MSUCP-63
August, 1992

The BEAM3D Code For the 3D Extraction of Multiply-Charged Ions from Ion Sources

Timothy A. Antaya
National Superconducting Cyclotron Laboratory
Michigan State University
East Lansing, MI 48824

Z.Q. Xie
Lawrence Berkeley Laboratory
Berkeley, CA 49720

MSUCP–

The BEAM3D Code–For the Extraction of Multiply-Charged Ions from Ion Sources

Timothy A. Antaya

National Superconducting Cyclotron Laboratory,
Michigan State University, East Lansing, MI 48842

Z.Q. Xie

Lawrence Berkeley Laboratory, Berkeley, CA 49720

1. Motivation

ECRIS for multiply-charged ions can produce multiple ion species simultaneously. The beam formation process involves the electric fields produced by the extraction electrodes, magnetic fields from source solenoid coils, iron yoke and multipole magnets, the charge state distribution (CSD), space charge force, the plasma boundary, the ion thermal energy and beam neutralization. One would like to know what parameters are the most important for beam quality and accelerator transport matching. To answer this question, one generally makes simplifying assumptions. The typical assumptions made are to study only a single ion species and azimuthal field symmetry, in order to reduce the calculations or apply an existing code, such as the SLAC Electron Trajectory Code [1]. One exception would be KOBRA3, developed at GSI, in which some calculations were done in 3D for ECRIS extraction [2], but only with extraction of the single species 0^{3+}. However, the existing codes do not include the total magnetic field in an ECRIS extraction region in a 3D calculation of the ion trajectories, while also taking the other parameters mentioned above into consideration at the same time. In order to support theoretically the ECRIS beam extraction experimental measurements of this distribution, a new code, BEAM3D has been developed.

2. General Organization

Shown in Figure 1 is a flow chart of the present BEAM3D code. First, primary data input, such as the field data files (which will be explained in a later section), CSD, ion thermal energy, extraction voltages and the focussing solenoid specifications. Second, ion ray tracing in the combined fields is performed with an axial step size of a few tenths field s is described by the Lorentz equation

$$\frac{d\vec{P}}{dt} = Q(\vec{E} + \vec{v} \times \vec{B}) \tag{1}$$

where Q is the charge that the particle in question carries. We change the independent variable from t to z through the relation

$$\frac{d}{dt} = v\frac{d}{zdz} \tag{2}$$

to obtain the following equation

$$\frac{d\vec{P}}{dz} = \frac{Q}{v_z}(\vec{E} + \vec{v} \times \vec{B}) \tag{3}$$

The general analytic solution of Eq. 3 is not possible, because of the coupling terms in the component equations, but a numerical solution is possible but is sensitive to the calculation step size. The Runge-Kutta integration technique [3] is used in BEAM3D to solve the integral equations for $\vec{v}$ and $\vec{r}$.

4. Interpolation of the External Fields

The electric fields of the extraction electrodes and magnetic fields due to the solenoid coils and iron yoke, which have azimuthal symmetry, are calculated by the POISSON code [4] with a small mesh spacing in the r and z directions. The fields at a point P(r,z) between grid points, as shown in Figure 2, are linearly interpolated using the following two dimensional formulae [5]

$$Fz(r,z) = \frac{(r-r_1)}{(r_2-r_1)(z_3-z_1)}[(Fz_4 + Fz_1 - Fz_3 - Fz_2)(z-z_1) + (Fz_2 - Fz_1)(z_3-z_1)] + \frac{(Fz_3 - Fz_1)}{(z_3-z_1)}(z-z_1) + Fz_1 \tag{4}$$

$$Fr(r,z) = \frac{(z-z_1)}{(r_2-r_1)(z_3-z_1)}[(Fr_4 + Fr_1 - Fr_3 - Fr_2)(r-r_1) + (Fr_3 - Fr_1)(r_2-r_1)] + \frac{(Fr_2 - Fr_1)}{(r_2-r_1)}(r-r_1) + Fr_1 \tag{5}$$

where $\vec{F}$ stands for $\vec{E}$ or $\vec{B}$ field.

By using POISSON for the magnetic field generation, BEAM3D automatically takes into consideration iron included in the problem. The hexapole field is incorporated into the total magnetic field by calling a subroutine "HEX" [6]. Our BEAM3D calculations have shown however, that the effect of the hexapole field on the beam formation is on the order of 1%, because the hexapole strength up to the typical extraction aperture radius is much less than the solenoid coil strength.

5. Estimation of Space Charge Force

The space charge force for multiple ion species is based on the assumptions (1) and (2), in the following way, similar to the model established for the beam of a single

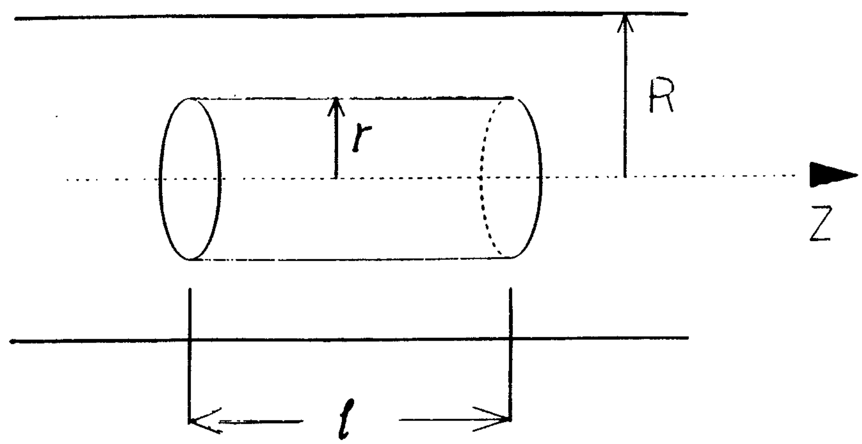

Figure 3. The geometry used in the calculation of the space charge force.

6. Geometry of Calculation

   BEAM3D can track the beam from the source extraction region through a first focussing solenoid and/or beam collimating slit down to some appropriate beamline location as shown in Figure 4. The z-axis is taken as the beamline optical axis. In this example, after the source extraction electrodes, is an X-Y steering magnet (which has not been taken into consideration), followed by the focussing solenoid. A beam defining slit is located at the image of the focussing solenoid. Computing the beam trajectories through the focussing solenoid to this slit position can be very important for a proper matching of the ion source extraction to the beam transport system, since by default, both the ion source extraction and first beamline element are included in the calculation.

7. Output Form

   Figures 5-8 show the graphic outputs of a helium beam calculation. The orbits of the ions are plotted in the r-z plane, with the extraction electrodes, shown in Fig. 5, the focussing solenoid and the beam defining slit indicated at the proper locations as well as the ion orbits in the transverse X-Y plane. In this calculation of both He^{1+} and He^{2+} ions, He^{2+} is over-focussed because the solenoid is set for He^{1+} transport and hits the beam pipe and the beam defining slit. The initial and final ion position distributions are shown in Fig. 6, while Fig. 7 shows the initial and final CSD and the emittance fittings are shown in Fig. 8.

3. Special Features

The special features of the BEAM3D code are summarized as follows:

1. The CSD can be user specified or based on actual ion source measurements. The focussing solenoid current is set for the focussing requirements of a selected ion

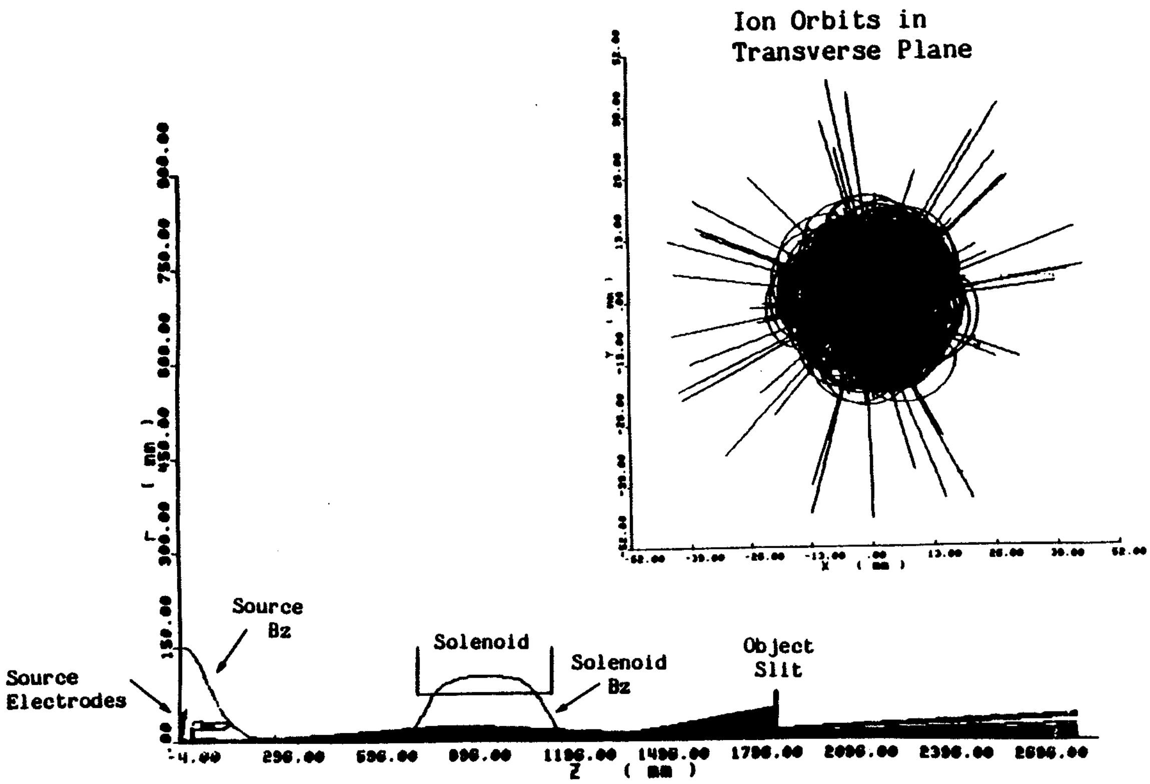

Figure 5. The axial and transverse ion orbits of He^{1+} and He^{2+} calculated by the BEAM3D code.

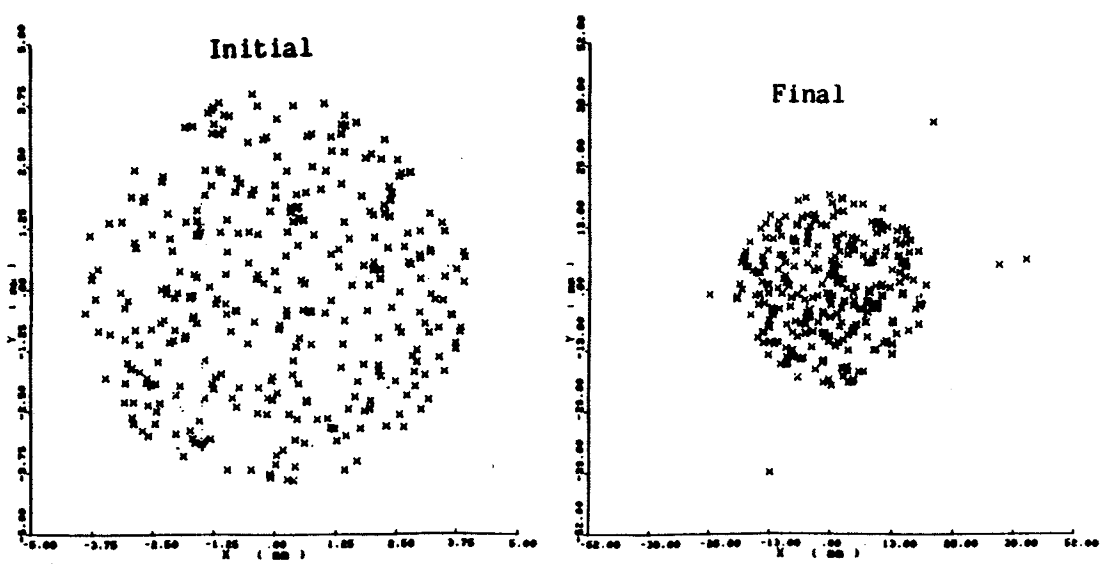

Figure 6. The initial and final ion position distributions.

References

[1] Herrmansfelt W 1979 *Electron Trajectory Program*

[2] Spadtke p 1985 *KOBRA3–Three Dimensional Raytracing Including Space-Charge Effects*(IEEE Trans. Nucl. Sci. NS32) p 2465

[3] Romanelli M 1966 *Mathematical Methods for Digital Computers* (edited by A. Ralston and H.S. Wilf), John Wiley & Sons, Inc. p 110

[4] Holsinger R F 1979 *New England Nuclear*

[5] Xie Z Q and Antaya T A 1987 *NSCL Report MSUCP–47* (Proc. Int. Conf. on ECR Ion Sources) p 420

[6] Xie Z Q and Antaya T A 1986 *NSCL Annual Report* p 166

[7] Brewer G R 1967 *Focusing of Charged Particles* (edited by A Septier) Academic Press Inc. p 74

Appendix A

Input Instructions

BEAM3D INPUT DATA

```
CN1   BSCOIL                        : Source magnetic field data filename.

CN2   EEND                          : Extraction electrode data filename.

CN3   EPULL                         : Puller electrode data filename.

CN4   BFCOIL                        : Focussing solenoid data filename.

CN5   9999                          : IR--seed for random generation of
                                    : initial particle positions,
                                    : IR=1, manually input all positions

CN6   400                           : NP- number of rays, (NPmax=400)

CN7   30*1,20*2,20*3,20*4,20*5,10*6,10*7,250*1,20*2
                                    : N(i)*Q(i) (No. rays of same ion
                                    : species with charge state Q(i))

CN8   30*3,20*4,20*5,20*6,20*7,10*2,10*1,250*1,20*1
                                    : N(i)*I(i) (N(i) No. rays, I(i)
                                    : Current (uA) carried by each ray),
                                    : Total Current = sum (N(i)*I(i))

CN9  130*14,270*4
                                    : N(j)*A(j) (No. rays of the same ion
                                    : mass number A(j))

CN10  130*2,270*3
                                    : N(j)*Tt(j) (No. rays with initial
                                    : thermal transverse energy Tt eV/per
                                    : charge state)

CN11  130*1.5,270*2
                                    : N(j)*Tl(j) (No. rays with initial
                                    : thermal longitudinal energy
                                    : (Tl eV/per charge state)

CN12  .1,50,120                     : DZ, NZ, M.  DZ (mm) is the stepsize of
                                    : calculation, NZ is the number of steps
                                    : to take before a data point will be
                                    : written into the output data file
                                    : OPTF.  The Z interval between writes
                                    : is then NZ*DZ.
                                    : (M+1) is the total data points written
                                    : to OPTF, and M*NZ*DZ is the total ray
                                    : tracing length.

CN13  0.004,.00508,.0327,.076,.156,.3,.7
                                    : EXR,PUR,EXSP,RPIP,EXEFL,EXBFL,FSFL (in M)
                                    : EXR - extraction aperture radius;
                                    : PUR - puller aperture radius;
                                    : EXSP - the spacing between the extraction
                                    :     aperture and the puller aperture;
                                    : RPIP - the radius of the transport beam pipe;
```

CN5 .392,.592 : Locations (M) that ellipses will be generated and plotted.

CN6 131,380 : Rays used to generate the ellipses.

```
**              Format of Source Magnetic Field Date File
*                (Br(r) and Bz(r) in Tesla, Z in M)
Z(1),Br(1),Br(2),Br(3),Br(4),Br(5),Br(6),Br(7),Br(8),Br(9),Br(10),Br(11)
Z(1),Br(12),Br(13),Br(14),Br(15),Br(16),Br(17),Br(18),Br(19)
Z(1),Bz(1),Bz(2),Bz(3),Bz(4),Bz(5),Bz(6),Bz(7),Bz(8),Bz(9),Bz(10),Bz(11)
Z(1),Bz(12),Bz(13),Bz(14),Bz(15),Bz(16),Bz(17),Bz(18),Bz(19)
......
Z(4),Br(1),Br(2),Br(3),Br(4),Br(5),Br(6),Br(7),Br(8),Br(9),Br(10),Br(11)
Z(4),Br(12),Br(13),Br(14),Br(15),Br(16),Br(17),Br(18),Br(19)
Z(4),Bz(1),Bz(2),Bz(3),Bz(4),Bz(5),Bz(6),Bz(7),Bz(8),Bz(9),Bz(10),Bz(11)
Z(4),Bz(12),Bz(13),Bz(14),Bz(15),Bz(16),Bz(17),Bz(18),Bz(19)
......
Z(n),Br(1),Br(2),Br(3),Br(4),Br(5),Br(6),Br(7),Br(8),Br(9),Br(10),Br(11)
Z(n),Br(12),Br(13),Br(14),Br(15),Br(16),Br(17),Br(18),Br(19)
Z(n),Bz(1),Bz(2),Bz(3),Bz(4),Bz(5),Bz(6),Bz(7),Bz(8),Bz(9),Bz(10),Bz(11)
Z(n),Bz(12),Bz(13),Bz(14),Bz(15),Bz(16),Bz(17),Bz(18),Bz(19)
* Note:  Z(1)=-.003 M
         Z(4)=0.0 M
         Z(n)=EXBFL
         DZ=1 mm, DR=.1 mm for i from 1 to 11, DR=.5 mm for i from 12 to 19.
    *** In the POISSON calculation, the magnetic field direction should point the
        first stage (opposit to the down stream beam line direction) direction,
        that means the Bz is positive and Br is negative. In BEAM_3D, the
        magnetic field is assumed point from source to the focussing solenoid,
        and the BEAM_3D rouine reverses the magnetic field direction
        accordingly. See the following sketches.
```

```
**              Format of End Electrode Electric Field Data File
*                (Er(r) and Ez(r) in V/M, Z matches the Z in Source Magnetic
*                  Field Data File)
Z(1),Er(1),Er(2),Er(3),Er(4),Er(5),Er(6),Er(7),Er(8),Er(9)
Z(1),Er(10),Er(11),Er(12),Er(13),Er(14),Er(15),Er(16),Er(17),Er(18)
Z(1),Ez(1),Ez(2),Ez(3),Ez(4),Ez(5),Ez(6),Ez(7),Ez(8),Ez(9)
Z(1),Ez(10),Ez(11),Ez(12),Ez(13),Ez(14),Ez(15),Ez(16),Ez(17),Ez(18)
......
Z(4),Er(1),Er(2),Er(3),Er(4),Er(5),Er(6),Er(7),Er(8),Er(9)
Z(4),Er(10),Er(11),Er(12),Er(13),Er(14),Er(15),Er(16),Er(17),Er(18)
```

```
**              Geometry Data File Definition and Format:
*              (All lengths in mm)
* Format

 N,Z(1)....Z(N),R(1)....R(N)

        * N is the number of points that a part (end plate or puller or
          focussing solenoid or object slit) will be drawn in the output plot.
          Z(i)s are in the BEAM_3D calculation coordinates except for the
          focussing solenoid, that is Z=0 is the location of the edge of the
          extraction aperture in the puller side.
       * Note Zs:  Since the focussing solenoid position can be varied in
          BEAM_3D calculation, special attention should be paid to this
          solenoid coordinates Zs. In the plot routine, the focussing solenoid
          will be plot at Z=SOL_LOC+Zs to match its real position.


* Data Sample

  6, 9.525,9.525,0.,-3.175,4.445,4.445,50.,21.59,4.,4.,21.59,50   (End plate)
  9, 32.7,37.785,51.76,89.86,89.86,69.54,69.54,59.38,32.7         (Puller)
  5.08,5.08,19.05,19.05,22.85,22.85,31.496,31.496,5.08
  8, 101.29,133.29,155.26,155.26,75.89,75.89,101.29,101.29      (Ground Ring)
  20.32,20.32,24.765,29.845,29.845,26.035,26.035,20.32
  4,  0.,0.,428.88,428.88,150.,75.,75.,150.        (Focussing Solenoid, note Zs)
  4,  1900.,1900.,1905.,1905.,75.,0.,0.,75.                  (Object Slit)



***  Plot Routine
     The Plot routine
```

Appendix B

BEAM3D.FOR

```
c                    PARTICLE RAY TRACING CODE BEAM3D
c
c          Written by Z.Q. Xie
c          6 June 1987
c
c
c   This code does the 3D particle ray tracing of beam of multiple ion
c species extracted from the RT-ECR by using the iterative numerical solution
c (Runge-Kutta method). The ray number limit of this version is 800, each ray
c carries a portion of the total beam current partitioned by the user at their
c desires.
c
c   In this version, all the particles initially are assumed to follow the
c magnetic field line and emitt from a plane surface with no radial velocity.
c
c   The magnetic filed (including the source solenoid and the first focussing
c solenoid coil magnetic fields) and the extraction electric field (including
c the extraciotn and the puller) components are linearly interpolated based on
c the input data files (which are generated with the help of POISSON Code, 4
c data files should be generated: 1. source solenoid magnetic field (fixed
c field, usually a typical field); 2. end electrode electric field data file
c in POISSON calculation, 10000 V is applied to the end electrode and 0 V is
c applied to the puller); 3. puller electrode electric field data file (in
c POISSON calculation -100 V is applied to the puller and 0 V is applied to the
c end electrode); 4. focussing solenoid magnetic data file (in POISSON
c calculation, 100 A should be applied to the calulation). Thus inside ERAY
c routine, factors of 1/10000, -1/100 and 1/100 are scaled to the input data
c accordingly to generate the real fields. The magnetic contribution from the
c hexapole magnet can be analytically calculated from the "PERMMAG" Code which
c is ingored in this first stage of development for the time being in order to
c save CPU time. PERMAG will be incorporated in ERAY as a subrouting in the
c future version.
c
c   The source solenoid field input data file is formated in specific step size

c of DZ=1 mm in the Z and DR=1 mm in the r direction when r<1 cm, and DR=5 mm
c when r>1 cm. The electric field data file have the same Z step size but DR=
c 1.27 mm in the region (Z=-.3 cm to EXEFL and R to 2.159 cm). The focussing
c solenoid field data file is formated DZ = DR = 5mm.
c The rest input data are: initial position in X and Y axises with Z=-3 mm is
c assumed to be the starting point. Z=0 is at one of the extraction aperture
c edges in the puller electrode side;
c   (In this version, the fields are tracted back to Z=-3 mm, the thikness
c   of the extraction aperture is assumed about 3 mm thick, in -3 mm to z=0.
c   region, considering the total current is contributed from the particles
c   under consideration which get through the extraction aperture, then in the
c   above mentioned region, the radial electric field is taken as zero.)
c   The extraction electric potentials (V); The input initial velocities in all
c 3 dimensions in the unit of eV; Charge state Q and mass number A.
c
c   The electric field inside the conductor should be zero but the POISSON
c calculation does not give the correct values due to the deficiency of the
c POISSON Code. When the particle get very close to the electrode within a
c grid unit, instead of putting pseudo values into those ponits, the values
c from POISSON just inside the conductor are used in the 2-D linearly
c interpoltation of the field strength outside the conductor but within a
c grid unit, and those particle ray will be notified in the output LOG data
c file.
c
c   The space charge are assumed to uniform distributed inside any circle of
```

```
      READ(5,*) IR
      TYPE *, '  HOW MANY RAYS, TYPE NP (NPmax=800)'
      READ (5,*) NP
      TYPE *, ' CHARGE STATES IN THE ORDER OF 1 TO NP'
      READ(5,*)(Q(I),I=1,NP)
      TYPE *, ' CURRENT ( UA ) PER RAY '
      READ(5,*)(AP(I),I=1,NP)
      TYPE *,' PARTICLE MASS IN THE ORDER OF 1 TO NP'
      READ(5,*)(A(I),I=1,NP)
      IF(IR.EQ.1) GOTO 65
      type *, ' INITIAL ENERGIES IN TRANSVERSE DIRECTION (eV) ?'
      READ(5,*) (E_trans(I),I=1,NP)
      TYPE *,' INITIAL ENERGIES IN LONGITUDINAL DIRECTION (eV) ?'
      read(5,*) (E_long(I),I=1,NP)
      GOTO 76
65    TYPE *, ' INITIAL POSITIONS IN THE ORDER X0 AND Y0 ( MM
     1) OF 1 TO NP'
      READ(5,*)(X01(I),Y01(I),I=1,NP)
      TYPE *,'  INITIAL TRANS. AND LONGI. ENERGIES IN eV IN THE ORDER OF
     1 1 TO NP'
      READ(5,*)(TT(I),TL(I),I=1,NP)
76    Z0=-3.
      TYPE *,'    Z STEP SIZE ( MM ) AND NUMBER OF POINTS WHICH WILL'
      TYPE *,'    BE WRITTEN INTO THE OUTPUT DATA FILE. THE POINT '
      TYPE *,'    STEP SIZE IS NZ*(Z STEP SIZE ).'
      TYPE *,'     TYPE  DZ , NZ , M '
      READ(5,*) DZ0,NZ,M
        TYPE *,'    EXTRACTION APERTURE (RADIUS), PULLER APERTURE'
        TYPE *,'    (RADIUS), EXTRACTION SPACING, RADIUS'
        TYPE *,'    OF BEAM PIPE, EXTRACTION ELETRIC FIELD LENGTH,'
        TYPE *,'    EXTRACTION MAGNETIC FIELD LENGTH AND FOCUSSING'
        TYPE *,'    SOLENOID FIELD LENGTH, ALL IN METER UNIT (m).'
       READ(5,*) EXR,PUR,EXSP,RPIP,EXEFL,EXBFL,FSFL
       EXRS=EXR*1000.
       WRITE(10,10)
10    FORMAT(5X,'  K  ',10X,' A ',10X,' Q ',5X,'I (UA)/RAY',/)
       DO K=1,NP
      WRITE(10,11)K,A(K),Q(K),AP(K)
11    FORMAT(4X,I4,10X,F4.0,9X,F4.0,5X,F8.4)
       ENDDO
       AP1=0.
       DO K=1,NP
         AP1=AP1+AP(K)*.001
       ENDDO
c
c     neutralzation factor is used to reduce the space force strength.
c     the second neutralization factor (FNEU_NF) will oversee the first
c     neutralization FNEU1 starting from ZN1 till ZN2.
c
      TYPE *, ' NEUTRALIZATION FACTOR ( < 1 ), IF NEUTRALIZATION  '
      TYPE *, ' OCCURS AT SOME REGION, TYPE 1 (ELSE 0), REGION SIZE'
      TYPE *, ' (ZN1 AND ZN2 (M)), NEUTRALIZATION FACTOR IN THAT REGION'
      READ(5,*)FNEU1,NF,ZN1,ZN2,FNEU_NF
      WRITE(10,13)DZ0,NZ,M,AP1,FNEU1
13    FORMAT(//,'  DZ = ',F8.5,2X,' POINT STEP SIZE = ',I4,
     12X,' MUNBER OF POINTS = ',I4,/,5X,' BEAM CURRENT = ',
     1 F5.2,' mA ',5X,' FNEU1 =',F4.2,/)
      type *,' VEXTRACTION ( V ) AND PULLER VOLTAGE ( in V )'
      type *, ' Percent of the source B field intensity?'
      READ(5,*) V_ext,V_pul,bpercent
```

```
C    DATE FIELD IS PRECALCULATED WITH 100 A. THE FOLLOWING EXF, EPUL AND BFSOL
C    ARE THE SCALE FIELD FACTORS WITH THE CORRESPONDING INPUTS.
C
        Exf=V_ext/10000.
        Epul=-V_pul/100.
        BFSOL=SOL_CURI/100.
C READ THE FIRST TWO INPUT DATA
      READ(11,*)ZC1,(BCR1(I),I=1,11)
      READ(11,*)ZC1,(BCR1(I),I=12,19)
      READ(11,*)ZC1,(BCZ1(I),I=1,11)
      READ(11,*)ZC1,(BCZ1(I),I=12,19)
      READ(11,*)ZC3,(BCR3(I),I=1,11)
      READ(11,*)ZC3,(BCR3(I),I=12,19)
      READ(11,*)ZC3,(BCZ3(I),I=1,11)
      READ(11,*)ZC3,(BCZ3(I),I=12,19)
      READ(12,*)ZZC1,(EXVR1(I),I=1,9)
      READ(12,*)ZZC1,(EXVR1(I),I=10,18)
      READ(12,*)ZZC1,(EXVZ1(I),I=1,9)
      READ(12,*)ZZC1,(EXVZ1(I),I=10,18)
      READ(13,*)ZZC1,(PULVR1(I),I=1,9)
      READ(13,*)ZZC1,(PULVR1(I),I=10,18)
      READ(13,*)ZZC1,(PULVZ1(I),I=1,9)
      READ(13,*)ZZC1,(PULVZ1(I),I=10,18)
      READ(12,*)ZZC3,(EXVR3(I),I=1,9)
      READ(12,*)ZZC3,(EXVR3(I),I=10,18)
      READ(12,*)ZZC3,(EXVZ3(I),I=1,9)
      READ(12,*)ZZC3,(EXVZ3(I),I=10,18)
      READ(13,*)ZZC3,(PULVR3(I),I=1,9)
      READ(13,*)ZZC3,(PULVR3(I),I=10,18)
      READ(13,*)ZZC3,(PULVZ3(I),I=1,9)
      READ(13,*)ZZC3,(PULVZ3(I),I=10,18)
      READ(14,*)ZSS1,(BSR1(I),I=1,8)
      READ(14,*)ZSS1,(BSR1(I),I=9,16)
      READ(14,*)ZSS1,(BSZ1(I),I=1,8)
      READ(14,*)ZSS1,(BSZ1(I),I=9,16)
      READ(14,*)ZSS3,(BSR3(I),I=1,8)
      READ(14,*)ZSS3,(BSR3(I),I=9,16)
      READ(14,*)ZSS3,(BSZ3(I),I=1,8)
      READ(14,*)ZSS3,(BSZ3(I),I=9,16)
      ZS1=SOL_LOC+ZSS1
      ZS3=SOL_LOC+ZSS3
      DO I=1,18
       VEXR1(I)=EXF*EXVR1(I)+EPUL*PULVR1(I)
       VEXZ1(I)=EXF*EXVZ1(I)+EPUL*PULVZ1(I)
       VEXR3(I)=EXF*EXVR3(I)+EPUL*PULVR3(I)
       VEXZ3(I)=EXF*EXVZ3(I)+EPUL*PULVZ3(I)
      ENDDO
       Bz0=BCZ1(1)*bpercent
       do K=1,NP
      write(10,200)K,Z0,X01(K),Y01(K),VX1(K),VY1(K),VZ1(K),TT(K),
     1TL(K),Bz0
       enddo
      CCK=1.79836*(100000.**2.)
      DO K=1,NP
      YI(k,4)=X0(K)
      YI(K,5)=Y0(K)
      YI(K,1)=VX0(K)
      YI(K,2)=VY0(K)
      YI(K,3)=VZ0(K)
      ENDDO
```

```
          READ(12,*)ZZC3,(EXVZ3(I),I=1,9)
          READ(12,*)ZZC3,(EXVZ3(I),I=10,18)
          READ(13,*)ZZC3,(PULVR3(I),I=1,9)
          READ(13,*)ZZC3,(PULVR3(I),I=10,18)
          READ(13,*)ZZC3,(PULVZ3(I),I=1,9)
          READ(13,*)ZZC3,(PULVZ3(I),I=10,18)
        DO I=1,18
          VEXR3(I)=EXF*EXVR3(I)+EPUL*PULVR3(I)
          VEXZ3(I)=EXF*EXVZ3(I)+EPUL*PULVZ3(I)
        ENDDO

       ENDIF
CC
C   TAKE THE CENTER MASS VELOCITY OF THE M PARTICLES AS THE BEAM VELOCITY
CC
301       DO 309 K=1,NP
         DO NS=1,NT
         IF(K.EQ.NSTOP(NS)) GOTO 309
         ENDDO
       RC=RM(K)
     if((Z.LT..0).and.(RC.GT.EXR)) then
         nt=nt+1
         nstop(nt)=k
         ZSTOP(NT)=Z
         QS(NT)=Q(K)
        goto 309
       endif
CC  RAYS WILL BE STOPPED IF ITS RADIUS LARGER THAN THE PULLER RADIUS AT THE
CC  VICINITY OF THE PULLER SINCE THESE RAYS WILL HIT THE PULLER. LACKING OF
CC  THE EXACT SHAPE OF THE PULLER, THE 45 DEGREE TILT SHAPE OF THE PULLER OF
CC  THE MSU RT-ECR IS USED HERE AS THE CRITERION TO STOP THOSE RAYS CAN NOT GO
CC  THROUGH THE PULLER. USERS CAN MODIFY THIS PART TO MEET THEIR DESIRES.
CC

    if((Z.ge.EXSP).and.(Z.le.EXSP+.01524)) AE=RC-PUR-(Z-EXSP)
       if(ABS(AE).le..0001) then
         nt=nt+1
         nstop(nt)=k
         ZSTOP(NT)=Z
         QS(NT)=Q(K)
         AE=1.
        goto 309
       endif
       if(RC.GT.RPIP) then
         nt=nt+1
         nstop(nt)=k
         ZSTOP(NT)=Z
         QS(NT)=Q(K)
        goto 309
       endif
       if((abs(Z-OSLT_LOC).le..001).and.(RC.gt.OSLIT)) then
         nt=nt+1
         nstop(nt)=k
         ZSTOP(NT)=Z
         QS(NT)=Q(K)
        goto 309
       endif
       IF(Z.GT.EXEFL) THEN
          ER=0.
          EZ=0.
```

```
          ENDIF
          IF(RC.LT.RCC(I)) THEN
             CR1=RCC(I-1)
             CR2=RCC(I)
             BBCZ1=((RC-CR1)*((BCZ3(I)-BCZ3(I-1)+BCZ1(I-1)-BCZ1(I))
     1          *(Z-ZC1)+(BCZ1(I)-BCZ1(I-1))*(ZC3-ZC1))/(ZC3-ZC1)
     1          /(CR2-CR1)+(BCZ3(I-1)-BCZ1(I-1))*(Z-ZC1)/(ZC3-
     1          ZC1)+BCZ1(I-1))*bpercent
             IF(RC.EQ.0.) THEN
              BBCX1=0.
              BBCY1=0.
              GOTO 39
             ENDIF
             BBCR1=((Z-ZC1)*((BCR3(I)-BCR3(I-1)+BCR1(I-1)-BCR1(I))
     1          *(RC-CR1)+(BCR3(I-1)-BCR1(I-1))*(CR2-CR1))/(ZC3-ZC1)
     1          /(CR2-CR1)+(BCR1(I)-BCR1(I-1))*(RC-CR1)/(CR2-CR1)
     1          +BCR1(I-1))*bpercent

             BBCX1=-BBCR1*YI(k,4)/RC
             BBCY1=-BBCR1*YI(K,5)/RC
             GOTO 39
          ENDIF
         ENDDO
39        IF((Z.GE.ZSOL).AND.(Z.LE.SOL_LOC+FSFL)) THEN
            DO I=1,16
             RCC(I)=.005*(I-1)
          IF(RC.LT.RCC(I)) THEN
             CR1=RCC(I-1)
             CR2=RCC(I)
             BBCZ2=((RC-CR1)*((BSZ3(I)-BSZ3(I-1)+BSZ1(I-1)-BSZ1(I))
     1          *(Z-ZS1)+(BSZ1(I)-BSZ1(I-1))*(ZS3-ZS1))/(CR2-CR1)
     1          /(ZS3-ZS1)+(BSZ3(I-1)-BSZ1(I-1))*(Z-ZS1)/(ZS3-
     1          ZS1)+BSZ1(I-1))*bfsol
             IF(RC.EQ.0.) THEN
              BBCX2=0.
              BBCY2=0.
              GOTO 53
             ENDIF
             BBCR2=((Z-ZS1)*((BSR3(I)-BSR3(I-1)+BSR1(I-1)-BSR1(I))
     1          *(RC-CR1)+(BSR3(I-1)-BSR1(I-1))*(CR2-CR1))/(ZS3-ZS1)
     1          /(CR2-CR1)+(BSR1(I)-BSR1(I-1))*(RC-CR1)/(CR2-CR1)
     1          +BSR1(I-1))*bfsol
             BBCX2=BBCR2*YI(k,4)/RC
             BBCY2=BBCR2*YI(K,5)/RC
             GOTO 53
          ENDIF
         ENDDO
         ENDIF
53      IF(RC.EQ.0.) THEN
          EX=0.
          EY=0.
         GOTO 54
        ENDIF
        TLD=0.
         DO 99 J=1,NP
          DO NQ=1,NT
          IF(J.EQ.NSTOP(NQ)) GOTO 99
          ENDDO
          IF(RM(J).GT.RC) GOTO 99
          TLD=TLD+AP(J)*.000001/YI(J,3)
```

```
            WRITE(6,1020)NSTOP(I)
          ELSE
            WRITE(6,1010)NSTOP(I)
        ENDIF
 1010   FORMAT(/,2X,'  PARTICLE',I4,' HIT THE PULLER.')
 1020   FORMAT(/,2X,'  PARTICLE',I4,' HIT THE BEAM PIPE.')
        ENDDO
 1030   DO I=1,NP
        IF(NC(I).EQ.0) GOTO 1110
        ZCL=ZC(I)*1000.
        ZRCL=ZRC(I)*1000.
        WRITE(6,1040) NC(I),ZCL,ZRCL
 1040   FORMAT(/,2X,'  PARTICLE',I4,' WAS CLOSE TO THE ELECTRODE WITHIN A
     1 GRID UNIT AT Z =',F7.2,' AND R =',F7.2,' mm')
        ENDDO
 1110   STOP
        END
```

Appendix C

BEAM3D_READ.FOR

```
      CHARACTER*132 JUNK
      CHARACTER*15 NAME,GEO
      real Z(405,600),R(405,600),X(405,600),Y(405,600)
      real VX(405,600),VY(405,600),PVX(405),PVY(405)
      real ZG(10,50),RG(10,50),PX(405),PY(405),BZ(405)
      REAL RAQ(405),RAQI(405),QQI(3,405),RQQI(3,405)
      REAL OA(405),OQ(405),VZ(405,600),EZ(10),ZB(405)
      REAL VVX(405),VVY(405),XXZ(405),YYZ(405),ZZE(10)
      REAL XV(405),YV(405),ZX(405),ZY(405),ALX(2),ALY(2)
      real tt(405),ttem(10),AP(405),API(405),APO(405)
      REAL OAP(405),XE(405),YE(405),TNOTX(10),TNOTY(10)
      real Q(405),AA(405),QI(405),AO(405),QO(405)
      CHARACTER*4 NUMB
      INTEGER ZAX(3),RAX(3),IZE(10),BZ0(3),NUB(405),LG(10)
      DATA BZ0/'B(Z)',' GAU','SS'/
      DATA ZAX/'Z   ','( mm',' )'/,RAX/'r   ','( mm',' )'/
      INTEGER XAX(3),YAX(3),NS(405)
      DATA XAX/'X   ','( mm',' )'/,YAX/'Y   ','( mm',' )'/

      INTEGER VAX(3),VAY(3)
      DATA VAX(1)/'X  ('/,VAX(2)/'mrad'/,VAX(3)/')  '/
      DATA VAY(1)/'Y  ('/,VAY(2)/'mrad'/,VAY(3)/')  '/
      INTEGER BI(3),AQR(3)
      DATA  BI/'I (',' euA',' )'/,AQR/' A/Q',' ',' '/
CTAA      TYPE *,' NAME'
CTAA      READ(5,FMT='(A15)')NAME
      TYPE *,' GEOMETRY FILE NAME'
      READ(5,FMT='(A15)')GEO
      TYPE *, ' HOW MANY RAYS ? TYPE M '
      READ(5,*) M
      TYPE *,'  SOLENOID LOCATION (M) AND SLIT WIDTH ?'
      READ(5,*) SOL_LOC,SLIT_WIDTH
      OPEN(100,FILE=GEO,STATUS='OLD')
       LGG=0
       DO I=1,10
       READ(100,*,END=110)LG(I),(ZG(I,J),J=1,LG(I)),
     &(RG(I,J),J=1,LG(I))
       LGG=LGG+1
       ENDDO
110    DO I=1,4
       ZG(4,I)=ZG(4,I)+SOL_LOC*1000.
      ENDDO
       DO I=2,3
         RG(5,I)=RG(5,I)+SLIT_WIDTH*1000.
       ENDDO
      TYPE *,'  EMITTANCE FITTING FRAC , TYPE NO OF LOCATIONS AND
     1 THE Z_STEP ( MM ) IN THE CALCULATION?'
      READ(5,*) FRAC,IZ,ZSTEP
      TYPE *,'   Z POSITIONS FOR THE EMITTANCE PLOTS?'
      READ(5,*) (EZ(I),I=1,IZ)
ctaa
ctaa  emittance locations are onput in meters- conver to mm
ctaa
      DO I=1,IZ
           EZ(I) = EZ(I)*1000.
      ENDDO
      TYPE *,'  ray number of the calculated main species?'
      READ(5,*) mm,Mr
      OPEN(1,STATUS='OLD')
      OPEN(11,STATUS='NEW',form='unformatted')
```

```
        do 1 I=1,M
      READ(1,*,END=999)N,Z(I,K),X(I,K),Y(I,K),VX(I,K),VY(I,K),VZ(I,K),
     1 tt(k),tl,BZ(K)
        R(I,K)=(X(I,K)*X(I,K)+Y(I,K)*Y(I,K))**.5
        AX=ABS(X(I,K))
        AY=ABS(Y(I,K))
        AZ=Z(I,K)
        if(tte.lt.tt(k)) tte=tt(k)
        IF(AX.GT.RXX) RXX=AX
        IF(AY.GT.RXX) RXX=AY
        IF(AZ.GT.RZX) RZX=AZ
1       IF(R(I,K).GT.RMAX) RMAX=R(I,K)
        BZ(K)=ABS(BZ(K))*10000.
        ZB(K)=RZX
        DO J=1,IZ
        IF((ABS(RZX-EZ(J)).LT.ZSTEP/2.).OR.(RZX-EZ(J).
     1   EQ.ZSTEP/2.).OR.(RZX.EQ.EZ(J))) THEN
         KEZ=KEZ+1
         ttem(kez)=tte
         IZE(KEZ)=K
         ZZE(KEZ)=RZX
        ENDIF
        ENDDO
2     continue
999   L=K-1
      l1=1
      NZ=RZX/25.4+1
      NR=RMAX/25.4+1
      I=6
      l=1
      a=float(NZ)
      if(a.lt.8.) a=8.
      b=float(NR)
      IF(b.lt.6.) b=6.
      write(11)I,l,a,b,a,b,d,d
      i=1
      l=3
      b=25.4
      c=0.
      d=-4.
      write(11)i,l,a,b,c,d,ZAX,ZAX
      a=float(NR)
      IF(a.lt.6.) a=6.
      i=2
      b=25.4
      d=0.
      write(11)i,l,a,b,c,d,RAX,RAX
      b=500.
      c=float(NZ)
      write(11)i,l,a,b,c,d,BZ0,BZ0
      I=7
      l=K-1
      B=25.4
      A=25.4
      C=-4.
      D=0.
      do 3 n1=1,M
3     WRITE(11)i,l,a,b,c,d,(Z(n1,j),j=1,l),(R(n1,j),j=1,l)
      IF(RZX.LT.SOL_LOC) LGG=3
      DO J=1,LGG
```

```
      write(11)i,l,a,b,c,d,BI,BI
      i=0
      l=2
      a=FLOAT(MA)
      DO NJ=1,LI
        RQQI(1,NJ)=RAQI(NJ)
         RQQI(2,NJ)=RAQI(NJ)
        QQI(1,NJ)=0.
        QQI(2,NJ)=API(NJ)
      ENDDO
      DO LJ=1,LI
      WRITE(11)i,l,a,b,c,d,(RQQI(j,LJ),j=1,l),(QQI(j,LJ),j=1,l)
      ENDDO
C
C
400   i=3
      l=1
      a=0.
      b=0.
      write(11)i,l,a,b,c,d,c,c
      NXY=RXX
        NDXY=FLOAT(NXY)/4.+2
        ND=NDXY

      i=6
      l=1
      a=8.
      b=8.
      write(11)i,l,a,b,A,B,d,d
      i=1
      l=3
      a=8.
      b=FLOAT(ND)
      c=0.
      d=-4.*b
      write(11)i,l,a,b,c,d,XAX,XAX
      i=2
      write(11)i,l,a,b,c,d,YAX,YAX
      i=7
      l=11
      a=FLOAT(ND)
      c=d
      do 5 n1=1,M
5     WRITE(11)i,l,a,b,c,d,(X(n1,j),j=1,l),(Y(n1,j),j=1,l)

C
      DO 1000 J=1,KEZ
      VMX=0.
      VMY=0.
      XMX=0.
      YMX=0.
      IA=0
      II=IZE(J)
      DO 800 IV=mm,Mr
      IF(Z(IV,II).LT.ZZE(J)) GOTO 800
      IA=IA+1
      NUB(IA)=IV
      VVX(IA)=VX(IV,II)/VZ(IV,II)*1000.
      VVY(IA)=VY(IV,II)/VZ(IV,II)*1000.
      XXZ(IA)=X(IV,II)
```

```
a=20.
c=-4.*a
WRITE(11)i,l,a,b,c,d,(ALX(j1),j1=1,l),(ALY(j1),j1=1,l)
ALX(1)=-4.*20.
ALY(1)=0.
ALX(2)=4.*20.
ALY(2)=0.
i=7
l=2
a=20.
c=-4.*a
WRITE(11)i,l,a,b,c,d,(ALX(j1),j1=1,l),(ALY(j1),j1=1,l)
CALL ELLCON(ANG,AE,BE,AREA,R12,XINT,YINT,XMAX,YMAX,YATXM,XATYM)
TNOTX(1)=ZZE(J)
TNOTX(2)=XINT+XC
TNOTX(3)=XMAX+XC
TNOTY(1)=YINT+YC
TNOTY(2)=YMAX+YC
TNOTY(3)=AREA
TNOTY(4)=YC
TNOTX(4)=XC
TNOTY(5)=ANG
TNOTX(5)=ANG
i=9
l=5
WRITE(11)i,l,a,b,c,d,(TNOTX(j1),j1=1,l),(TNOTY(j1),j1=1,l)

i=3
l=1
a=0.
b=0.
write(11)i,l,a,b,c,d,c,c
i=6
l=1
a=8.
b=8.
write(11)i,l,a,b,A,B,d,d

i=1
l=3
a=8.
b=20.
c=0.
d=-4.*b
write(11)i,l,a,b,c,d,XAX,XAX
i=2
b=20.
d=-4.*b
l=3
write(11)i,l,a,b,c,d,VAX,VAX
i=10
l=IA
a=20.
c=-4.*a
WRITE(11)I,L,A,B,C,D,(XXZ(J1),J1=1,L),(VVX(J1),J1=1,L)

i=7
l=IA+1
WRITE(11)i,l,a,b,c,d,(XE(j1),j1=1,l),(YE(j1),j1=1,l)
ALX(1)=0.
```

```
WRITE(11)(NUB(INB),INB=1,L)
CALL ELLFIT(YYZ,VVY,IA,FRAC,XC,YC,AE,BE,ANG)
CALL ELLIPS(AE,BE,XC,YC,ANG,IA,XE,YE,NUSE)

i=7
l=IA+1
XE(L)=XE(1)
YE(L)=YE(1)
WRITE(11)i,l,a,b,c,d,(XE(j1),j1=1,l),(YE(j1),j1=1,l)

ALX(1)=0.
ALY(1)=-4.*20.
ALX(2)=0.
ALY(2)=4.*20.
i=7
l=2
a=20.
c=-4.*a
WRITE(11)i,l,a,b,c,d,(ALX(j1),j1=1,l),(ALY(j1),j1=1,l)
ALX(1)=-4.*20.
ALY(1)=0.
ALX(2)=4.*20.
ALY(2)=0.
i=7
l=2
a=20.
c=-4.*a
WRITE(11)i,l,a,b,c,d,(ALX(j1),j1=1,l),(ALY(j1),j1=1,l)
CALL ELLCON(ANG,AE,BE,AREA,R12,XINT,YINT,XMAX,YMAX,YATXM,XATYM)
TNOTX(1)=ZZE(J)
TNOTX(2)=XINT+XC
TNOTX(3)=XMAX+XC
TNOTY(1)=YINT+YC
TNOTY(2)=YMAX+YC
TNOTY(3)=AREA
TNOTX(4)=XC
TNOTY(4)=YC
TNOTY(5)=ANG
TNOTX(5)=ANG
i=9
l=5
WRITE(11)i,l,a,b,c,d,(TNOTX(j1),j1=1,l),(TNOTY(j1),j1=1,l)

i=3
l=1
a=0.
b=0.
write(11)i,l,a,b,c,d,c,c
i=6
l=1
a=8.
b=8.
write(11)i,l,a,b,A,B,d,d
i=1
l=3
a=8.
b=20.
c=0.
d=-4.*b
write(11)i,l,a,b,c,d,YAX,YAX
```

```
      KL=0
      do 10 LI1=1,M
      IF(Z(LI1,L1).LT.RZX) GOTO 10
       KL=KL+1
      NUB(KL)=LI1
      PX(KL)=X(LI1,L1)
      PY(KL)=Y(LI1,L1)
      OQ(KL)=QO(LI1)
      OA(KL)=AO(LI1)
      OAP(KL)=APO(LI1)
      RAQ(KL)=OA(KL)/OQ(KL)
10    CONTINUE

      i=5
      l=KL
      a=FLOAT(ND)
      c=d
      DO J=1,L
      PX(J)=PX(J)/A+4.
      PY(J)=PY(J)/B+4.
      ENDDO
      WRITE(11)i,l,a,b,c,d,(PX(J),J=1,L),(PY(J),J=1,L)
      WRITE(11)(NUB(INB),INB=1,L)
C
       DO I=2,KL
         DO J=I,KL
         IF(RAQ(J).LT.RAQ(I-1)) THEN
            RAT=RAQ(I-1)
            QT=OQ(I-1)
            APT=OAP(I-1)
            RAQ(I-1)=RAQ(J)
            OQ(I-1)=OQ(J)
            OAP(I-1)=OAP(J)
            OAP(J)=APT
            RAQ(J)=RAT
            OQ(J)=QT
          ENDIF
         ENDDO
       ENDDO
        NT=0
        LI=0.
        DO 601 I=1,KL
           DO LN=1,NT
           IF(I.EQ.NS(LN)) GOTO 601
           ENDDO
           LI=LI+1
           API(LI)=0.
           RAQI(LI)=RAQ(I)
           DO 501 J=I,KL
           DO LN=1,NT
           IF(J.EQ.NS(LN)) GOTO 501
           ENDDO
           IF(RAQ(J).EQ.RAQ(I)) THEN
            API(LI)=API(LI)+OAP(J)
            NT=NT+1
            NS(NT)=J
           ENDIF
501        CONTINUE
601        CONTINUE
      i=3
```

```
C==  INPUT  PARAMETERS: A,B,XC,YC,ANG,N.                                  ==
C==                                                                       ==
C==  OUTPUT PARAMETERS: X,Y,NUSE.                                         ==
C==                                                                       ==
C=========================================================================
C=========================================================================
C==                                                                       ==
C==  SUBROUTINES CALLED:                                                  ==
C==                                                                       ==
C==  NAME   TYPE              EXPLANATION                                 ==
C==                                                                       ==
C==  COS    R*4         COSINE FUNCTION.                                  ==
C==  SIN    R*4         SINE FUNCTION.                                    ==
C==                                                                       ==
C=========================================================================
C=========================================================================
C==                                                                       ==
C==  VARIABLES:                                                           ==
C==                                                                       ==
C==  NAME   TYPE              EXPLANATION                                 ==
C==                                                                       ==
C==  A      R*4         SEMI-MAJOR AXIS LENGTH.                           ==
C==  ANG    R*4         ANGLE OF ORIENTATION W.R.T TO THE X-AXIS.         ==
C==                     MEASURED IN DEGREES.                              ==
C==  B      R*4         SEMI-MINOR AXIS LENGTH.                           ==
C==  CANGR  R*4         COSINE OF ANG.                                    ==
C==  I      I*4         DO LOOP INDEX.                                    ==
C==  N      I*4         NUMBER OF POINTS ELLIPS TRYS TO POPULATE THE      ==
C==                     ELLIPSE WITH. ACTUAL NUMBER OF POINTS RETURN-     ==
C==                     ED IN X,Y IS NUSE=N/4*4.                          ==
C==  ND2M1  I*4         N/2-1.                                            ==
C==  ND4    I*4         N/4.                                              ==
C==  ND4P1  I*4         N/4+1.                                            ==
C==  NUSE   I*4         ACTUAL NUMBER OF POINTS USED TO POPULATE THE      ==
C==                     ELLIPSE. NUSE=N/4*4.                              ==
C==  PI     R*4         WELL KNOWN CONSTANT: 3.14159265.                  ==
C==  SANGR  R*4         SINE OF ANGLE ANG.                                ==
C==  THINC  R*4         THETA INCREMENT.                                  ==
C==  X      R*4(N)      NUSE X-COORDINATES OF THE ELLIPSE.                ==
C==  XC     R*4         X-COORDINATE OF THE CENTER OF THE ELLIPSE.        ==
C==  XX     R*4         X-COORDINATE OF THE ELLIPSE.                      ==
C==  Y      R*4(N)      NUSE Y-COORDINATES OF THE ELLIPSE.                ==
C==  YC     R*4         Y-COORDINATE OF THE CENTER OF THE ELLIPSE.        ==
C==  YY     R*4         Y-COORDINATE OF THE ELLIPSE.                      ==
C==                                                                       ==
C=========================================================================
C=========================================================================
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      REAL X(N),Y(N)
      PI=3.14159265
      ND4=N/4
      NUSE=4*ND4
      IF(ND4.EQ.0)RETURN
      ND4P1=ND4+1
      THINC=PI/2./ND4
C=========================================================================
C==                                                                       ==
C==   POPULATE 1/4 OF THE ELLIPSE WITH N/4 POINTS IN QUADRANT 1 OF        ==
C==   THE STANDARD COORDINATE SYSTEM FOR THE ELLIPSE.                     ==
C==                                                                       ==
```

```
C==   OUTPUT PARAMETERS: AREA,R12,XINT,YINT,XMAX,YMAX,YATXM,    ==
C==                      XATYM (R*4).                           ==
C==                                                             ==
C==   ANGLE : ANGLE OF THE ELLIPSE SEMI-MAJOR AXIS A RELATIVE   ==
C==           TO THE POSITIVE X-AXIS IN DEGREES.                ==
C==   A     : SEMI-MAJOR AXIS OF THE ELLIPSE.                   ==
C==   B     : SEMI-MINOR AXIS OF THE ELLIPSE.                   ==
C==           NOTE: B MAY BE GREATER THAN A.                    ==
C==   AREA  : AREA OF THE ELLIPSE = PI*A*B.                     ==
C==                                                             ==
C==   FOR THE FOLLOWING PARAMETERS THE X-AXIS AND Y-AXIS ARE    ==
C==   ASSUMED TO PASS THROUGH THE CENTER OF THE ELLIPSE:        ==
C==                                                             ==
C==   R12   : CORRELATION COEFFICIENT OF THE ELLIPSE.           ==
C==   XINT  : POSITIVE X-INTERCEPT OF THE ELLIPSE.              ==
C==   YINT  : POSITIVE Y-INTERCEPT OF THE ELLIPSE.              ==
C==           NOTE: EQUATION OF ELLIPSE IS:                     ==
C==           X**2/XI**2-2*R12*X*Y/XI/YI+Y**2/YI**2 = 1         ==
C==           WHERE XI = X-INTERCEPT, YI = Y-INTERCEPT.         ==
C==   XMAX  : MAXIMUM X-COORDINATE OF THE ELLIPSE (X 1/2 WIDTH)==
C==   YMAX  : MAXIMUM Y-COORDINATE OF THE ELLIPSE (Y 1/2 WIDTH)==
C==   YATXM : Y COORDINATE OF THE ELLIPSE AT XMAX.              ==
C==   XATYM : X COORDINATE OF THE ELLIPSE AT YMAX.              ==
C==                                                             ==
C==   THE ABOVE PARAMETERS ARE CALCULATED FROM ANGLE,A,B USING ==
C==   THE FOLLOWING FORMULAS:                                   ==
C==                                                             ==
C==   AREA  = PI*A*B.                                           ==
C==   XMAX  = SQRT((A*COS(ANGLE))**2+(B*SIN(ANGLE))**2).        ==
C==   YMAX  = SQRT((A*SIN(ANGLE))**2+(B*COS(ANGLE))**2).        ==
C==   XINT  = (A*B)/YMAX  IF YMAX .NE. 0.                       ==
C==         = 0.          IF YMAX = 0.                          ==
C==   YINT  = (A*B)/XMAX  IF XMAX .NE. 0.                       ==
C==         = 0.          IF XMAX = 0.                          ==
C==   R12   = SIGN*SQRT(1.-(A*B/XMAX/YMAX)**2),                 ==
C==           WHERE SIGN = SIGN(1.,SINCOS*(A**2-B**2))          ==
C==                      = +1 OR -1 IF SINCOS*(A**2-B**2) NE 0.==
C==                      =  0       IF SINCOS*(A**2-B**2) =  0.==
C==               SINCOS = SIN(ANGLE) * COS(ANGLE).             ==
C==                                                             ==
C==   YATXM = R12*YMAX.                                         ==
C==   XATYM = R12*XMAX.                                         ==
C==                                                             ==
C=================================================================
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL ANGLE,A,B,AREA,R12,XINT,YINT,XMAX,YMAX,YATXM,XATYM
      DATA PI/3.14159265358979323846264D0/
C=================================================================
C==   CALCULATE AREA.                                           ==
C=================================================================
      AB=ABS(A*B)
      AREA=PI*AB
C=================================================================
C==   CALCULATE XMAX AND YMAX.                                  ==
C=================================================================
      ANGL=ANGLE*PI/180.D0
      COSA=DCOS(ANGL)
      SINA=DSIN(ANGL)
      SINCOS=SINA*COSA
```

```
      REAL XIN(NPTS),YIN(NPTS)
      REAL XPP(5000),YPP(5000),EFFA(5000)
C*****************************************************************
C********  RETURN THROUGH 1 IF INVALID VALUES FOR NPTS OR FRAC.
C*****************************************************************
      IF(NPTS.LE.1 .OR. FRAC.LT..0 .OR. FRAC.GT.1.) RETURN 1
      IF(FRAC*NPTS .LE. .5) RETURN 1
C*****************************************************************
C********  FIT LEAST PERPENDICULAR SQUARED LINE TO DATA.
C********  EQUATION OF LINE IS U*X + V*Y + W = 0, WHERE
C********  U*U + V*V = 1
C*****************************************************************
      CALL LQLINE(XIN,YIN,NPTS,2,U,V,W,XCENTR,YCENTR,&10)
      GO TO 20
 10   RETURN 1
C*****************************************************************
C******  FIND ANGLE OF ELLIPSE (ANGLE FROM X-AXIS COUNTERCLOCKWISE).
C*****************************************************************
 20   ANGLE=ATAN2(U,-V)
C*****************************************************************
C******  THE X" AXIS IS THE AXIS ALONG THE LINE U*X + V*Y + W = 0
C******  WITH ZERO AT THE POINT (XCENTR,YCENTR).
C******  THE Y" AXIS IS PERPENDICULAR TO THE X" AXIS.
C******  CALCULATE NEW ARRAYS X" (CALLED XPP) AND Y" (CALLED YPP).
C*****************************************************************
 30   SINA=SIN(ANGLE)
      COSA=COS(ANGLE)
      ANGLE2=ANGLE*180.0/3.14159265
      DO 40 I=1,NPTS
      XP=XIN(I)-XCENTR
      YP=YIN(I)-YCENTR
      XPP(I)=XP*COSA + YP*SINA
      YPP(I)=YP*COSA - XP*SINA
 40   CONTINUE
C*****************************************************************
C******  FIND VARIANCES OF XPP, AND OF YPP
C*****************************************************************
      SYPPSQ=0.0
      SXPPSQ=0.0
      DO 60 I=1,NPTS
      SYPPSQ=SYPPSQ + YPP(I)*YPP(I)
      SXPPSQ=SXPPSQ + XPP(I)*XPP(I)
 60   CONTINUE
      XVARI=SXPPSQ/NPTS
      YVARI=SYPPSQ/NPTS
      IF(XVARI.LT..0001) RETURN 1
      BOVERA=SQRT(YVARI/XVARI)
      BOASQ=BOVERA*BOVERA
C*****************************************************************
C******  NOW FIND VALUES EFFA FOR EACH POINT, WHICH IS THE A
C******  OF ELLIPSE WHICH WOULD PLACE EACH POINT ON THE PERIMETER.
C*****************************************************************
      DO 80 I=1,NPTS
      EFFA(I)=ABS(XPP(I))
      IF(ABS(BOASQ).GT..0001) EFFA(I) =
     X SQRT(XPP(I)*XPP(I)  + YPP(I)*YPP(I)/BOASQ)
 80   CONTINUE
C
C********  SORT EFFECTIVE A'S.
C
```

```
C*     ALSO: ROUTINE RETURNS XCM (X CENTER OF MASS) AND YCM (Y
C*     CENTER OF MASS).
C*     IN CASE OF BAD INPUT DATA, ROUTINE RETURNS THROUGH 1.
C*
C**********************************************************************
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      REAL XIN(NPTS),YIN(NPTS)
      IF(NPTS.LT.2 .OR. IFLAG.LT.1 .OR. IFLAG.GT.2) RETURN 1
C
C******  CALCULATE SUMS OF:  (XIN), (YIN), (XIN**2), (YIN**2),
C******   AND (XIN*YIN)
C
      SUMX=0.0
      SUMY=0.0
      SUMXSQ=0.0
      SUMYSQ=0.0
      SUMXY=0.0
      DO 20 I=1,NPTS
      SUMX=SUMX+XIN(I)
      SUMY=SUMY+YIN(I)
      SUMXSQ=SUMXSQ + XIN(I)*XIN(I)
      SUMYSQ=SUMYSQ + YIN(I)*YIN(I)
      SUMXY =SUMXY  + XIN(I)*YIN(I)
 20   CONTINUE
C
C********  CALCULATE CENTER OF MASSES.
C
      XCM = SUMX/NPTS
      YCM = SUMY/NPTS
      IF(IFLAG.EQ.2) GO TO 40
C
C******  FIND A AND B FOR LEAST Y**2 LINE Y=AX+B
C
      A = (NPTS*SUMXY - SUMX*SUMY) / (NPTS*SUMXSQ - SUMX*SUMX)
      B = (SUMY - A*SUMX) / NPTS
      RETURN
C
C********  FIND A, B, AND C FOR LEAST PERPENDICULAR**2 LINE
C********  A*X + B*Y + C = 0
C
 40   A=0
      B=1
      C=YCM
      D=NPTS*SUMXSQ - NPTS*SUMYSQ - SUMX*SUMX + SUMY*SUMY
      E=NPTS*SUMXY - SUMX*SUMY
      IF(ABS(D).LT.1.E-6 .AND. ABS(E).LT.1.E-6) RETURN
      F = E*E / (D*D + 4*E*E)
      ISIGN=1
      IF(D.GT.0) ISIGN=-1
      ASQ = (1. + ISIGN*SQRT(1.-4.*F)) / 2.
      A=SQRT(ASQ)
      B=1.
      IF(A.GT.1.E-6) B=SIGN(1.,-E) * SQRT(1.-ASQ)
      IF(A.GT.1.E-6 .AND. ABS(D).GT.1.E-6) B=((ASQ+ASQ-1.)*E) / (A*D)
      C= -(A*SUMX + B*SUMY)/NPTS
      RETURN
      END
```

Appendix D

BEAM3D_PLOT.FOR

```
C
C   25 APRIL 1985     AUXILARY PLOTTING ROUTINE FOR THE SLAC ELECTRON
C   (T.A.Antaya)                     TRAJECTORY CODE
C
C   SLACETC puts plot data on FOR001 when control variable MI is non-
C   zero.   This plot data includes trajectories, equipotentials, fields
C   and axes for one or more cycles of the poisson calculation.  The plot
C   data format is specified in the SLACETC header, which have been excised
C   and re-specified here:
C
C   ----------------------------------------------------------------
C   SUMMARY OF FILE 1 FORMAT FOR PLOT DATA OUTPUT
C   ----------------------------------------------------------------
C WRITE(1)I,L,A,B,C,D,(X(J),J=1,L),(Y(J),J=1,L)
C
C WHERE:
C I=0 THROUGH 10
C FOR I=0,7,8  PLOT A LINE
C      L=NUMBER OF DATA POINTS TO BE PLOTTED
C      X, Y ARE ARRAYS OF LENGTH >= L, WITH X,Y DATA
C  FOR I=1, PLOT X AXIS, FOR I=2, PLOT Y AXIS
C      L=NUMBER OF COMPUTER WORDS IN TITLE
C         FOR IBM/360  L=(N+3)/4 IF N=NUMBER OF CHARS
C      A=SCALE (DATA UNITS/INCH)
C      B=AXIS LENGTH (INCHES)
C      C=X COORD OF Y AXIS, OR Y COORD OF X (OTHER COORD IS 0.)
C      D=DATA VALUE TO APPEAR ON LOWER END OF AXIS
C FOR I=3,  END OF PICTURE, GET A CLEAN AREA ON PAPER, ETC.
C       L=1;   A,B,C,D,X,Y=0.0
C FOR I=4, CLOSE PLOT, THIS IS THE LAST RECORD OF THE FILE
C       L=1;   A,B,C,D,X,Y=0.
C FOR I=5, PLOT NUMBERS  (OR X'S, OR SOME SYMBOL)
C       L,A,B,C,D,X,Y   SAME AS FOR I=0 (LINES)
C FOR I=10, PLOT CROSSES  (OR X'S, OR SOME SYMBOL)
C       L,A,B,C,D,X,Y   SAME AS FOR I=0 (LINES)
C FOR I=6, SET SCALE FACTOR
C          A=X AXIS LENGTH
C          B=Y AXIS LENGTH
C          C=SX (FROM &INPUT5)
C          D=SY
C FOR I=9, SPECIAL TITLE FOR EMITTANCE PLOTS
C       PLOT AREA MUST BE AT LEAST -0.5<X<A+0.5  -0.5<Y<B+0.5
C       C AND D CAN BE USED IF NEEDED.
C       THE TITLE ON THE AXIS SHOULD BE UNDER THE X AXIS,
C       AND TO THE LEFT OF THE Y AXIS  (THE PROGRAM CAN PLOT
C       MORE THAN ONE Y AXIS ON A PLOT, SO BE CAREFUL.)
C  I LESS THAN 0, OR GREATER THAN 9 SHOULDN'T HAPPEN, BUT CHECK IT.
C  -----------------------------------------------------------------
C
C
C  I/O:
C            FOR006     SYS$OUTPUT
C            FOR010     SLACETC plot data file
C
C
     real x(1000),y(1000)
     real*8 titles(11)
    */'LINE','X-AXIS','Y-AXIS','NEW PIC.','CLOSE','POINTS','SCALE',
    *'LINE','LINE','TITLE','CROSS'/
     INTEGER NUB(1000)
```

```
      goto 1
 17   IF(X(5).GT.90.) GOTO 18
      CALL NOTE (5.75,2.15,'Z = ',4)
      CALL NOTE (9999.,9999.,X(1),1002)
      CALL NOTE (5.75,1.9,'XC = ',5)
      CALL NOTE (9999.,9999.,X(4),1002)
      CALL NOTE (5.75,1.65,'YC = ',5)
      CALL NOTE (9999.,9999.,Y(4),1002)
      CALL NOTE (9999.,9999.,' mm',3)
      CALL NOTE (5.75,1.4,'Xint = ',7)
      CALL NOTE (9999.,9999.,X(2),1002)
      CALL NOTE (9999.,9999.,' mm',3)
      CALL NOTE (5.75,1.15,'Xmax = ',7)
      CALL NOTE (9999.,9999.,X(3),1002)
      CALL NOTE (9999.,9999.,' mm',3)
      CALL NOTE (5.75,.9,'Yint = ',7)
      CALL NOTE (9999.,9999.,Y(1),1002)
      CALL NOTE (9999.,9999.,' mrad',5)
      CALL NOTE (5.75,.65,'Ymax = ',7)
      CALL NOTE (9999.,9999.,Y(2),1002)
      CALL NOTE (9999.,9999.,' mrad',5)
      CALL NOTE (5.75,.40,'Area = ',7)
      CALL NOTE (9999.,9999.,Y(3),1002)
      CALL NOTE (9999.,9999.,' mm mrad',8)
      goto 1
 18   CALL NOTE (.4,2.15,'Z = ',4)
      CALL NOTE (9999.,9999.,X(1),1002)
      CALL NOTE (.4,1.9,'XC = ',5)
      CALL NOTE (9999.,9999.,X(4),1002)
      CALL NOTE (.4,1.65,'YC = ',5)
      CALL NOTE (9999.,9999.,Y(4),1002)
      CALL NOTE (9999.,9999.,' mm',3)
      CALL NOTE (.4,1.4,'Xint = ',7)
      CALL NOTE (9999.,9999.,X(2),1002)
      CALL NOTE (9999.,9999.,' mm',3)
      CALL NOTE (.4,1.15,'Xmax = ',7)
      CALL NOTE (9999.,9999.,X(3),1002)
      CALL NOTE (9999.,9999.,' mm',3)
      CALL NOTE (.4,.9,'Yint = ',7)
      CALL NOTE (9999.,9999.,Y(1),1002)
      CALL NOTE (9999.,9999.,' mrad',5)
      CALL NOTE (.4,.65,'Ymax = ',7)
      CALL NOTE (9999.,9999.,Y(2),1002)
      CALL NOTE (9999.,9999.,' mrad',5)
      CALL NOTE (.4,.4,'Area = ',7)
      CALL NOTE (9999.,9999.,Y(3),1002)
      CALL NOTE (9999.,9999.,' mm mrad',8)
      goto 1
20    x(l+1)=C
      x(l+2)=A
      y(l+1)=D
      y(l+2)=B
      call line (x,y,l,1,-1,4)
      goto 1
c

99    write (6,102)
102   format (' end of file found, hope plot was finished ...')
      stop
      end
```

Appendix E

Sample VMS/DCL Command File for Executing BEAM3D code group

```
$!
$! EXECUTE BEAM3D FOR TRAJECTORY COMPUTATIONS
$!
$SET DEFAULT 'P1'
$ASSIGN 'P2'BM.DAT FOR005
$ASSIGN 'P2'BM.OUT FOR010
$RUN [my_directory]BEAM3D.EXE
$!
$! EXECUTE BEAM3D_READ FOR PLOT DEVELOPMENT
$!
$ASSIGN 'P2'BM.OUT FOR001
$ASSIGN 'P2'RD.DAT FOR005
$ASSIGN 'P2'RD.OUT FOR011
$RUN [my_directory]BEAM3D_READ.EXE
$!
$! EXECUTE BEAM3D_PLOT FOR PLOT RASTER CONSTRUCTION
$!
$ASSIGN 'P2'RD.OUT  FOR010
$!
$RUN [my_directory]BEAM3D_PLOT.EXE
$!
$! EXECUTE PLOT10 CONVERSION CODE PASS2
$! THIS STEP IS HIGHLY SYSTEM DEPENDENT
$! PASS2 PREPARES PLOT FOR TRILOG PRINTER AT NSCL
$!
$RUN PASS2
$DELETE FOR002.DAT;*
$DELETE FOR003.DAT;*
$!
$! LINE PRINTER OUTPUT
$!
$PRINT 'P2'BM.DAT
$PRINT 'P2'RD.DAT
$PRINT/NOFEED/DELETE LP.LIS;0
$!
$!  CLEANUP
$!
$DELETE 'P2'BM.OUT;0
$DELETE 'P2'RD.OUT;0
```