UC20
PFC/RR/82-8 DOE/ET/51013-36

USER'S MANUAL FOR LINEAR, A COMPUTER PROGRAM THAT CALCULATES THE LINEAR CHARACTERISTICS OF A GYROTRON

Kenneth E. Kreischer
Plasma Fusion Center
Massachusetts Institute of Technology
Cambridge, MA. 02139

January 1982

This work is supported by DOE Contract DE-AC02-78ET-51013.

I. GENERAL DESCRIPTION

A. Purpose

Calculates the linear characteristics of a gyrotron. This program is capable of:

1) calculating the starting current or frequency detuning for each gyrotron mode.
2) generating mode spectra
3) plotting these linear characteristics as a function of device parameters (e.g., beam voltage)
4) doing the above for any axial RF field profile

B. Compatibility

ANSI Standard FORTRAN IV, 1966, with some features of FORTRAN 77. This program was written using the MULTICS operating system on the Honeywell 68/DPS computer.

C. Limitations

1) The electron beam parameters are calculated assuming that adiabatic theory is valid in the gun region [1].
2) The electron beam is assumed to be annular, azimuthally symmetric, and to have no radial thickness. It is also monoenergtic and has no velocity spread.
3) The cavity is assumed to be a cylindrical open resonator.

D. Method

The starting current of a gyrotron [2] can be written as:

$$I_{ST} = - \frac{\varepsilon_o}{2} \frac{\omega}{Q_T} \left|P_o\right|^2 \frac{m_e}{e} \frac{(K_{\parallel} V_{\parallel})^2}{G(r_e)} \left[J_n'(K_{\perp} r_L)\right]^{-2} \left[F_c - \frac{1}{2} \frac{S V_{\perp}^2}{c^2} \frac{dF_c}{dx}\right]^{-1} \quad (1$$

where Q_T is the total Q of the cavity, ω is the resonator frequency, m_e is the relativistic electron mass, r_L is the electron Larmor radius, $S = \omega/K_{\parallel}V_{\parallel}$, and $V_{\parallel}$ and $V_{\perp}$ are the electron beam's parallel and perpendicular velocity respectively. The parameter $|p_o|^2$ represents the stored energy in the cavity and contains the factor:

$$E_S = \frac{1}{K_{\parallel} L} \int_{\bar{Z}_{IN}}^{\bar{Z}_{OUT}} d\bar{Z} \left[g(\bar{Z})\right]^2 \qquad (2)$$

where $g(\bar{Z})$ = axial RF field profile

$\bar{Z} = K_{\parallel} Z$

$\bar{Z}_{IN}$, $\bar{Z}_{OUT}$ = limits of the beam-field interaction

$K_{\parallel}$ = qK/L = axial wavenumber (K is a constant)

The variable $G(r_e)$ is the beam-field coupling term, and can be written as follows depending on the structure of the field in the azimuthal direction:

$$\text{(Standing)} \qquad G(r_e) = 0.25 \left[J^2_{m+n}(K_{\perp} r_e) + J^2_{m-n}(K_{\perp} r_e)\right] \qquad (3)$$

$$\text{(Rotating)} \qquad G(r_e) = 0.5 \left[J^2_{m\pm n}(K_{\perp} r_e)\right]$$

where n is the harmonic number and r_e is beam radius. The gain function F_c is expressed in terms of the detuning parameter $x = (n\omega_c - \omega)/(K_{\parallel}V_{\parallel})$ and has the following form:

$$F_c(X) = 0.5 \left[\int_{\overline{Z}_{IN}}^{\overline{Z}_{OUT}} g(\overline{Z}) e^{ix\overline{Z}} \, d\overline{Z} \right]^2 \qquad (4)$$

The frequency detuning that a gyrotron experiences can be written as:

$$\frac{2Q_T(\omega-\omega')}{\omega} = \frac{F_s - \frac{1}{2}\left(\frac{S\,V_\perp^2}{c^2}\right)\frac{dF_s}{dx}}{F_c - \frac{1}{2}\left(\frac{SV_\perp^2}{c^2}\right)\frac{dFc}{dx}} \qquad (5)$$

where ω = cavity resonant frequency with beam

ω' = cavity resonant frequency without beam

F_s = detuning gain function

In Eq.(5), the assumption $s \ll Q_{ohmic}$ is made.

E. Calculational procedure

In this code, the starting current or frequency detuning is considered the y parameter. It is calculated as a function of two other parameters x and z. A range of x must be supplied, as well as discrete values of z. As a typical example, for a mode spectra the starting current would be calculated as a function of resonator magnetic field (x) for each gyrotron mode (z).

F. Notes

1) All parameters are in MKS units.

2) All data is inputted as real numbers unless otherwise noted.

REFERENCES

1. A.L. Gol'denberg and M.I. Petelin, Radiophysics and Quantum Electonics 16, No. 1 (1973) 106-111.

2. K.E. Kreischer and R.J. Temkin, Int. J. Infrared and Millimeter Waves 1, No. 2 (1980) 195-223. Also, K.E. Kreischer, M.I.T. Plasma Fusion Center Research Report PFC/RR-81-1, to be published in Infrared and Millimeter Waves, edited by K.J. Button.

3. K.E. Kreischer and R.J. Temkin, Int.J. Infrared and Millimeter Waves 2, No. 2 (1981) 175.

II. INPUT AND OUTPUT

A. Variables

Eleven device parameters are required to calculate linear characteristics. Each is referenced by a two letter abbreviation. The following is a table of these parameters with their abbreviations and dimensions:

PARAMETER	ABBREVIATION	DIMENSIONS
Beam voltage	VB	volts
Anode voltage	VA	volts
Cavity magnetic field	BR	testla
Cathode magnetic field	BC	tetla
Mode	MO	none
Cavity radius	RR	meter
Cavity length	LE	meter
Cathode/anode distance	DC	meter
Ohmic Q	QO	none
Diffractive Q	QD	none
Cathode radius	RC	meter

B. File INPUT.DAT

This file, which consists of three lines, provides information needed by the program

1. PAR(1), PAR(2),...., PAR(11)/

   PAR(N) = default value of n^{th} variable listed in A.

2. MODE(1), ZERO(1), MODE(2), ZERO(2),....,MODE(N), ZERO(N)/

   MODE(N) = index of n^{th} mode. For TE_{mpq}, MODE = mp.

   ZERO(N) = transverse index of n^{th} mode. For TE_{mpq}, the p^{th} zero of $J_m'(x) = 0$

3. K, E_s, QTOT, Q(1), X(1), Q(2), X(2),...,Q(QTOT),X(QTOT)/

   K = axial wavenumber constant (see Section I.D.).

   E_x = stored energy factor (see Section I.D.).

   QTOT = number of q branches calculated for each TE_{mp} mode.

   Q(N) = value of q for n^{th} branch.

   X(N) = approximate value of detuning parameter X corresponding to n^{th} branch [3].

C. Other Information required during program execution

1. VAR, VALUE

   An opportunity to change default value is given.

2. IMODE

   IMODE = parameter indicating whether starting current or detuning is desired.

3. PARX, XMIN, XMAX

   PARX = two letter representation of x parameter.

   XMIN = lower limit of x parameter.

   XMAX = upper limit of x parameter.

4. YMIN, YMAX

   YMIN = minimum value of starting current or frequency detuning.

   YMAX = maximum value of starting current or frequency detuning.

5. ZPAR, ZNO, Z(1), Z(2),...,Z(ZNO),/

   ZPAR = two letter representation of z parameter.

   ZNO = number of z values to be inputted.

   Z(N) = value of n^{th} z parameter.

6. ICROS

   ICROS = parameter indicating cross-sectional structure of RF field.

7. IPROF

   IPROF = parameter specifying type of axial RF profile.
   (see Section II.D if you are specifying profile)

8. NHAR

   NHAR = harmonic number.

At this point, desired data will be calculated, followed by a prompt for type of output.

9. IOUT

   IOUT = parameter specifying type of output. IOUT = 3 terminates program.

D. Files and information needed if profile is specified

1. LINPROF. FORTRAN

   A Fortran subroutine must be included with the program that contains the axial profile of the RF field. It should have the following form:

```
SUBROUTINE LINPROF (Q,Z)
Z = G (Q,Z)
RETURN
END
```

   In this routine, you would provide the axial profile as the function G. Note that Q is the axial index of TE_{mpq}.

2. If the profile is specified, the parameters K and E_s of INPUT.DAT are ignored. The constant K is asked for specifically, and E_s is calculated within the program.

E. Tabular output

1. Heading title.
2. Date and time of program execution.
3. List of variables and their values. The x and z parameters are set to zero.
4. Tables of detuning or starting current data for each case requested. Each table is headed by q and z parameter values.

III. SAMPLE RUN

A. The file INPUT.DAT

```
6.5e4,2.e4,5.57,.219,3.,3.47e-3,1.5e-2,4.e-3,2.e4,1.16e3,9.18e-3,30./
11.,1.8412,21.,3.0542,1.,3.8317,31.,4.2012,41.,5.3176,12.,5.3314,
51.,6.4156,22.,6.7061,2.,7.0156,61.,7.5013,32.,8.0152,13.,8.5363,
71.,8.5778,42.,9.2824,81.,9.6474,23.,9.9695,3.,10.1735,52.,10.5199,
91.,10.7114,33.,11.3459,14.,11.7060,62.,11.7349,101.,11.7709,
43.,12.6819,111.,12.8265,72.,12.9324,24.,13.1704,4.,13.3237,121.,
13.8788,53.,13.9872,82.,14.115,34.,14.5858,15.,14.8636,131.,14.9284,
63.,15.2682,92.,15.2867,44.,15.9641,141.,15.9754,25.,16.3475,102.,
16.4479,5.,16.4706,73.,16.5294,151.,17.0203,54.,17.3128,112.,17.6003/
3.14159,0.5,3.,1.,-1.,2.,.4,2.,-1.4/
```

B. Terminal input

```
vb= 0.650E+05      va= 0.200E+05      br= 0.557E+01
bc= 0.219E+00      mo= 0.300E+01      rr= 0.347E-02
le= 0.150E-01      dc= 0.400E-02      ao= 0.200E+05
ad= 0.116E+04      rc= 0.918E-02

Any changes? print 'no,/' or 'parameter,value'

no,/
Choose plot type:(integer)
(1)Starting current vs. two parameters
(2)Detuning vs. two parameters(sine profile)

1
Input x parameter:aa,minimum,maximum

br,5.2,5.8
Input y parameter minimum,maximum

0.,10.
Input z parameter:aa,no. of values,value1,value2,...,/

mo,2.,3.,23.,/
Choose cross-sectional structure:(integer)
(1)Rotating
(2)Standing

2
Choose axial profile:(integer)
(1)Sine
(2)Gaussian
(3)Profile you have specified

1
Input harmonic number(integer)

1




Select output format:(integer)
(1)Table
(2)Graphics
(3)Neither

1
Input table heading in apostrophes

  Starting current for TE(031) and   TE(231) modes'
```

C. Terminal Output

```
 Starting current for TE(031) and TE(231) modes
01/21/82  1131.5 est Thu



vb= 0.650E+05       va= 0.200E+05      br= 0.000E+00
bc= 0.219E+00       mo= 0.000E+00      rr= 0.347E-02
le= 0.150E-01       dc= 0.400E-02      ao= 0.200E+05
ad= 0.116E+04       rc= 0.918E-02

Standing cross sectional structure.



All data is in MKS units.
```

mo= 0.300E+01 a = 0.100E+01		mo= 0.300E+01 a = 0.200E+01		mo= 0.300E+01 a = 0.200E+01	
br	ISTART	br	ISTART	br	ISTART
0.5625E+01	0.1290E+02	0.5800E+01	0.3155E+01	0.5504E+01	0.1465E+03
0.5610E+01	0.2891E+01	0.5795E+01	0.2984E+01	0.5492E+01	0.1730E+02
0.5596E+01	0.1739E+01	0.5790E+01	0.2862E+01	0.5480E+01	0.9847E+01
0.5581E+01	0.1320E+01	0.5785E+01	0.2781E+01	0.5468E+01	0.7300E+01
0.5567E+01	0.1123E+01	0.5780E+01	0.2735E+01	0.5456E+01	0.6113E+01
0.5552E+01	0.1028E+01	0.5774E+01	0.2721E+01	0.5444E+01	0.5515E+01
0.5538E+01	0.9934E+00	0.5769E+01	0.2739E+01	0.5432E+01	0.5249E+01
0.5523E+01	0.1003E+01	0.5764E+01	0.2789E+01	0.5420E+01	0.5209E+01
0.5509E+01	0.1051E+01	0.5759E+01	0.2873E+01	0.5408E+01	0.5357E+01
0.5494E+01	0.1140E+01	0.5754E+01	0.2997E+01	0.5396E+01	0.5688E+01
0.5480E+01	0.1279E+01	0.5749E+01	0.3169E+01	0.5384E+01	0.6224E+01
0.5465E+01	0.1481E+01	0.5744E+01	0.3401E+01	0.5372E+01	0.7015E+01
0.5451E+01	0.1776E+01	0.5739E+01	0.3714E+01	0.5360E+01	0.8150E+01
0.5436E+01	0.2206E+01	0.5733E+01	0.4140E+01	0.5348E+01	0.9778E+01
0.5422E+01	0.2854E+01	0.5728E+01	0.4733E+01	0.5336E+01	0.1216E+02
0.5408E+01	0.3869E+01	0.5723E+01	0.5594E+01	0.5324E+01	0.1577E+02
0.5393E+01	0.5562E+01	0.5718E+01	0.6928E+01	0.5313E+01	0.2155E+02
0.5379E+01	0.8651E+01	0.5713E+01	0.9223E+01	0.5301E+01	0.3161E+02
0.5364E+01	0.1520E+02	0.5708E+01	0.1401E+02	0.5289E+01	0.5175E+02
0.5350E+01	0.3395E+02	0.5703E+01	0.2974E+02	0.5277E+01	0.1050E+03

C. Terminal Output (continued)

mo= 0.230E+02 a = 0.100E+01		mo= 0.230E+02 a = 0.200E+01		mo= 0.230E+02 a = 0.200E+01	
br	ISTART	br	ISTART	br	ISTART
--	--------	--	--------	--	--------
0.5512E+01	0.3672E+02	0.5730E+01	0.1652E+02	0.5387E+01	0.4064E+03
0.5497E+01	0.4108E+01	0.5723E+01	0.9316E+01	0.5377E+01	0.2895E+02
0.5482E+01	0.2340E+01	0.5715E+01	0.6624E+01	0.5367E+01	0.1588E+02
0.5467E+01	0.1744E+01	0.5708E+01	0.5250E+01	0.5358E+01	0.1146E+02
0.5453E+01	0.1472E+01	0.5701E+01	0.4443E+01	0.5348E+01	0.9329E+01
0.5438E+01	0.1343E+01	0.5693E+01	0.3935E+01	0.5338E+01	0.8156E+01
0.5423E+01	0.1296E+01	0.5686E+01	0.3611E+01	0.5328E+01	0.7487E+01
0.5408E+01	0.1308E+01	0.5679E+01	0.3410E+01	0.5318E+01	0.7130E+01
0.5393E+01	0.1372E+01	0.5671E+01	0.3303E+01	0.5308E+01	0.6996E+01
0.5379E+01	0.1490E+01	0.5664E+01	0.3276E+01	0.5298E+01	0.7042E+01
0.5364E+01	0.1674E+01	0.5657E+01	0.3324E+01	0.5289E+01	0.7249E+01
0.5349E+01	0.1943E+01	0.5649E+01	0.3450E+01	0.5279E+01	0.7621E+01
0.5334E+01	0.2334E+01	0.5642E+01	0.3671E+01	0.5269E+01	0.8171E+01
0.5319E+01	0.2906E+01	0.5634E+01	0.4015E+01	0.5259E+01	0.8933E+01
0.5304E+01	0.3767E+01	0.5627E+01	0.4536E+01	0.5249E+01	0.9958E+01
0.5290E+01	0.5118E+01	0.5620E+01	0.5341E+01	0.5239E+01	0.1133E+02
0.5275E+01	0.7368E+01	0.5612E+01	0.6659E+01	0.5230E+01	0.1316E+02
0.5260E+01	0.1147E+02	0.5605E+01	0.9082E+01	0.5220E+01	0.1564E+02
0.5245E+01	0.2015E+02	0.5598E+01	0.1473E+02	0.5210E+01	0.1909E+02
0.5230E+01	0.4474E+02	0.5590E+01	0.4092E+02	0.5200E+01	0.2402E+02

```
Select output format:(integer)
(1)Table
(2)Graphics
(3)Neither

3
r 11:32 12.899 90
```

IV. PROGRAM LISTING

```
c         Program Linear
c         Written by K. E. Kreischer, M.I.T.
c         Latest Revision: 12/1/81
c         Purpose: Calculates linear characteristics of gyrotron.
c
c
c
c  external references for Multics graphics
          external plot_$setup(descriptors)
          external plot_$scale(descriptors)
          external plot_ (descriptors)
c  external reference for date in table
          external date_time
          common par,xdat
          common/prof/zmin,ph,istep
          common/gun/wnoz,wnop,u,w,frer,freb,x,gfac,se,q,xi,xm,xfun(4)
          dimension chi(60,2),par(12),data(2400,2),zparam(20),xdat(50)
          &,xpts(2),xlim(2),ica(3),ija(3),iia(3)
          character parx*2,parz*2,ytitle*50,mtitle*50,xtitle*50
          dimension xdati(50),ydati(50)
          data data,zparam/4820*0./
c  file input.dat opened for reading
          open(7,form='formatted',file='input.dat')
c  first line of data from input.dat is read and displayed
          read(7,)(par(i),i=1,12)
          write(6,5)(par(i),i=1,11)
   5      format(' vb=',e10.3,5x,'va=',e10.3,5x,'br=',e10.3/' bc=',e10.3,
          &5x,'mo=',e10.3,5x,'rr=',e10.3/' le=',e10.3,5x,'dc=',e10.3,5x,
          &'qo=',e10.3/' qd=',e10.3,5x,'rc=',e10.3/)
          write(6,6)
   6      format(' Any changes? print "no,/" or "parameter,value"'/)
c  changes to default values accepted
   7      read(5,40) parx,parval
          if(parx .eq. 'no') goto 15
          call iden(parx,i)
          if(i.eq.99) goto 10
          par(i)=parval
          goto 7
  10      write(6,11)
  11      format(' Error in last entry'/)
          goto 7
c  rest of input.dat is read
  15      read(7,)((chi(i,j),j=1,2),i=1,60)
          read(7,)(xdat(i),i=1,50)
c  type of calculation is chosen
          write(6,25)
  25      format(' Choose plot type:(integer)'/' (1)Starting current vs. two
          &parameters'/' (2)Detuning vs. two parameters(sine profile)'/)
          read(5,) imode
c  info on x,y parameters and their ranges is obtained
          write(6,35)
  35      format(' Input x parameter:aa,minimum,maximum'/)
          read(5,40) parx,xmin,xmax
  40      format(v)
          call iden(parx,ixpar)
          write(6,43)
  43      format(' Input y parameter minimum,maximum'/)
          read(5,) ymin,ymax
c  z parameter and its discrete values are inputted
          write(6,50)
```

```
50      format(' Input z parameter:aa,no. of values,value1,value2
       &,...,/'/)
        read(5,40) parz,zno,(zparam(i),i=1,20)
        call iden(parz,izpar)
c  info on beam and rf field are inputted
        write(6,55)
55      format(' Choose cross-sectional structure:(integer)'/'
       &(1)Rotating'/' (2)Standing'/)
        read(5,) icros
        write(6,60)
60      format(' Choose axial profile:(integer)'/' (1)Sine'/' (2)Gaussian'/
       &' (3)Profile you have specified'/)
        read(5,) iprof
        if(iprof.ne.3) goto 80
c  info on specified profile is obtained
        write(6,61)
61      format(' Input k(parallel) constant'/)
        read(5,) xdat(1)
        write(6,62)
62      format(' Input z limit of axial profile:lower,upper'/)
        read(5,) zmin1,zmax1
        write(6,63)
63      format(' Input number of integration steps(integer)'/)
        read(5,) istep
c  info on harmonic is inputted
80      write(6,81)
81      format(' Input harmonic number(integer)'/)
        read(5,) nhar
c  determine if detuning parameter is a function of x parameter
        ixvrx=0
        if(ixpar .le. 8) ixvrx=1
        l1=2
        if(icros .ne. 1) l1=1
c  do loop varying q of TE(mpq)
        do 500j=1,xdat(3)
        q=xdat(2*j+2)
        if(ixvrx.eq.0) goto 95
c  search for detuning parameter limits of mode
        if(iprof.ne.3) goto 87
        zmin=q*zmin1*xdat(1)/par(7)
        zmax=q*zmax1*xdat(1)/par(7)
        ph=(zmax-zmin)/istep
87      x=xdat(2*j+3)
        call xcal(iprof,imode)
        df=xfun(2)
        call gun(nhar,iprof)
        fac=0.5*nhar*freb*w**2/wnoz/u/9.e16
        do 90 k1=1,2
        x=xdat(2*j+3)
        do 85k2=1,2
        step=(0.2/q)/10**(k2-1)
        do 82k3=1,10
        x=x+(-1.)**(k1+k2)*step
        call xcal(iprof,imode)
        test=-xfun(1)+fac*xfun(2)
        if(k2.eq.1.and.test.lt.df/10) goto 85
        if(k2.eq.2.and.test.gt.df/10) goto 85
82      continue
85      continue
        xlim(k1)=x
```

```
90      continue
c  branch to separate algorithm if profile is specified
        if(iprof.eq.3) goto 401
c  do loop varying z parameter
95      do 500k=1,zno
        par(izpar)=zparam(k)
c  select correct mode index
        do 70j1=1,60
        if(int(par(5)) .ne. int(chi(j1,1))) goto 70
        xi=chi(j1,2)
        xm=int(chi(j1,1)/10)
        goto 75
70      continue
c  branches depending on whether x parameter is a function of
c  detuning parameter
75      if (ixvrx .eq. 1) goto 130
c  direct calculation of data/ixvrx=0
        call gun(nhar,iprof)
        call xcal(iprof,imode)
        do 120l=1,11
        rot=(-1.)**l
        do 120i=1,50
        par(ixpar)=(i-1)*(xmax-xmin)/49.+xmin
        call istart(nhar,imode,iprof,icros,rot,val,1,0,1)
        ic=(l-1)*zno*xdat(3)+(k-1)*xdat(3)+j
c  input of final data into file
        data((ic-1)*51+i,1)=par(ixpar)
120     data((ic-1)*51+i,2)=val
        igap=51
        goto 500
c  direct calculation of data/ixvrx=1
130     del=.01/q**2
c  vary x parameter until detuning parameter is matched
c  determine range for each mode in terms of x parameter
        do 160i=1,2
        ex=xlim(i)
        icnt=0
        call match(xmin,xmax,par(ixpar),del)
155     call gun(nhar,iprof)
        test=abs(x-ex)
        call matchi(test,par(ixpar),icnt)
        goto (155,157,158,159) icnt
157     xpts(i)=par(ixpar)
        goto 160
158     xpts(i)=xmax
        goto 160
159     xpts(i)=xmin
160     continue
        do 170l=1,11
        rot=(-1.)**l
        do 170i=1,20
        par(ixpar)=(i-1)*(xpts(2)-xpts(1))/19.+xpts(1)
        call istart(nhar,imode,iprof,icros,rot,val,1,1,1)
        ic=(l-1)*zno*xdat(3)+(k-1)*xdat(3)+j
c. input of final data into file
        data((ic-1)*21+i,1)=par(ixpar)
170     data((ic-1)*21+i,2)=val
        igap=21
        goto 500
c  calculate data for case of iprof=3 and ixvrv=1.
```

```
c  minimize number of integrations in calculation of x functions
c  in subroutine xcal.
c  energy constant is calculated
  401      icnt=0
           val=0
  420      call integ(zmin,ph,istep+1,val,icnt)
           if(icnt.eq.1) goto 425
           call linprof(val,q)
           val=val**2
           goto 420
  425      xdat(2)=val/xdat(1)/q
           do 409i=1,20
           ex=xlim(1)+(i-1)*(xlim(2)-xlim(1))/19.
           x=ex
           call xcal(iprof,imode)
           do 409kl=1,zno
           par(izpar)=zparam(kl)
           do 406j1=1,60
           if(int(par(5)).ne.int(chi(j1,1))) goto 406
           xi=chi(j1,2)
           xm=int(chi(j1,1)/10.)
           goto 407
  406      continue
c  find x parameter corresponding to each detuning parameter
  407      icnt=0
           del=.01/q**2
           call match(xmin,xmax,par(ixpar),del)
  404      call gun(nhar,iprof)
           test=abs(x-ex)
           call matchi(test,par(ixpar),icnt)
           goto(404,403,405,405) icnt
  405      val=-999.
  403      do 409l=1,11
           rot=(-1.)**l
           if(icnt.ge.3) goto 408
           call istart(nhar,imode,iprof,icros,rot,val,0,0,1)
  408      mm=i+21*((l-1)*zno*xdat(3)+(k-1)*xdat(3)+j-1)
c  input of final data into file
           data(mm,1)=par(ixpar)
  409      data(mm,2)=val
           igap=21
  500      continue
c  selection of output format
  501      write(6,502)
  502      format(' Select output format:(integer)'/' (1)Table'/'
           & (2)Graphics'/' (3)Neither'/)
           read(5,) iout
           goto(504,600,900) iout
c  output of data in tabular form
  504      write(6,506)
  506      format(' Input table heading in apostrophes'/)
           read(5,40) mtitle
           write(6,508) mtitle
  508      format(10(/),1x,a50)
           call date_time
           do 510j = 1,12
  510      if(j.eq.ixpar.or.j.eq.izpar) par(j) = 0
           write(6,511)
  511      format(///)
           write(6,5)(par(j),j=1,11)
```

```
      if(icros.eq.1) goto 513
      write(6,512)
512   format(' Standing cross sectional structure.'/)
      goto 517
513   write(6,516)
516   format(' Rotating cross sectional structure.'/)
517   write(6,511)
514   write(6,515)
515   format(' All data is in MKS units.'///)
      n=0
      do 550k=1,xdat(3)*zno*11
      n=n+1
      iia(n)=aint(1.01+(k-1)/(xdat(3)*11))
      ija(n)=aint(1.01+(k-1)/11)-xdat(3)*(iia(n)-1)
      ica(n)=k-1
      if(n.eq.3.or.k.eq.(xdat(3)*zno*11)) goto 520
      goto 550
520   write(6,522)parz,zparam(iia(1)),parz,zparam
      &(iia(2)),parz,zparam(iia(3))
522   format(8x,2(a2,"=",e10.3,13x),a2,'=',e10.3)
      write(6,524)(xdat(2*ija(kk)+2),kk=1,3)
524   format(8x,2('q =',e10.3,13x),'q =',e10.3/)
      ytitle=" ISTART "
      if(imode.eq.2)ytitle="DETUNING"
      write(6,526)parx,ytitle,parx,ytitle,parx,ytitle
526   format(7x,2(a2,8x,a8,8x),a2,8x,a8)
      write(6,528)
528   format(7x,2('__',8x,8('_'),8x),'__',8x,8('_')/)
      do540j=1,igap-1
      write(6,530)((data(ica(m)*igap+j,kk),kk=1,2),m=1,n)
530   format(6(2x,e11.4))
540   continue
      write(6,511)
      n=0
550   continue
      goto 501
c  output of data in graphic form
600   write(6,605)
605   format(' Input heading title in apostrophes'/)
      read(5,40) mtitle
      write(6,610)
610   format(' Input x-axis title in apostrophes'/)
      read(5,40) xtitle
      write(6,615)
615   format(' Input y-axis title in apostrophes')
      read(5,40) ytitle
      write(6,620)
620   format(' Input plot type:(integer)'/' (1)Linear-linear'/'
      & (2)Log x-linear y'/' (3)Linear x-log y'/' (4)Log-log'/)
      read(5,40) itype
      base=10.
      if(itype.eq.1) base=0.
      call plot_$setup(mtitle,xtitle,ytitle,itype,base,0,0)
      call plot_$scale(xmin,xmax,ymin,ymax)
      do 640i=0,xdat(3)*zno*11-1
      ixydim=0
      do 630j=1,igap-1
      if (data(i*igap+j,2).lt.ymin.or.data(i*igap+j,2).gt.ymax)
      & goto 630
      ixydim=ixydim+1
```

```
        xdat1(ixydim)=data(i*igap+j,1)
        ydat1(ixydim)=data(i*igap+j,2)
 630    continue
        if(ixydim.eq.0) goto 640
        call plot_ (xdat1,ydat1,ixydim,1,'.')
 640    continue
        goto 501
 900    continue
        end
        subroutine iden(a,ib)
c  determines numerical code for device variables
        character*2 a
        ib=99
        if(a .eq. 'vb') ib=1
        if(a .eq. 'va') ib=2
        if(a .eq. 'br') ib=3
        if(a .eq. 'bc') ib=4
        if(a .eq. 'mo') ib=5
        if(a .eq. 'rr') ib=6
        if(a .eq. 'le') ib=7
        if(a .eq. 'dc') ib=8
        if(a .eq. 'qo') ib=9
        if(a .eq. 'qd') ib=10
        if(a .eq. 'rc') ib=11
        if(a .eq. 'cb') ib=12
        return
        end
        subroutine xcal(iprof,imode)
c  calculates gain functions and their derivatives
        common par(12),xdat(50)
        common/prof/zmin,ph,istep
        common/gun/wnoz,wnop,u,w,frer,freb,x,gfac,se,q,x1,xm,xfun(4)
        do 2j=1,4
 2      xfun(j)=0.
        pi=3.141592654
        do 50k=1,2
        x1=x-.001*(-1)**k
        if (iprof .ne. 1) goto 5
        fc=2*sin((x1+1)*pi*q/2)**2/(1-x1**2)**2
 5      if(iprof .ne. 1 .or. imode .ne. 2) goto 10
        fs=-1/(1-x1**2)**2*(sin(q*pi*(x1-1))+x1*(1-x1**2)*q*pi/2)
 10     if(iprof .ne. 2) goto 15
        fc=pi/2*exp(-.5*x1**2)
 15     if(iprof .ne. 3) goto 20
c  gain function calculated by integration
        fc=0
        do 19j=1,2
        icnt=0
        val=0
 17     call integ(zmin,ph,istep+1,val,icnt)
        if(icnt .eq. 1) goto 19
        val1=val
        call linprof(val,q)
        if(j.eq.1) val=val*cos(x1*val1)
        if(j.eq.2) val=val*sin(x1*val1)
        goto 17
 19     fc=0.5*val*val+fc
 20     xfun(1)=xfun(1)+.5*fc
        xfun(2)=xfun(2)-500*(-1)**k*fc
        xfun(3)=xfun(3)+.5*fs
```

```
50      xfun(4)=xfun(4)-500*(-1)**k*fs
        return
        end
        subroutine integ(smin,h,istep,val,icnt)
c  integration done by Simpson approximation
        if(icnt .eq. 0) accum=0
        if(icnt .eq. 0) index=0
        if(index .eq.1 .or. index .eq. istep) val=val/2
        accum=accum+val*(3+(-1)**index)*h/3
        icnt=2
        index=index+1
        val=smin+h*(index-1)
        if(index .eq. (1+istep)) val=accum
        if(index .eq. (1+istep)) icnt=1
        return
        end
        subroutine bessel(xm,arg,func)
c  Bessel function J(arg) of order xm is calculated
c  result returned as func
        icnt=0
        val=0
        istep=10*max1(abs(xm),arg)+1
        istep=max0(istep,30)
        h=3.141592654/(istep-1)
5       call integ(0.,h,istep,val,icnt)
        if(icnt .eq. 1) goto 10
        val=cos(arg*sin(val)-xm*val)
        goto 5
10      func=val/3.14159
        return
        end
        subroutine gun(nhar,iprof)
c  remaining device variables and detuning parameter are calculated
c  assuming adiabatic theory
        common par(12),xdat(50)
        common/gun/wnoz,wnop,u,w,frer,freb,x,gfac,se,q,xi,xm,xfun(4)
        ec=par(2)/par(8)
        gam=1+par(1)/5.11e5
        al=par(3)/par(4)
        beta=sqrt(1-1/gam**2)
        w=sqrt(al)*ec/par(4)
        wnop=xi/par(6)
        wnoz=xdat(1)/par(7)*q
        frer=3.e8*sqrt(wnoz**2+wnop**2)
        freb=1.75758e11*par(3)/gam
        arg=9.e16*beta**2-w**2
        if(arg.lt.0.) goto 10
        u=sqrt(arg)
        x=(nhar*freb-frer)/wnoz/u
        return
10      x=0.
        u=1.e10
        return
        end
        subroutine crosec(nhar,icros,rot,iprof)
c  beam-field coupling and stored energy factors are calculated
        common par(12),xdat(50)
        common/gun/wnoz,wnop,u,w,frer,freb,x,gfac,se,q,xi,xm,xfun(4)
        call bessel(xm,xi,func)
        se=1.57*par(7)/wnop**2*(xi**2-xm**2)*
```

```
      &func**2*xdat(2)
      rbeam=sqrt(par(4)/par(3))*par(11)
      if(rbeam.ge.par(6)) gfac=-1.e-10
      if(rbeam.ge.par(6)) goto 50
      arg=wnop*rbeam
      if(icros .ne. 1) goto 15
      call bessel(xm+rot*nhar,arg,func)
      gfac=.5*func**2
      goto 50
15    har=nhar
      call bessel(xm-har,arg,func)
      call bessel(xm+har,arg,func1)
      gfac=.25*(func1**2+func**2)
50    return
      end
      subroutine match(amin1,amax1,var,del1)
c  'var'is varied Until 'val' is within limit set by 'del1'
c  also determines if match is outside parameter limits 'amin1,amax1'
      amin=amin1
      amax=amax1
      del=del1
      del2=(amax-amin)/400.
      fmax=amax
      fmin=amin
      var=amin+0.618034*(amax-amin)
      return
      entry match1(val,var,icnt)
      if(icnt .ne. 0) goto 10
      val1=val
      var1=var
      icnt=1
      goto 20
10    if(val .gt. val1) goto 15
      if(var1 .le. var) amin=var1
      if(var1 .gt. var) amax=var1
      var1=var
      val1=val
      goto 20
15    if(var .le. var1) amin=var
      if(var .gt. var1) amax=var
20    if(abs(fmax-var1).lt.del2) icnt=3
      if(abs(fmin-var1).lt.del2) icnt=4
      if(val1.lt.del) icnt=2
      if(icnt .eq. 2) var=var1
      if(icnt .eq. 2) val=val1
      if(icnt .gt. 1) goto 30
      varf=amin+0.381966*(amax-amin)
      var=amin+0.618034*(amax-amin)
      if(abs(var-var1).lt.abs(varf-var1)) var=varf
30    return
      end
      subroutine istart(nhar,imode,iprof,icros,rot,val,i1,i2,i3)
c  starting current or frequency detuning is calculated
      common par(12),xdat(50)
      common/gun/wnoz,wnop,u,w,frer,freb,x,gfac,se,q,xi,xm,xfun(4)
      if(i1.eq.1) call gun(nhar,iprof)
      if(i2.eq.1) call xcal(iprof,imode)
      if(i3.eq.1) call crosec(nhar,icros,rot,iprof)
      fac=.5*nhar*freb*w**2/wnoz/u/9.e16
      if(imode.eq.2) goto 5
```

```
      arg=wnop*w/freb
      har=nhar
      call bessel(har-1.,arg,func1)
      call bessel(har+1.,arg,func2)
      harfac=func1-func2
      qt=par(10)/q**2*(xl/10.1735)**2
c  starting current is calculated
      val=-1.e-22*frer/qt*se*wnoz**2*u**2/gfac/harfac**2/
      &(xfun(1)-fac*xfun(2))*(1+par(1)/5.11e5)
      goto 10
c  frequency detuning is calculated
 5    val=(xfun(3)-fac*xfun(4))/(xfun(1)-fac*xfun(2))
10    return
      end
```

PFC BASE LIST

INTERNAL MAILINGS (MIT)

G. Bekefi
36-213

A. Bers
38-260

D. Cohn
NW16-250

B. Coppi
26-201

R.C. Davidson
NW16-202

T. Dupree
38-172

S. Foner
NW14-3117

J. Freidberg
38-160

A. Gondhalekar
NW16-278

M.O. Hoenig
NW16-176

M. Kazimi
NW12-209

L. Lidsky
38-174

E. Marmar
NW16-280

J. McCune
31-265

J. Meyer
24-208

D.B. Montgomery
NW16-140

J. Moses
NE43-514

D. Pappas
NW16-272

R.R. Parker
NW16-288

N.T. Pierce
NW16-186

P. Politzer
NW16-286

M. Porkolab
36-293

R. Post
NW21-

H. Praddaude
NW14-3101

D. Rose
24-210

J.C. Rose
NW16-189

R.M. Rose
4-132

B.B. Schwartz
NW14-5121

R.F. Post
NW21-203

L.D. Smullin
38-294

R. Temkin
NW16-254

N. Todreas
NW13-202

J.E.C. Williams
NW14-3210

P. Wolff
36-419

T.-F. Yang
NW16-164

Industrial Liaison Office
ATTN: Susan Shansky
Monthly List of Publications
39-513

MIT Libraries
Collection Development
ATTN: MIT Reports
14E-210

B. Colby
PFC Library
NW16-255

EXTERNAL MAILINGS

International

Professor M.H. Brennan
Willis Plasma Physics Dept.
School of Physics
University of Sydney
N.S.W. 2006, Australia

Division of Plasma Physics
Institute of Theoretical Physics
University of Innsbruck
A-6020 Innsbruck
Austria

c/o Physics Section
International Atomic Energy Agency
Wagramerstrasse 5
P.O. Box 100
A-1400 Vienna, Austria

Laboratoire de Physique des Plasmas
c/o H.W.H. Van Andel
Dept. de Physique
Universite de Montreal
C.P. 6128
Montreal, Que H3C 3J7
Canada

Plasma Physics Laboratory
Dept. of Physics
University of Saskatchewan
Saskatoon, Sask., Canada S7N OWO

The Library
Institute of Physics
Chinese Academy of Sciences
Beijing, China

Mrs. A. Wolff-Degives
Kernforschungsanlage Julich GmbH
Zentralbibliothek - Exchange Section
D-5170 Julich - Postfach 1913
Federal Republic of Germany

Preprint Library
Central Research Institute for Physics
H-1525 Budapest, P.O. Box 49
Hungary

Plasma Physics Dept.
Israel Atomic Energy Commission
Soreq Nuclear Research Center
Yavne 70600
Israel

The Librarian (Miss DePalo)
Associazione EURATOM - CNEN Fusione
C.P. 65-00044 Frascati (Rome)
Italy

Librarian
Research Information Center
Institute of Plasma Physics
Nagoya University
Nagoya, 464
Japan

Dr. A.J. Hazen
South African Atomic Energy Board
Private Bag X256
Pretoria 0001
South Africa

EXTERNAL MAILINGS

National

Argonne National Laboratory
Argonne, IL 60439
ATTN: Library Services Dept.

Battelle-Pacific Northwest Laboratory
P.O. Box 99
Richland, WA 99352
ATTN: Technical Information Center

Brookhaven National Laboratory
Upton, NY 11973
ATTN: Research Library

U.S. Dept. of Energy
Washington, D.C. 20545
ATTN: D.O.E. Library

Roger Derby
Oak Ridge National Lab.
ETF Design Center
Bldg. 9204-1
Oak Ridge, TN 37830

General Atomic Co.
P.O. Box 81608
San Diego, CA 92138
ATTN: Library

Lawrence Berkeley Laboratory
1 Cyclotron Rd.
Berkeley, CA 94720
ATTN: Library

Lawrence Livermore Laboratory
UCLA
P.O. Box 808
Livermore, CA 94550

Oak Ridge National Laboratory
Fusion Energy Div. Library
Bldg. 9201-2, ms/5
P.O. Box "Y"
Oak Ridge, TN 37830

Dr. D. Overskei
General Atomic Co.
P.O. Box 81608
San Diego, CA 92138

Princeton Plasma Physics Laboratory
Princeton University
P.O. Box 451
Princeton, NJ 08540
ATTN: Library

Plasma Dynamics Laboratory
Jonsson Engineering Center
Rensselaer Polytechnic Institute
Troy, NY 12181
ATTN: Ms. R. Reep

University of Wisconsin
Nuclear Engineering Dept.
1500 Johnson Drive
Madison, WI 53706
ATTN: UV Fusion Library