - 6.*/
  F  * B(I)                THE ALPHA2 PARAMETER IN THE DISTRIBUTION.*/
  F  * C                   THE ELECTRON CYCLOTRON FREQ. IN KHZ.*/
  F  * D(I)                THE DELTA PARAMETER IN THE DISTRIBUTION*/
  F  * F                   FREQUENCY, START VALUE FOR ITERATION.*/
  F  * L           L=1     THE P AND Z PARAMETERS ARE INTERPRETED*/
  F  *                     AS LOGARITHMS OF THE WAVE NUMBERS. THIS*/
  F  *                     OPTION ALLOWS FOR LOGARITHMIC STEPS.*/
  F  *             L=0     DEFAULT VALUE. LINEAR STEPS.*/
  F  * M(I)                MASS IN UNITS OF PROTON MASS.*/
  F  * N(I)                NUMBER DENSITY IN PART./CUBIC METER*/
  F  * P(I)                PERPENDICULAR WAVE VECTOR COMPONENTS.*/
  F  *                     P(1) IS THE SMALLEST VALUE, P(2) THE*/
  F  *                     LARGEST VALUE, AND P(3) THE INCREMENT.*/
  F  * S                   STOP! TERMINATES THE PROGRAM.*)
   PRINT 32
32 FORMAT(  * T(I)                TEMPERATURE IN KEV*/
  F  * V(I)                DRIFT VELOCITY / THERMAL VELOCITY.*/
  F  * Z(I)                Z-COMPONENT OF WAVE VECTOR. I HAS THE*/
  F  *                     SAME MEANING AS FOR P(I).*/
  F  * A NAME WITHOUT INDEX REFERS TO THE FIRST ELEMENT, "A" IS */
  F  * THUS EQUIVALENT TO "A(1)". THE VALUES V11,V12,.. MAY BE */
  F  * SPECIFIED IN I-, F-, OR E-FORMAT, SEPARATED BY COMMA(,).*/
  F  * THE "=" IS OPTIONAL, BUT MAKES THE INPUT MORE READABLE.*/
  F  * EXAMPLE: INPUT:A1.,2. B(3).5,P=.1,.2,1.E-2*/
  F  * THIS SETS A(1)=1., A(2)=2., B(3)=.5, P(1)=.1, P(2)=.2,*/
  F  * AND P(3)=.01. IF THE INCREMENT P(3)/Z(3) IS NEGATIVE, P/Z*/
  F  * WILL FIRST BE SET TO P(2)/Z(2) AND THEN STEPPED DOWN TO*
  F  * P(1)/Z(1)*/
  F  * THE LAST SPECIFIED OF P AND Z WILL VARY FIRST.*/
  F  * IF THE LETTER "O" (WITHOUT VALUE) IS INCLUDED, YOU WILL*/
  F  *  BE ASKED TO SPECIFY A NEW OUTPUT FORMAT.*/)
   GOTO 1
50 IF(ARRAY(86).GT.0.) GOTO 51
   PRINT*,'$ START FREQUENCY'
   READ*,ARRAY(86)
51 GOTO(52,53,54,55) KP
52 PRINT*,'$ PERP. WAVE VECTOR UNDEFINED!'
   GOTO 1
53 ARRAY(81)=ARRAY(80)
54 ARRAY(82)=ARRAY(81)-ARRAY(80)
55 IF(ARRAY(82).EQ.0.) ARRAY(82)=10.
   GOTO (56,57,58,59) KZ
56 PRINT*,'$ PARALLEL WAVE VECTOR UNDEFINED!'
   GOTO 1
57 ARRAY(84)=ARRAY(83)
58 ARRAY(85)=ARRAY(84)-ARRAY(83)
59 IF(ARRAY(85).EQ.0.) ARRAY(85)=10.
   IF(IOUT.NE.1) CALL INOUT
   IOUT=1
   RETURN
   END
```

```
C     TIME= 82/06/14 - 14.30.49


      SUBROUTINE OUTPT
      CHARACTER IOU*20
      COMPLEX X,EFL(3),BFL(3),DI(4),EPS(6,4),RI
      COMMON /COUT/ X,P,Z,EFL,BFL,DI,EPS,VG(2),SG(2),RI
C
      K=0
    1 K=K+1
      IF(K.GT.KMX) PRINT 6
      IF(K.GT.KMX) RETURN
      IC=ICHAR(IOU(K:K)).AND.177B
      IF(IC.EQ.57B) GOTO 5
      IF(IC.LT.100B) GOTO 1
      IC=IC-100B
      IF(IC.GT.32B) GOTO 1
      IB=IC/9
      ID=IC-IB*9+1
      GOTO (2,3,4) IB+1
    2 GOTO(1,10,18, 1,24,16,10,20, 1) ID
C       .      @  A  B  C  D  E  F  G  H
    3 GOTO(1, 1, 1, 1, 1, 1, 1,12, 1) ID
C           I  J  K  L  M  N  O  P  Q
    4 GOTO(26,22,28, 1, 1, 1, 1, 1,14) ID
C            R  S  T  U  V  W  X  Y  Z
    5 PRINT 6
    6 FORMAT(*  *)
      GOTO 1
C
   10 PRINT 11,X
   11 FORMAT(*$ FREQ=*,F7.4,E10.2,*  *)
      IF(IC.GT.1) GOTO 1
   12 PRINT 13,P
   13 FORMAT(*$ P=*,F8.4,*  *)
      IF(IC.GT.1) GOTO 1
   14 PRINT 15,Z
   15 FORMAT(*$ Z=*,F8.5,*  *)
      IF(IC.GT.1) GOTO 1
      PRINT 6
   16 PRINT 17,EFL
   17 FORMAT(*$ EX=*,F7.4,F8.4,*  EY=*,F7.4,F8.4,*  EZ=*,F7.4,F8.4,* *)
      IF(IC.GT.1) GOTO 1
      PRINT 6
   18 PRINT 19,BFL
   19 FORMAT(*$ BX=*,F7.4,F8.4,*  BY=*,F7.4,F8.4,*  BZ=*,F7.4,F8.4,* *)
      IF(IC.GT.1) GOTO 1
      PRINT 6
   20 PRINT 21,VG
   21 FORMAT(*$ VGP=*,E9.2,*  VGZ=*,E9.2,*  *)
      IF(IC.GT.1) GOTO 1
   22 PRINT 23, SG
   23 FORMAT(*$ SGP=*,E9.2,*  SGZ=*,E9.2,*  *)
      IF(IC.GT.1) GOTO 1
      PRINT 6
   24 PRINT 25,DI
   25 FORMAT(*$ D=*,2E10.2,*  DX=*,2E10.2,*  DZ=*,2E10.2,
     F * DP=*,2E10.2,/)
      IF(IC.GT.1) GOTO 1
   26 PRINT 27,RI
   27 FORMAT(*$ RI=*,2E10.2)
      IF(IC.GT.1) GOTO 1
```

```
      PRINT 6
   28 DO 30 J=1,6
      N=1+J/4+J/6
      M=J-J/4*2-J/6
      PRINT 29,((N,M,EPS(J,I)),I=1,4)
   29 FORMAT(*$ E*,2I1,*=*,2E10.2,*  EX*,2I1,*=*,2E10.2,
     F  *EZ*,2I1,*=*,2E10.2,*  EP*,2I1,*=*,2E10.2,/)
   30 CONTINUE
      PRINT 6
      GOTO 1
C
      ENTRY INOUT
  101 PRINT*,'$ OUTPUT:'
      READ 102, IOU
  102 FORMAT(A20)
      DO 103 K=1,20
      IF(IOU(K:K).EQ.' ') GOTO 103
      IF(IOU(K:K).EQ.'H') GOTO 104
      KMX=K
  103 CONTINUE
      RETURN
C
  104 PRINT 105
  105 FORMAT(* THE OUTPUT IS DETERMINED BY A STRING OF LETTERS:*//
     F  * A      ALL AVAILABLE OUTPUT.*/
     F  * B      WAVE MAGNETIC FIELD COMPONENTS.*/
     F  * D      DISPERSION FUNCTION AND DERIVATIVES.*/
     F  * E      WAVE ELECTRIC FIELD COMPONENTS.*/
     F  * F      FREQUENCY.*/
     F  * G      GROUP VELOCITY COMPONENTS.*/
     F  * P      PERPENDICULAR COMPONENT OF WAVE VECTOR.*/
     F  * R      REFRACTIVE INDEX.*/
     F  * S      SPATIAL GROWTH-RATES.*/
     F  * T      DIELECTRIC TENSOR AND DERIVATIVES.*/
     F  * Z      Z-COMPONENT OF WAVE VECTOR.*//
     F  * THE RESULTS ARE NORMALLY PRINTED ON ONE LINE IN THE ORDER*
     F  * THEY ARE*/* SPECIFIED. A NEW LINE IS OBTAINED BY INSERTING A*
     F  * "/" IN THE STRING.*/
     F  * EXAMPLE: OUTPUT: PZF/E*/
     F  * THE WAVE NUMBERS AND THE FREQUENCY ARE PRINTED ON ONE LINE,*/
     F  * AND THE ELECTRIC FIELD COMPONENTS ON THE NEXT.*//)
      GOTO 101
      END
```

VIII. Discussion

In this report, the following general assumptions are made about the plasma and the waves:

a) Homogeneity
b) The particles are non-relativistic
c) The plasma distribution function is of the form (II-8)
d) The waves are linear
e) The frequency ω is complex but the wave vector $\underline{k}$ real
f) The waves are not to heavily damped, i e Im $\omega > -k_{||}$ or Im $\omega > -$ Re$(\omega - n\Omega)$.

Apart from this, there are no restrictions on ω or $\underline{k}$. Some of the assumptions above can be further relaxed without much effort. Instead of Equation (II-8) we may in c) take any linear combination of Maxwellian components. The assumption e) can probably be omitted if all variables associated with $\underline{k}$ in the program are declared as COMPLEX. This may be desirable if the method is used in a ray-tracing program.

The main approximation made is the introduction of a Padé approximant for the plasma dispersion functon Z. This forces us to make the assumption f) above, and will cause small but striking errors in Im ω when the frequency is almost real. In practice, these difficulties are not to severe, since strongly damped waves usually are of limited interest, and the weak spurious "numerical" instabilies which sometimes appear are easily recognized and disregarded. It is however important to keep their existence in mind. When Im $\omega > 0$, the error in the Padé approximant is negligible compared to the error in $R(y,\lambda)$, which is $\lesssim 10^{-6}$ $R(y,\lambda)$.

Although the errors in the basic functions Z and R are rather easily estimated, it is in general very difficult to say anything about the accuracy of the dispersion function $D(\omega, \underline{k})$ or the solution $\omega(\underline{k})$ of the dispersion rela-

tion. Since the interesting case is $D(\omega, \underline{k}) \approx 0$, we can expect that the most significant part of $\underline{\underline{\varepsilon}}$ will cancel in the calculation of D, and truncation errors may thus become important. No sign of this has been seen in the tests made up to now, but since only a small corner of the whole 47-dimensional parameter space has been investigated, the possibility can not be ruled out. Users of the program are therefore adviced to handle unexpected results with reasonalbe suspicion, and to try to confirm them by independent methods if possible.

I would appreciate if significant errors in this description, and in the results produced by the program WHAMP, were reported to me.

IX. Acknowledgement

I am grateful to Mats André for his valuable assistance during the writing, debugging, and testing of this program.

X. References

Aamodt, R.E., Plasma Physics, 9, 573, 1967.

Abramowitz, M., and I. Stegun, Handbook of Mathematical Functions, (Dover, New York, 1965).

Akhiezer, A.I., I.A. Akhiezer, R.V. Polovin, A.G. Sitenko, and K.N. Stephanov, Plasma Electrodynamics, (Pergamon Press, Oxford, 1975).

Baker, G.A., Essentials of Padé Approximants, (Academic, New York, 1975).

Clemmow, P.C., and J.P. Dougherty, Electrodynamics of Particles and Plasmas (Addison-Wesley, London, 1969).

Fredricks, R.W., J. Plasma Physics, 2, 197, 1968.

Fried, B.D., and S.D. Conte, The Plasma Dispersion Function, (Academic, New York, 1961).

Ichimaru, S., Basic Principles of Plasma Physics, (W.A. Benjamin Inc., Reading, Massachusetts, 1973).

Karpman, V.I., Ju. K. Alekhin, N.D. Borisov, and N.A. Rajabova, Astrophys. Space Sci., 22, 267, 1973.

Martín, P., G. Donoso, and J. Zamudio-Christi, J. Math. Phys., 21, 280, 1980.

Martín, P., and M.A. Gonzáles, Phys. Fluids, 22, 1413, 1979.

Németh, G., A. Ag, and Gy. Páris, J. Math. Phys., 22, 1192, 1981.