# ROOTS, A DISPERSION EQUATION SOLVER

by

Michael J. Gerver

Memorandum No. ERL-M77-27

October 31, 1976

Electronics Research Laboratory

College of Engineering
University of California, Berkeley
94720

# ROOTS, A DISPERSION EQUATION SOLVER*

Michael J. Gerver*

Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory,
University of California, Berkeley, California 94720

October 31, 1976

I. Purpose of ROOTS
II. How to use
III. Input format
IV. What to do if the code does not work
V. Making changes in the code

This report complements Memorandum No. ERL-M602 of 24 September 1976, Appendix I, which outlines the main program ROOTS and the subroutines used, along with a listing. The physics of the problem is in ERL-M602 and in the Physics of Fluids, 20, beginning on page 291, Feb. 1977; as noted therein, ROOTS was begun by Dieter Fuss in 1969, working with C.K. Birdsall, with great help from A.B. Langdon. The current ROOTS is much more powerful and is due to Gerver.

Communications on use of ROOTS et al, should be sent to:

Professor C. K. Birdsall
EECS Dept, Cory Hall
University of California
Berkeley, CA 94720

Memorandum No. ERL-M77-27

* Work supported by ERDA Contract #EY-76-S-03-0034-PA128
** Present address: Laboratory of Plasma Studies, Cornell University, Ithaca, N.Y. 14853

## I. PURPOSE OF ROOTS

The code ROOTS finds and plots the linear normal mode frequencies $\omega(k_y)$ of electrostatic perturbations of a Vlasov plasma with unperturbed ion guiding center density varying in the x-direction as

$$n_i(x_{gc}) = n_o [1 + \epsilon_i \cos(k_o x_{gc})] \qquad (1)$$

and a uniform magnetic field in the z-direction. Complex values of $\omega$ are found for real $k_y$; $k_z$ is assumed to be zero. The code also finds and plots the linear normal mode potential $\phi(x,y)$ and perturbed ion and electron charge density $\rho_i(x,y)$ and $\rho_e(x,y)$ for the fastest growing normal mode, as well as $|\phi(x)|^2$ for other normal modes; these are all assumed to be periodic in x with period $2\pi/k_o$. The unperturbed electron guiding center density is assumed to be such that there is no net unperturbed charge density.

The perpendicular velocity distribution of each species can be either Maxwellian

$$f_s(v_\perp) = \exp(-v_\perp^2/2v_s^2)/2\pi v_s^2 \qquad (2)$$

or a ring distribution

$$f_s(v_\perp) = \delta(v_\perp - v_s)/2\pi v_s \qquad (3)$$

or a loss-cone distribution:

$$f_s(v_\perp) = \frac{R_s}{2\pi v_s^2 (R_s - \Gamma_s)} [\exp(-v_\perp^2/2v_i^2) - \Gamma_s \exp(-R_s v_\perp^2/2v_i^2)] \qquad (4)$$

Up to four species can be used; however, loss-cone distributions count as two species (since internally the code treats loss-cone species as two Maxwellian species).

In general, there are an infinite number of normal modes for a given $k_y$. The code finds only those in a specified frequency range, and

also assumes that the normal mode potentials are of the form

$$\phi(x,y) = \exp(ik_y y) \sum_{p=-p_{max}}^{p_{max}} \phi_p \exp(ipk_o x) \qquad (5)$$

Thus the results are only to be trusted if $\phi_p$ is small for $|p| \sim p_{max}$.

If the input parameter PMAX is set equal to zero, then instead of a sinusoidal density profile, the plasma is assumed to have a slight constant density gradient,

$$n_i(x) = n_o(1 + \varepsilon x) \qquad (6)$$

where $\varepsilon a_i << 1$

and the local approximation is used (i.e. perturbed quantities are assumed to be independent of x, while unperturbed quantities are evaluated at x = 0). The ring distribution option has not yet been implemented for the local approximation. (If you try using a ring distribution with PMAX=0, the code will run but will give you garbage).

The code UNIFORM is similar to ROOTS, but uses a plasma uniform in space. It also differs from ROOTS in that it stores roots $(\omega,k_y)$ in LCM, so that it has room to store more roots; and, in addition, it can handle warm ring distribution functions of the form

$$f_s(v_\perp) = \frac{\exp[-2j_s(v_s - v_\perp)^2/v_s^2]j_s^{\frac{1}{2}}}{2^{\frac{1}{2}}\pi^{3/2}v_s} \quad \text{for } j_s >> 1 \qquad (7)$$

but only when the straight-line orbit approximation is used.

## II. HOW TO USE

The Fortran listings and sample input files for ROOTS and UNIFORM can be obtained by typing (on the NMFECC, at LLL, Livermore)

filem 1205 rds .public filelist

where "filelist" is a list of the files wanted.

rootsf - Fortran listing of ROOTS

inputrl - sample input for ROOTS

inputr2 - another sample input for ROOTS

uniform - Fortran listing of UNIFORM

inputu - sample input for UNIFORM

To run one of the codes (say ROOTS) in its present form, type

filem 1205 rds .public rootsf inputrl

(if desired, inputr2 could have been specified instead).

When you get an "all done" from filem, type

switch inputrl input

You should then make whatever changes you want to make in the input file, using your favorite text editor. The input format is described in Sec. III.

Then, to compile ROOTSF, type

CHATR(ORDERLIB, CF76LIB, TV80LIB, SYMT↑) ROOTSF RUNROOTS C D box M M

where "box" is your box number. This will produce the executing file RUNROOTS, as well as files called SYMT, C, D, and D1, which won't be needed. Since the compilation takes the better part of a minute of run time, RUNROOTS should be saved in your own filem file to avoid having to recompile it in the future. RUNROOTS will not destroy itself when it is executed.

Then, to execute, type

runroots

The execution time in seconds is very roughly given by

$$\text{Time} = 0.01*(2*\text{PMAX} + 1)^2\,(\text{FKYAI} - 0.5*\text{SKYAI})*(\text{RMAX}-\text{RMIN})*\text{NSPEC}$$

where PMAX is the cutoff in $k_x/k_o$, discussed above

FKYAI is the maximum value of $k_y a_i$

SKYAI is the minimum value of $k_y a_i$

RMAX and RMIN are the maximum and minimum values of $\omega/\omega_{ci}$

NSPEC is the number of species (loss cones count as two species), excluding cold species

The above estimate is for runs with magnetized particle orbits. For straight-line orbits, the term (FKYAI -0.5*SKYAI)*(RMAX-RMIN) should be replaced by $4*\log_{10}(2*\text{FKYAI}/\text{SKYAI})$.

This estimate assumes that each branch of $\omega(k_y)$ is a fairly smooth function; some runs, with very "hairy" branches of $\omega(k_y)$, have taken four times longer, while a nearly flat $\omega(k_y)$ might take less than half the time.

The code will produce a DD80 file called ROOTPLOTAX (or, if you already have an active file ROOTPLOTAX, the DD80 file will be called ROOTPLOTBX, etc.), as well as an ASCII file called OUTPUT. The DD80 file can be viewed with TEKPLOT, or disposed to microfiche or hardcopy with FROG. It will list, for the fastest growing mode of each branch of $\omega(k_y)$, the complex frequency $\omega/\omega_{ci}$ and wavenumber $k_y a_i$, the absolute value of each $\phi_p$ for $-p_{max} \leq p \leq p_{max}$, and the argument (in degrees) of each $\phi_p$. For the branch containing the fastest-growing mode of any branch, this listing is preceded by a list of the real and imaginary parts of $\phi_p$ for each p, the corresponding information for the perturbed ion and electron charge densities, $\rho_i$ and $\rho_e$ are contour plots of $\phi(x,y)$, $\rho_i(x,y)$ and $\rho_e(x,y)$. Following all these listings and plots are plots of $|\phi(x)|^2$ and cross-section plots of $\phi(x,y)$ for the fastest-growing mode of each branch, as well as

for other modes if requested (via the input parameter EIGV). Finally there is a plot of the real part of $\omega(k_y)$ and a plot of the imaginary part of $\omega(k_y)$. The last two plots have the input parameters listed beneath the plot on the same frame. The plot of $Im\omega(k_y)$ also has the complex frequency and wavenumber of the fastest growing mode, and of the mode with maximum $\gamma/k^2$, listed beneath it.

The file OUTPUT is normally not needed unless something goes wrong. Among other things, it tells how long it took to follow each branch of $\omega(k_y)$, whether or not each branch was followed successfully, and whether the execution ended normally or due to running out of time or running out of memory to store roots.

III. INPUT FORMAT

First line:

1 - must be "T"

2 - UNMAG = "T" means straight-line orbit approximation is used for all non-cold species.

"F" means magnetized particle orbits are used for all species

3 and 4 - blank

5 - OPT(1) = 1 means that branches $\omega(k_y)$ for which $\phi_0 = 0$ ($\phi_0$ is the p=0 component of the perturbed potential) will not be followed. Half of the branches have this property, and usually (though not always) the fastest-growing branch has $\phi_0 \neq 0$, so setting OPT(1) = 1 will cut computation time in half if you are only interested in the fastest growing mode or branch.

"0" or blank otherwise

6 - OPT(2) = 1 means that if you are using straight-line orbits, only branches with $Re\omega/k_y < v_i$ initially will be followed. Since the dispersion function takes much longer to calculate when $\omega/k_y > v_i$, OPT(2) = 1 is desirable if you know from physical considerations that the modes of interest must have $\omega/k_y < v_i$.

"0" or blank otherwise.

OPT(2) is ignored if you are not using straight-line orbits.

Second line:

1 to 5 - RMAX is the highest frequency that you want searched initially, in units of the ion cyclotron frequency; integer format

6 to 9 - blank

10 - PMAX; the sum over p (i.e. $k_x/k_o$) is cut off at $\pm$ PMAX.

PMAX must be no greater than 9. PMAX up to 9 has been used successfully with straight-line orbits and PMAX up to 5 has been used

successfully with magnetized orbits.

If PMAX = 0, the local approximation is used with constant density gradient, rather than the normal mode calculation with the sinusoidal density profile.

If you are running UNIFORM, PMAX must be set equal to zero.

11 to 14 - blank

15 - EIGV. This should normally be 2, in which case only the potentials $|\phi(x)|^2$ of the fastest-growing mode of each branch will be plotted.

If EIGV = 3, $|\phi(x)|^2$ will be plotted for a representative sample of all unstable modes.

If EIGV = 4, $|\phi(x)|^2$ will be plotted for a representative sample of all modes.

16 to 20 - SKYAI is the initial (minimum) value of $k_y a_i$ wanted. Usually we have used 0.5. Higher or lower values should be used if the desired branch is not found. If SKYAI is set too high, branches may be missed because their growth rate is too high. If SKYAI is set too low, branches may be missed because they are too close together.

21 to 25 - DK is the initial relative increment of $k_y a_i$ used in following each branch. The increment is later adjusted up or down by the code depending on how smooth the branch is. We have always used 0.1.

26 to 30 - FKYAI is the maximum value of $k_y a_i$ wanted.

31 to 34 - blank

35 - NSW1 - If NSW1 = 1, information is printed out (in the file OUTPUT) on each root $\omega$ found at the initial $k_y a_i$. This includes information on unsuccessful attempts made to converge on roots.

This information may be useful if the code failed to find a branch you wanted.

36 to 49 - blank

50 - NSPEC; NSPEC = 0 should be used if you are using one species of ions (either Maxwellian, loss cone or ring) and electrons (either Maxwellian, loss cone, or ring). If you are using more complicated distributions (e.g. ring + Maxwellian, or warm Maxwellian + hot loss cone, or two ion species), then NSPEC is the number of species used (either Maxwellian or ring; loss-cone distributions are now constructed from two Maxwellians, so count as two species). NSPEC may be as great as 4.

51 to 55 - IRMIN is the minimum frequency to be searched at initial $k_y a_i$, in terms of the ion cyclotron frequency. Integer format.

Third line:

1 to 5 - AIL is $k_o a_i/2\pi$, i.e. the ratio of ion Larmor radius, $a_i = v_i/\omega_{ci}$, to the wavelength of the sinusoidal density variation. Note that, in the case of a loss cone distribution, $v_i$ is not exactly the thermal velocity, but is defined by Eq.(4). AIL is ignored if PMAX = 0 (i.e. if local approximation is used).

6 to 10 - AEAI is $a_e/a_i$, the ratio of electron to ion Larmor radius. Ignored if NSPEC > 0. If NSPEC > 0, $a_e$ is read in later.

11 to 15 - OMPSQ is $\omega_{pi}^2/\omega_{ci}^2$. If NSPEC > 0, then this is $\omega_{pi}^2/\omega_{ci}^2$ for a species of unit density; the density of each species is read in later if NSPEC > 0.

16 to 20 - EPSI is $\varepsilon_i$, the density variation of the ion guiding center profile, in Eq.(1). If NSPEC > 0, and $\varepsilon_s$ is different for each ion species, then EPSI may be superceded by values for $\varepsilon_s$ read in later.

$\varepsilon_e$ is automatically adjusted so that there is no net unperturbed charge density.

21 to 25 - MASS is $m_i/m_e$. Ignored if NSPEC > 0.

26 to 30 - EP is $\varepsilon a_i = (a_i/n)\, dn/dx$ for the local approximation, in Eq.(6). (ignored when PMAX > 0 or NSPEC > 0). If NSPEC > 0, $\varepsilon a_i$ is read in later.

31 to 35 - DFUI is an ion diffusion coefficient $D_i$, in units of $a_i^2 \omega_{ci}$, which results in $\omega$ being replaced by $\omega + iD_i k_y^2$ for ion terms. Ignored when NSPEC > 0. If NSPEC > 0, $D_i$ is read in later.

36 to 40 - DFUE is electron diffusion coefficient $D_e$. Same remarks as apply to DFUI.

41 to 45 - BETA is $\beta$, ratio of plasma pressure to magnetic field pressure. Used only for local approximation, ignored when PMAX > 0. Finite $\beta$ corrections to cold electrons are included, not to ions or hot electrons. Not yet implemented for UNIFORM, only for ROOTS.

45 to 50 - CH is the width of the regions into which the complex $\omega$-plane is divided for the initial root search. If left blank, the default value CH = 5 is used. If roots are being lost, CH = 3 or 2 may help. This has only been implemented with UNIFORM so far, not for ROOTS.

51 to 55 - COARSE;

56 to 60 - COARSI. These parameters, implemented only for UNIFORM, determine how fine or coarse the $\omega$ vs. $k_y$ plots are. Reasonable default values are used if left blank. If code is running out of room to store roots, higher values may be used.

Fourth line:

If NSPEC > 0, line 4 is blank.

If NSPEC = 0;

1 to 5 - R is the effective ion mirror ratio $R_i$ (actually the square of the ratio of the hole in the velocity distribution to the thermal velocity) for an ion loss-cone distribution. If R = 0, then a ring ion distribution (δ-function) is used. For a Maxwellian, any R ≠ 0 should be used.

6 to 10 - GAM is $\Gamma_i$, which says how empty the ion loss cone is. For a completely empty ion loss cone, use $\Gamma_i = 0$; for a Maxwellian ion distribution (completely filled loss cone), use $\Gamma_i = 1$.

11 to 15 - RE is $R_e$, the effective electron mirror ratio; ignored if AEAI = 0.

16 to 20 - GAME is $\Gamma_e$; ignored if AEAI = 0.

If NSPEC = 0, Line 4 is the last line of the INPUT file. If NSPEC > 0, then there are NSPEC additional cards, one for each species. The electron species must be read in last, and must have mass less than 0.1 times the mass of the first species read in (otherwise, the electron density gradient or guiding center density variation $\varepsilon_e$, will not be adjusted to give no net unperturbed charge density).

The remaining lines have the following form, for each species I (where I runs from 1 to NSPEC):

Fifth line, Sixth line; etc.

1 to 10 - RLARM(I) is the Larmor radius of species I.

11 to 20 - AMASS(I) is the mass of species I.

21 to 30 - CHARG(I) is the charge of species I.

31 to 40 - DENSE(I) is the density of species I.

41 to 45 - JAY(I) = 0 for a Maxwellian distribution.

JAY(I) = -1 for a δ-function ring distribution. Otherwise JAY(I)

is the quantity $j_s$ in Eq.(7), a quantity indicating the thermal spread of a ring with slight thermal spread. This is only valid for JAY(I) >> 1. If magnetized orbits are used (UNMAG = "F") then a δ-function ring is used for any JAY(I) ≠ 0.
Integer format.

46 to 55 - EPS(I) is the density variation ε for species I [in Eq.(1)], if PMAX > 0. If PMAX = 0, EPS(I) is ε, the density gradient of species I [in Eq.(6)], using the local approximation.

56 to 65 - DFU(I) is a diffusion coefficient of species I; see remarks on DFUI in the third line.

The quantities RLARM, AMASS, CHARG, and DENSE are normalized to the Larmor radius, mass, charge, and density of a reference species which has Larmor radius $a_i$, cyclotron frequency $\omega_{ci}$, and plasma frequency $\omega_{pi}$, where OMPSQ = $\omega_{pi}^2/\omega_{ci}^2$, and $a_i$ and $\omega_{ci}$ are the quantities referred to earlier in this section, e.g. SKYAI is the initial value of $k_y a_i$, RMAX is the maximum value of $\omega/\omega_{ci}$, etc. DFU is in units of $a_i^2\omega_{ci}$, roughly the Bohm diffusion coefficient for the reference species.

## IV. WHAT TO DO IF THE CODE DOES NOT WORK

Occasionally the code will fail to find one of the branches $\omega(k_y)$; this happens more often when you are using parameters that are very different from those that have been used before. If the code fails to find a branch that you are interested in, there are several things that can be done.

(1) Try changing SKYAI

(2) Try changing IRMIN. The starting points of the root search depend on IRMIN.

(3) Try reducing CH from its default value of 5, to 2 or 3, say.

(4) Try varying any of the parameters slightly. Missing roots is sometimes a matter of "chance", and if the parameters are varied slightly the root may be found.

(5) Use NSPEC > 0, and adjust the masses, Larmor radii, etc., of the species so that the physics is unchanged, but the frequencies are normalized to something other than the ion cyclotron frequency. This will result in the starting points for the root search being located at different places with respect to the ion cyclotron frequency, and may result in the missing branch being found.

This same result may be accomplished more directly and with more degrees freedom by changing P1, P2, and P3 in the Fortran listing, and re-compiling the code.

(6) As a very last resort, rip into the code and change it around as much as you dare, e.g., try finding branches initially at high $k_y a_i$ and following them down to low $k_y a_i$. Or, change the criteria or method used for finding roots.

## V. MAKING CHANGES IN THE CODE

In order to make changes in the code, use your favorite text editor on the Fortran listing file ROOTSF ( or UNIFORM). The new version may then be compiled by typing

CHATR(ORDERLIB, CF76LIB,TV80LIB, SYMT↑) ROOTSF RUNROOTS C D box M M where "box" is your box number. Here, ROOTSF is the Fortran file to be compiled, RUNROOTS will be the name of the dropfile, and SYMT will be a symbol table, to be used with DBUG if necessary. The computer will also produce a binary file called C, and ASCII files D (a listing) and D1 (a load map).

In making changes, the following points should be noted.

1) The different parts of the code are delicately intertwined. For example, in changing the dispersion function, it may not be sufficient just to change DISP, since MRAF and FOLLOW make certain assumptions about the properties of the dispersion function, e.g., whether or not it is Hermitian.

2) RUNROOTS, the execution file for ROOTS, is near the limit of size allowed for small core memory. Any substantial increase in size will require the use of large core memory. UNIFORM already uses LCM.