Converting 2D Trak distributions to 3D

Stanley Humphries, Ph.D.

Field Precision LLC
E mail: techinfo@fieldp.com
Internet: https://www.fieldp.com

Trak is a two-dimensional finite-element code for electric/magnetic field calculations and charged-particle trajectory tracing. It is widely used for the design of cylindrically symmetric, space-charge-limited electron and ion guns. This report explains methods for converting the 2D distributions generated by the code to a 3D format. Applications for such conversions include the following:

- Input to **OmniTrak** to model beam dynamics through a transport system with 3D elements (for example, bending magnets, electrostatic deflectors,...).
- Input to **GamBet** to model a 3D X-ray source.
- Transfer to other beam transport codes.

Conversions are performed with **GenDist**, a utility for generating, analyzing and transforming particle distributions. The program can be applied in two ways:

- Direct conversion with the *Beam Sector Tool*.
- Parametric conversion using analysis and beam generation functions.

This report follows benchmark calculations to illustrate the techniques and relative advantages of the approaches. The project involves three tasks:

- Design an electron gun and solenoid lens with **Trak** to generate a 0.6 A, 10 keV beam with adjustable focal spot.
- Convert the output distribution to 3D format with **GenDist**.
- Propagate the beam through an electrostatic scanner to a target with **OmniTrak**.

The solenoid lens in the **Trak** calculation controls the exit beam so it is not necessary to design a converging beam gun[1]. Instead, we will consider a modified planar gap design where the beam diverges after passing through the exit aperture. The initial constraint is a moderate source current density, $j_e = 2.0$ A/cm^2. For a planar gap, the current density is given by:

$$j_e = \left(\frac{4\epsilon_o}{9}\right) \sqrt{\frac{2e}{m_e}} \; \frac{V_o^{3/2}}{d^2}. \qquad (1)$$

The target value of j_e with $V_0 = 10.0$ kV implies a gap width $d \cong 1.1$ cm. The total current $I_e = 0.6$ A implies a cathode radius of about $r_c = 0.3$ cm.

[1]https://www.fieldp.com/tutorials/egundesign.pdf

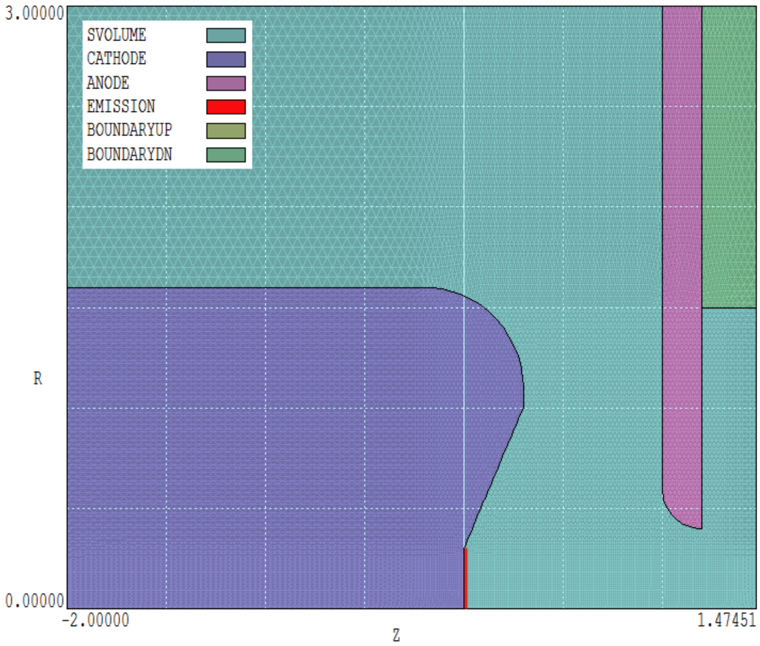

Figure 1: Expanded view of the electron gun region showing the variable mesh resolution and emission surface.

Figure 1 shows a closeup view of the gun region. The cathode support has potential -10 kV. A disk of radius 0.3 cm on the front surface is defined as an emission surface. The cathode support shape defines a focusing extension at the Pierce angle (22.5^o). Fine mesh resolution is used in the acceleration gap and at small radius so that the emission surface includes 30 element facets.

Figure 2 shows model electron orbits and electric and magnetic fields of the complete solution. The beam current is 0.608 A. The beam diverges passing through the exit aperture and enters a drift region where trajectories are determined by self-consistent space-charge electric fields and the magnetic field of the solenoid lens. The lens, modeled on an independent larger mesh, consists of a coil inside a iron magnetic shield. At a lens current of 800 A-turn, the beam trajectories are approximately parallel to the axis at the end of the drift region. At this setting, the peak lens field is $B_z = 136$ G.

The number of model electrons per emission facet can be adjusted. One calculation was carried out with a moderate number of particles, one per facet for a total of 30 particles. A second run used 10 particles per facet for a total of 300. The number chosen has only a small effect on the total current and output beam dynamics, but is a consideration in the 2D/3D conversion. The input script for the **Trak** calculation calls for the creation of a `PRT` file listing the positions, momenta and currents of model particles at the exit point. The data can be loaded into **GenDist** to create a variety of plots. Figure 3 shows

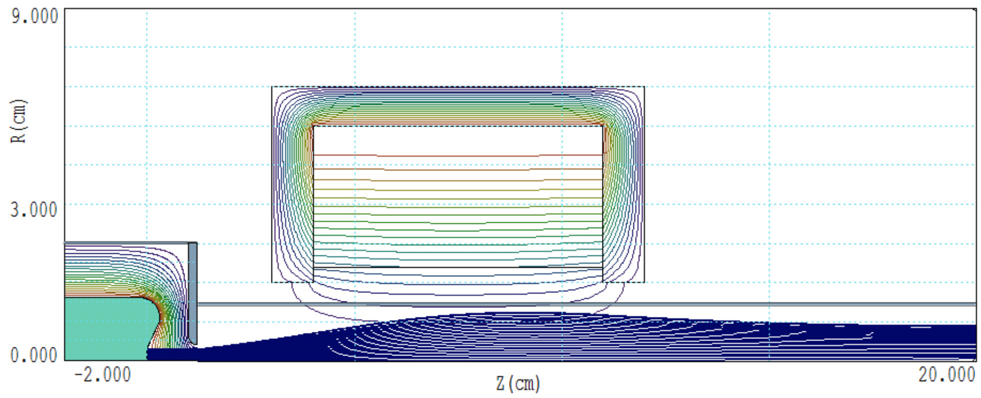

Figure 2: Trak solution for an electron gun with injection into a beam pipe passing through a solenoid lens. The plot shows model electron trajectories, electrostatic equipotential lines and lines of magnetic flux density.

the beam distribution in the transverse x-y plane at the exit of the transport tube, $z = 20.0$ cm. This type of plot often confuses users who expect to see a circular beam distribution. The explanation is that in a 2D simulation with cylindrical symmetry, there is no point in calculating multiple trajectories at different azimuths because they are all identical. **Trak** applies the following conventions:

- Representative model particles may have different radii but trajectories need be calculated only at one azimuth.
- The trajectory calculations are performed in Cartesian rather than cylindrical coordinates to avoid singularities at the axis.
- For space-charge assignment and the calculation of other quantities, each model particle is treated as an annular section centered at its position in r.
- **Trak** starts model trajectories along the x axis.

With these considerations, the distribution of Fig. 3 does represent a circular beam that rotates $\theta \geq 85.4^o$ in the θ direction passing through the solenoid lens. The plot also gives a sensitive indication of non-linear forces at the gun aperture and in the lens. Because of over-focusing at larger radii, the rotation at the beam envelope is 87.0^o. Figure 4 shows the radial phase distribution at the system exit. Particle trajectories are parallel to z-axis to within $\pm 0.25^o$. The over-focused electrons are visible as a portion of the distribution that folds back (red arrow).

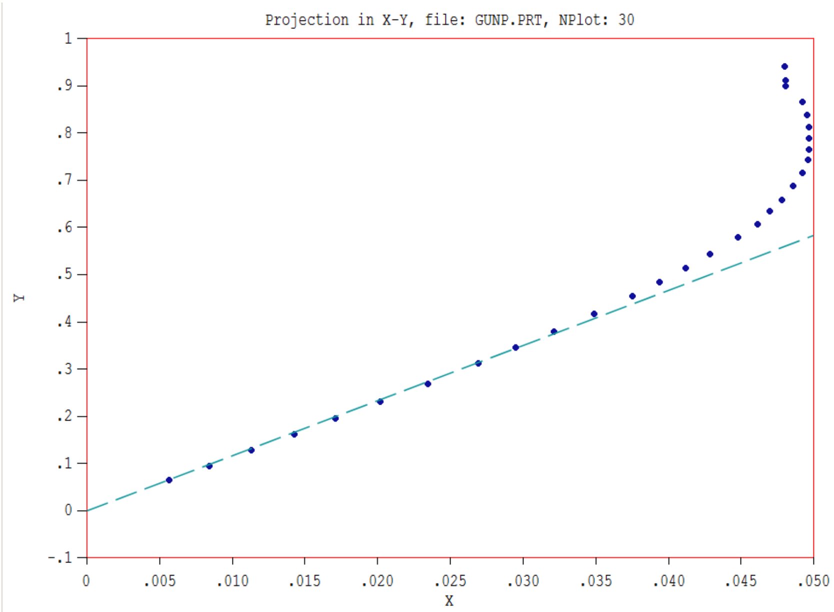

Figure 3: Model electron positions in the x-y plane at the transport tube exit ($z = 20.0$ cm) for a lens current of 800 A-turn. The dashed line represents a rotation of 85.4°. (30 model electrons)

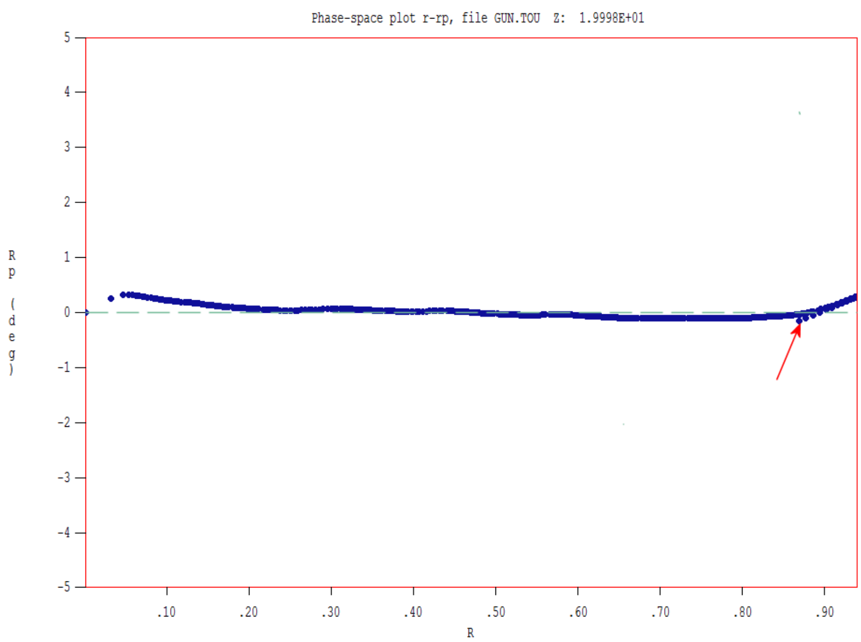

Figure 4: Radial phase-space distribution in the exit plane for a lens current of 800 A-turn. (300 model electrons.)

Although the distribution of Fig. 3 makes sense in a cylindrical code, it is not directly useful for calculations of space-charge and other effects in three-dimensional space. Nonetheless, it contains the information necessary to construct a 3D distribution. We have the radial position and momentum of a set of representative particles and we know that $p_\theta \cong 0.0$ at the solenoid lens exit through conservation of angular momentum. We simply need to project the trajectory parameters to a set of azimuthal positions with the proper rotations. This operation can be carried out automatically in **GenDist** using the `Beam Section Tool`. We load the 30 particle `PRT` file into **GenDist** and then click `Tools/Beam Section Tool` in the menu. Figure 5 shows the settings applied in the dialog:

- The reference axis is along z.
- The input distribution is two-dimensional.
- For each input particle, 60 particles are uniformly distributed in θ.

Clicking *OK* creates the additional particles and stores the distribution in memory. **GenDist** divides the current carried by each input model electron by 60 to preserve the beam net current. We can then save the data as a `PRT` file suitable for input into **OmniTrak** or **GamBet**. The data can be reloaded into **GenDist** for plotting and analysis. Figure 5 shows the circular beam distribution plotted in the x-y plane. In applications using the distribution as input, it may be convenient to set the particle entrance position to $z = 0.0$ cm. For this, we can use the `Transform Distribution Tool` in **GenDist**. In the dialog, set $ZShift = -20.0$, click `OK` to exit and save the modified `PRT` file.

As an illustration consider an **OmniTrak** application, the electrostatic deflector of Fig. 6. The converted circular beam distribution created from the **Trak** calculation enters a transport region from the grounded 1.5 cm radius beam pipe in the entrance plate. Shaped electrodes at $\pm V_0$ sweep the beam over an extended region of a grounded target at $z = 23.0$ cm. Electron trajectories in the drift region are determined by the combined effects of the applied electric field and the self-consistent space-charge field. The goal is to find the spatial distribution of electron beam power density incident on the target which might then be used in an X-ray or thermal calculation with **GamBet** or **HeatWave**. Figure 6 shows the conformal mesh generated by **MetaMesh** from data input from **Geometer**. The electrodes are shaped extrusions. The applied electric field is determined by a **HiPhi** calculation. The script to define the **OmniTrak** calculation contains the following information:

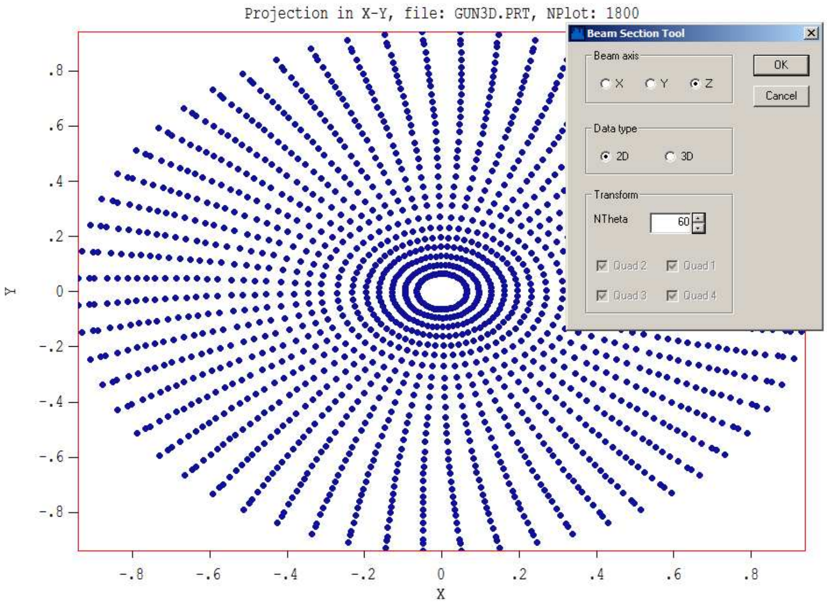

Figure 5: Converted beam distribution with 1800 electrons created from the **Trak** simulation with 30 model particles. Plot in the x-y plane with the `Beam Section Tool` dialog superimposed.

```
FIELDS
  EFIELD3D: Deflector.HOU
  DUNIT:   1.0000E+02
  MAXCYCLE:    500
  RESTARGET:   5.0000E-08
  PARALLEL 4
END
PARTICLES SCHARGE
  NCYCLE:  8
  AVG:   8.5000E-01
  PFILE: Gun3D
  PLOTSKIP 5 10
END
DIAGNOSTICS
  PARTFILE: DeflectorP
  EDUMP: DeflectorP.HOU
END
ENDFILE
```

The `FIELDS` section commands tell the program to read geometry and field data from the **HiPhi** output file `Deflector.HOU` and to interpret dimensions in cm. The other commands set parameters to control modeling beam space-charge. The `PFile` command of the `PARTICLES` section tell the program to read the file we prepared with **GenDist** (`Gun3D.PRT`). The information sets the positions, momenta and current of the incident electrons. In response to

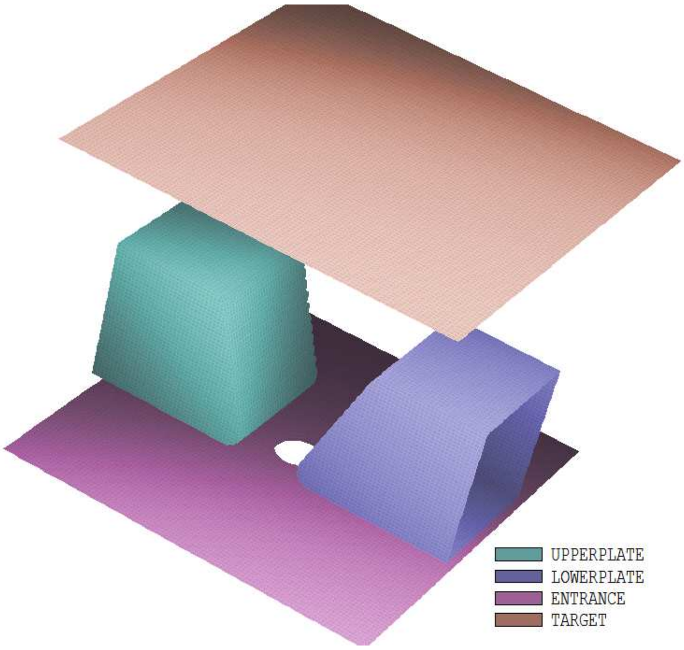

Figure 6: Electostatic deflector, z axis in the vertical direction.

the `PartFile` command of the `DIAGNOSTIC` section, **OmniTrak** creates the particle file `DeflectorP.PRT` for analysis in **GenDist** or transfer to another program such as **GamBet**. Figure 7a shows the result for $V_0 = \pm 1500.0$ V. The plot shows a sampling of electron paths projected in the x plane and equipotential lines of the total electric field. The initially parallel beam expands from space-charge repulsion as it is deflected by the applied electric field. The beam has a pronounced halo from the over-focused electrons on the periphery (Fig. 3).

A second approach to conversion of **Trak** results to a 3D format is to analyze the average radial properties of the output beam and to use the beam generation functions of **GenDist** to create an approximation. To get improved statistics for the analysis, we specify 10 electrons per facet in the **Trak** calculation for a total of 300 model particles. Loading the resulting `PRT` file into **GenDist**, we can generate the bin plot of the radial probability distribution of model particles weighted by their assigned currents of Fig. 8. The distribution corresponds to a hollowed beam with the flux density at the center about one half that at the edge.

Applying current weighting to the probability distribution (the default in **GenDist**) is an important consideration. Model particles in a cylindrical **Trak** run carry different currents because areas of the annular segments they represent vary with radius. Therefore, in an approximately uniform-flux beam, particles near the axis carry small current compared to those

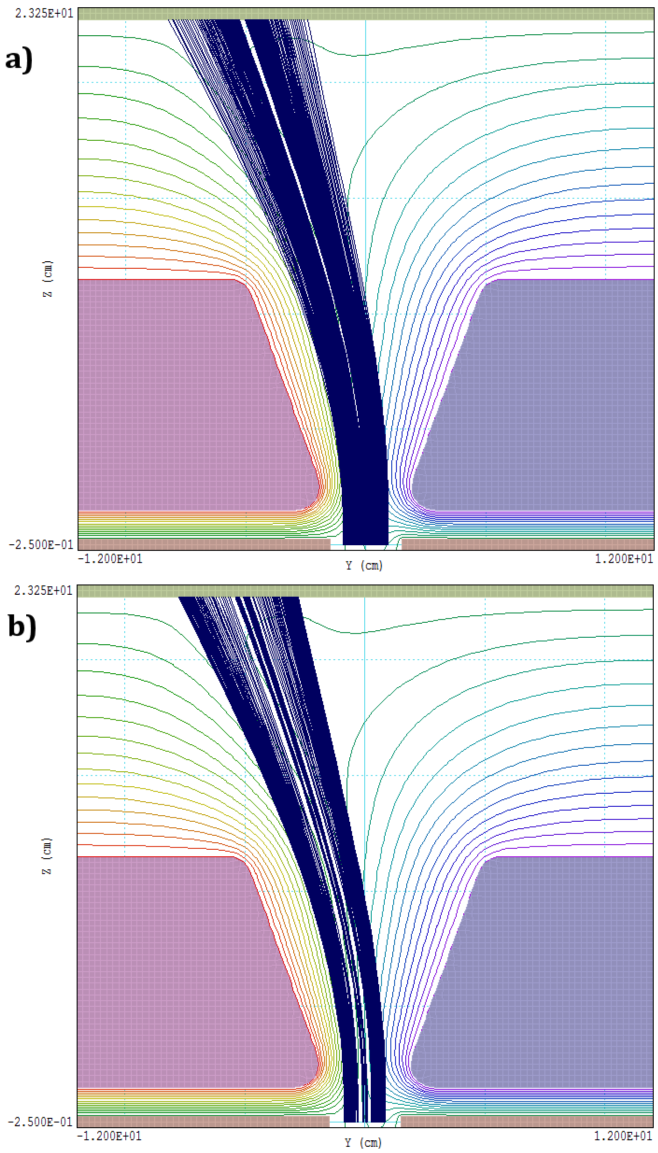

Figure 7: Projection of particle orbits in the x plane with equipotential lines for the total electric field. a) Input distribution derived from a **Trak** calculation with 30 model electrons converted with the **GenDist** `Beam Section Tool`. b) Parametric beam distribution created by **GenDist** from analysis of a **Trak** calculation with 300 model electrons.

near the periphery. On the other hand, all model particles carry the same current in the distribution we create with **GenDist**. Therefore, we need to work with information on beam flux density (not the model particle density) from the **Trak** run. With the current-weighted distribution displayed, we open an output data file, click the `Record` command and close the file to create the following probability data:

Bin	RMin	RMax	p(R)	P(R)	dP(R)/dR
1	8.3379E-04	1.1819E-01	5.4782E-03	5.4782E-03	1.2484E-01
2	1.1819E-01	2.3554E-01	2.9789E-02	3.5267E-02	2.2842E-01
3	2.3554E-01	3.5290E-01	4.9741E-02	8.5008E-02	2.2928E-01
4	3.5290E-01	4.7025E-01	7.6612E-02	1.6162E-01	2.5245E-01
5	4.7025E-01	5.8760E-01	1.1484E-01	2.7646E-01	2.9447E-01
6	5.8760E-01	7.0496E-01	1.7611E-01	4.5258E-01	3.6957E-01
7	7.0496E-01	8.2231E-01	2.3493E-01	6.8750E-01	4.1722E-01
8	8.2231E-01	9.3966E-01	3.1250E-01	1.0000E+00	4.8106E-01

The quantities used to prepare a **GenDist** input file are the outer radius of the bin (R_{max}) and the cumulative probability, $P(R)$. The **GenDist** script has the following content:

```
FileType = PRT
RestMass =   0.0000E+00
Charge =  -1.0000E+00
Energy =   1.0000E+04
Current =   6.0790E-01
Def(Circ) =   0.0000E+00  0.940E+00     30     60
Distribution Random
RDist
  0.1258 0.0055
  0.2507 0.0353
  0.3756 0.0850
  0.5004 0.1616
  0.6253 0.2765
  0.7502 0.4526
  0.8751 0.6875
  1.0000 1.0000
EndFile
```

The first five commands specify an electron beam with 10 keV kinetic energy and total current 0.608 A with output to a file in `PRT` format. The `Def` command calls for a solid circular beam with outer radius 0.940 units where electrons are distributed at 30 positions in the r direction and at 60 uniform intervals in θ (Note that coordinates will be interpreted in cm when input to **OmniTrak** or **GamBet** if $DUnit = 100.0$). The script represents a beam moving parallel to z, but there is option to add convergence or divergence. The `Distribution` and `RDist` commands define the statistics of

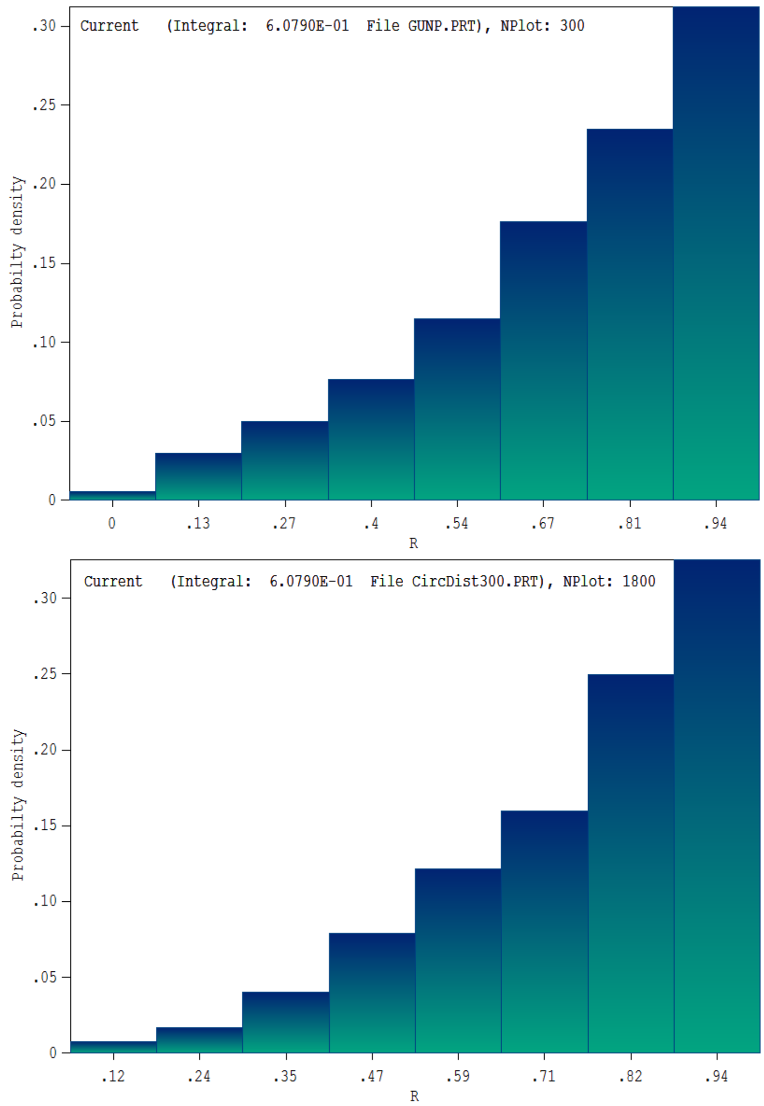

Figure 8: Histograms of the relative radial distribution of beam flux $f(r)$, model electron density weighted by the model particle current. a) Output of a 2D distribution create by a **Trak** run with 300 particles. b) 3D distribution create by **GenDist** with 1800 model particles.

radial particle assignment. The data columns in the `RDist` command are the relative radius (0.0 to 1.0) and the cumulative probability (the fraction on the distribution contained inside the relative radius) derived from the **GenDist** analysis.

The resulting `PRT` file can be used as input to the 3D electrostatic deflector calculation. Figure 7*b* shows the result. The solutions are almost identical except for the beam halo in the solution using the `Beam Section Tool`. The reason is that the distribution constructed from the **GenDist** script did not include the detailed structure of the distribution on the periphery (Fig. 4). To conclude, we can compare the relative advantages of the two conversion methods:

- Use of the `Beam Section Tool` to project a 2D circular beam distribution to 3D preserves many details of the radial distribution. This feature may be important in beam transport calculations because peripheral particles carry a high fraction of the current.

- Creating a distribution approximation with **GenDist** would be a better approach for a Monte Carlo calculation with **GamBet** (*e.g.*, X-ray generation, electron beam welding,...). Here, details of the electron distribution may not be important because electrons undergo large-angle scattering when they enter materials. The inner model particles in a **Trak** calculation carry a relatively small fraction of the current. In a `Beam Section Tool` conversion, small current particles are replicated as many times as the high current particles at large radii. In this case, **GamBet** devotes as much computational effort to particles that make only a small contribution to the dose as to particles that make a large contribution. Creating a distribution where all particles have the same weight can lead to a more efficient Monte Carlo simulation.

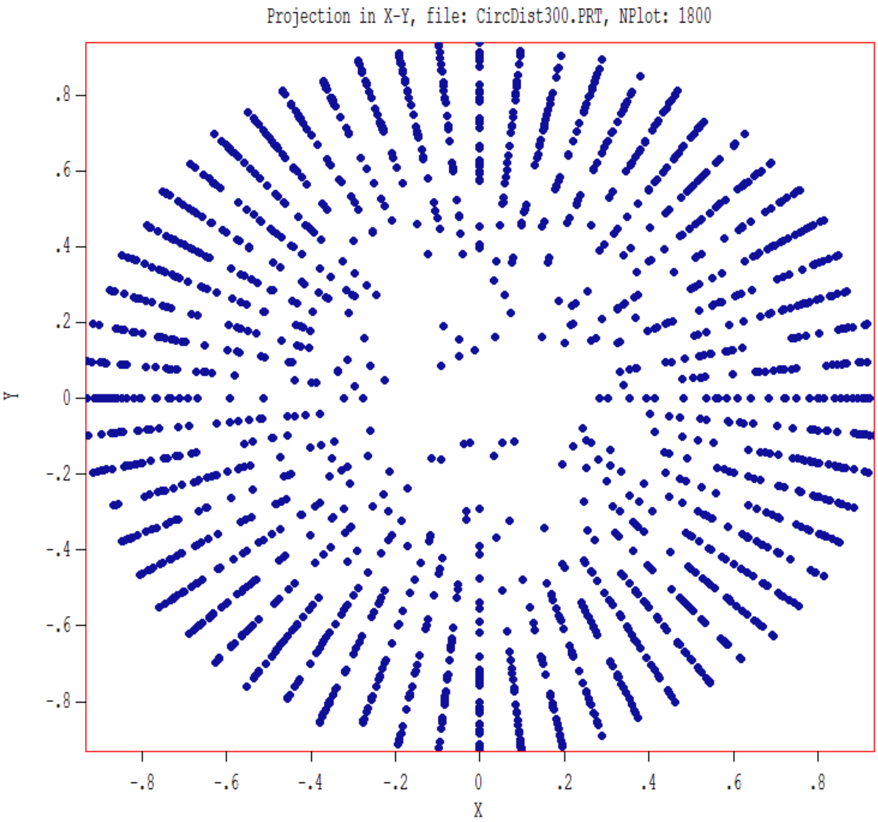

Figure 9: Particle distribution created by the **GenDist** script, plot of particle positions in the x-y plane. Each particle carries a current 3.377×10^{-4} A.