PFC/JA-89-52

PROGRAMS FOR COMPUTING CYCLOTRON-RESONANCE-MASER ABSOLUTE INSTABILITY PROPERTIES

Ge Zhang*
John A. Davies*
and
George L. Johnston

October, 1989

Plasma Fusion Center
Massachusetts Institute of Technology
Cambridge, MA 02139

*Permanent address: Clark University, Worcester, MA 01610

Research supported in part by the Department of Energy High Energy Physics Division, the Naval Research Laboratory Plasma Physics Division and the Office of Naval Research.

PROGRAMS FOR COMPUTING CYCLOTRON-RESONANCE-MASER ABSOLUTE INSTABILITY PROPERTIES

Ge Zhang and John A. Davies
Clark University
Worcester, MA 01610

and

George L. Johnston
Plasma Fusion Center
Massachusetts Institute of Technology
Cambridge, MA 02193

ABSTRACT

This report contains descriptions of programs for computing absolute instability properties of the cyclotron resonance maser (CRM). These programs, which can be read from [DAVIES.CRM] on the MIT PFCVAX, are based on the linear theory of Briggs and Bers. Included are a program for computing temporal CRM instability growth rates, a program for computing images of the (complex ω-plane) inverse Laplace contour in the complex k-plane, and several programs for rapidly computing approximate and exact pinch point coordinates (k_s,ω_s). Also included are programs for computing parameter regions of absolute instability for the CRM. Some of the programs are readily modified to accommodate dispersion relations other than that for the CRM.

I . INTRODUCTION

This report describes several computer programs which are useful for computing absolute instability properties of the cyclotron resonance maser (CRM). Fortran files of these programs can be copied from [DAVIES.CRM] on the MIT PFCVAX. The basic formalism upon which most of the programs are based is the pinch-point theory of Briggs[1] and Bers[2]. Specific formulas used by the programs are discussed in the remainder of this section. For detailed derivations of these formulas see Refs. 3-7. An overview of the relationships among the programs is given in Sec. 2. Individual programs are discribed in detail in Sec. 3.

The approximate CRM dispersion relation employed is[5]

$$D(\hat{k},\hat{\omega}) = \left\{\hat{\omega}^2 - \hat{k}^2 - \left[1-(1+i)(1+\frac{m^2}{\nu_{mn}^2-m^2}\hat{\omega}^2)\frac{\delta}{r_w}\right]\right\}(\hat{\omega}-\hat{k}\beta_{\|}-sb)^2+\epsilon = 0. \quad (1)$$

which couples the TE_{mn} waveguide mode and the beam mode. In the above equation, the normalized frequency $\hat{\omega}$ and wave number $\hat{k}$ are given by $\hat{\omega} = \omega/\omega_c$ and $\hat{k} = k_{\|}/k_{mn}$, where $k_{\|}$ is the longitudinal wavenumber and $\omega_c = ck_{mn}$ is the cutoff frequecy for the TE_{mn} waveguide mode. The quantity k_{mn} is given by $k_{mn} = \nu_{mn}/r_w$, where ν_{mn} is the n-th zero of the first derivative of the Bessel function J_m. The quantity δ is the skin depth of the waveguide wall, $\beta_{\|}$ is the dimensionless axial electron velocity, and s is the beam harmonic. The quantity $b = \Omega_c/\omega_c$, where $\Omega_c = eB_0/\gamma_0 mc$ is the relativistic electron cyclotron frequency and $\gamma_0 mc^2$ is the relativistic electron energy. For an annular beam with all electron guiding centers located at $r = r_b$, the coupling constant ϵ is given by

$$\epsilon = \frac{4\beta_{\perp}^2}{\gamma_0\beta_{\|}}\frac{[J_{s-m}(k_{mn}r_b)J_s'(k_{mn}r_L)]^2}{(\nu_{mn}^2-m^2)J_m^2(\nu_{mn})}\frac{I}{I_A}, \quad (2)$$

where $\beta_{\perp}$ is the dimensionless perpendicular electron velocity, and $r_L = c\beta_{\perp}/\Omega_c$ is the electron Larmor radius. The beam current is given by I, and $I_A = mc^3/e = 17.045kA$. In the case of a solid beam with a spatially uniform distribution of guiding centers for $r < r_b$ the coupling constant ϵ is obtained by multipling the expression in Eq. (2) by factor κ given by

$$\kappa = 2\left(\frac{r_w}{r_b}\right)^2 J_{s-m}^{-2}(k_{mn}r_b)\int_0^{r_b/r_w} yJ_{s-m}^2(\nu_{mn}y)dy. \quad (3)$$

To test for the presence of absolute instability, first consider an inverse Laplace contour in the complex $\hat{\omega}$-plane which is above all of the zeros of $D(\hat{k},\hat{\omega})$ for real $\hat{k}$. [Here it is

assumed that the Fourier-Laplace components used to derive the dispersion relation are of the form $exp(ikz - i\omega t)$.] Images of the inverse Laplace contour in the complex $\hat{k}$-plane are obtained by solving the dispersion relation $[D(\hat{k},\hat{\omega}) = 0]$ for $\hat{k}$. Absolute instability is present if an image from above the real $\hat{k}$-axis merges with one from below to form a pinch point as the inverse Laplace contour is lowered to the real $\hat{\omega}$-axis.[1,2] A pinch point is a saddle point of the function $\hat{\omega}(\hat{k})$ (as determined from dispersion relation) in the complex $\hat{k}$-plane. However, a saddle point is not necessarily a pinch point. Conditions for a saddle point are[1,2]

$$D(\hat{k},\hat{\omega}) = 0,$$
$$\frac{\partial D(\hat{k},\hat{\omega})}{\partial \hat{k}} = 0. \tag{4}$$

From Eqs. (1) and (4), the following saddle point conditions are readily derived for the CRM,

$$\hat{k}(\hat{\omega} - \beta_{\|}\hat{k} - sb) + \beta_{\|}\left\{\hat{\omega}^2 - \hat{k}^2 - \left[1 - (i+1)\left(1 + \frac{m^2}{\nu_{mn}^2 - m^2}\hat{\omega}^2\right)\frac{\delta}{r_w}\right]\right\} = 0,$$
$$\beta_{\|}\epsilon = \hat{k}(\hat{\omega} - \beta_{\|}\hat{k} - sb)^3. \tag{5}$$

Pinch-point coordinates $(\hat{k}_s,\hat{\omega}_s)$ can be obtained by solving the conditions in Eq. (5) simultaneously for $\hat{k}$ and $\hat{\omega}$. However, care must be taken that the result is a pinch point and not some other saddle point which is not a pinch point.

Exact analytic conditions for the existence of absolute instability have been determined for the case of zero skin depth[6] [i.e., $\delta = 0$ in Eq. (1)]. If $sb > 1$ in Eq. (1), then absolute instability will be present in the CRM mode. In Ref. 6, conditions are derived for the existence of absolute instability when $sb < 1$. Absolute instability will be present when the coupling constant ϵ is greater than a critical value ϵ_c. Over most of the useful range of values of $\beta_{\|}$ and sb, it is given by

$$\epsilon_c = 27\beta_{\|}^2\hat{k}_s^4, \tag{6}$$

where $\hat{k}_s$ is the pinch-point coordinate at the onset of absolute instability. It is given by

$$\hat{k}_s = \frac{1}{2}(1+8\beta_\parallel^2)^{-1}\{-4\beta_\parallel sb + [16\beta_\parallel^2 s^2b^2 + 2(1+8\beta_\parallel^2)(1-s^2b^2)]^{\frac{1}{2}}\}. \tag{7}$$

The corresponding frequency is

$$\hat{\omega}_s = 4\beta_\parallel \hat{k}_s + sb. \tag{8}$$

However, Eqs. (6)-(8) are applicable if and only if

$$\hat{\omega}_s - \frac{\hat{k}_s}{\beta_\parallel} > 0. \tag{9}$$

If the condition in Eq. (9) is not obeyed, then the critical coupling constant is given by

$$\epsilon_c = \frac{\hat{k}_s'}{\beta_\parallel^4}(b_0^2\hat{k}_s' - \beta_\parallel sb)^3, \tag{10}$$

where

$$b_0^2 = 1 - \beta_\parallel^2. \tag{11}$$

The coordinates of the pinch point at the onset of absolute instability are given by

$$\hat{k}_s' = \frac{\beta_\parallel}{4b_0^2}[sb + (s^2b^2 + 8b_0^2)^{\frac{1}{2}}], \tag{12}$$

and

$$\hat{\omega}_s' = \frac{\hat{k}_s'}{\beta_\parallel}. \tag{13}$$

A plot showing regions of absolute instability is presented in Fig. 1. Each curve on the plot shows ϵ_c as a function of $\beta_\parallel$ for the value of sb shown against the curve. A CRM mode is absolutely unstable if its parameter point ($\beta_\parallel$, ϵ) lies above that curve corresponding to its value of sb. The curve AB on the plot separates the left-hand region where Eqs. (10), (12), and (13) are applicable from the right-hand region where Eqs. (6)–(8) are applicable.

Approximate pinch-point coordinates are readily obtained from Eq. (5) for the case of $sb > 1$ and $\delta = 0$. These approximate expressions are given by[7]

$$\begin{aligned}
\hat{\omega}_s &= \hat{\omega}_- + \frac{3}{4}\frac{(\hat{k}_-^2\beta_\parallel\epsilon)^{\frac{1}{3}}}{(\beta_\parallel\hat{\omega}_- - \hat{k}_-)}(1+i\sqrt{3}), \\
\hat{k}_s &= \hat{k}_- + \frac{(\hat{k}_- + 2\beta_\parallel\hat{\omega}_-)}{3\beta_\parallel\hat{k}_-}\delta\hat{\omega}_-,
\end{aligned} \tag{14}$$

where

$$\hat{k}_- = \frac{1}{b_0^2}\left[sb\beta_\| - (s^2b^2 - b_0^2)^{\frac{1}{2}}\right],$$
$$\hat{\omega}_- = \beta_\|\hat{k}_- + sb. \tag{15}$$

Some of the CRM programs presented in this report are designed to compute CRM absolute instability properties relative to a reference frame moving with arbitrary velocity β_v in the axial direction. Such computations are useful for determining pulse shapes[8] for absolute and convective CRM instabilities. Details for the CRM are given in Ref. 7. All of the formulas discussed in this section can be extended to the general frame by replacing noninvariant quantities by primed quantities. The quantities $\hat{k}'$ and $\hat{\omega}'$ are related to $\hat{k}$ and $\hat{\omega}$ by the usual Lorentz transformation of wave numbers and frequencies. The velocities $\beta_\|'$ and $\beta_\|$ are related by the velocity addition rule. In addition,

$$b' = b\gamma_v^{-1}(1 - \beta_v\beta_\|)^{-1},$$
$$\epsilon' = \epsilon\gamma_v^{-2}(1 - \beta_v\beta_\|)^{-2}, \tag{16}$$

where $\gamma_v = (1 - \beta_v^2)^{-\frac{1}{2}}$. [However, see Eqs. (20) and (25)–(27) of Ref. 7 for details of the extension of Eqs. (14) and (15) to the general frame.] Those readers interested only in laboratory-frame results should set $\beta_v = 0$ in programs referring to the general frame.

A final remark is that three of the programs (CRMDISP.FOR, CRMWK.FOR and CRMPIN.FOR) are readily modifed to accommodate dispersion relations other than Eq. (1). In each of these programs the dispersion relation is contained in a single subroutine.

II . SUMMARY OF PROGRAMS

Fortran programs which can be copied from [DAVIES.CRM] are briefly described below.

A. CRMDISP.FOR: Using the dispersion relation in Eq. (1), this program computes complex $\hat{\omega}$ vs. real $\hat{k}$ over a range of $\hat{k}$ selected by the user. For each $\hat{k}$, all $\hat{\omega}$ roots are computed, but only that root having the the largest value of $Im(\hat{\omega})$ is printed out. Because the CRM dispersion relation in Eq. (1) appears only in subroutine FN, it can easily be replaced with other dispersion relations. CRMDISP.FOR is useful for determining ranges of $\hat{\omega}$ and $\hat{k}$ to be used in the program CRMWK.FOR.

B. CRMWK.FOR: This program locates pinch points graphically by plotting images in the complex $\hat{k}$-plane of the lowered Laplace contour in the complex $\hat{\omega}$-plane. Numerical results obtained from the program CRMDISP.FOR are useful in selecting the range of $\hat{\omega}$ to be used in CRMWK.FOR. The interesting range of $Im(\hat{\omega})$ lies between $Im(\hat{\omega}) = 0$ and the maximum value of $Im(\hat{\omega})$ obtained from CRMDISP.FOR. The interesting range of $Re(\hat{\omega})$ roughly coincides with the interval of $Re(\hat{\omega})$ for which $Im(\hat{\omega}) > 0$. The CRM dispersion relation in Eq. (1) appears only in the subroutine FCN and is easily replaced by other dispersion relations. The graphical method of determining pinch-point coordinates accurately is very time consuming. Consequently, it is recommendated that, once an overview of the stability properties of a dispersion relation are obtained through the use of CRMWK.FOR, more rapid methods of computing pinch-point coordinates be derived. In particular, some of the programs listed below (CRMPIN.FOR and CRMPINCHV.FOR) are used to determine CRM pinch-point coordinates rapidly. [Methods for rapidly calculating pinch-point coordinates for the free electron laser are discussed in Ref. 9.] CRMWK.FOR runs very slowly on the MIT PFCVAX. A CRAY version of the program called CRMWKC. can also be read from [DAVIES.CRM].

C. CRMWKC.: This program is the CRAY version of CRMWK.FOR.

D. CALEPS.FOR: This program calculates the coupling constant ϵ in Eqs.(2) and (3) for the cases of a thin annular beam and of a uniform solid beam.

E. CRMPIN.FOR: This program computes exact pinch-point coordinates for the CRM

by solving the conditions in Eq. (4) simultaneously for $\hat{k}_s$ and $\hat{\omega}_s$. Newton's method is used. The result will be the pinch-point coordinates (if a pinch point exists) provided that the initial estimates for solutions used in Newton's method are sufficiently close to the exact solutions. Otherwise, the result will be the coordinates of a saddle point that is not a pinch point. The programs APCVPIN.FOR and CRMPINCHV.FOR provide initial estimates for solutions. CRMPIN.FOR can determine pinch points for all values of sb and δ in Eq. (1). However, it is recommended that CRMPINCHV.FOR be used instead of CRMPIN.FOR if $sb < 1$ and $\delta = 0$, because the former program is much easier to use than the latter. The CRM dispersion relation is used only in subroutine of FCN of CRMPIN.FOR. Consequently, the program is readily modified to accommodate other dispersion relations.

F. APCVPIN.FOR: This program computes approximate CRM pinch-point coordinates relative to a general inertial frame of reference moving in the axial direction, provided that $sb' > 1$ and $\delta = 0$. Users interested in general-frame computations should refer to Eqs.(25)–(27) of Ref. 7. Users interested only in laboratory-frame computations should set the input variable BETV = 0.0. Then, the program solves Eq. (14). The laboratory-frame approximation is good to excellent for $sb > 1.2$, but becomes increasingly bad as sb approaches one. The results are invalid if $sb < 1$. APCVPIN.FOR provides initial estimates for the pinch-point coordinates in CRMPIN.FOR for the case of $sb > 1$.

G. CRMPINCHV.FOR: This program calculates exact pinch-point coordinates for a general reference frame moving in the axial direction for the case of $sb' < 1$ and $\delta = 0$. Users interested only in the laboratory frame should set the input variable BETV = 0.0. Then the program first computes the critical coupling constant ϵ_c given in Eq. (6) or (10) and the critical pinch-point coordinates $(\hat{k}_c,\hat{\omega}_c)$ at the onset of absolute instability given by Eqs. (7) and (8) or Eqs. (12) and (13). The interval between the coupling constant ϵ and the critical coupling constant ϵ_c is divided into small increments. After a small positive imaginary part is added to $\hat{\omega}_c$, Newton's method is used to successively compute pinch-point coordinates after each increment of the coupling constant until the value ϵ is reached. If $sb < 1$ and $\delta = 0$, then it is suggested that CRMPINCHV.FOR be used instead of CRMPIN.FOR. If $sb < 1$ and $\delta > 0$, then coordinates obtained from CRMPINCHV.FOR

can be used as initial estimates for solutions in CRMPIN.FOR.

The program functions in a similar way for general-reference-frame computations (BETV $\neq 0.0$). The general-reference-frame critical coupling constant ϵ_c' is given in Eq. (30) or (34) of Ref. 7 and the pinch-point coordinates at the onset of absolute instability are given in Eqs. (31) and (32) or Eq.(33) of Ref. 7.

H. EXEPSCRIT.FOR: This program computes regions of absolute instability for the CRM for the case of $\delta = 0$. (Note that for $\delta = 0$ all CRM modes are absolutely unstable if $sb > 1$.) The plot in Fig. 1 was obtained using EXEPSCRIT.FOR. For a given value of sb, EXEPSCRIT.FOR computes one curve on the plot over a specified range of $\beta_{||}$ using Eqs. (6) and (7) or Eqs. (10) and (11).

I. GENCARM.FOR: This program computes parameter regions of absolute instability for the CRM for arbitrary values of δ. Like EXEPSILON.FOR, the program GENCARM.FOR computes ϵ_c over a range of $\beta_{||}$ for a given value of sb. However, EXEPSCRIT.FOR should be used for the case of $\delta = 0$, because it is much less difficult to run and uses much less CPU time than GENCARM.FOR. Input for GENCARM.FOR includes approximate pinch-point coordinates for one value of $\beta_{||}$ and a value of ϵ which is estimated to be larger than any value of ϵ_c over the range of $\beta_{||}$ of the computation. These coordinates can be obtained from CRMPIN.FOR, APCVPIN.FOR, and CRMPINCHV.FOR.

J. GENCARMC.: This program is the CRAY version of GENCARM.FOR.

III . DESCRIPTIONS OF PROGRAMS

A. Description of Program CRMDISP.FOR

PURPOSE: CRMDISP.FOR computes complex $\hat{\omega}$ for a sequence of real values of $\hat{k}$ by solving CRM the dispersion relation in Eq. (1). All roots $\hat{\omega}$ are computed but only that root with the maximum value of $Im(\hat{\omega})$ is printed out.

ALGORITHM: The real value of $\hat{k}$ is incremented from an initial value XKINIT to a final value XKFINL in steps of DLTAK. For each value of $\hat{k}$, Muller's method is used to compute the four roots ($\hat{\omega}$) of the CRM dispertion relation . A sort routine selects that root having the maximum $Im(\hat{\omega})$.

FILES: The source file, the input file and the output file are described in the following:

1. CRMDISP.FOR: This is the source file written in double precision. Since the IMSL library is employed, the execute file is generated by the link command: LINK CRMDISP, IMSL$:IMSL/LIB.

2. INCRMDISP.DAT: This is the input file which contains the input data. The following values are input in the format 4F15.5 in the following sequence.

BEE	EPSIL	BETAP	DELTA
XKINIT	XKFINL	DLTAK	XN

The quantities in Eq. (1) represented by the above FORTRAN input variables are b=BEE, ϵ = EPSIL, $\beta_{\|}$=BETAP and δ/r_w = DELTA. The initial and final values of $\hat{k}$ and the increment in $\hat{k}$ are given by XKINIT, XFINL and DLTAK, respectively. The quantity XN is the number of roots solved for by the Muller's method routine. In this program, its value is 4.0, because Eq. (1) is a fourth-degree polynomial in $\hat{\omega}$.

3. OPCRMDISP.DAT: This is the output file. First, it lists the input parameters. This list followed by a table giving the real and imaginary parts of $\hat{\omega}$ for each value of $\hat{k}$. The format of the table is:

xk=	reomega=	imomega=
- - - -	- - - -	- - - -
- - - -	- - - -	- - - -
- - - -	- - - -	- - - -

In the table, xk=$\hat{k}$, reomega=$Re(\hat{\omega})$, and imomega=$Im(\hat{\omega})$. The output is also printed on the screen.

REMARKS: The dispersion relation is in subroutine FN and is easily replaced by another dispersion relation. Users also may find it is necessary to modify the read statements, write statements, variable declaration statements, and common statements.

B. Description of Program CRMWK.FOR

PURPOSE: Program CARMWK locates pinch points graphically by plotting images in the complex $\hat{k}$-plane of the lowered inverse Laplace contour in the complex $\hat{\omega}$-plane for the CRM dispersion relation in Eq. (1). For selected values of $Im(\hat{\omega})$ for the lowered Laplace contours, the program prints out the $\hat{k}$-coordinates of points on the image curves. The program also prints a graph of the $\hat{k}$-plane images for each of the lowered inverse Laplace contours.

ALGORITHM: The program computes $\hat{k}$-plane images for each member of a sequence of lowered inverse Laplace contours. These inverse Laplace contours are shown in Fig. 2 by solid lines. The inverse contours are confined in a rectangular region of the complex $\hat{\omega}$-plane defined by the input variables WINIT and WFINL as shown in Fig. 2. The distance between any two successive contours is given by an input variable DLTIW. Along any particular contour, the $\hat{k}$-plane image is determined by solving Eq. (1) by Muller's method for $\hat{k}$ at intervals of $Re(\hat{\omega})$ given by the input variable DLTRW. CRMWK first computes the image of the contour connecting point WINIT and A in Fig. 2. Then the program repeats the same procedure for the next contour. This procedure is repeated until the imaginary part of the contour exceeds Im(WFINL). For each inverse Laplace contour, both a numerical table of coordinates of image points and a graph of the images are printed out.

FILES: The source file, the input file, and the output file are described in the following:

1. CRMWK.FOR: This is the source file written in double precision. Since the IMSL library is employed, the execute file is generated by the link command: LINK CRMWK, IMSL$:IMSL/LIB.

2. INCRMWK.DAT: This is the input file which contains the input data. The following values are input in the format 4F15.5 in the following sequence:

BEE	EPSIL	BETAP	DELTA	TST
Re(WINIT)	Im(WINIT)	Re(WFINL)	Im(WFINL)	
DLTRW	DLTIW			
XN				
HWD	DRKMIN	DRKMAX	DIKMIN	DIKMAX
CH	CV			

The quantities in Eq. (1) represented by the above FORTRAN input variables are b=BEE, ϵ = EPSIL, $\beta_{\|}$=BETAP and δ/r_w = DELTA. The variable TST controls the amount of numerical data which is printed out by the program. As the routine traverses an inverse Laplace contour in steps of DLTRW, data is printed out only at the end of every TST-th step. The quantity XN is the number of roots solved for by Muller's method. In this program, its value is 4.0, because Eq. (1) is a fourth degree polynomial in $\hat{k}$. The graph of the $\hat{k}$-plane images is printed by the printer with the $Re(\hat{k})$-axis printed vertically on the page. Several of the above input variables control the size of this graph and the maxima and minima of the axes. HWD is the length of the $Re(\hat{k})$-axis in terms of printer line spaces. A value of 30.0 to 100.0 is recommended. The limits of the $Re(\hat{k})$-axis are controlled by the input variables CH, DRKMIN, and DRKMAX. There are just two values for CH, which are 0.0 and 1.0. If CH = 1.0, then the minimum and maximum of the $Re(\hat{k})$-axis are given by DRKMIN and DRKMAX, respectively. If CH = 0.0, then the limits of the axis are the minimum and maximum values of $Re(\hat{k})$ computed. Similarily, the input variables CV, DIKMIN, and DIKMAX control the limits of the $Im(\hat{k})$-axis. The remaining input variables have been described in the summary of the algorithm.

3. OPCRMWK.DAT: This is the output file. The output lists the input parameters. Then for each lowered inverse Laplace contour a table and a graph are printed.

The format of the table is as the following:

rw=	iw=	real k=	imag k=	matrh=	matrv=
- - - -	- - - -	- - - -	- - - -	- - - -	- - - -
- - - -	- - - -	- - - -	- - - -	- - - -	- - - -

In the above table rw=$Re(\hat{\omega})$ and iw=$Im(\hat{\omega})$, where $\hat{\omega}$ is a point on the lowered inverse Laplace contour. Moreover, real k=$Re(\hat{k})$ and imag k=$Im(\hat{k})$, where $\hat{k}$ is an image of $\hat{\omega}$. The graph of the $\hat{k}$-plane images immediately follows the table. This graph is printed by the printer. The quantities matrh and matrv, appearing in the table, give the respective coordinates of $Re(\hat{k})$ and $Im(\hat{k})$ in terms of printer spaces. If the graph is held such that the $Re(\hat{k})$-axis is horizontal, then the printer-space coordinates of the lower-left corner of the graph are (1, 1). Isolated dots appear on the graph. Their printer-space coordinates are (11,11), (11,21), (21,11), etc. Matrh and matrv allow the user to determine the value of $Re(\hat{\omega})$ associated with each point on the graph.

All of the output is also printed on the screen of the terminal.

REMARKS: The dispersion relation appears only in subroutine FN and is easily replaced by another dispersion relation. Users may find that it is also neccessary to modify the read statements, the write statements, the variable declaration statements, and the common statements.

C. Description of Program CRMWKC.

This program, written in single precision, is the CRAY version of CRMWK.FOR. Directions for running it are the same as those for CRMWK.FOR, except that a CRAY routine such as COSMOS must be used to compile the program. The input file is named INPUT, and the output file is named OUTPUT.

D. Description of Program CALEPS.FOR

PURPOSE: CALEPS.FOR computes the coupling constant (ϵ) for the CRM dispersion relation in Eq. (1). Two values of ϵ are computed; that for a thin annular beam and that for a solid beam with a uniform distribution of guiding centers.

ALGORITHM: This program calculates the coupling constant for a hollow beam using Eq. (2). Then Simpson's rule is used to evaluate the integral in Eq. (3) in order to obtain the coupling constant for a uniform solid beam.

FILES: The source file, the input file and the output file are described in the following:

1. CALEPS.FOR: This is the source file written in double precision. Since the IMSL library is employed, the execute file is generated by the link command: LINK CALEPS, IMSL$:IMSL/LIB.

2. INCALEPS.DAT: This file is the input file. The following parameters are input in the format 4F15.5 in the following sequence.

BEE	BETPAR	BETPER	XI	RW
RB	XMM	XNN	XS	XNF

Input variables appearing in Eqs. (1)-(3) are b=BEE, $\beta_{\|}$= BETPAR, $\beta_{\perp}$=BETPER, I=XI in units of Amperes, r_w=RW in centimeters , r_b=RB in centimeters, and the s=XS. The TE_{mn} mode indices are given by m= XMM and n= XNN. XNF is the number of increments of the interval between 0 to r_b/r_w to be used in the evaluation of the integral in Eq. (3) by Simpson's rule. XNF must be odd. The suggested value is XNF=101.0. If XNF is even, then the program stops and a notice 'CHANGE XNF TO AN ODD INTEGER' shows on the screen.

3. OPCALEPS.DAT: This is output file. The input parameters are listed first, followed by the coupling constant for the hollow beam, the coupling constant for the solid beam, and the ratio of the coupling constant for the hollow beam to the coupling constant for the solid beam. The output is also printed on the screen of the terminal.

E. Description of Program CRMPIN.FOR

PURPOSE: This program computes exact pinch-point coordinates $(\hat{k}_s, \hat{\omega}_s)$ of CRM dispersion relation in Eq. (1) by solving the conditions in Eq. (4) simultanously. Newton's method is used. The result will be the pinch-point coordinates if the initial estimate for the solutions used in Newton's method are sufficiently close to the exact solutions. These initial estimates can be obtained from the programs APCVPIN.FOR or CRMPINCHV.FOR.

ALGORITHM: Newton's method is used to compute pinch-point coordinates $(\hat{k}_s, \hat{\omega}_s)$. At each step of the computation, corrections $(\delta\hat{k},\ \delta\hat{\omega})$ are obtained to the values $\hat{k}$ and $\hat{\omega}$ obtained at the end of the previous step. To make the procedure stable, each of the corrections is multipled by the factor WEIT to obtain the new approximate values $\hat{k} + \delta\hat{k} *$ WEIT and $\hat{\omega} + \delta\hat{\omega} *$ WEIT. WEIT should be less than one. A value of 0.2 works well in most applications. The iteration terminates in either one of two ways:

(1) The procedure terminates when every one of the ratios $Re(\delta k)/(Re(\hat{k}) + a)$, $Im(\delta k)/(Im(\hat{k}) + a)$, $Re(\delta w)/(Re(\hat{\omega}) + a)$, and $Im(\delta w)/(Im(\hat{\omega}) + a)$ is less than the input value XTOL, where $a = 10^{-8}$. A value of XTOL$\simeq 10^{-7}$ is recommended.

(2) The procedure terminates when the number of iterations exceeds the input variable XMAXIT. The statement "THE ITERATION DID NOT CONVERGE" appears in the output if this type of termination has occurred. The suggested value for the input variable is XMAXIT = 1000.0

FILES: The source file, the input file and the output file are described in the following:

1. CRMPIN.FOR: This is the source file written in double precision. Since the IMSL library is employed, the execute file is generated by the link command: LINK CRMPIN, IMSL$:IMSL/LIB.

2. INCRMPIN.DAT: This is the input file which contains the input data. The following values are input in the format 4F15.5 in the following sequence:

BEE	EPSIL	BETAP	DELTA
S	XM	XN	
Re(X(1))	Im(X(1))	Re(X(2))	Im((X(2))
XTOL	XMAXIT	WEIT	

The quantities in Eq. (1) represented by the input variables are b=BEE, ϵ = EPSIL, $\beta_{\|}$= BETAP and δ/r_w=DELTA. The TE_{mn} mode indices are given by m = XMM and n= XNN, and s=S is the beam harmonic. X(1), X(2) are the initial estimates for the respective pinch-point coordinates $\hat{k}_s$ and $\hat{\omega}_s$ to be used in Newton's method. These initial estimates can be obtained by running APCVPIN.FOR or CRMPINCHV.FOR. The input variables XTOL, XMAXIT, and WEIT have been described in the discussion of the algorithm. Suggested values are XTOL=0.0000001, XMAXIT=1000.0, and WEIT=0.2.

3. OPCRMPIN.DAT: This is output file. It gives input pararmeters and the computed pinch-point coordinates. A warning is printed if the iteration failed to converge (i.e., that the number of iterations exceeded XMAXIT.) The output is also printed on the screen.

REMARKS: The dispersion relation is in subroutine FN and is easily replaced by another dispersion relation. Users also may find it is necessary to modify the read statements, the write statements, the variable declaration statements, and the common statements.

F. Description of Program APCVPIN.FOR

PURPOSE: This program calculates approximate pinch-point coordinates for the CRM dispersion relation in Eq. (1) in a general reference frame, moving with velocity β_v in the axial direction relative to the laboratory frame. Use of the program is limited to the case of $\delta = 0$ and $sb' > 1$, where b' is defined in Eq. (16). General-frame pinch-point coordinates are useful for determining pulse-shapes for absolute and convective CRM instabilities.[7,8] Users interested only in the laboratory frame should set the input variable BETV=0.0.

ALGORITHM: The CRM dispersion relation in a general frame has the same form as in the laboratory frame in Eq. (1), but with all the noninvariant quantities primed. First the program transforms the laboratory-frame input parameters b, ϵ and $\beta_{\|}$ in Eq.(1) to their general frame values, b', ϵ' and $\beta'_{\|}$ with the aid of Eq. (16) and the velocity addition rule. If $sb' < 1$, then a notice 'SBV< 1 HERE. THIS PROGRAM IS ONLY APPLICABLE FOR SBV > 1' appears on the screen of the terminal and the program stops. If $\beta_v - \beta_{\|} > 0$, then the approximate pinch-point coordinates are computed using Eq.(25)–(26) of Ref. 7. Otherwise the approximate coordinates are computed using Eq. (25) and Eq.(27) of Ref. 7. Eqs. (25)–(27) of Ref. 7 are the generalization of Eq. (14) to the general frame.

FILES: The source file, the input file, and the output file are described in the following:

1. APCVPIN.FOR: This file is the source file written in double precision.

2. INAPCVPIN.DAT: This file is the input file. The following values are input in the format 4F15.5 in the following sequence:

BEE	EPSIL	BETPAR	S
BETV			

The quantities in Eq. (1) represented by the FORTRAN input variables are b=BEE, ϵ = EPSIL, and $\beta_{\|}$= BETAP. It is emphasized that these are laboratory frame parameters. BETV represents the dimensionless velocity of the general reference frame (β_v) relative to the laboratory frame. Users interested only in the laboratory frame should set BETV=0.0.

3. OPAPCVPIN.DAT: This is output file containing the following values:

a. Laboratory frame input parameters.

b. The general frame values of the parameters.

c. The upshifted $(\hat{k}'_+, \hat{\omega}'_+)$ or the downshifted $(\hat{k}'_-, \hat{\omega}'_-)$ intersections of the uncoupled beam and waveguide general-frame dispersion relations.[7]

d. The approximate pinch-point coordinates: $Re(\hat{k}'_s)$, $Im(\hat{k}'_s)$, $Re(\hat{\omega}'_s)$, and $Im(\hat{\omega}'_s)$. The format for those values is:

rkp= xikp= rwp= xiwp=

The output is also written on the screen of the terminal.

G. Description of Program CRMPINCHV.FOR

PURPOSE: This program calculates pinch-point coordinates for the CRM dispersion relation in a general reference frame moving with velocity β_v in the axial direction. Use of the program is limited to the case of $\delta = 0$ and $sb' < 1$, where b' is defined in Eq. (16). General-frame pinch-point coordinates are useful for determining pulse-shapes[8] for absolute and convective CRM instabilities. Users interested only in the laboratory frame should set the input variable BETV=0.0.

ALGORITHM: The CRM dispersion relation in a general frame has the same form as in the laboratory frame [Eq. (1)] but with all the noninvariant quantities primed. The program transforms the laboratory-frame input parameters b, ϵ and $\beta_{\|}$ in Eq. (1) into their general-frame (primed) values b', ϵ' and $\beta'_{\|}$ with the aid of Eq. (16) and the velocity addition rule. If $sb' > 1$, a notice appears on the screen of the terminal and the program stops. Next, the program computes the general-frame critical coupling constant ϵ'_c and the corresponding pinch-point coordinates $(\hat{k}'_c, \hat{\omega}'_c)$ in the general frame using Eqs. (28)–(34) of Ref. 7. [These equations are the same as Eq. (6)–(13) with all the noninvariant variables primed.] If $\epsilon' < \epsilon'_c$, then the program prints the message "THERE IS NO PINCH POINT" and stops. The interval $\epsilon' - \epsilon'_c$ is broken into a sequence of values of the coupling constant in the general frame starting with $(\epsilon'_c + \delta\epsilon')$, ending with ϵ', and incremented by $\delta\epsilon'$. Here $\delta\epsilon'$ is $(\epsilon' - \epsilon'_c)$/XNEPS, where XNEPS is an input variable. The program successively computes pinch-point coordinates for each member of the sequence by using Newton's method to solve the conditions in Eq. (4) simultaneously. Initial estimates for each value of the coupling constant are those obtained for the previous value of the coupling constant in the sequence. Initial estimates for the first value in the sequence $(\epsilon'_c + \delta\epsilon')$ are $(\hat{k}'_c,\ \hat{\omega}'_c + i\text{SIMAG})$. Here, SIMAG is a small positive number, which the program instructs the user to input from the screen. This small positive value is needed to insure that the computation leads to the pinch-point coordinates instead of the coordinates of a saddle point which is not a pinch point. If $s^2 b'^2 > b'^2_0 = 1 - \beta'^2_{\|}$, then the program computes an approximate value of the maximum of the growth rate curve ($Im(\hat{\omega}')$ vs. real $\hat{k}'$), obtained from the CRM dispersion relation in Eq. (1), before requesting a value

for SIMAG. It is suggested that SIMAG be set equal to this maximum value divided by XNEPS. If $s^2 b'^2 < b_0'^2$, then no aid is given the user in the selection of SIMAG. Values in the range of 10^{-3} to 10^{-6} usually result in successful computation of the pinch-point coordinates. [Note that an unsuccessful computation is indicated by a zero or negative value of $Im(\hat{\omega}_s')$.]

FILES: The source file, the input file and the output file are described in the following:

1. CRMPINCHV.FOR: This is the source file written in double precision.

2. INCRMPINCHV.DAT: This is the input file. The following values are input in the format 4F15.5 in the following sequence:

BEE	BETAP	EPSIL	S
BETV	XNEPS		
XTOL	XMAXIT	WEIT	

The quantities in Eq. (1) represented by the above FORTRAN input variables are b=BEE, ϵ = EPSIL, $\beta_{\|}$=BETAP, δ/r_w = DELTA and s=S. These are laboratory frame values. BETV represents β_v, the velocity of the general reference frame in the axial direction relative to the laboratory frame. Users interested only in the laboratory frame should set BETV=0.0. XNEPS is the number of increments between ϵ' and ϵ_c', discussed in the description of the algorithm. The value suggested is XNEPS=100.0. The input variables XTOL, XMAXIT, and WEIT are described in the description of the program CRMPIN.FOR. Suggested values for these parameters are XTOL=0.0000001, XMAXIT=1000.0, and WEIT=0.2.

3. OPCRMDISP.DAT: This is output file. It first lists the laboratory-frame values of the input parameters. This list is followed by a list of their values relative to the general frame. The general-frame critical coupling constant ϵ_c' and the corresponding pinch-point coordinates are given next. Finally, the general-frame pinch-point coordinates ($\hat{\omega}_s'$, $\hat{k}_s'$) corresponding to the general-frame coupling constant ϵ' are listed in the following format:

rekp= imkp= rewp= imwp=

The output is also typed on the screen of the terminal.

H. DESCRIPTION OF PROGRAM EXEPSCRIT.FOR

PURPOSE: Program EXEPSCRIT.FOR computes the critical coupling constant ϵ_c for the CRM absolute instability as a function of $\beta_{\|}$, over an interval of $\beta_{\|}$, for a given value of sb. The program was used to construct the plot in Fig. 1. Use of EXEPSCRIT.FOR is limited to the case of zero waveguide skin depth [i.e., $\delta = 0$ in Eq. (1)]. The program GENCARM.FOR is applicable to the case of a nonzero skin depth.

ALGORITHM: The program calculates the critical coupling constant ϵ_c, for a sequence of values of $\beta_{\|}$ starting from BETPAR, incremented by DBETP, and ending with BETPF. If the condition in Eq. (9) is satisfied, then ϵ_c, and the corresponding pinch-point coordinates $(\hat{k}_s, \hat{\omega}_s)$ are computed using Eqs. (6)–(8). Otherwise Eqs. (10), (12), and (13) are used.

FILES: The source file, the input file and the output file are described in the following:

1. EXEPSCRIT.FOR: This is source file written in double precision.

2. INEXEPSCRIT.DAT: This is the input file. Input the following parameters in format 4F15.5 in the folowing sequence:

BEE	BETPAR	BETPF	DBETP

In the above , BEE $=sb$, given in Eq. (1). BETPAR, BETPF and DBETP have been described in the ALGORITHM above.

3. OPEXEPSCRIT.DAT: This is the output file. It gives the input value of sb, followed by a table giving the critical coupling constant ϵ_c, and the corresponding pinch-point coordinates $(\hat{k}_s, \hat{\omega}_s)$ for each value of $\beta_{\|}$. The format of the table is:

betpar=	epsilon=	xk=	xw=
- - - -	- - - -	- - - -	- - - -
- - - -	- - - -	- - - -	- - - -
- - - -	- - - -	- - - -	- - - -

Here, betpar=$\beta_{\|}$, epsilon=ϵ_c, xk =$\hat{k}_s$, and xw=$\hat{\omega}_s$. The output is also typed to the screen of the terminal.

I. Description of Program GENCARM.FOR

PURPOSE: Program GENCARM computes the critical coupling constant ϵ_c for the CRM absolute instability as a function of $\beta_\parallel$, over an interval of $\beta_\parallel$, for a given value of sb. The program is useful in constructing graphs similar to that in Fig. 1 for arbitrary values of the skin depth δ. If $\delta = 0$, then the user should employ the program EXEPSCRIT.FOR instead of GENCARM.FOR, because the former program is much easier to use and requires much less CPU time than the latter program.

ALGORITHM: The program computes ϵ_c for the initial value of $\beta_\parallel$ specified by the input variable BETAP. This value is successively incremented by the input variable DBETAP until $\beta_\parallel$ exceeds the input variable BETAPF. For each value of $\beta_\parallel$, the program first computes pinch-point coordinates for a value of the coupling constant specified by the input parameter ϵ_0=XI. XI must exceed ϵ_c for all values of $\beta_\parallel$ over the interval of the computation. XI is then decremented by the value of the input parameter DXI, and the pinch-point coordinates $(\hat{k}_s,\hat{\omega}_s)$ are recalculated. This process continues until $Im(\hat{\omega}_s) \leq 0$. Then the current value of XI is increased by DXI, DXI is reduced by a factor of 10, and the process of decrementing XI is continued. The critical coupling constant ϵ_c is selected to be that value of XI for which $Im(\hat{\omega}_s) \leq 0$ and DXI/XI first becomes less than DXIM, where DXIM is an input parameter.

FILES: The source file, the input file, and the output file are described in the following.

1. GENCARM.FOR: This is the source file written in double precision. Since the IMSL library is employed, the execute file is generated by the link command: LINK GENCARM, IMSL$:IMSL/LIB.

2. INGENCARM.DAT: This is the input file. Input the following parameters in format 4F15.5 in the folowing sequence:

BEE	DELTA		
BETAP	BETAPF	DBETAP	
XMM	XNN		
XI	DXI	DXIM	
Re(X(1))	Im(X(1))	Re(X(2))	Im(X(2))
XTOL	XMAXIT	WEIT	

Quantities in the CRM dispersion relation in Eq. (1) represented by the FORTRAN input variables are b=BEE and δ/r_w=DELTA. The initial value, increment, and final value of $\beta_{||}$ are BETAP, DBETAP and BETAPF, respectively. The TE_{mn} mode is specified by m=XMM and n=XNN. The FORTRAN variables XI, DXI, and DXIM have been described in the description of the algorithm. DXIM determines the accuracy to which ϵ_c is determined. The value DXIM=0.0001 is typical. However, selecting a smaller value does not greatly lengthen CPU time. The selection of a proper value for XI requires some experience in running the program and familiarity with Refs. 6 and 7. The use of Fig. 1 is also helpful. The selection of a value of XI that is too large results in undue use of CPU time. If the value selected is too small, then XI will not exceed ϵ_c over the full range of $\beta_{||}$ and the program will stop before completing the computation. The proper choice of DXI also requires some experience in the use of the program. In order to calculate the pinch-point coordinates when decrementing XI by DXI, the program uses the coordinates obtained in the previous step as the estimates for Newton's method in the current step. Consequently, a choice of DXI which is too large may results in the calculation of coordinates for a saddle-point which is not a pinch-point. The program will then compute an incorrect result for ϵ_c. On the other hand, a choice of DXI which is too small will result in undue use of CPU time. In order to compute pinch-point coordinates at the first step of the algorithm (i.e., for ϵ=XI and $\beta_{||}$=BETAP), the program requires initial pinch-point coordinate estimates $(\hat{k}_s,\hat{\omega}_s)$ for Newton's method. These are provided by the complex input variables X(1) and X(2) for $\hat{k}_s$ and $\hat{\omega}_s$, respectively. Values for X(1) and X(2) can be obtained with the aid of the programs CRMPIN.FOR, APCVPIN.FOR, and CRMPINCHV.FOR. The input variables XTOL, XMAXIT, and WEIT are described

in the description of the program CRMPIN.FOR. Suggested values for these parameters are XTOL=0.0000001, XMAXIT=1000.0, and WEIT=0.2.

3. OPGENCARM.DAT: This is output file. The output first lists the input variables. Then follows a table of three columns, marked as I, betap and epsilonc. Here betap = $\beta_{\|}$, and epsilonc=ϵ_c. A value I=1 indicates that the iteration in Newton's method converged for the last step of in the calculation of ϵ_c for the corresponding value of $\beta_{\|}$. A value I=0 shows that the computation terminated after MAXIT steps, indicating that the iteration did not converge. The format of the table is as below:

I	betap	epsilonc
-	- - - -	- - - -
-	- - - -	- - - -
-	- - - -	- - - -

The output is also typed on the screen of the terminal.

REMARKS: For reasons that have not been investigated, it is very difficult to make the computation in GENCARM.FOR converge for small values of sb (i.e., $sb < 0.4$). If the skin depth is zero, then the program EXEPSCRIT.FOR should be used instead of GENCARM.FOR, because the former program is much easier to run and uses much less CPU time than the latter. GENCARM.FOR runs very slowly on the PFCVAX. The CRAY version of the program is GENCARMC.. The dispersion relation is in subroutine FN and is easily replaced by another dispersion relation. Users also may find it is necessary to modify the read statements, write statements, variable declaration statements, and common statements.

J. Description of Program GENCARMC. This program, written in single precision, is the CRAY version of GENCARM.FOR. Directions for running it are the same as for GENCARM.FOR, except that a CRAY routine such as COSMOS must be used to compile the program. The name of the input file is INPUT, and the name of the output file is OUTPUT.

ACKNOWLEDGEMENTS

The authors wish to acknowledge J. J. M. Williams and W. Zhang for contributions made to these programs. This work was supported in part by the Department of Energy High Energy Physics Division, the Naval Research Laboratory Plasma Physics Division, and the Office of Naval Research.

REFERENCES

1. R. J. Briggs, *Electron Stream Interactions with Plasmas* (MIT Press, Cambridge, MA, 1964).
2. A. Bers, in *Basic Plasma Physics* 1, Handbook of Plasma Physics, edited by A. A. Galeev and R. N. Sudan (North-Holland, Amsterdam, 1983), Vol. 1, Chap. 3.2.
3. Y. Y. Lau, K. R. Chu, L. R. Barnett, and V. L. Granatstein, Int. J. Infrared Millimeter Waves 2, 373 (1981).
4. Y. Y. Lau, K. R. Chu, L. R. Barnett, and V. L. Granatstein, Int. J. Infrared Millimeter Waves 2, 395 (1981).
5. K. R. Chu and A. T. Lin, IEEE Trans. Plasma Science 16, 90 (1988).
6. J. A. Davies, Phys. Fluids B1, 663 (1989).
7. J. A. Davies, R. C. Davidson and G. L. Johnston, 1989 MIT Plasma Fusion Center Report No. JA-89-31.
8. A. Bers, A. K. Ram and G. Francis, Phys. Rev. Lett. 53, 1457 (1984).
9. J. A. Davies, R. C. Davidson and G. L. Johnston, J. Plasma Phys. 40, 1 (1988).

FIGURE CAPTIONS

1. Plot showing parameter regions of absolute instability for the CRM for the case of zero waveguide-wall resistance. The coupling of the beam mode and the waveguide mode produces absolute instability if the point $(\beta_{\|}, \epsilon)$ lies above the curve of sb for the beam-mode harmonic.
2. Schematic plot illustrating the input variables WINIT, WFINL, and DLTIW for the programs CRMWK.FOR and CRMWKC..

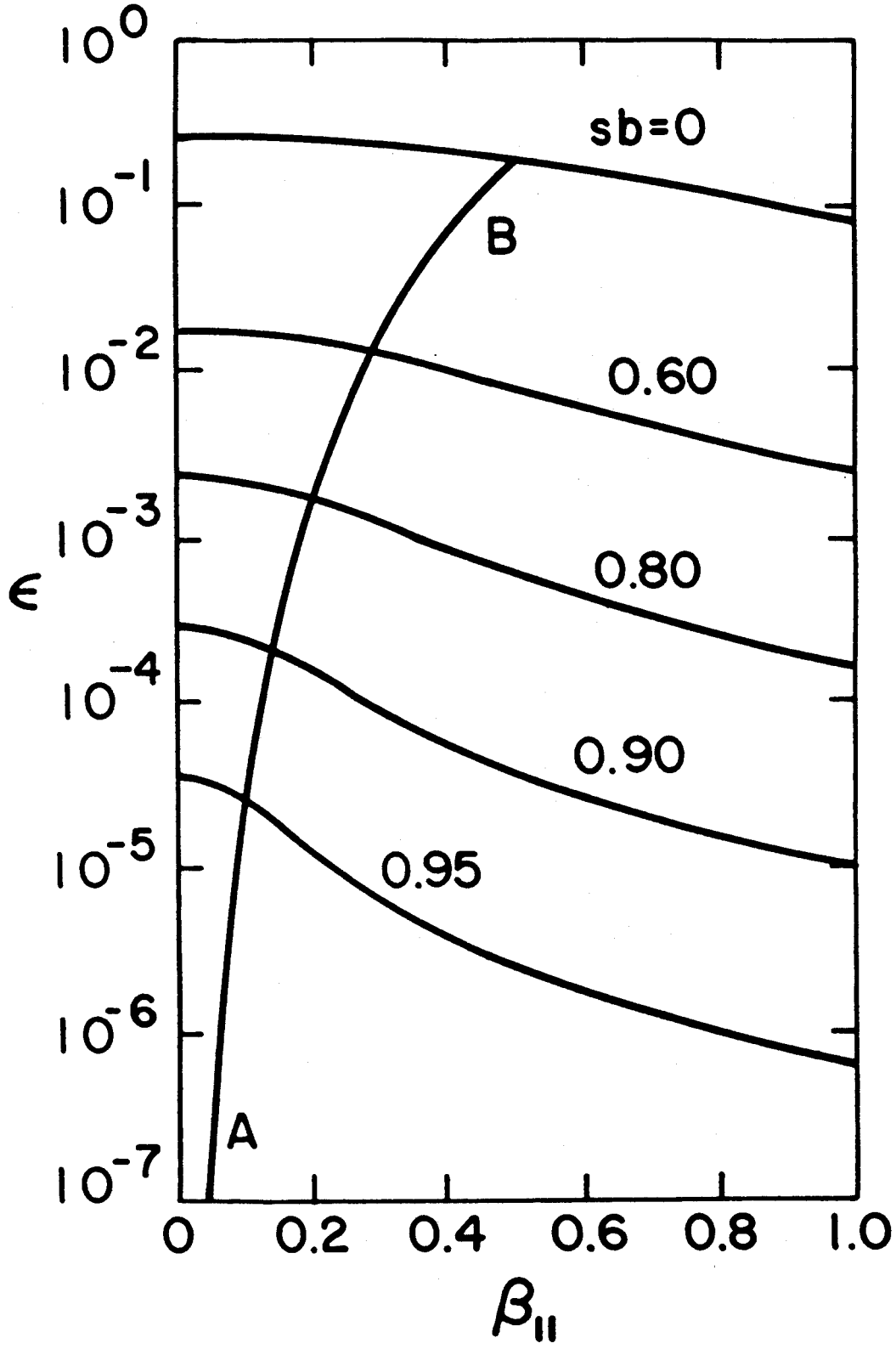
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
ϵ
sb=0
B
0.60
0.80
0.90
0.95
A
0
0.2
0.4
0.6
0.8
1.0
β_{11}

Fig. 1

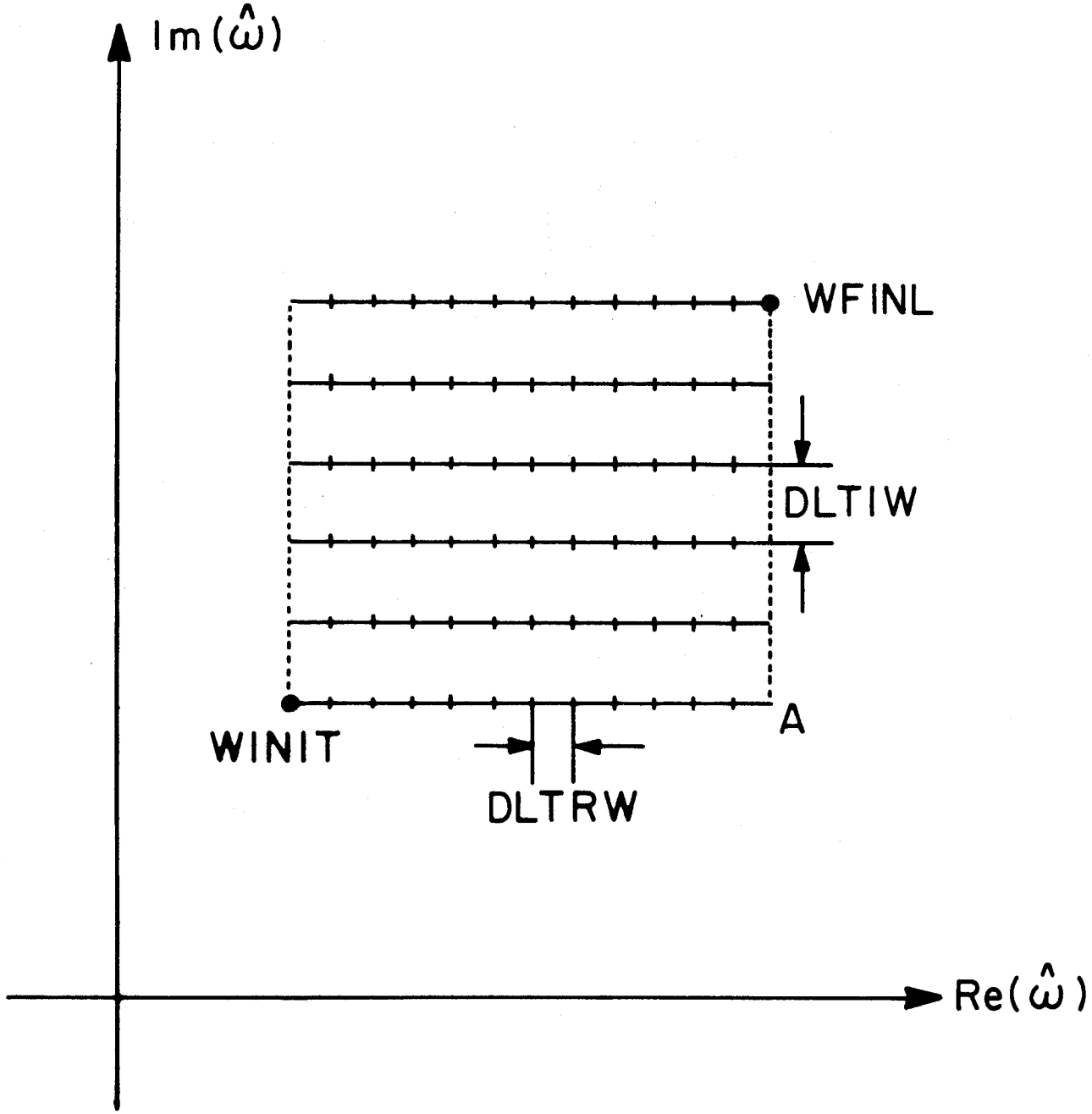

Fig. 2