



OmniTrak 1.0
Three-dimensional Charged-particle
Optics and Gun Design

Field Precision
Copyright 2003
Internet: www.fieldp.com
E Mail: techninfo@fieldp.com
PO Box 13595, Albuquerque, New Mexico 87192 U.S.A.
Telephone: 505-220-3975, FAX: 505-294-0222

OmniTrak

Three-dimensional Charged-particle Optics and Gun Design

Table of contents

Chapter 1. Introduction to OmniTrak

- 1.1. Program function
- 1.2. Program requirements and installation
- 1.3. Overview

Chapter 2. Walkthrough example

- 2.1. Geometry and mesh definition
- 2.2. Field and orbit solutions
- 2.3. Analysis and plotting

Chapter 3. OmniTrak structure

- 3.1. Organizing **OmniTrak** simulations
- 3.2. Control script syntax

Chapter 4. Electric fields - Fields section

- 4.1. Loading and modifying **HiPhi** solutions
- 4.2. Controlling electric field recalculation
- 4.3. Program control commands
- 4.4. Loading electric field information from a table

Chapter 5. Applied magnetic fields - Fields section

- 5.1. Uniform field components
- 5.2. Loading and modifying two-dimensional solutions
- 5.3. Loading and modifying **Magnum** solutions
- 5.4. Loading magnetic field information from a table

Chapter 6 Temporal field modulations - Fields section

Chapter 7. Single-particle tracking in the Track mode - Particles section

- 7.1. Starting orbits from a list
- 7.2. Starting orbits from a source surface
- 7.3. Orbit integration controls
- 7.4. Stopping orbits

- 7.5. Setting material properties
- 7.6. Application example

Chapter 8. Self-consistent beam forces in the SCharge mode - Particles section

- 8.1. Beam-generated forces
- 8.2. Control of orbit and field updates
- 8.3. Application example

Chapter 9. Self-consistent Child-law emission - Particles section

- 9.1. Mechanics of the space-charge emission model
- 9.2. Space-charge emission control commands
- 9.3. Application examples
- 9.4. Restarting runs with space-charge and Child-law emission.

Chapter 10. Beam-generated magnetic fields - Particles section

- 10.1. Modeling relativistic beams
- 10.2. Using the relativistic mode
- 10.3. Physical basis of the *RelBeam* tracking mode
- 10.4. Applied fields in the *RelBeam* tracking mode
- 10.5. Material fields in the *RelBeam* tracking mode
- 10.6. Symmetry conditions in the *RelBeam* tracking mode
- 10.7. Commands for the *RelBeam* tracking mode
- 10.8. Application example - planar beam gun.

Chapter 11. Self-consistent field emission - Particles section

Chapter 12. Space-charge-limited emission from plasma surfaces - Particles section

(In preparation)

Chapter 13. Electron secondary emission - Particles section

- 13.1. Theoretical background
- 13.2. Control commands
- 13.3. Benchmark test

Chapter 14. Tracing field lines - Particle section

- 14.1. Commands for field line tracing
- 14.2. Conventions for field line tracing

Chapter 15. Diagnostics section

- 15.1. Field diagnostics
- 15.2.. Particle diagnostics
- 15.3. Particle termination conditions
- 15.4. Field line diagnostics

Chapter 16. Running OmniTrak

- 16.1. Interactive mode
- 16.2. Command line operation
- 16.3. Batch files
- 16.4. Introduction to **GCon**
- 16.5. Running **GCon**

Chapter 17. OmniView - file operations

Chapter 18. OmniView - two-dimensional plots

- 18.1. Setting the view
- 18.2. Setting plot properties
- 18.3. Exporting plots

Chapter 19. OmniView - three-dimensional plots

Chapter 20. OmniView - distribution plots and analysis

- 20.1. Analysis commands
- 20.2. Plot commands
- 20.3. Beam analysis equations

Chapter 21. OmniTrak output files

Chapter 22. Features for ion mobility spectrometry

Appendix 1. Fine-tuning cathode surfaces in MetaMesh

1.1 Program function

OmniTrak is a versatile three-dimensional software suite for charged-particle optics. Applications include electron and ion guns, particle accelerators, ion sources, microwave sources, acceleration columns, electrostatic and magnetostatic lenses, vacuum tubes, and electro-optical devices. The program package includes the **OmniView** postprocessor for analysis of beam distributions and orbit plotting (2D and 3D). **OmniTrak** features include the following:

- Option to combine the effects of calculated electric and magnetic fields on independent 3D meshes.
- Conformal mesh representations to give accurate orbit calculations near shaped surfaces.
- **TDist** utility for automatic creation of input particle distributions.
- Support for self-consistent space-charge-limited emission (Child law) of multiple species from multiple emission surfaces of arbitrary shape.
- Precision interpolation to stopping planes and material surfaces for lens characterization.
- Compatibility with applied magnetic field solutions from the 2D **TriComp** series of programs (**BStat**, **PerMag**, **Pulse** and **Nelson**).
- Self-consistent field emission from multiple sites to model electron microscope sources and vacuum microelectronic devices.
- Option to create zoomed solutions from large-scale field solutions for high accuracy representation of small features.
- Standardized particle file format makes it easy to create input distributions from spreadsheets or experimental data and to incorporate output from previous **Trak** or **OmniTrak** runs.
- Easy definition of emission surfaces using the *COAT* capability in **MetaMesh**.
- Flexible methods to add temporal modulations to fields in the single particle tracking mode.

Some features of the **OmniView** postprocessor include:

- Interactive analysis of beam distributions with extensive plot support
- Two- and three-dimensional orbit plots.
- Particle filtering by charge, mass, position or energy.
- Publication-quality hardcopy with support for all installed Windows print devices.
- Easy porting of results to your own analysis programs

OmniTrak conforms to the standards of the Field Precision **AMaze** series of finite-element programs. The program requires **MetaMesh**, the **AMaze** universal mesh generator and the **HiPhi** program to generate electric field input files. You can also incorporate magnetic field information generated by the **TriComp** two-dimensional programs **BStat**, **PerMag** and **Pulse** as well as the **AMaze** three-dimensional **Magnum** program. Install the other **AMaze** and/or **TriComp** programs before installing **OmniTrak**. You should be familiar with mesh generation and field solution procedures before attempting **OmniTrak** solutions.

1.2 Program installation

OmniTrak runs on Pentium-class personal computers under Linux or any 32 bit version of Windows (95/98/ME/XP/NT/2000). The program utilizes dynamic memory allocation so that mesh size is limited only by the installed memory. We recommend processor speeds greater than 1 GHz and installed memories of 512 MB or higher.

We assume that you have already installed the **AMaze** program launcher (**AMAZE . EXE**), **MetaMesh**, **HiPhi** and/or **Magnum**. If you downloaded the **OmniTrak** programs electronically, unzip the files to a temporary directory. To install the programs, copy the executable files **OMNITRAK . EXE**, **OMNIVIEW . EXE** and **TDIST . EXE** and the file **OMNITRAK . HTML** from the temporary directory or the distribution disk to the directory **\AMAZE** on your hard disk. You may also want to copy the sample input files in the directory **\EXAMPLES** of the distribution

disk to a directory \AMAZE\OMNITRAK\EXAMPLES. Please review licensing regulations for the programs in the file INSTRUCTIONS . TXT.

1.3 Overview

OmniTrak is a powerful tool for charged particle optics. An extensive set of control commands is required to achieve the program's versatility. As a result, this manual is more detailed than those of the other **AMaze** programs. We have tried to make the learning process relatively painless through comprehensive documentation, walkthrough descriptions of application examples, and a library of read-to-run simulations. This section gives a brief overview of the **OmniTrak** solution process. The application example in the following chapter shows how the pieces connect.

The software package contains a PDF instruction manual, an online help file (`omnitrak.html`), ready-to-run examples and the following three programs:

- **OMNITRAK . EXE** – tracks charge particles in a three-dimensional field environment defined by solutions from **HiPhi** and/or **Magnum**.
- **OMNIVIEW . EXE** – generates plots of and performs analyses on field solutions and orbit traces from **OmniTrak**.
- **TDIST . EXE** – utility to create particle input distributions for **OmniTrak**.

The **TDist** utility is described in a separate manual. Use of **OmniTrak** requires **MetaMesh** (the **AMaze** conformal mesh generator) and **HiPhi** for the generation of an input electrostatic solution. The **Magnum** program is required if you want to include effects of three-dimensional magnetic fields.

An **OmniTrak** simulation consists of three steps: 1) generation of conformal meshes for the field solutions, 2) calculation and analysis of the field solutions and 3) orbit tracking and analysis. A run may involve the programs **MetaMesh**, **HiPhi**, **PhiView**, **Magnum**, **MagView**, **TDist**, **OmniTrak** and **OmniView**. Several input files may be necessary for a complete run:

EFieldName.MIN (Input to **Metamesh**, mesh for the **HiPhi** solution)

EFieldName.HIN (Control and material parameters for the **HiPhi** solution)

BFieldName.MIN (Input to **Metamesh**, mesh for the **Magnum** solution)

BFieldName.HIN (Control and material parameters for the **Magnum** solution)

CoilName.CDF (Applied currents for the **Magnum** solution)

ParticleName.PRT (Input starting parameters for the **OmniTrak** solution)

TrackName.OIN (Control parameters for the **OmniTrak** solution)

The prefixes in the list represent arbitrary names of up to 20 characters. The suffixes refer to the file function and must have the forms shown.

In an **OmniTrak** solution there are three basic tasks: 1) loading and modifying field solutions, 2) defining particle starting parameters and tracking orbits, and 3) performing diagnostics. The tasks are reflected in the structure of the **OmniTrak** control script (TrackName.OIN), which has three sections: *FIELDS*, *PARTICLES* and *DIAGNOSTICS*. Within the *PARTICLES* section there are six possible run modes:

TRACK

In this mode **OmniTrak** performs single-particle tracking with negligible contribution from beam-generated fields. Particles can be initiated from a list of input parameters and/or automatically generated from one or marked emission surfaces. Lists that can be prepared with an editor, spreadsheet, the **TDist** utility or a previous Trak or OmniTrak run. The mode is useful for low-current devices.

SCHARGE

Orbit tracking with self-consistent effects of beam space-charge. Particles can be initiated from a list of input parameters and/or automatically generated from one or marked emission surfaces. The

current of particles from emission surfaces is assigned to satisfy the condition of space-charge-limited emission (Child law). The program must update the electric field solution to reflect the presence of beam space-charge. The mode has application to high-current beam transport as well as low-voltage electron guns and ion guns with thermionic emitters.

RELBEAM

Orbit tracking with self-consistent effects of both beam charge and current. Particles can be initiated from a list and/or automatically generated from one or more marked Child-law emission surfaces . In *RelBeam* calculations **OmniTrak** must update both the electric and magnetic field solutions. Application to gun and transport systems for high-current, high-voltage electron beams.

FEMIT

Start particles from a list or electrons from an emission surface. The current of emitted electrons is assigned to satisfy conditions of the Fowler-Nordheim equation for field emission. The program must update the electric field solution.

PLASMA

Start particles from a list or marked emission surfaces,. In the latter case the program chooses currents to satisfy conditions for space-charge-limited emission on a plasma surface. The iteration cycle includes adjustment of the plasma surface to achieve a condition of uniform ion flux. This mode applies to extraction of high-current ion beams from magnetic-field-free sources.

FLINE

Trace electron or magnetic field lines in three-dimensional solutions. **OmniTrak** generates a file of coordinates so that 2D or 3D field-line plots can be created in **OmniView**. The program can also list values of field components along the lines.

Chapter 2 gives a detailed description of an **OmniTrak** application. You can learn many of the features of the code by running the example using the input files supplied in the example library. Chapters 3 through 15 cover preparation of the **OmniTrak** control script while Chapter 16 describes options to run **OMNITRAK . EXE**. Chapters 17 through 20 detail features of the **OmniView** postprocessor for the analysis of output files from **OmniTrak**. Finally, Chapter 21 describes the format of **OmniTrak** output files to help you create interfaces for your own analysis programs.

This manual concentrates on procedures to run **OmniTrak** and does not give detailed explanations of the physics of charged-particle beams. A comprehensive treatment of charged particle optics is given in two texts by the program author that are supplied in PDF format with the code:

S. Humphries, Jr., Principles of **Charged Particle Acceleration** (Wiley, New York, 1986)

S. Humphries, Jr., **Charged Particle Beams** (Wiley, New York, 1990).

The latter work discusses beam distributions, emittance, beam-generated fields, propagation of high-current beams, the design of electron and ion guns and other topics. Principles of finite-element electric and magnetic field solutions are reviewed in

S. Humphries, Jr., **Field Solutions on Computers** (CRC Press, Boca Raton, 1997).

This book is available from CRC Press.

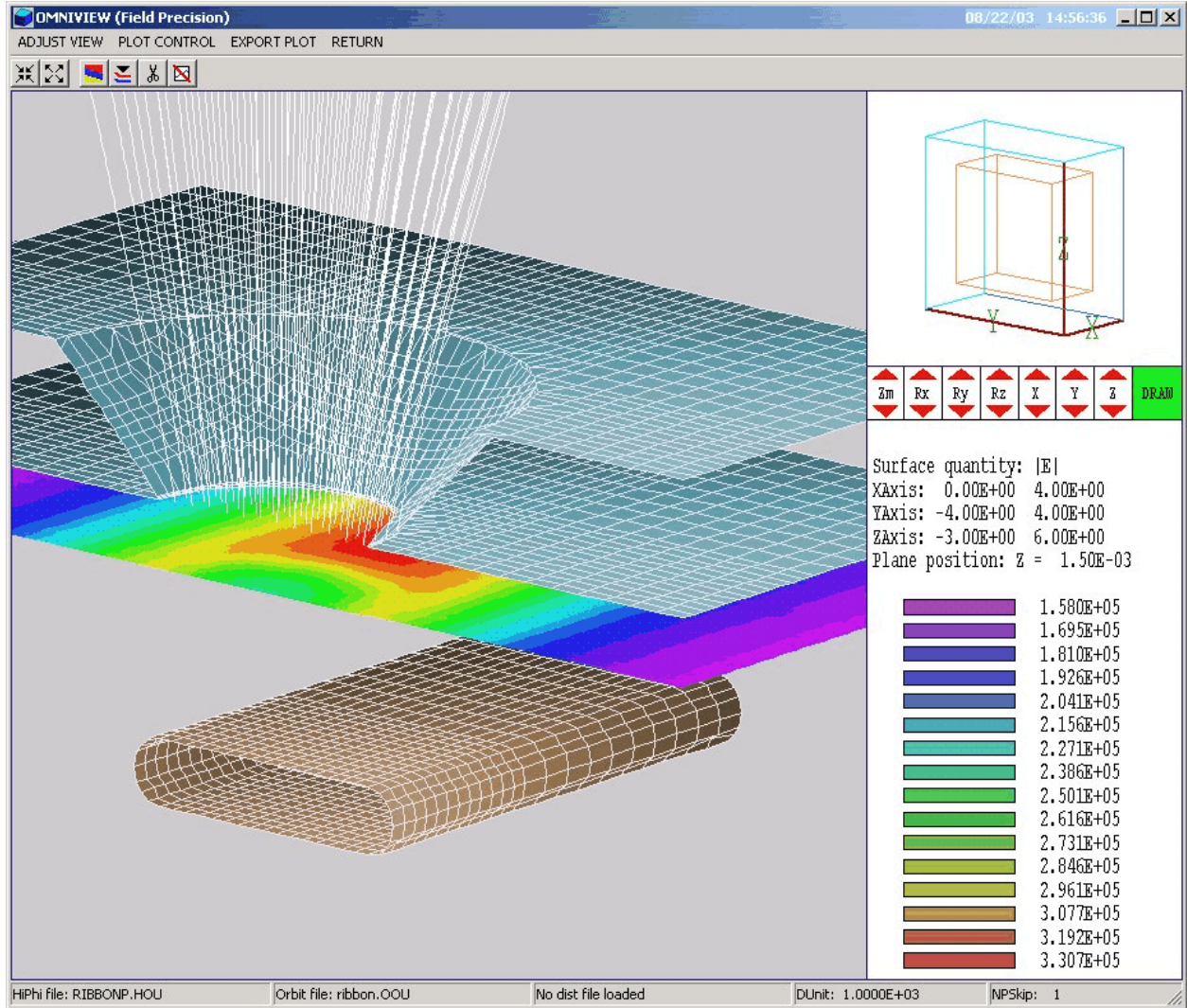


Figure 2.1. Working environment of the OmniView postprocessor for OmniTrak

Chapter 2. Walkthrough example

2.1. Geometry and mesh definition

This chapter reviews an application example to give a quick overview of **OmniTrak** procedures and advanced features. Figure 2.1 shows the geometry – electrons from a ribbon cathode are extracted through a circular aperture. The example is a good exercise of the three-dimensional capabilities of the code. We want to find space-charge limits on current and the distribution of the extracted beam. We shall assume uniform

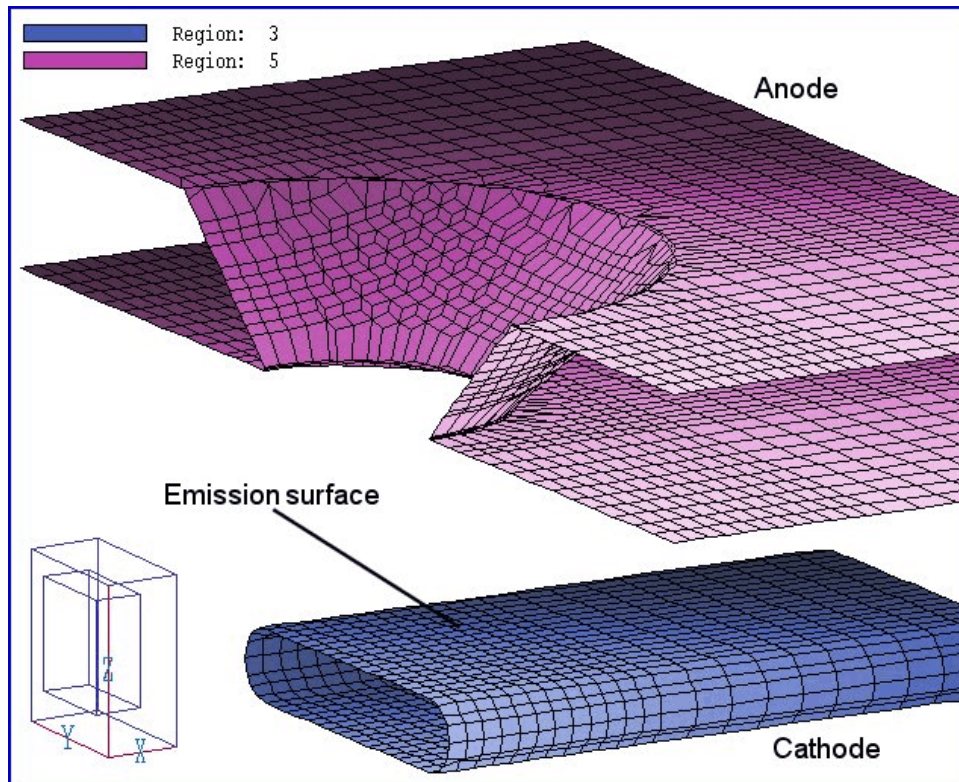


Figure 2.2. Anode and cathode shapes – ribbon example.

emission along the length of the ribbon in the solution volume. The flat face of the ribbon has width 2.0 mm and the aperture has a radius 1.0 mm. The distance between the emission face of the ribbon and the adjacent anode surface is 1.75 mm. We shall use a symmetry boundary at $x = 0.0$ but treat the full width of the system in y . This choice gives flexibility to model effects of cathode misalignments. Neumann boundaries at $y = \pm 4.0$ mm and $x = +4.0$ mm approximate the effect of extended electrodes.

Move the input files for the example (`ribbon.min`, `ribbon.hin`, `ribbon.oin` and `ribbon.gcn`) to a working directory (*i.e.*, `\amaze\buffer`). The files control the programs **MetaMesh**, **HiPhi**, **OmniTrak** and **GCon**. Run **MetaMesh** and load `ribbon.min`. The file contents are listed in Table 2.1. This example illustrates several useful MetaMesh techniques and the special features required for an **OmniTrak** simulation.

- The foundation mesh is created with variable resolution to concentrate elements in x - y in the region of beam extraction and in z near the acceleration gap.

Table 2.1. MetaMesh input file ribbon.min

```

* File RIBBON.MIN
Global
  XMesh
    0.00 1.50 0.125
    1.50 4.00 0.250
  End
  YMesh
    -4.00 -1.50 0.250
    -1.50 1.50 0.125
    1.50 4.00 0.250
  End
  ZMesh
    -3.00 -1.00 0.250
    -1.00 3.00 0.125
    3.00 6.00 0.250
  End
  RegName 1 Vacuum upper
  RegName 2 Vacuum lower
  RegName 3 Ribbon cathode
  RegName 4 Emission surface
  RegName 5 Extractor anode
  RegName 6 Lower boundary
  RegName 7 Upper boundary
End
* Solution volume
* Lower solution volume
Part 1
  Type Box
  Region 1
  Fab 8.00 8.00 6.00
  Shift 0.00 0.00 -3.00
End
* Lower solution volume
Part 2
  Type Box
  Region 2
  Fab 8.00 8.00 6.00
  Shift 0.00 0.00 3.00
End

* Ribbon cathode
Part 3
  Type Extrusion
  L -1.00 -0.25 1.00 -0.25 S
  A 1.00 -0.25 1.25 0.00 1.0 0.0 S
  A 1.25 0.00 1.00 0.25 1.0 0.0 S
  L 1.00 0.25 -1.00 0.25 S
  A -1.00 0.25 -1.25 0.00 -1.0 0.0 S
  A -1.25 0.00 -1.00 -0.25 -1.0 0.0 S
  End
Region 3
  Fab 10.00
  Rotate 90.0 90.0 0.0 YXZ
  Surface Region 1 0.98
  Surface Region 2 0.98
  Coat 2 4
End
* Extractor anode
Part 4
  Region 5
  Type Turning
  L 2.00 6.00 2.00 1.00 SE
  L 2.00 1.00 3.00 2.00 SE
  L 3.00 2.00 3.00 6.00 S
  L 3.00 6.00 2.00 6.00 S
  End
  Fab 0.0 360.0
  Surface Region 2 Edge
End
* Lower boundary
Part 5
  Type BoundZDn
  Region 6
  End
* Upper boundary
Part 6
  Type BoundZUp
  Region 7
  End
EndFile

```

■ Unfilled regions of the types *BoundZUp* and *BoundZDn* are used to set nodes on the top and bottom surfaces of the solution volume. The nodes assume fixed-potential values in the **HiPhi** solution.

■ The anode with a shaped extraction aperture is a *Turning*. The complex three-dimensional structure requires only a few data lines for its definition. The ribbon cathode is an extrusion. Here, rotation operations are applied so that the long axis extends along *x*. We could

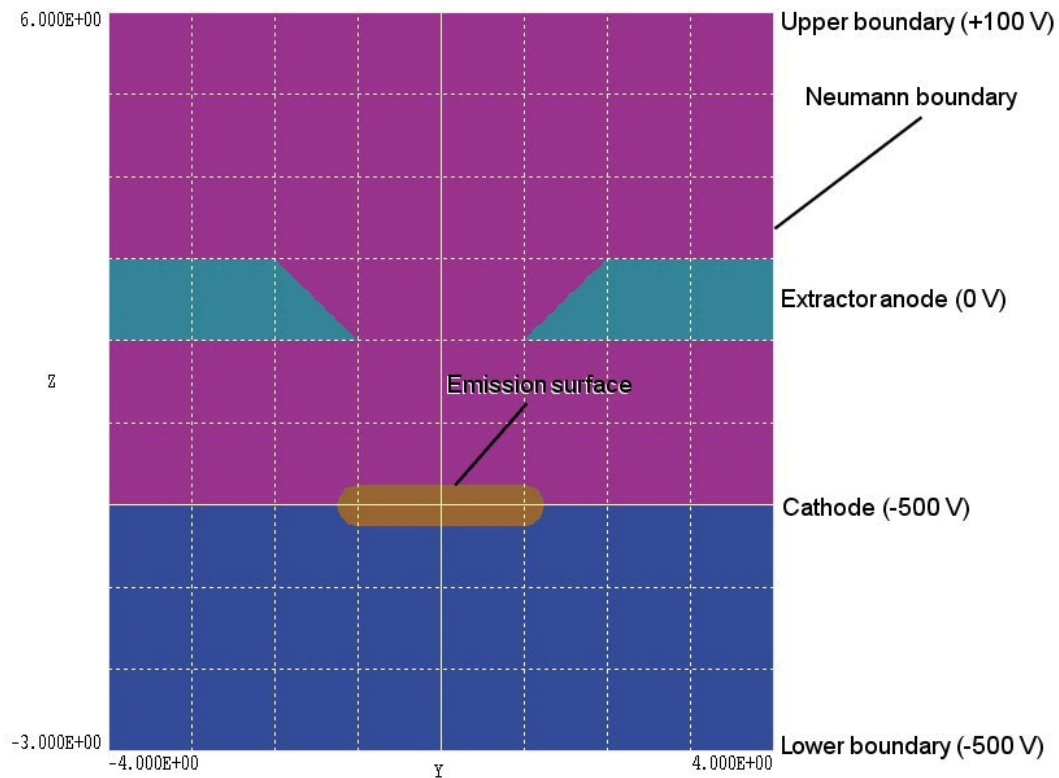


Figure 2.3. Cross-section of the geometry of example `ribbon.min` in a plane normal to x at $x = 0.0$. Lower vacuum volume: region 1 (blue). Upper vacuum volume: region 2 (violet). Cathode: region 3 (orange). Anode: region 5 (green).

add a non-zero y displacement in the *Shift* command to investigate cathode misalignment.

■ To implement automatic generation of electrons in **OmniTrak** we must define an *emission surface*, a region of nodes on the surface on an electrode with a unique region number. In the **OmniTrak** run, all element facets whose four nodes have the designated region number are treated as sites for electron initiation. Chapter 7 describes the process in detail. In the present example, we shall define the upper surface of the ribbon cathode as an emission region. Figure 2.3 shows a cross-section of the system in a plane normal to x at position $x = 0.0$. Note that the vacuum volume has been divided into upper and lower regions in z with a boundary at the cathode midplane. In response to the statement `Coat 2 4` in the definition of the cathode (Part 3), the program checks all facets that border on Region 2 (upper vacuum volume) and changes the associated nodes to Region 4. The result is that all facets on the top of the cathode will be treated as emission facets.

Process and save the mesh following instructions in the **MetaMesh** manual. You can use the plot capabilities of **MetaMesh** to confirm features discussed in this section.

2.2. Field and orbit solutions

The next step is to run **HiPhi** to generate a baseline electrostatic solution. The short input file `ribbon.hin` sets Regions 1 and 2 to the vacuum condition and defines the potential values shown in Fig. 2.3. The file also specifies that mesh input coordinates are given in millimeters. The solution takes about 21 seconds on a 1.5 GHz computer and generates the output file `ribbon.hou`. We can now proceed to the **OmniTrak** simulation. You can start the program from the **AMaze** program launcher (`amaze.exe`). Click on *Edit input files* in the *Files* menu and pick `ribbon.o.in` to show the content listed in Table 2.2. Commands enclosed in the *Fields* section define properties of electric and/or magnetic fields. The *DUnit* command specifies that spatial quantities in succeeding statements should be converted from millimeters. The command *EField3D* loads the file `ribbon.hou` as the three-dimensional electrostatic solution. This solution will be modified in **OmniTrak** to reflect the presence of beam space charge. We use default values for most of the field relaxation parameters. The line

```
NCheck E 5
```

specifies frequent checks of convergence to avoid wasted time during latter relaxation cycles when the solution has only small changes..

Commands of the *Particles* section control tracking of particle orbits and space-charge updates of the field solution. Chapter 10 gives a detailed discussion of the methods used to simulate self-consistent Child law emission. The *Dt* command explicitly sets the orbit integration time step. The value is chosen so that maximum-velocity electrons travel less than the width of one element in a time step. The *NCycle* command specifies twelve cycles of orbit tracking and field updates while the command *AvgE* controls space-charge averaging between cycles. The parameters were chosen to achieve good solution convergence for the specific problem. The *Material* commands set properties of elements in Regions 3 (cathode) and 5 (anode). Electron orbits are terminated when they enter a material element. The command

Table 2.2. OmniTrak input file ribbon.oin

```
* File RIBBON.OIN
FIELDS
  DUnit = 1000.0
  EField3D = RIBBON.HOU
  NCheck E 5
END
PARTICLES SCharge
  NCycle 12
  Emit 4 0.0 -1.0 0.10 2
  Material E 3
  Material E 5
  AvgE 0.40
  Dt 2.5E-12
END
DIAGNOSTICS
  EDump RIBBONP.HOU
  PTable RIBBON
END
ENDFILE
```

```
Emit 4 0.0 -1.0 0.10 2
```

states that element facets bounded by nodes with region number 4 are emission facets. The other parameters in the command line have the following functions: 1) set the mass and charge of emitted particles to correspond to electrons, 2) set the distance between the emission boundary and the physical cathode surface to 0.10 mm, and 3) set the number of electrons per facet to 2^2 . The *Diagnostics* section controls the creation of output files.

Exit the editor and choose *Start run* from *Run* menu. Processing starts automatically and takes about 230 seconds on a 1.5 GHz computer. During the simulation **OmniTrak** performs several activities: 1) read the electrostatic input file and create coupling coefficients for the field update, 2) identify emission facets and generate 2112 particle starting points at the emission surface, and 3) track orbits and recalculate the field solution over multiple cycles. During the run the program creates four output files:

ribbon.ols

The file contains an extensive listing of useful and not-so-useful quantitative information. You can inspect the file with a text editor or abstract information using the **GCon** utility.

ribbon.ooa

This file is a text listing of orbit vectors calculated on the final cycle. You can load the file into **OmniView** to create plots or to transfer the data to your own analysis programs.

ribbonp.hou

This binary file in standard **HiPhi** output format is generated in response to the *EDump* command. It contains information on modified electrostatic fields including contributions from beam space charge. You can create field plots from the file with **PhiView** or **OmniView**. You can also reload the file into an **OmniTrak** simulation to trace single particle orbits in the self-consistent fields.

ribbon.prt

This text file contains particle parameters at the orbit termination points calculated on the final cycle. The file can be used as input for subsequent **OmniTrak** runs, it can be loaded in **OmniView** for distribution analyses or it can be ported to your own analysis programs.

In the remainder of this section, we will review features of the solution and analysis tools in the **OmniTrak** package. First, we shall extract data from the listing file to determine the convergence history of the run and the final value of total current emitted from the ribbon. Launch **GGon** from the **AMaze** program launcher. Choose *Run GCon script* and pick the file `ribbon.gcn`. The file content is shown in Table 2.3.

Table 2.3. GCon input file ribbon.gcn

```

ABSTRACT
  INDATA \SIMULATIONS\AMAZE\BUFFER\RIBBON.OLS
  OUTDATA \SIMULATIONS\AMAZE\BUFFER\RIBBONCURRENT.DAT
  ITEM Total emitted current
ENDABSTRACT

```

GCon opens the file `ribbon.ols` (1.7 MB in length) and extracts each line that contains the string *Total emitted current*. You can use the *Edit files* command in the *Files* menu of **GCon** to check the resulting data file `ribboncurrent.dat`. It has the following content:

```
Total emitted current: 1.3594E-01
Total emitted current: 1.2515E-01
Total emitted current: 1.7510E-01
Total emitted current: 1.9928E-01
Total emitted current: 1.9264E-01
Total emitted current: 1.4928E-01
Total emitted current: 1.5194E-01
Total emitted current: 1.5142E-01
Total emitted current: 1.5148E-01
Total emitted current: 1.5152E-01
Total emitted current: 1.5156E-01
```

The results show that there is good convergence of the emitted current between cycles 7 and 11.

2.3. Analysis and plotting

Our final task is to use the **OmniView** postprocessor to make plots and to perform quantitative analyses of the solution. Run the program from the **AMaze** program launcher and choose *Load OmniTrak dist file* from the *Files* menu. Pick the file `ribbon.prt` and click *OK*. The *Distributions* menu becomes active and the status bar at the bottom of the window is updated. Go to the *Distributions* menu. By default, the program shows a plot of final particle positions in the x - y plane (at all values of z). The display is confusing because the positions of extracted particles (that travel through the aperture and travel to the boundary at $z = 6.0$ mm) are superimposed on those of electrons that strike the anode plate near $z = 2.0$ cm. To improve the display, click on the *Set particle filter* command in the *Analysis* menu. The default setting is that all orbits are included, regardless of charge, mass, final position or final energy. Set the filter entry *ZMin* to 5.0, thereby eliminating electrons that strike the anode. You will see the display in the background of Fig. 2.4 showing a slightly elliptical beam. To get quantitative information about the filtered particles, click on the *Beam analysis command* in the *Analysis* menu. **OmniTrak** displays the data window shown in Fig. 2.4 that reports calculated beam quantities relative to a reference axis. In this case, the program uses the default z axis as the reference. Note that the total extracted current is 0.011 A, less than 10 per cent of the current emitted from the ribbon. The extracted current for a full circular aperture is 0.022 A.

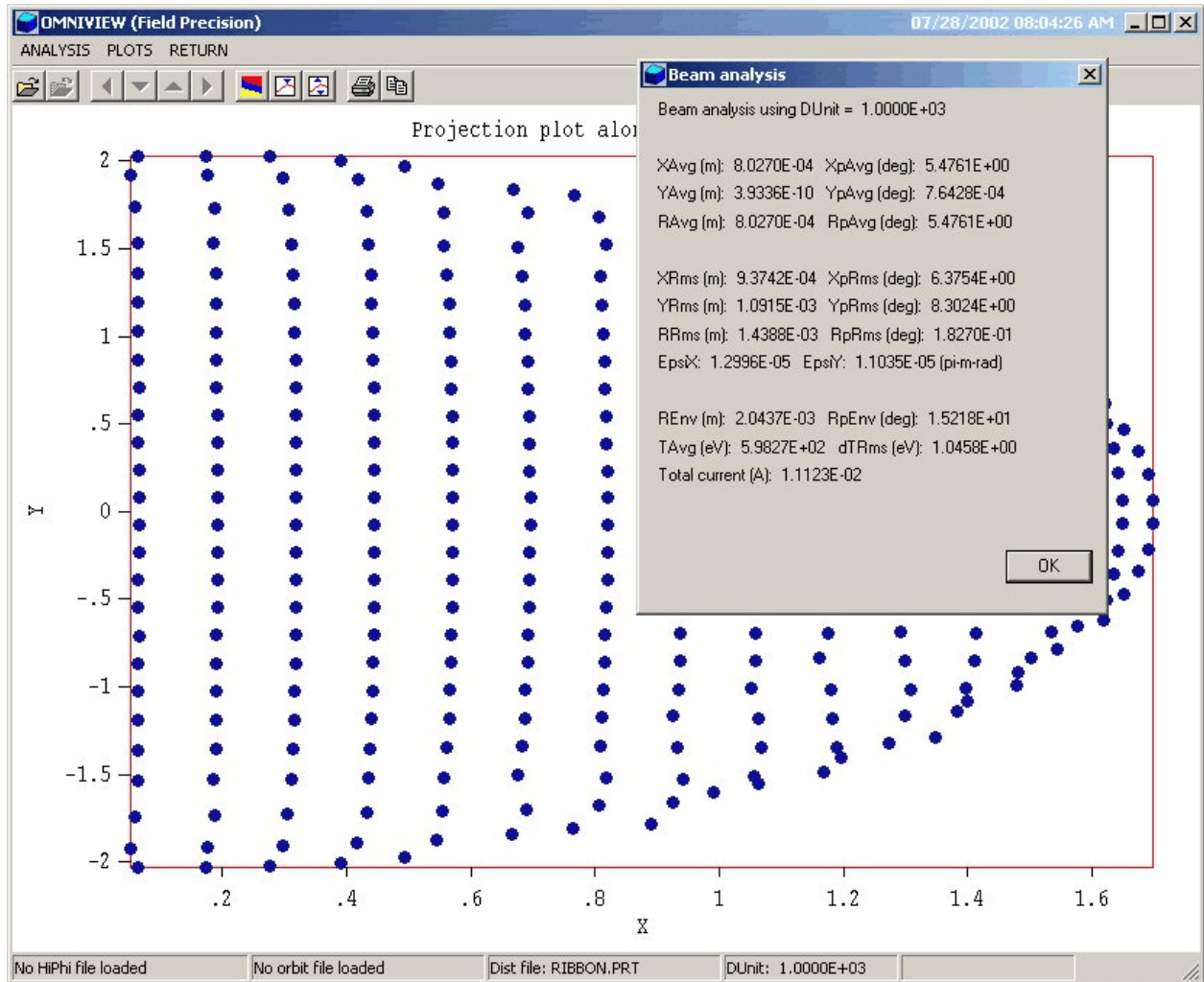


Figure 2.4. Distribution of extracted particles in the x - y plane for the *ribbon* example with beam analysis information window.

OmniTrak can create two- and three-dimensional plots of orbits superimposed on the three-dimensional electric field solution. Return to the main menu and click on the command *Load HiPhi solution file* in the *File operations* menu. Pick the file *ribbonp.hou*, the electrostatic solution with space-charge effects. Go to the *2D plots* menu. **OmniTrak** makes a default equipotential contour plot normal to the z -axis at the midpoint of the solution volume. In the *Change view* menu, click on the command *Set slice plane properties*. In the dialog, activate X in the *Normal axis* group and move the *Position along axis* slide all the way to the left. You will see the equipotential plot of Fig. 2.5. We shall now add projected particle orbits. Return to the main menu and click on the *Load OmniTrak plot file* command in the *File operations* menu. Pick the file

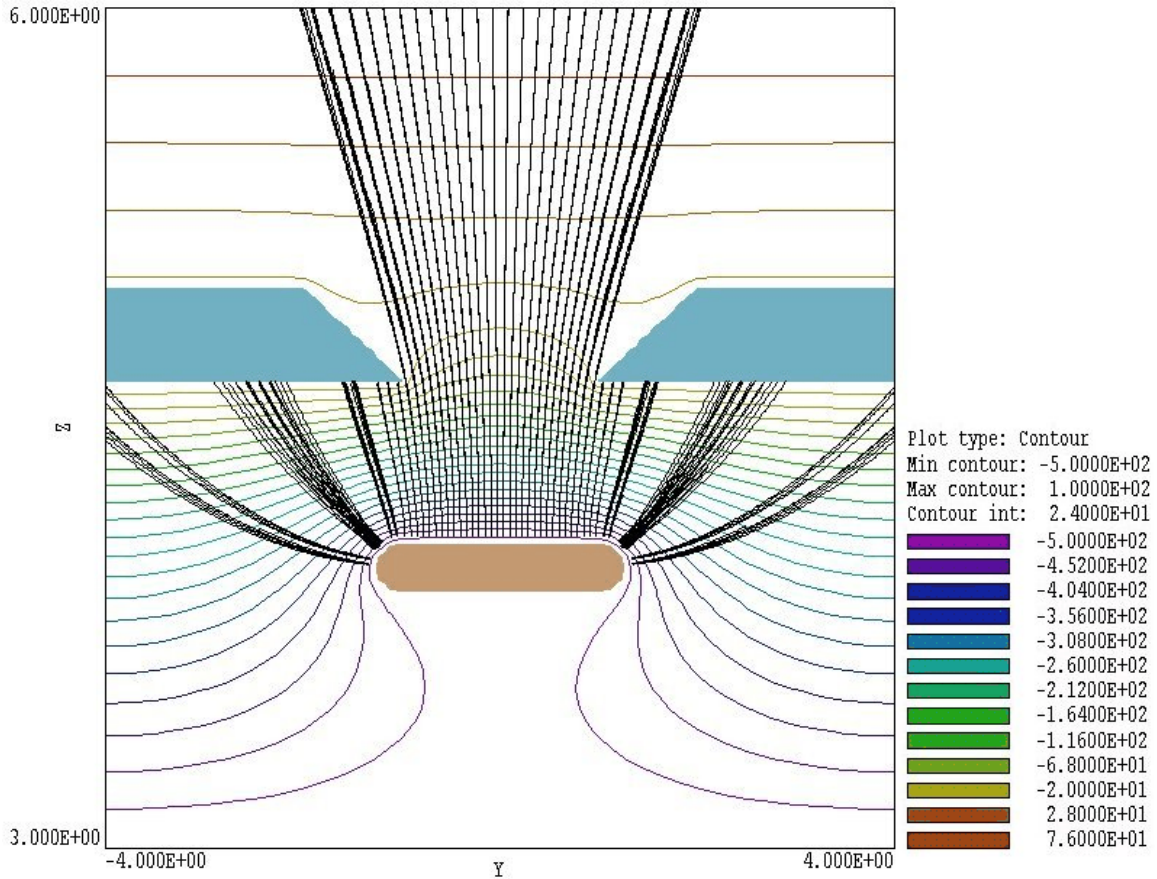


Figure 2.5. Two-dimensional plot of electrostatic equipotential contours and projected orbits, example *ribbon*.

`ribbon.000` and then return to the *2D plots* menu. Initially there are no plotted orbit vectors because the particle filter is still active. In the *Plot control* menu click on *Set particle filter*. Deactivate the present setting with the *Remove filters* button. Although the field plot shows information near $x = 0.0$, the superimposed orbit plot will show vector projections in the y - z plane from all x positions. To avoid a confusing display, define a new filter by setting $x_{\max} = 0.2$. The orbit plot will therefore be limited to vectors near $x = 0.0$.

After you click *OK* to exit the dialog, the program updates the plot. Note the only a few orbits are plotted. By default, **OmniTrak** includes only a fraction of the available orbits in large sets to save time during setup. To plot all the orbits, click on the *Set NPSkip* command in the *Plot control* menu. In the dialog, change the value of *NPSkip* (number of particles to

skip between plots) to 1. After you click *OK* you should see the display of Fig. 2.5. The plot clearly shows how the combined effects of the beam-space charge and the negative lens effect at the aperture, leading to a diverging extracted beam. To create a record of the plot, you can click on the *Default printer* command of the *Export plot* menu. A hardcopy plot will be created on the default Windows printer device. **OmniTrak** also has three-dimensional plotting capabilities as shown in Fig. 2.1. Chapter 20 discusses procedures to generate such plots.

Chapter 3. OmniTrak run structure

3.1. Organizing OmniTrak simulations

OmniTrak simulations require **MetaMesh** (the universal mesh generator for the **AMaze** series of programs) and **HiPhi** to prepare baseline electrostatic solutions. The **Magnum** program is required if you want to include effects of three-dimensional magnetostatic fields. **OmniTrak** can also incorporate information on two-dimensional applied magnetic fields from files generated by the **TriComp BStat**, **PerMag**, **Pulse** or **Nelson** programs. The package consists of two programs. **OmniTrak** handles particle generation, orbit tracking and self-consistent field updates. **OmniView** is a dedicated post-processor for plotting and analyzing results from **OmniTrak**. The package also includes the **TDist** utility to generate input particle distributions. The **OmniTrak** program can run in two modes: interactively in a window or autonomously in the background under batch file control. Similarly, **OmniView** can run as an interactive application under user control or autonomously under script file control. The autonomous modes allow automatic processing of large or repetitive data sets.

Two types of input files are required for all **OmniTrak** solutions:

- A **HiPhi** (FPrefix1.HOU) and/or **Magnum** (FPrefix2.GOU) output file that describes the conformal hexahedron mesh and supplies values to calculate electric and/or magnetic fields.
- A script (FPrefix.OIN) that sets control parameters and initial properties of particle orbits.

Other input files that may appear in specific simulations are listed below. An **OmniTrak** simulation may consist of the following steps:

- Prepare a **MetaMesh** script (NameE.MIN) that defines the spatial distribution of regions for the electrostatic solution.
- Prepare a **MetaMesh** script (NameB.MIN) for a the magnetostatic solution.

- Run **MetaMesh** to create the files NameE . MDF and NameB . MDF of standard mesh information.
- Prepare a **HiPhi** script (NameE . HIN) that sets control parameters and defines the material properties of regions for the electrostatic solution.
- Run **HiPhi** to create a solution file NameE . HOU. This file contains information on node spatial coordinates and the applied electrostatic potential ϕ .
- Prepare a **Magnum** script (NameB . MIN) that sets control parameters and defines the material properties of regions for the applied magnetostatic solution.
- Run **Magnum** to create a solution file NameB . HOU. This file contains information on node spatial coordinates, applied magnetic field H_s , and the magnetostatic scalar potentials ϕ and ψ .
- (*Optional*) Depending on the type of run, prepare a list of starting particle parameters using a text editor, spreadsheet or the **TDist** utility (NameP . PRT).
- Prepare a script (Name . OIN) to control the **OmniTrak** run. The script controls loading and modification of fields, initiation of particle orbits, orbit racking, field updates and diagnostic.
- Run **OmniTrak** to create the files Name . OLS (listing of run parameters and particle data), Name . OOU (vector file for orbit plotting) and/or NameP . PRT (optional listing of final particle parameters).

The sequence of discrete operations may seem complex, but the approach is effective in the long run. You can check the validity of the solution at each stage. Furthermore, the same field files (NameE . HOU or NameB . MOU) could be used for several **OmniTrak** runs. Table 3.1 summarizes the input and output file that may be used in **OmniTrak**.

| Table 3.1. OmniTrak files | |
|----------------------------------|--|
| Name form | Function |
| NameE . MIN | Geometry of electric materials, input to MetaMesh |
| NameE . MDF | Output from MetaMesh , input to HiPhi |
| NameE . HIN | Control and description of electrical materials, input to HiPhi |
| NameE . HOU | Output from HiPhi , input to OmniTrak |
| NameB . MIN | Geometry of magnetic materials, input to MetaMesh |
| NameB . MDF | Output from MetaMesh , input to Magnum |
| NameC . CDF | Geometry of drive coils, input to Chopper |
| NameC . FSC | Output from Chopper , input to Magnum |
| NameB . MIN | Control and description of magnetic materials, input to Magnum |
| NameB . MOU | Output from Magnum , input to OmniTrak |
| Name . OIN | Program control script, input to OmniTrak |
| NameP . PRT | Listing of initial particle parameters, input to OmniTrak |
| Name . OLS | Run parameters and diagnostics, output from OmniTrak |
| Name . OOU | Orbit vectors, output from OmniTrak |
| NamePP . PRT | Final particle parameters, output from OmniTrak |

3.2. Control script syntax

The input script for **OmniTrak** is a text file with data lines containing commands and parameters. The script has the following general format:

```
FIELDS
  (Field commands)
END

PARTICLES [Option]
  (Particle commands)
END

DIAGNOSTICS
  (Diagnostic commands)
END

ENDFILE
```

The script is divided into three groups: *Fields*, *Particles* and *Diagnostics*. The groups must appear the order shown. Commands are valid only within their group. In most cases the commands of a group may appear in any order. **OmniTrak** reads and analyzes all commands in a group before initiating operations. Following chapters give detailed descriptions of allowed commands for the three groups.

OmniTrak makes no distinction between upper and lower case. Entries on a line can be separated by the following delimiters

```
Space, blank
Comma [,]
Tab
Colon [:]
Equal sign [=]
Left parenthesis [(]
Right parenthesis [)]
```

You can use any number of delimiters in a line. This means that you can add indentations and customize the appearance of the script. For example, the two lines

```
MARKREG 4 0.0 -1.0 0.10 2
MarkReg (4) = (0.0, -1.0, 0.10, 2)
```

have the same effect.

OmniTrak ignores blank lines and comment lines. Comment lines begin with the symbol [*] (asterisk). Most parameters are real numbers. The following formats are valid.

```
1.000
5.67E6
6.8845E+09
5
```

The final number is interpreted as 5.0.

The following example illustrates a complete script for **OmniTrak**:

```
* File DETECTOR.OIN
FIELDS
  DUnit = 39.37
  EField3D = DETECTOR.HOU
END
PARTICLES Track
  Dt = 1.2E-10
  TMax = 8.0E-9
  Material(E) = 1
  Material(E) = 3
  Material(E) = 4
  Material(E) = 5
  StopPlane XUp 4.0
  PList
*   Mass Chrg   Eng      x      y      z      px      py      pz
    4.0  2.0  3.52E+06  0.01  0.0  0.001  0.9397 -0.3420  0.0
    4.0  2.0  3.52E+06  0.01  0.0  0.001  0.9455 -0.3256  0.0
    4.0  2.0  3.52E+06  0.01  0.0  0.001  0.9511 -0.3090  0.0
  End
END
DIAGNOSTICS
  PartList Detector
END
ENDFILE
```

You can place any amount of text in any format after the *EndFile* command. Therefore you can add extended documentation that can be helpful when you return to a simulation.

4. Electric field commands - FIELDS section

4.1. Loading and modifying HiPhi solutions

The first step in a **OmniTrak** run is to load electric and/or magnetic field information by reading one or more **AMaze** solution program files. The associated meshes are used for field interpolations and to determine stopping conditions when a particle strikes a material object. This section introduces commands to load and to modify electric field solutions from **HiPhi**. Commands are displayed with symbolic parameters and also in a form in which they may appear in the script.

EFIELD3D = FileName [FMultE3D]

EFIELD3D = KlyGun.HOU

This command loads an electric field solution from **HiPhi**. The quantity *FileName* is the full name of the file (i.e., EFTEST.HOU). The optional real-number parameter *FMultE3D* is a field scaling factor. Values of electrostatic potential ϕ are multiplied by *FMultE3D* when the file is loaded. The default value is *FMultE3D* = 1.0. The **HiPhi** file must be available in the working directory.

SHIFT E XShift YShift ZShift

SHIFT E = (-5.00, 2.50, 0.00)

This command shifts node positions in the 3D electric field mesh or table according to

$$x_{\text{new}} = x_{\text{old}} + XShift.$$

$$y_{\text{new}} = y_{\text{old}} + YShift.$$

$$z_{\text{new}} = z_{\text{old}} + ZShift.$$

Enter the real-number parameters *XShift*, *YShift* and *ZShift* in the units set by the *DUnit* command (see Section 4.3). This command may be useful to shift an electric field mesh to match a magnetic field mesh or to do a piecewise simulation of a periodic system. Shifts are performed after rotations. Electric field shifts are valid only in the *TRACK* orbit tracing mode. Shifts affect only values used within the code and are not included in a field file created by the *EDUMP* command.

ROTATE E XRot YRot ZRot [RotOrder]
ROTATE(E) = 90.0 0.0 45.0 ZX

This command rotates the coordinates and field components of the 3D electric field mesh or table. The parameters *XRot*, *YRot* and *ZRot* are rotation angles (in degrees) about the current *x*, *y* or *z* axes. The optional parameter *RotOrder* gives the order in which rotations should be performed. *RotOrder* contains from one to three of the characters *X*, *Y* or *Z*. The default is *RotOrder* = 'XYZ'. In the example above, the field solution is rotated 45° about the *z* axis and then 90° about the new *x* axis. The value *YRot* is ignored. Rotations are performed before shifts. Electric field rotations are value only in the *TRACK* orbit-tracing mode. Rotations affect values used within the code and are not included in a field file created by the *EDUMP* command.

CHANGE POT(RegNo) = PotNew
CHANGE POT(6) = 25000.0

This command can be used to adjust potentials on individual electrodes. The quantity *RegNo* (integer) corresponds to the number of a region defined in **MetaMesh**. The region must have the fixed-potential condition (set by the *Potential* command in **HiPhi**). The quantity *PotNew* (real) is the new value of potential (in volts). Several *ChangePot* commands may appear in the file. After reading the complete *Fields* section, **OmniTrak** performs an iterative field solution using the new values of potential. Commands in the next section control the relaxation process.

4.2. Controlling electric field calculations

OmniTrak must update the electric field solution when the *ChangePot* command appears in the control script or if space-charge is associated with particle orbits. Space-charge contributions are always present under the *SCharge*, *RelBeam*, *FEmit* and *Plasma* options. The field solution routines are the same as those used in **HiPhi**. Therefore, the following commands are identical to those described in the **HiPhi** manual except for the appearance of the symbol *E* which designates that the parameters apply to the electric field solution.

RESTARTGET E ResTargE
RESTARTGET(E) = 5.0E-6

The numerical calculation of electrostatic fields requires the solution of a large set of coupled linear equations, one for each active node in the solution volume. **OmniTrak** uses an iterative technique based on

corrections that reduce the error in electrostatic potential at a point compared to predictions from values at neighboring nodes. The *residual* is an average of the relative errors over all nodes in the solution space during an iteration. This command sets a target value for the residual. The program stops if the error drops below the value. For good accuracy, the relative residual should be less than 10^{-6} . Default value: $ResTargE = 5.0 \times 10^{-7}$.

MAXCYCLE E MaxCyE

MAXCYCLE(E) = 2500

The integer parameter *MaxCyE* is the maximum number of iteration cycles. **OmniTrak** saves the solution and stops when it reaches *MaxCyE* even if *ResTargE* has not been attained. Default value: *MaxCyE* = 2500.

OMEGA E OmegaE

OMEGA(E) = 1.92

The real-number parameter *OmegaE* is the over-relaxation factor used to correct potential errors during the solution. This quantity may have a significant effect on the run time. If the *Omega* command does not appear, **OmniTrak** automatically picks values that vary with the iteration cycle following the Chebyshev acceleration prescription. With this command, you can set a value manually. The value must lie between 0.0 and 2.0 for numerical stability. Higher values generally give faster convergence.

NCHECK E NCheckE

NCHECK(E) = 20

The operations to calculate *ResTargE* take time; therefore, **OmniTrak** usually does not make a check on each iteration cycle. The integer parameter is the number of relaxation cycles between convergence checks. Default value: *NCheckE* = 25.

NGAUSS E NGaussE

NGAUSS E 5

The parameter *NGaussE* controls the number of intervals for the Gaussian-quadrature integrals for the electric field recalculation. Usually it should not be necessary to make any change from the default value of *NGaussE* = 3. Higher values may give somewhat better accuracy at the expense of longer computation times.

4.3. Program control commands

The following two commands perform general program control functions and apply to runs with either electric or magnetic fields.

DUNIT DUnit
DUNIT = 39.37

This command controls conversion of input spatial quantities in the **OmniTrak** script. It does not affect coordinate values read from **HiPhi** and **Magnum** solution files which have already been converted to meters. The quantity *DUnit* (real) is the number of distance units per meter. For example, to enter values in cm for the *SHIFT* command, set

DUnit = 100.0. In contrast to other *FIELDS* commands, the order of appearance of *DUNIT* is significant. The present value of *DUnit* is applied in commands that follow. Multiple *DUNIT* commands may appear in the *FIELDS*, *PARTICLES* and *DIAGNOSTICS* sections.

BOUNDARY X1 Y1 Z1 X2 Y2 Z2
BOUNDARY = (0.0, 0.0, 0.0) (5.0, 2.5, 2.5)

Orbit calculations are performed inside a three-dimensional solution box with opposite corners defined by (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2) . Enter coordinates in units set by *DUnit*. Particle orbits terminate if they move outside the solution box. **OmniTrak** interpolates the final orbit step so that the stopping point lies exactly on the box surface. The values of (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2) are recorded in the plot file (FPrefix.OOU) and are used as default plot boundaries in **OmniView**. If the *BOUNDARY* command does not appear the program automatically sets the solution volume as the largest box that overlaps the electrical and/or magnetic solutions. If the *BOUNDARY* command does appear **OmniTrak** sets the *ballistic flag*. In this case orbits continue even if they leave the boundaries of the electric and/or magnetic solution volume. The program takes $\mathbf{E} = 0$ and $\mathbf{B} = 0$ in regions that are inside the boundary box but outside the field solution volumes. The ballistic mode is useful, for example, if you want to trace orbits to a focal point distant from a lens with localized field.

4.4. Loading electric field information from a table

As an alternative to electric field solutions from **HiPhi**, **OmniTrak** can read field information from a text file in **AMaze** Field Exchange format. With this feature, you can track orbits in electric fields generated mathematically, by your own software, or by other finite-element programs.

The field exchange file contains a set of data lines with electric field values (E_x, E_y, E_z) calculated at points in a rectangular mesh. The mesh has the following structure shown in Fig. 4.1. The quantity I is the column index (x direction), J is the row index (y direction) and K is the depth index (z direction). The indices have the ranges $0 \leq I \leq I_{\max}$, $0 \leq J \leq J_{\max}$, $0 \leq K \leq K_{\max}$.

The file consists of a two-line header followed by $(I_{\max}+1)(J_{\max}+1)(K_{\max}+1)$ data lines.

```
IMax, JMax, KMax
XMin, YMin, ZMin, XMax, YMax, ZMax
Ex Ey Ez [0,0,0]
Ex Ey Ez [1,0,0]
Ex Ey Ez [2,0,0]
...
Ex Ey Ez [IMax,0,0]
Ex Ey Ez [0,1,0]
...
Ex Ey Ez [I,J,K]
...
Ex Ey Ez [IMax,JMax,KMax]
```

Any number of comment lines (starting with an asterisk, '*') may appear before the header. Line entries may be separated by any of the valid **AMaze** delimiters (Space, Comma, Equal sign, Tab, Left parenthesis, Right parenthesis). The quantities $IMax$, $JMax$ and $KMax$ are integers. All other entries are real numbers in any valid format. The order of data line storage is

```
DO K=0, KMax
  DO J=0, JMax
    DO I=0, IMax
      READ Ex(I,J,K) Ey(I,J,K) Ez(I,J,K)
    END DO
  END DO
END DO
```

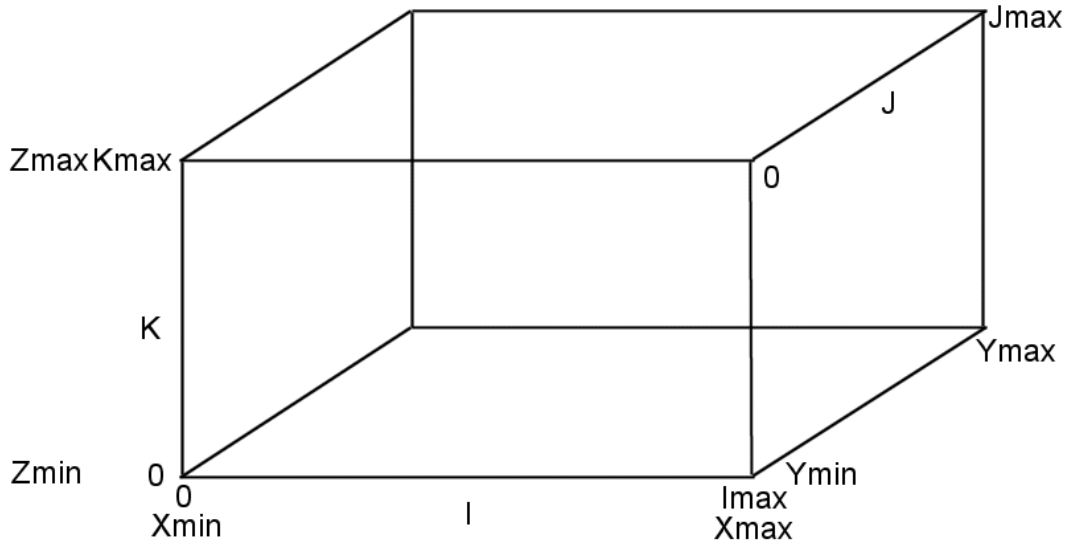


Figure 4.1. Mesh conventions for recording values in an AMaze Field Exchange file.

Data lines may contain additional quantities after the three electric field values. For example, the data lines in matrix files created by **PhiView** contain

```
Ex Ey Ez Phi EpsiR NReg
```

The last three quantities are ignored if matrix file is loaded into **OmniTrak**.

ETABLE3D = TableFileName [FMultET3D]

ETABLE3D = ElecLens.DAT 100.0

This command loads values from an electric field table in AMaze Field Exchange format. The quantity *FileTableName* is the full name of a data file available in the current directory. The optional real-number parameter *FMultET3D* is a field scaling factor. Values of the electric field are multiplied by *FMultET3D* when the file is loaded. The quantity can also be used to convert units if the electric field values are not in the standard units of V/m. The default value is *FMultET3D* = 1.0.

The following rules apply to the use of tabular electric field in **OmniTrak**:

- The program accepts either a **HiPhi** solution or tabular file information. The code gives an error message if both the *EFIELD3D* and *ETABLE3D* commands appear in the *FIELDS* section.
- The field exchange file contains no conformal mesh information. Consequently, **OmniTrak** does not detect mesh regions along the particle orbit. Stopping conditions based on region properties are not functional.
- Without mesh information, the electric field cannot be recalculated. Therefore, tabular electric fields can be used only in the *TRACK* and *FLINE* modes.
- Tabular fields can be translated with the *SHIFT E* command.
- If the *BOUNDARY* command does not appear in the script, orbits terminate if they leave the bounds of the table. The presence of the *BOUNDARY* command sets the ballistic mode. In this case, particle orbits continue outside the boundaries of the table with the condition $(E_x, E_y, E_z) = (0.0, 0.0, 0.0)$.
- Field interpolations are faster on the regular meshes of tables compared to the conformal meshes of **HiPhi** solutions. For intensive orbit calculations in the *TRACK* and *FLINE* modes, you may be able to reduce the run time by using a tabular file created with **PhiView** rather than the direct **HiPhi** field solution. This technique should be used only for orbits in a vacuum volume. Field interpolations on a table are usually inaccurate near conducting or dielectric boundaries.

5. Applied magnetic fields - FIELDS section

5.1. Uniform field components

Magnetic fields within the solution volume for particle orbits may arise from two sources:

- Applied fields created by specified currents in external coils.
- Beam-generated fields that result from the current carried by the particle orbits.

Generally beam-generated fields are important only in simulations of high-current relativistic electron beams. In this case the program must carry out complex recalculations of the magnetic fields that include the self-consistent contributions of the particle orbits. Chapter 11 describes commands for magnetic field recalculation.

In most applications only applied magnetic fields are significant, and there is no need for recalculation within **OmniTrak**. This chapter discusses the options for including magnetic force contributions in the particle equations of motion. The simplest is the addition of a magnetic flux-density that is uniform over the solution space. This option may be useful, for example, to find the effect of the Earth's magnetic field on an electron optics system.

BUNI Bx0 By0 Bz0

BUNI = (0.25E-4 0.0 0.0)

This command defines Cartesian components of a spatially-uniform magnetic flux density. Enter the values in tesla. Uniform flux-density values can be combined with either two-dimensional (Sect. 5.2) or three-dimensional (Sect. 5.3) magnetic solutions. The default values are $B_{x0} = 0.0$, $B_{y0} = 0.0$ and $B_{z0} = 0.0$.

5.2. Loading and modifying two-dimensional solutions

Beam systems with three-dimensional variations of electric fields often have magnetic fields with cylindrical or planar symmetry. For example, an electron optics device may be located in the field created by Helmholtz coils or a finite-length solenoid. For this reason we have included the option to include two dimensional magnetic field solutions from the Field Precision **TriComp** series of programs. **OmniTrak** can read solution files from **BStat**, **PerMag**, **Pulse** and **Nelson**. Note that you can load either a two-dimensional or three-dimensional magnetic field solution. Both solution types cannot be loaded simultaneously.

BFIELD2D = FileName [FMultB2D]

BFIELD2D = HelmHoltz.BOU 0.75

This command loads a magnetic field solution from a **TriComp** program. The quantity *FileName* is the full name of the file (*i.e.*, PPM_SYSTEM.POU). The optional real-number parameter *FMultB2D* is a field scaling factor. Values of vector potential A_z or stream-function rA_θ are multiplied by *FMultB2D* when the file is loaded. The default value is $FMultB2D = 1.0$. The solution file must be available in the working directory.

The following default conventions for mapping the 2D solution onto the 3D space apply when the file is loaded.

■ **Planar solution** (*Rect* option). The z axis of the 2D field solution (out of the page in a plot) corresponds to the z -axis of the 3D solution space. The two spaces share the same x and y axes and origin in the x - y plane. The field components B_x and B_y are uniform in z .

■ **Cylindrical solution** (*Cylin* option). The z axis of the 2D field solution corresponds to the z axis of the 3D solution space with the same origin ($z = 0.0$). The fields have azimuthal symmetry about the z axis of the 3D solution volume. The B_x and B_y field components in the 3D space are computed from the B_r component of the 2D solution.

The default orientation can be modified with the following two commands. The commands give considerable latitude to rotate and to shift the 2D solution in the 3D space.

SHIFT B XShift YShift ZShift

SHIFT(B) = (-5.00, 2.50, 0.00)

This command adds displacements to the transformation between 3D and 2D space. For a planar solution, the values *XShift* and *YShift* move the *x-y* origin of a non-rotated 2D solution to the position (*XShift*, *YShift*) in the 3D space, while the quantity *ZShift* has no effect. For a non-rotated cylindrical solution, the command moves the *z* axis of the 2D solution to the position (*XShift*, *YShift*) in the *x-y* plane of the 3D solution and shifts the origin of the *z*-axis of the 2D solution to the position *ZShift* in the 3D space. Enter the real-number parameters *XShift*, *YShift* and *ZShift* in units set by the *DUnit* command. Note that translations are performed *after* any rotations specified in the *ROTB2* command

ROTATE B RotX RotY RotZ [RotOrder]

ROTATE(B) = (90.0, 0.0, 45.0) (ZX)

This command rotates the 2D solution in the 3D space. The parameters *XRot*, *YRot* and *ZRot* are rotation angles (in degrees) about the current *x*, *y* or *z* axes. The optional parameter *RotOrder* gives the order in which rotations should be performed. *RotOrder* contains from one to three of the characters *X*, *Y* or *Z*. The default is *RotOrder* = 'XYZ'. In the example above, the field solution is rotated 45° about the *z* axis and then 90° about the new *x* axis. The value *YRot* is ignored..

Rotations are performed before shifts.

Rotation operations can be confusing. You can use the *BPOINT* and *BSCAN* commands in the *DIAGNOSTICS* section to check the validity of the field transformation.

5.3. Loading and modifying Magnum solutions

This section introduces commands to load and to modify three-dimensional magnetic field solutions from the **AMaze** program **Magnum**. Magnetic field information can be incorporated for particle tracking, either individually or in conjunction with an electrostatic solution from **HiPhi**. The option for magnetic field only is useful for single-particle tracking in septum magnets or magnetic spectrometers. The associated mesh is used for magnetic field interpolations and may be queried to determine stopping conditions when a particle strikes a material object.

BFIELD3D = FileName [FMultB3D]

BFIELD3D = MSpect.GOU

This command loads a magnetic field solution from **Magnum**. The quantity *FileName* is the full name of the file (i.e., BEAMSEP.GOU). The optional real-number parameter *FMultB3D* is a field scaling factor. Values of applied field (\mathbf{H}_s), reduced potential ϕ and dual potential ψ are multiplied by *FMultB3D* when the file is loaded. The default value is *FMultB3D* = 1.0. The **Magnum** file must be available in the working directory.

SHIFT B XShift YShift ZShift

SHIFT B = (-5.00, 2.50, 0.00)

This command shifts node positions in the magnetic field mesh according to

$$x_{\text{new}} = x_{\text{old}} + XShift.$$

$$y_{\text{new}} = y_{\text{old}} + YShift.$$

$$z_{\text{new}} = z_{\text{old}} + ZShift.$$

Enter the real-number parameters *XShift*, *YShift* and *ZShift* in the units set by the *DUnit* command. This command may be useful to shift an magnetic field mesh to match an electric field mesh or to do a piecewise simulation of a periodic system. Shifts are performed after rotations.

ROTATE B XRot YRot ZRot [RotOrder]

ROTATE(B) = 90.0 0.0 45.0 ZX

This command rotates the coordinates and field components in the 3D magnetic field mesh or table. The parameters *XRot*, *YRot* and *ZRot* are rotation angles (in degrees) about the current x, y or z axes. The optional parameter *RotOrder* gives the order in which rotations should be performed. *RotOrder* contains from one to three of the characters X, Y or Z. The default is *RotOrder* = 'XYZ'. In the example above, the field solution is rotated 45° about the z axis and then 90° about the new x axis. The value *YRot* is ignored.. Rotations are performed before shifts.

5.4. Loading magnetic field information from a table

In addition to magnetic field information from **Magnum**, **OmniTrak** can load field information from a text file in AMaze Field Exchange format. With this feature, you can track orbits in magnetic fields calculated from any source. The nature of the field exchange file for electric fields was described in Sect. 4.4. The only difference is that the first three quantities in data lines are (B_x, B_y, B_z) in standard units of tesla. For reference, data lines created by the matrix file feature of **MagView** contain the quantities:

Bx By Bz MuR NReg

BTABLE3D = TableFileName [FMultBT3D]

BTABLE3D = HelmHoltz.DAT

This command loads values from a magnetic field table in AMaze Field Exchange format. The quantity *FileTableName* is the full name of a data file available in the current directory. The optional real-number parameter *FMultBT3D* is a field scaling factor. Values of the magnetic field are multiplied by *FMultBT3D* when the file is loaded. The quantity can also be used to convert units if the magnetic field values are not in the standard units of tesla. For example, use $FMultBT3D = 1.0 \times 10^{-4}$ if the value is tabulated in gauss.

The following rules apply to the use of tabular magnetic fields in **OmniTrak**:

- The program accepts either a **Magnum** solution or tabular file information. The code gives an error message if both the *BFIELD3D* and *BTABLE3D* commands appear in the *FIELDS* section.
- The field exchange file contains no conformal mesh information. Consequently, **OmniTrak** does not detect mesh regions along the particle orbit. Stopping conditions based on region properties are not functional.
- **OmniTrak** does not recalculate applied magnetic fields. Therefore, tabular magnetic fields may be used in all tracking modes.
- Tabular fields can be translated with the *SHIFT B* command.

■ If the *BOUNDARY* command does not appear in the script, orbits terminate if they leave the bounds of the table. The presence of the *BOUNDARY* command sets the ballistic mode. In this case, particle orbits continue outside the boundaries of the table with the condition $(B_x, B_y, B_z) = (0.0, 0.0, 0.0)$.

■ Field interpolations are faster on the regular meshes of tables compared to the conformal meshes of **Magnum** solutions. For intensive orbit calculations, you may be able to reduce the run time by using a tabular file created with **MagView** rather than the direct **Magnum** field solution. This technique should be used only for orbits in a vacuum volume. Field interpolations on a table are usually inaccurate near the boundaries of iron regions.

6. Temporal modulations of fields - FIELDS section

OmniTrak can add arbitrary time variations to electric and/or magnetic fields (both 2D and 3D). Temporal effects may of interest if the field changes significantly over a time scale comparable with the particle transit time. **OmniTrak** multiplies static field solution values by a modulation function to introduce time variations. You must determine whether these variations are physically correct for your simulation. Generally, solutions for time-varying fields will be close to the static solutions if

$$\Delta t \ll \frac{L\epsilon^{1/2}}{c} .$$

In the equation, Δt is the time scale for field variations, L is the approximate dimension of the solution volume, ϵ is the local relative dielectric constant, and c is the speed of light. Temporal modulations are invalid if the particles make significant contributions to the fields. Therefore, modulations can be applied only in the *Track* mode for single-particle tracking.

MODFUNC E SIN A0 A1 F T0

MODFUNC(E) = SIN (0.25, 2.00, 4.6E4, 5.3E-6)

This command activates a harmonic variation of all components of the electric field (E_x , E_y , E_z). The time variation of the modulation function is

$$ModFuncE = A_0 + A_1 \sin[2\pi f (t-t_0)].$$

The quantities in the equation are the offset (A_0), the modulation amplitude (A_1), the frequency in Hz (f) and the time offset in seconds (t_0). The time offset determines the phase of the oscillation. Values of **E** calculated from the static **HiPhi** solution are multiplied by $ModFuncE(t)$. Note that **OmniTrak** starts each orbit at $t = 0$, resetting the modulation function.

MODFUNC B SIN A0 A1 F T0

MODFUNC B SIN 0.25 2.00 4.6E4 5.3E-6

This command activates a harmonic variation of all components of the calculated magnetic flux density (B_x, B_y, B_z). The modulation is applied to the components of total magnetic flux density which may include uniform contributions defined by the *BUNI* command and either a two- or three-dimensional magnetic solution.

MODFUNC E TABLE TableName

MODFUNC(E) TABLE 50NSPULSE.DAT

OmniTrak can represent an arbitrary temporal variation of electric fields. The program uses cubic spline interpolation on a user-specified table. The table is contained in a text file that may include blank lines and comment lines beginning with ‘* ‘ (asterisk). The file must end with an *ENDFILE* command. A table may contain up to 128 data line. Each line has two real numbers:

Time ModFunc

Time values must start at $t = 0.0$. The time span of the table (t_{\max}) normally extends up to or beyond the maximum particle transit time. If the particle transit time t exceeds t_{\max} , **OmniTrak** restarts the table and returns the value $ModFuncE(t')$, where $t' = \text{MODULUS}(t, t_{\max})$. This feature makes it possible to define arbitrary periodic functions.

MODFUNC B TABLE TableName

MODFUNC(B, TABLE) = 50NSPULSE.DAT

In this case, the modulation function defined by a table is applied to the components of the total magnetic field.

To illustrate, consider a pulse with the following variation in the range $0 \leq t \leq 2\Delta t$, where the width at half-height is $\Delta t = 0.01$ s:

$$ModFuncE(t) = \frac{1 - \cos(\pi t / \Delta t)}{2},$$

The following table defines a single instance of the pulse:

| | |
|-------|-------|
| 0.000 | 0.000 |
| 0.001 | 0.024 |
| 0.002 | 0.095 |
| 0.003 | 0.206 |
| 0.004 | 0.345 |
| 0.005 | 0.500 |
| 0.006 | 0.655 |
| 0.007 | 0.794 |
| 0.008 | 0.905 |
| 0.009 | 0.976 |
| 0.010 | 1.000 |
| 0.011 | 0.976 |
| 0.012 | 0.905 |
| 0.013 | 0.794 |
| 0.014 | 0.655 |
| 0.015 | 0.500 |
| 0.016 | 0.345 |
| 0.017 | 0.206 |
| 0.018 | 0.095 |
| 0.019 | 0.024 |
| 0.020 | 0.000 |
| 0.030 | 0.000 |
| 0.100 | 0.000 |
| 1.000 | 0.000 |

ENDFILE

Note that the time interval need not be uniform. The extra values at the end of the table ensure that the modulation function equals 0.0 for $t > 0.02$ seconds. The next example illustrates how the table would be modified to create a series of pulses separated by an interval 0.05 seconds.

| | |
|-------|-------|
| 0.000 | 0.000 |
| 0.001 | 0.024 |
| 0.002 | 0.095 |
| 0.003 | 0.206 |
| 0.004 | 0.345 |
| 0.005 | 0.500 |
| 0.006 | 0.655 |
| 0.007 | 0.794 |
| 0.008 | 0.905 |
| 0.009 | 0.976 |
| 0.010 | 1.000 |
| 0.011 | 0.976 |

| | |
|-------|-------|
| 0.012 | 0.905 |
| 0.013 | 0.794 |
| 0.014 | 0.655 |
| 0.015 | 0.500 |
| 0.016 | 0.345 |
| 0.017 | 0.206 |
| 0.018 | 0.095 |
| 0.019 | 0.024 |
| 0.020 | 0.000 |
| 0.030 | 0.000 |
| 0.050 | 0.000 |

ENDFILE

7. Single-particle tracking in the Track mode - PARTICLES section

The *PARTICLES* section of the **OmniTrak** control script follows the *FIELDS* section. The commands of the *PARTICLES* section instruct the program to launch and to follow particle orbits. For high-current beams, **OmniTrak** may also modify electric and/or magnetic fields. Again, several commands may follow in arbitrary order. **OmniTrak** reads and processes all input information in the *PARTICLES* section before starting orbit calculations. The script section begins with the *PARTICLES* command followed by one the following keywords:

Track

The particle properties, initial positions and initial directions are specified in a list or source surface. Particle orbits are calculated in the single-particle limit.

SCharge

Particle starting properties are entered from a list or automatically generated over marked source surfaces. In the latter case the code assigns position, current and kinetic energy to satisfy conditions for space-charge-limited emission over multiple surfaces of arbitrary shapes. An iterative process determines the self-consistent electric fields with contributions from beam space charge

RelBeam

In addition to the capabilities of the *SCharge* mode, the *RelBeam* mode handles contributions of beam current to the three-dimensional magnetic field.

FEmit

This mode is similar to the *SCEmit* mode except that the particle current is determined by the Fowler-Nordheim equation.

Plasma

A space-charge-limited flow of ions is emitted from a flexible plasma surface. Particle parameters are adjusted to satisfy the Child law, while the shape of the plasma surface is adjusted to ensure uniform flux.

FLine

In this mode **OmniTrak** carries out precise calculations of electric or magnetic field lines.

Table 7.1. Allowed commands for tracking modes

| | |
|--|---|
| <p>ALL MODES STOPPLANE XUp XStopUp DIAGPLANE XUp XStopUp DUNIT DUnit SMAX SMax LISTON ListSkip ORBINFO PLOT OFF MATERIAL E NReg VACUUM E NReg NSTEPMAX NStepMax</p> | <p>ALL PARTICLE MODES DT Dt TMAX TMax SECONDARY E NReg Delta0 Eng0 SECONDPARAM KECutOff MinFact PLIST PFILE PFileName LISTTYPE Particle</p> |
| <p>TRACK MODE EMIT NReg Mass Charge NDivide</p> | <p>SCHARGE, RELBEAM, FEMIT, PLASMA MODES AVGE AvgE NCYCLE NCycle SUPPRESS SFract(1) SFract(2) ...</p> |
| <p>SCHARGE MODE EMIT NReg Mass Charge DEmit NDivide JLIMIT JLimit DTHETA DTheta EDIRECT [Surface, Field] RELMODE</p> | <p>RELBEAM MODE EMIT NReg Mass Charge DEmit NDivide JLIMIT JLimit DTHETA DTheta MU NReg Mu BBELEM LElem RElem RELMODE</p> |
| <p>FLINE MODE FLIST FFIELD DS EMIT NReg NDivide [Polarity]</p> | <p>FEMIT MODE EMIT NReg WorkFunc NDivide [Beta]</p> |

This chapter concentrates on the *Track* mode. The mode handles single-particle tracking. Effects of space-charge and beam-generated magnetic fields are covered in the following chapter. Section 7.1 introduces the format of the particle list and discusses options to enter a list as part of the command script or from a separate file. Section 7.2 discusses how to define a source surface and how particle orbits are generated from the surface. The material in Sections 7.3, 7.4 and 7.5 applies to all tracking modes. Section 7.3 describes commands to set the integration time step and other control parameters. Section 7.4 covers assignment of material properties on electric or magnetic field mesh. Particles propagate through *Vacuum* elements but stop if they enter a *Material* element (i.e., electrode). Stopping and diagnostic planes are described in Section 7.5. The option for accurate stopping conditions is useful for lens characterization or the design of isochronous systems.

7.1. Starting orbits from a list

OmniTrak follows up to 5000 particle orbits in a run. You can use a list to start distributions of particles from arbitrary locations. The particle list has a standard format compatible with the Field Precision two-dimensional code **Trak**. You can create lists with an editor or spreadsheet. **OmniTrak** can generate a standard list of particle parameters at the end of a run that can be used as input to a succeeding run. Entry of initial particle parameters from a list is through text statements. You must supply a line for each orbit. The particle information lines can be inserted directly into the input command file with the *PList* command or they can be read from an independent file with the *PFile* command. *PFile* is useful if you set up a spreadsheet for automatic generation of input beam distributions or use the **TDist** utility program. Also, each **OmniTrak** run can produce an output file that can act as the input particle file for another run. In this way, it is possible to stack runs to study long optical systems.

For both the *PList* and *PFile* commands, **OmniTrak** expects a series of text file lines consisting of nine real numbers. The *PList* section has the format

```
PLIST
  <Particle Line 1>
  <Particle Line 2>
  ...
  <Particle Line NPart>
END
```

Real numbers can be expressed in any valid floating point form. Input is the standard Field Precision free format using the delimiters listed in Sect. 3.2. For a *TRACK* mode simulation each particle line has the following components.

```
Rest   Charge   Kinetic   x   y   z   fx   fy   fz
mass   energy
```

The quantities in the line have the following meanings.

1. Rest mass

The particle rest mass in AMU (atomic mass units), where 1.0 AMU = 1.65979 x 10⁻²⁷ kg. The rest mass of a proton is about 1.00. If 0.00 appears in this position, **OmniTrak** will automatically insert the rest mass of the electron.

2. Charge

The particle charge in units of elementary charge 1.60210 x 10⁻¹⁹ coulombs. The charge of a proton is 1.00 while the electron has a charge -1.00.

3. Kinetic energy

The particle initial kinetic energy in electron volts, where 1.0 eV = 1.6021 x 10⁻¹⁹ joules.

4. x, y, z

The initial particle position in meters or units set by *DUnit*.

5. fx, fy, fz

The fraction of particle momentum directed along the three Cartesian axes. **OmniTrak** will automatically normalize these quantities if the the sum of the squares does not equal unity. For, example, an *x*-directed beam has $f_x = 1.0$, $f_y = 0.0$, and $f_z = 0.0$. A particle orbit in the *x-y* plane initially inclined at an angle α with respect to the *x* axis has $f_x = \cos\alpha$ and $f_y = \sin\alpha$. For a particle with zero kinetic energy use $f_x = f_y = f_z = 1.0$.

Note that line entries can be separated by any of the valid delimiters listed in Sect. 3.2. The **TDist** program and spreadsheets are useful to generate input lines for distributions of many particles. You can develop templates to generate beam distributions with given dimensions, directions or emittance.

PLIST

This command signals that a series of standard particle parameter lines will follow in the **OmniTrak** script. The list may contain blank and comment lines and must terminate with the *End* command. Comment lines are marked by the asterisk (*) symbol. The *PList* sequence can appear anywhere in the *Particles* section.

PFILE FPrefix

PFILE TCathode

This command specifies that **OmniTrak** should read particle parameter lines from an external file with a name of the form `FPrefix.PRT`. The file name must start with a prefix from 1 to 20 characters in length and end with the suffix `PRT`. The file must be available in the working directory. A particle file may consist of blank lines, comments and from 1 to 5000 data lines. The file must terminate with the *End* command.

```
<Particle Line 1>
<Particle Line 2>
...
<Particle Line NPart>
END
```

7.2. Starting orbits from a source surface

Particles with zero kinetic energy can be generated automatically near special areas on the surfaces of electrodes. Use of the areas, *source surfaces*, requires that a properly-prepared electric field file has been loaded with the *EFILE* command. The feature is activated by the presence of an *EMIT* command in the *PARTICLES* section. A source surface is defined by a set of nodes with a unique region number on the surface of a fixed-potential region. **OmniTrak** can handle up to 20 source surfaces. Surface marking is performed in **MetaMesh** using the `COAT` command. As an example, consider the converging electron beam source with focusing electrode shown in Fig. 7.1. Part *a* is a three-dimensional view of element facets on region boundaries and part *b* is section through the midplane showing the region identities of nodes. The assembly was created with the following **MetaMesh** script commands:

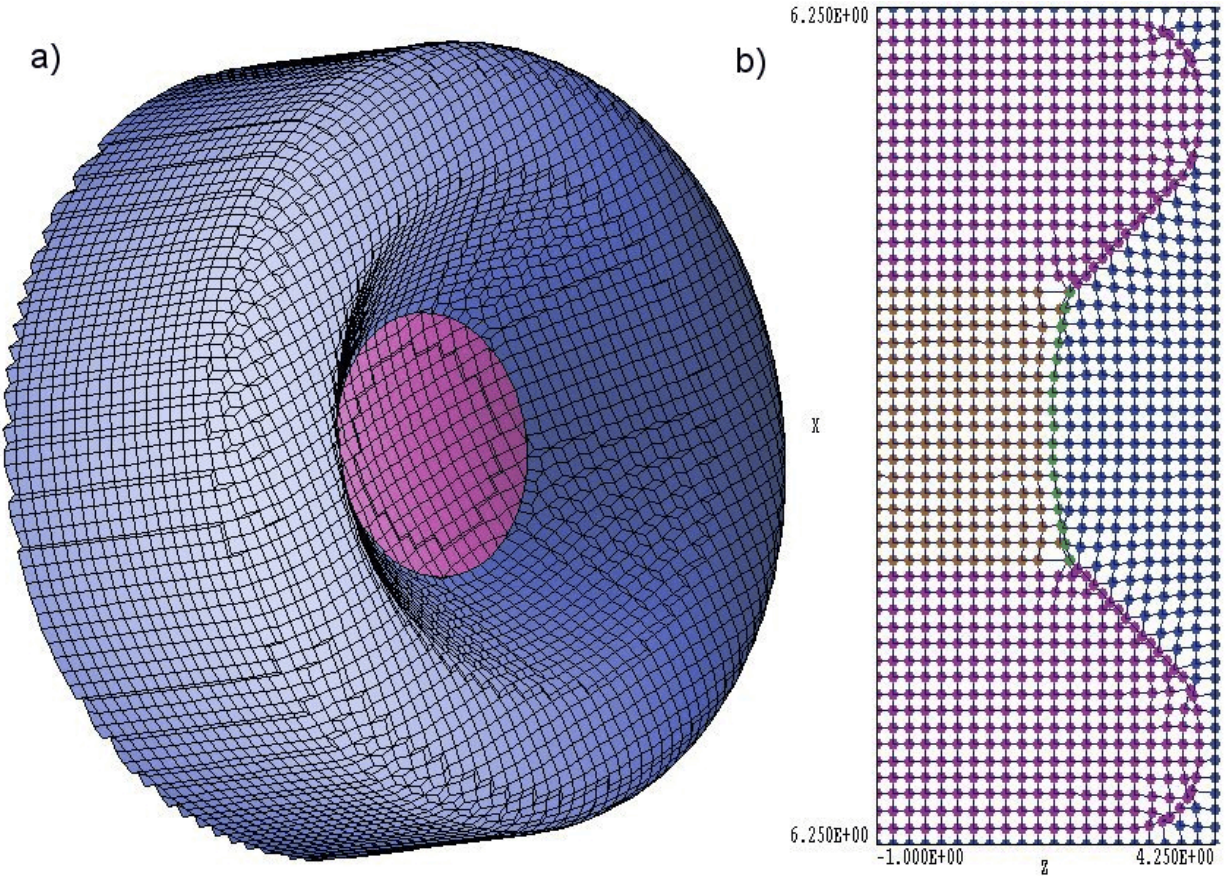


Figure 7.1. Illustration of the COAT operation. *a)* Three dimensional view of facets on region boundaries: region 2 (blue), region 3 (volet). *b)* Cross section in midplane showing node identities: region 1 (blue), region 2(violet), region 3(brown), region 4(green).

```

Part 2
Type Turning
L 2.00000 2.00000 3.70711 3.70711 S
A 3.70711 3.70711 4.00000 4.41421 3.00000 4.41421 S
L 4.00000 4.41421 4.00000 5.00000 S
A 4.00000 5.00000 3.00000 6.00000 3.00000 5.00000 S
L 3.00000 6.00000 -1.00000 6.00000 S
L -1.00000 6.00000 -1.00000 2.00000
L -1.00000 2.00000 2.00000 2.00000
End
Region 2
Fab 0.0 360.0
Shift 0.00 0.00 0.00
Surface Region 1 1.00
End

```

```

Part 3
Type Turning
  L  -1.00000  0.00000  1.71989  0.00000
  A  1.71989  0.00000  2.00000  2.00000  9.00000  0.00000  SE
  L  2.00000  2.00000 -1.00000  2.00000  SE
  L -1.00000  2.00000 -1.00000  0.00000  SE
End
Region 3
Fab 0.0  360.0
Shift 0.00 0.00 0.00
Surface Region 1 Edge 1.00
Surface Region 2 Edge 1.00
Coat 1 4
End

```

The command `Coat 1 4` in the definition of Part 3 instructs the program to change the region identity of nodes connected to facets on the boundary between Part 3 and Region 1 to Region 4. The resulting set of nodes that define the surface of a concave cathode is shown in green in Fig. 7.1*b*.

Given the identity of one or more marked regions, **OmniTrak** locates the *facets* (element faces) that comprise the source surface(s). The program checks each fixed-potential element in the mesh and inspects the six facets to see if: 1) the four nodes of the facet have the same region number and 2) the region number corresponds to one of the marked regions. When a facet is identified **OmniTrak** checks the surface to ensure that it lies between a fixed-potential (*Material*) and a *Vacuum* element. The code reports an error if both adjacent elements are *Material* or *Vacuum*. The program records the following facet information in a temporary file: the marked region number, the mesh indices and the facet orientation. The last quantity is one of the options *XUp*, *XDn*, ..., *ZDn*.

Once the source facets have been identified, the next step is to generate a set of initial particle parameters. **OmniTrak** sequentially reads the facet list. For each facet the program uses the orientation information to assign node coordinates that ensure that surface normal vectors will point out of the fixed potential element and into the vacuum element. The facet is divided into $N_{Divide} \times N_{Divide}$ subsections using the normal coordinates (u, v) of the surface. A model particle is created at the center of each facet subsection. The initial position of each particle is shifted a slight distance from the facet into the vacuum element using the surface normal vector to ensure a valid initial electric field interpolation.

Note that the density of particles from a source surface depends on the size and shapes of the facets. Generally the density will not be uniform over a surface with conformal elements (*i.e.*, the particles may not represent equal areas on the surface). In the *TRACK* mode the source surface functions as a quick way to start a large number of particle orbits from a portion of an electrode. Use the *PLIST* or *PFILE* commands if you want to define a precise distribution of particles.

EMIT NReg Mass Charge NDivide

EMIT(5) = (0.0,-1.0, 2)

Generate particles of a given species with zero kinetic energy near the facets of a source surface marked with region number *NReg*. The real-number parameter *Mass* is the particle mass in AMU. If you enter 0.0, **OmniTrak** applies the mass of an electron. The quantity *Charge* is the particle charge in units of elementary charge, 1.60210×10^{-19} coulombs. The charge of a proton is 1.00 while the electron has a charge -1.00. The quantity *NDivide* determines the number of particles per facet. Each facet is divided into *NDivide* sections along the two normal coordinate directions, so that the program creates $NDivide^2$ particles. In the example **OmniTrak** searches for valid facets that have all four nodes marked with region number 5 and generates 4 electrons per facet. Note that this form of the *EMIT* command applies only in the *TRACK* mode.

7.3. Orbit integration controls

The following commands set options for control of orbit integrations. The most important parameter is the time step, *Dt*. Small values of *Dt* will result in long run times. Conversely, the solution may be inaccurate if *Dt* is too large. By default, **OmniTrak** tries to estimate a good value for the time step when all particles have the same rest mass. In this case, the program applies the following procedure:

- Search the electric and/or magnetic meshes to find the minimum average element dimension along the *x*, *y* and *z* axes.
- If an electric mesh has been loaded, find the maximum potential difference in the solution space.
- Find the maximum initial kinetic energy (in eV) for particles

- Using the maximum of the potential difference and initial kinetic energy, estimate the maximum velocity that particles can attain in the solution space.
- Divide the minimum element size by the maximum velocity for a conservative time step estimate.

You can set a fixed value of the time step manually with the following command:

DT Dt
DT 5.0E-12

The *DT* command overrides automatic selection of the time step by **OmniTrak**. This option can be used to improve computational accuracy (smaller time step) or reduce the run time (larger time step). Enter the quantity *Dt* in seconds. When this form of the *DT* command appears, the program uses the same value of *Dt* for all orbits.

When runs include particles with different masses, you *must* set the time step manually. The situation occurs when different entries appear in the *RestMass* column for *Track* input or when emission surfaces are associated with different types of particles.

DT DtRef Mass
DT 5.0E-10 Mass

This form of the *DT* command has the added keyword *Mass*. Apply the command when runs include particles with different mass (*i.e.*, ions and electrons). Here the value *DtRef* is a reference time step for particles with a mass equal to 1 AMU (atomic mass unit). If a particle has a rest mass m_p (expressed in AMU), then **OmniTrak** picks a time step $Dt = DtRef \times m_p^{1/2}$ for the integration. For reference, 1 AMU equals 1.6598×10^{-27} kg. The electron rest mass is 9.1091×10^{-31} kg or $m_e = 5.494 \times 10^{-4}$ AMU. Therefore, the time step for integration of an electron orbit would be $0.0234 \times DtRef$

LISTON NStep
LISTON 5

Normally, **OmniTrak** records only the initial and final parameters of particles in the listing file `RunName.TLS`. Use the *LISTON*

command to include a listing of information at intermediate points along the orbit. The information is more detailed than that in the plot file (`RunName.OOU`) which contains only position information. Depending on the setting in the *ListType* command, you can record detailed information on particle momentum, electric field or magnetic field. The integer parameter *NStep* is the number of time steps between entries. This command is useful mainly for runs with a few particles or for initial run checks. Activation of the *LISTON* command for runs with a large number of particles could result in a huge listing file.

LISTTYPE ListOption

LISTTYPE EField

This command determines what type of information is recorded when the *LISTON* command has been issued. The parameter *ListOption* is a string with one of the following values: *PARTICLE*, *EFIELD* or *BFIELD*. The particle position at each point is listed under all options. Under the *PARTICLE* option (the default) **OmniTrak** lists normalized momentum, kinetic energy, elapsed time and transit distance. The two other options list local field values if an electric and/or magnetic field has been loaded.

PLOTOFF

This command suppresses the recording of orbit vectors in the plot file `FPrefix.OOU`. By default, **OmniTrak** records the orbit position vector at each time step for all particles. For runs with large numbers of particles, the recording process can create a very large file and slow the program operation. If you want plots of a few representative orbits from a complex solution, you can set up a second run with single-particle tracking using the self-consistent electric and magnetic fields created by the *EDUMP* and/or *BDUMP* commands in the first run.

ORBINFO

This command sets a program flag so that **OmniTrak** records detailed information on the termination conditions at the end of each orbit. Information includes final positions and momenta, estimates of energy conservation, and the stopping condition at termination. This information is useful for debugging runs with a moderate number of particles. For example if particles do not advance in a run, you can use the *ORBINFO* command to check the error condition.

7.4. Stopping orbits

DUNIT DUnit **DUNIT = 39.37**

This command is described in Section 4.3.

Accurate information on where and when orbits stop is critical to many applications. Therefore, **OmniTrak** has an extensive set of options to define stopping conditions. An orbit could terminate by striking a material object like an electrode or dielectric. In this case, we often want to know the precise location of the collision. The following section describes the definition of material properties in the computational meshes and the mechanisms in **OmniTrak** to determine entrance positions. This section concentrates on alternative methods to set stopping conditions such as elapsed time and stopping planes. Note that when multiple conditions are defined, an orbit terminates when any single condition is satisfied.

OmniTrak always ends a particle orbit when it leaves the box defined by the *BOUNDARY* command of Section 4.3. In this case the program performs an interpolation over the last integration step to estimate values for position, momentum, kinetic energy, transit time and orbit length at the box boundary. Note that you can set the size of the orbit solution box larger than the size of the field solution box. For example suppose you have an electrostatic lens with localized fields, and you want to find the particle distribution in a focal plane that is outside the boundaries of the **HiPhi** solution. In this case, define an orbit solution box using the *BOUNDARY* command where one side corresponds to the desired focal plane. When an orbit leaves the **HiPhi** solution box the field interpolation routines return $E_x = 0.0$, $E_y = 0.0$ and $E_z = 0.0$. Therefore the orbit continues ballistically until it strikes the boundary of the orbit solution box or a stopping plane.

TMAX TMax **TMAX 50.0E-9**

Orbits stop when their transit time exceeds the quantity *TMax*. Enter the real-number value in seconds. **OmniTrak** performs an interpolation over the final integration step to determine particle parameters at the exact termination time. With this feature, the program can be useful to study time-of-flight detectors or to design isochronous systems. The *TMax* parameter can also be used to avoid infinite orbits (such as electrons trapped in a magnetic field). The default value is $TMax = \infty$.

SMAX SMax**SMAX 4.567**

Orbits stop when their total track length exceeds the quantity *SMax*. Enter the real-number value in units set by *DUnit*. For example, if the command `DUNIT = 100.0` appears in the script before the *SMAX* command, enter the track length in centimeters. **OmniTrak** performs an interpolation over the final integration step to determine particle parameters at the exact termination distance. The *SMax* parameter can also be used to avoid infinite orbits. The default value is $SMax = \infty$.

NSTEPMAX NStepMax**NSTEPMAX = 20000**

Orbits stop when the total number of integration steps exceeds *NStepMax*. The reason to include this parameter in **OmniTrak** is to prevent an infinite loops for circular or reflex motions. The default value is $NStepMax = 200,000$. Use this command to set an alternate value.

STOPPLANE POrient PPos**STOPPLANE ZUp 99.0**

The *STOPPLANE* command sets up a plane perpendicular to one of the axes for precise particle orbit termination. The real-number parameter is the position of the plane along the normal axis. Enter the value in units set by *DUnit*. The string parameter is a keyword that determines the function of the plane:

XDn: Particle stops if it crosses the plane (normal to *x*) while moving in the -*x* direction.

XUp: Particle stops if it crosses the plane (normal to *x*) while moving in the +*x* direction.

YDn: Particle stops if it crosses the plane (normal to *y*) while moving in the -*y* direction.

YUp: Particle stops if it crosses the plane (normal to *y*) while moving in the +*y* direction.

ZDn: Particle stops if it crosses the plane (normal to *z*) while moving in the -*z* direction.

ZUp: Particle stops if it crosses the plane (normal to *z*) while moving in the +*z* direction.

You can use up to six *STOPPLANE* commands, one for each option. **OmniTrak** interpolates over the final integration step to project particle parameters to the first plane encountered.

The next two commands do not terminate particle orbits but are closely related to the *STOPPLANE* command.

DIAGPLANE POrient PPos

DIAGPLANE XDn -2.0

In contrast to the action of a *STOPPLANE*, a particle orbit does not terminate when it crosses a *DIAGPLANE*. Instead, **OmniTrak** writes interpolated particle parameters (position, momentum, energy, transit time and track length) at the plane to the data arrays that normally hold the final particle parameters. These arrays are used in diagnostic routines such as *PARTLIST* and *PARTFILE*. The *DIAGPLANE* command is useful, for example, in solutions with space charge. Here particles must traverse the full solution-volume length to generate the correct total electric field; nonetheless, it may be desirable to find the beam distribution at a special location (such as a waist) rather than at the system boundary. The second parameter (a string) is one of the keywords introduced in the discussion of the *STOPPLANE* command: *XDn*, *XUp*, *YDn*, *YUp*, *ZDn* or *ZUp*. The real-number parameter is the position of the plane along the normal axis. Enter the value in units set by *DUnit*. You can include multiple *DIAGPLANE* commands – quantities for the orbit are recorded at the first plane encountered.

REFLECTPLANE POrient PPos

REFLECTPLANE ZDn 0.0

Reflection planes can be used to model periodic structures or systems with field symmetries. **OmniTrak** performs the following operations if a particle crosses a reflection boundary: 1) orbit parameters are interpolated to the plane, 2) the momentum normal to the reflection plane is reversed and 3) the orbit integration continues. For physical validity, both the particle distribution and the fields should have mirror symmetry about the reflection plane. In equilibrium, every particle that leaves through the reflection plane is replaced by an identical particle traveling in the opposite direction. In addition the field solution should satisfy the special Neumann condition.

The second parameter (a string) is one of the keywords introduced in the discussion of the *STOPPLANE* command: *XDn*, *XUp*, *YDn*, *YUp*, *ZDn* or *ZUp*. The real-number parameter is the position of the plane along the normal axis. Enter the value in units set by *DUnit*. You can include multiple *REFLECTPLANE* commands in the script.

7.5. Setting material properties

During an orbit integration, **OmniTrak** must continually identify elements occupied by the particle to collect appropriate points for field interpolations. As a result the program always knows the region numbers of occupied elements on the electric and/or magnetic field meshes. This information can be used to determine if the particle has entered a material object. Region numbers defined on the electric or magnetic meshes are associated with one of the following properties in **OmniTrak**:

- *Vacuum*: orbits pass unimpeded through vacuum elements

- *Material*: an orbit terminates when it enters a material element.

- *Secondary*: an electron terminates when it enters a secondary material and is replaced by a secondary electron with emission parameters and weighting determined by Monte Carlo techniques.

Chapter 13 covers models for electron secondary emission.

When an electric field solution from **HiPhi** is loaded, **OmniTrak** sets fixed-potential regions (electrodes) and dielectrics with $\epsilon_r \neq 1.0$ to the *Material* condition and regions with $\epsilon_r = 1.0$ to the *Vacuum* condition. You can over-ride these assignments with the following commands.

VACUUM E RegNo

VACUUM (E) = 5

Sets all elements in the electric field mesh with region number *RegNo* (an integer from 1 to 127) to the *Vacuum* condition.

MATERIAL E RegNo

MATERIAL E 5

Sets all elements in the electric field mesh with region number *RegNo* to the *Material* condition.

Similarly, when a magnetic field solution from **Magnum** or a **TriComp** program is loaded, **OmniTrak** regions with fixed vector potential or with

$\mu_r \neq 1.0$ to the *Material* condition and regions with $\mu_r = 1.0$ to the *Vacuum* condition. You can reassign regions with the the following commands.

VACUUM B RegNo

VACUUM B 3

Sets all elements in the two- or three-dimensional magnetic field mesh with region number *RegNo* to the *Vacuum* condition.

MATERIAL B RegNo

MATERIAL(B) = 3

Sets all elements in the two- or three-dimensional magnetic field mesh with region number *RegNo* to the *Material* condition.

In solutions with both electric and magnetic fields, an orbit stops in the first material region it encounters in either mesh. If a particle enters a *Material* element, **OmniTrak** identifies the position on the surface facet corresponding to the particle entry point. The final particle parameters (position, momentum, energy, transit time, track length) are projected back to this point. The calculation is quite complex on an arbitrary conformal mesh. Nonetheless, **OmniTrak** exhibits good reliability for surface identification. Ideally, the choice of time step *Dt* should guarantee that the particle moves a distance less than an element width in a step. The procedure may fail for larger time steps where the particle penetrates several element layers into a material in a step.

7.6. Application example

To illustrate some single-particle tracking techniques, we shall consider the design of a miniature electrostatic spectrometer to estimate the energy of ions leaking from a low-density plasma chamber. Figure 7.2 shows the geometry. Ions enter through a small aperture in the grounded wall of the chamber (1). A shield electrode (2) biased to -100 V prevents electrons from flowing into the device. Two 90° cylindrical-section electrodes (3 and 4) with a voltage difference of V_0 create a field that varies approximately as:

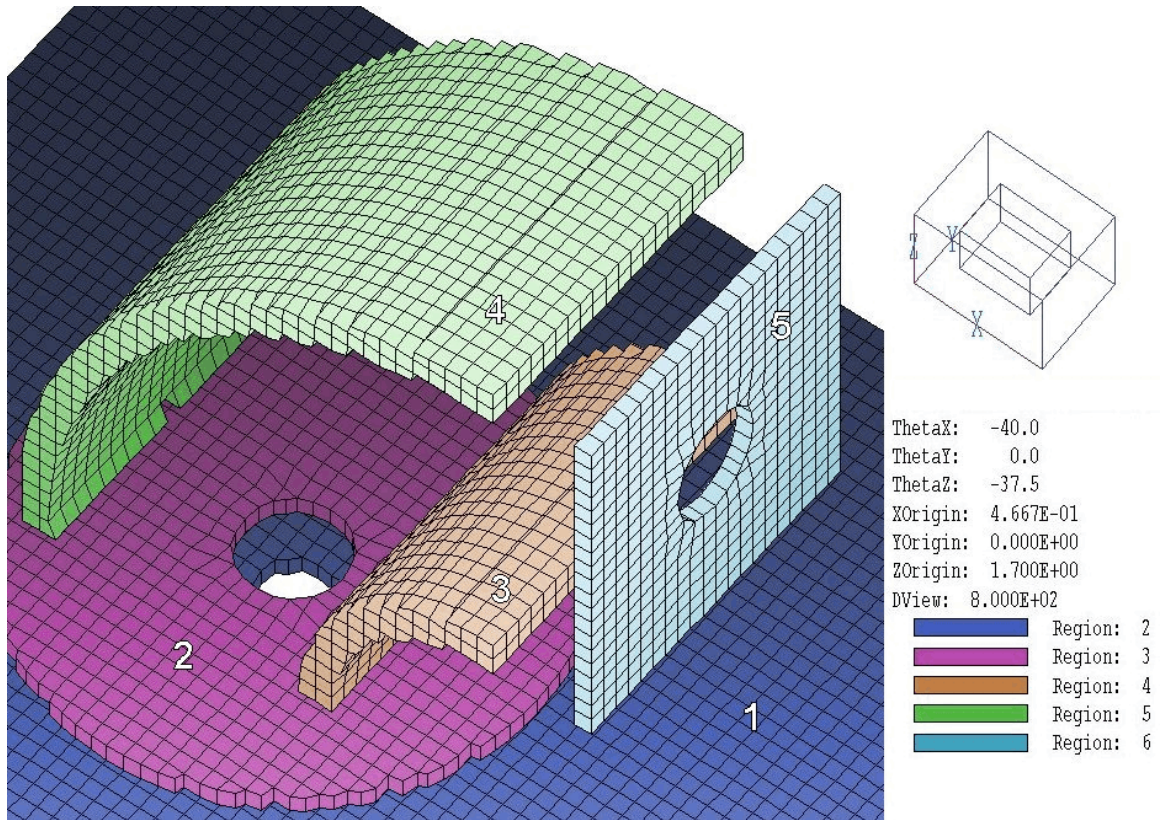


Figure 7.2. Geometry of the plasma ion energy analyzer. 1) Plasma chamber wall with extraction aperture. 2) Electron suppressor. 3) Inner deflection plate. 4) Outer deflection plate. 5) Detector shield. The assembly is located inside a grounded box (not shown).

In the equation r is the distance from the axis of the cylinders and r_i and r_o are the radii of the inner and outer surfaces. An exit plate (5) shields the detector volume from the bending field. The axes of the entrance and exit apertures enter the deflection electrodes at the radius $R = (r_i + r_o)/2 = 2.0$ cm. In an ideal cylindrical field, an ion passes from the entrance to the exit aperture when

$$\frac{m_o v_o^2}{r} = qE_r ,$$

or

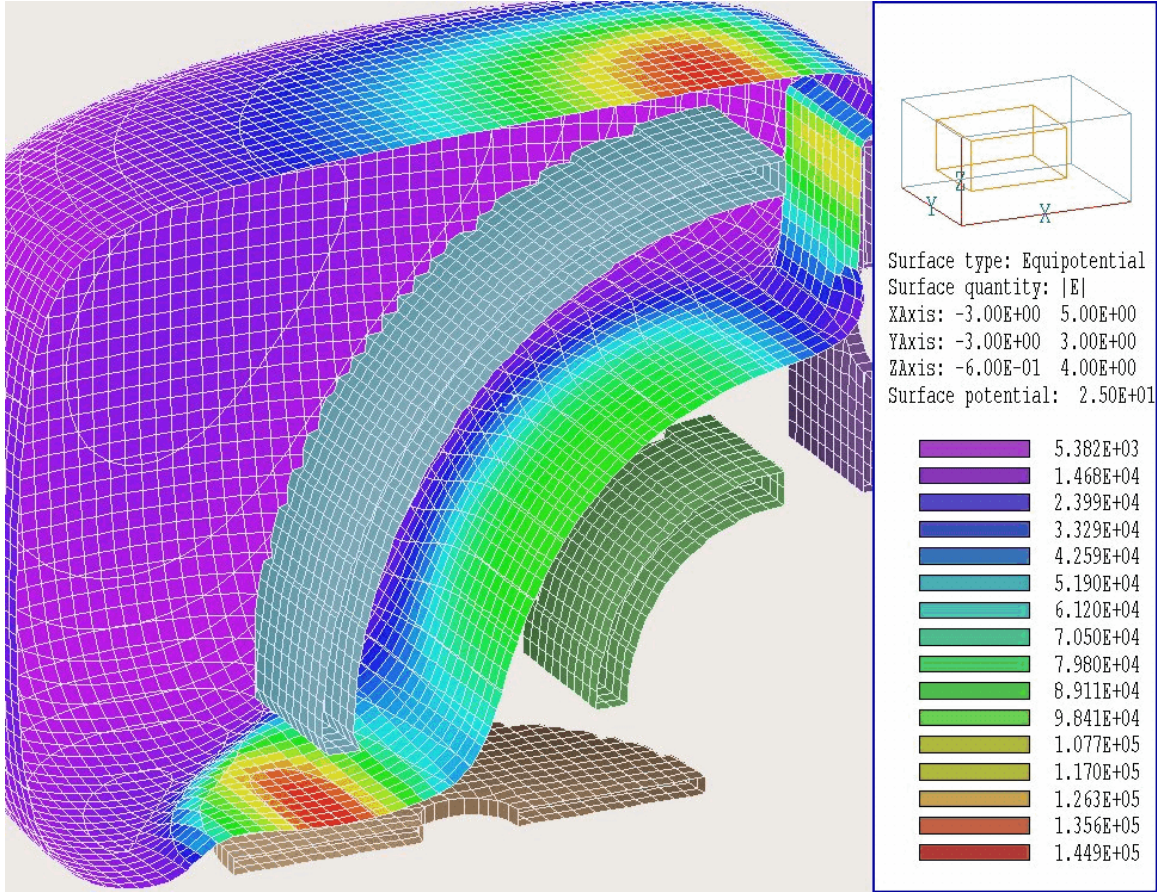


Figure 7.3. Equipotential surface at $\phi = 25$ V, color-coding on the surface according to $|E|$.

$$T_o = \frac{qV_o}{2 \ln (r_o / r_i)} .$$

In the second equation T_o is the kinetic energy of ions that pass through the center of the output aperture. For $r_i = 1.2$ cm and $r_o = 2.8$, the kinetic energy (in eV) is related to the electrode voltage by $T_o = 0.59V_o$. The mesh file `Plasma_Analyzer.MIN` defines the geometry shown in Fig. 7.1. We have employed a fairly coarse mesh so the example runs quickly. The voltage levels in the file `Plasma_Analyzer.HIN` are optimized for ions with $T_o = 1000$ eV. The total voltage difference is 1356 V and the bipolar electrode voltages are chosen so that the potential along the beam axis (at R) is approximately zero. Figure 7.3 shows the electrostatic solution. The equipotential surface at $\phi = 25$ V encloses the outer

deflection electrode. Color coding shows the variation of $|\mathbf{E}|$ over the surface. The electric field magnitude in the deflection gap is smaller than the value for ideal cylindrical electrodes because: 1) the electrodes have finite length along y and 2) the entrance and exit plates act to short the fields.

We shall first use **OmniTrak** to calibrate the detector. Using a C^+ ion with $T_0 = 1 \text{ keV}$, we seek to adjust the electrode potentials so that the ion passes through the middle of the exit aperture. The following input file is used for the run:

```
* File PLASMA_ANALYZER
FIELDS
  DUnit = 100.0
  EField3D = Plasma_Analyzer.HOU 1.000
END
PARTICLES Track
  Dt = 5E-9
  TMax = 1.0E-6
  Material(E) = 2
  Material(E) = 3
  Material(E) = 4
  Material(E) = 5
  Material(E) = 6
  StopPlane XUp 3.5
  PList
* Mass Chrg   Eng      x      y      z      px      py      pz
  12.0  1.0  1000.0  0.00  0.00 -0.50  0.0  0.0  1.0
  End
END
DIAGNOSTICS
  PartList Detector
END
ENDFILE
```

Consider first the commands of the *FIELDS* section. The line `DUnit = 100.0` specifies that all coordinates will be entered in units of centimeters. The *EField3D* command loads the electrostatic solution of Fig. 7.3. The unity value of the parameter *EMult3D* means that there is no scaling of the electrostatic potential. Later we will vary this number to adjust the spectrometer. In the *PARTICLES* section we set a time step manually. The velocity of C^+ ions at 1 keV is $1.3 \times 10^5 \text{ m/s}$. The time to cross an element of width 10^{-3} m is 7.9 ns. The choice $Dt = 5.0 \text{ ns}$ means that particles move less than one element per time step. The next command, `TMax = 1.0E-6`,

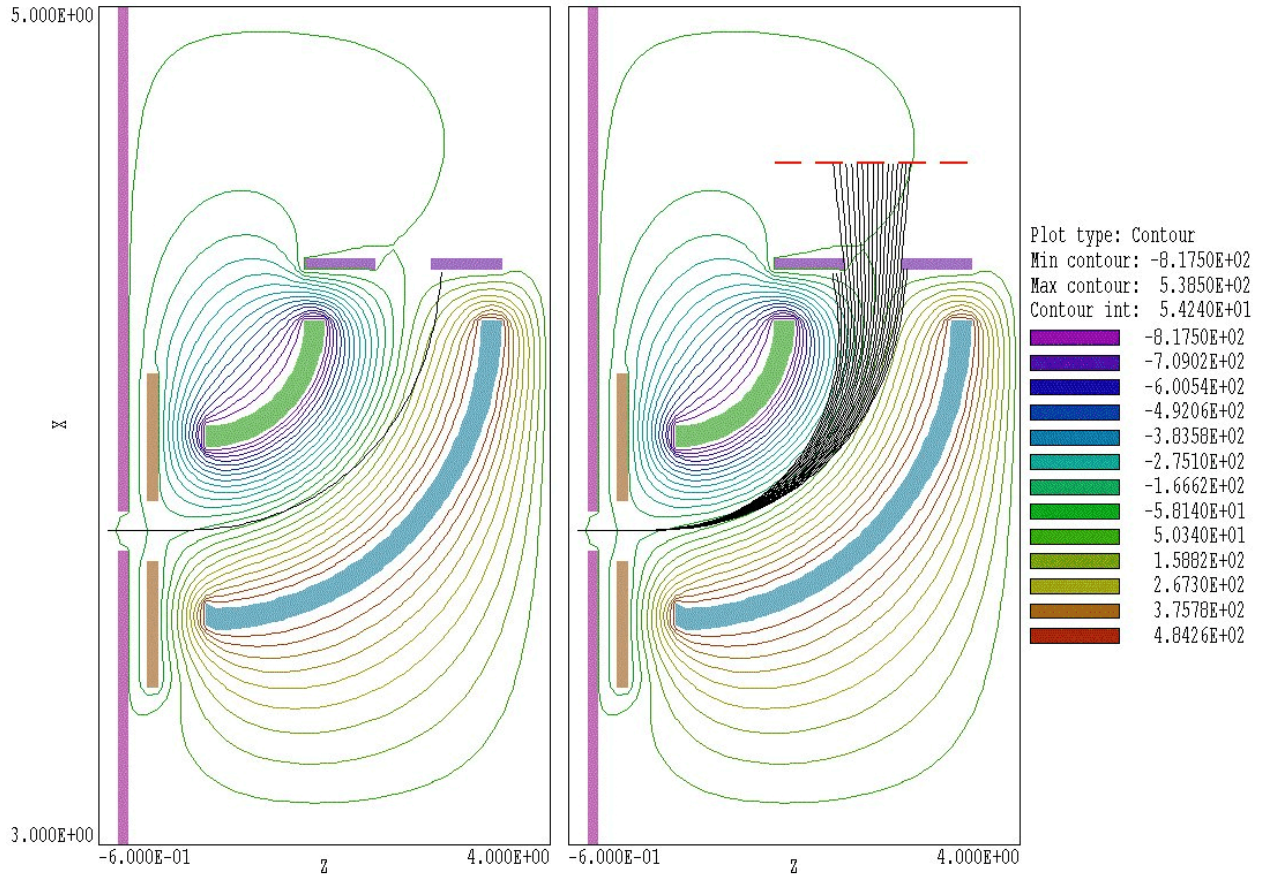


Figure 7.4. Equipotential lines and projected particle orbits in the plane $y = 0.0$. a) Single C^+ ion with $T_0 = 1000$ eV. b) Multiple orbits over the kinetic energy range $800 \text{ eV} \leq T_0 \leq 1200$ eV. Dashed red line shows the stopping plane.

limits the maximum orbit lifetime. Setting $TMax$ is a useful precaution while testing a run to avoid a runaway solution. Commands of the form `MATERIAL (E) = 2` set all regions in the solution volume except the vacuum to the *Material property*. Therefore particles will stop if they strike an electrode. These commands are included in this case for illustration – **OmniTrak** sets all fixed-potential regions to the *Material* condition by default. The command `StopPlane XUp 3.5` sets a precise stopping plane at the detector location. The *PList* command structure starts a single C^+ ion with a kinetic energy of 1 keV. The ion is created on the detector beam axis slightly upstream of the aperture in the plasma chamber wall. The initial velocity is directed along the local beam axis (z

direction). The single command in the *DIAGNOSTICS* section calls for a listing of particle parameters and the creation of a standard PRT file of parameters at the stopping plane.

Figure 7.4a shows equipotential lines and the resulting particle orbit projected in the $y = 0.0$ plane. Because the fields are smaller than the ideal values for a cylindrical system, the particle is not deflected to the detector axis. Additional runs with values of *EMult3D* larger than unity show that a multiplication factor of 1.228 ensures that outgoing particle orbits pass through the center of the detector aperture parallel to the x axis. The adjusted detector energy calibration is $T_0 = 0.48V_0$. We can get an idea of the dispersive properties of the spectrometer by expanding the *PList* structure to include ions with different energies over the range 800 to 1200 eV. The following values were prepared with a spreadsheet and pasted into the script, giving the orbits show in Fig. 7.4b.

```
* Mass Chrg   Eng      x      y      z      px      py      pz
* =====
12.0  1.0    800.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0    820.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0    840.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0    860.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0    880.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0    900.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0    920.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0    940.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0    960.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0    980.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0   1000.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0   1020.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0   1040.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0   1060.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0   1080.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0   1100.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0   1120.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0   1140.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0   1160.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0   1180.0   0.00   0.00  -0.50   0.0   0.0   1.0
12.0  1.0   1200.0   0.00   0.00  -0.50   0.0   0.0   1.0
```

To complete the study we shall introduce a spread in initial particle position and energy to get a sense of the detector energy resolution. For this purpose we need a large number of orbits for good statistics. One option is to use the **TDist** utility that automatically generates PRT files. We want to create a distribution of particles initially

Particle distribution in T, file plasma_analyzer.PRT

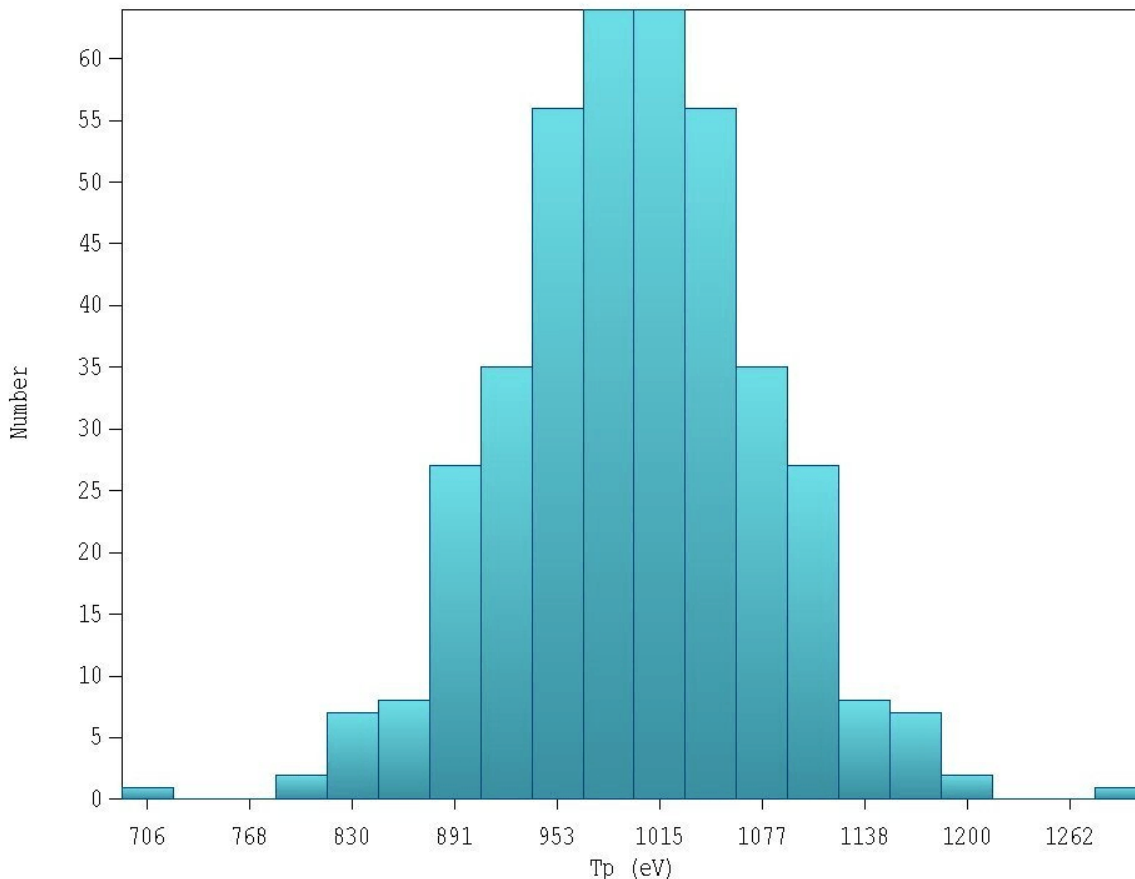


Figure 7.5. Energy distribution created by **TDist** and displayed with the distribution capabilities of **OmniView**.

moving in the z direction distributed over the area of the extraction port in the plasma chamber. To make the simulation more interesting, we also introduce a Gaussian spread of kinetic energy about the mean value with a half-width at half-maximum of 100 eV.

The **TDist** input file is shown below. The **TDist** manual gives a complete explanation of commands. Briefly, the first set of commands defines a beam of 400 C^+ ions distributed over circular area of radius 0.2 cm at position $z = -0.5$ cm. The central energy of 1250 eV will be varied for the study. The data lines in the **TDist** structure define an energy distribution with Gaussian weighting. We run **TDist** to create the file `Plasma_Analyzer.PRT`. We can

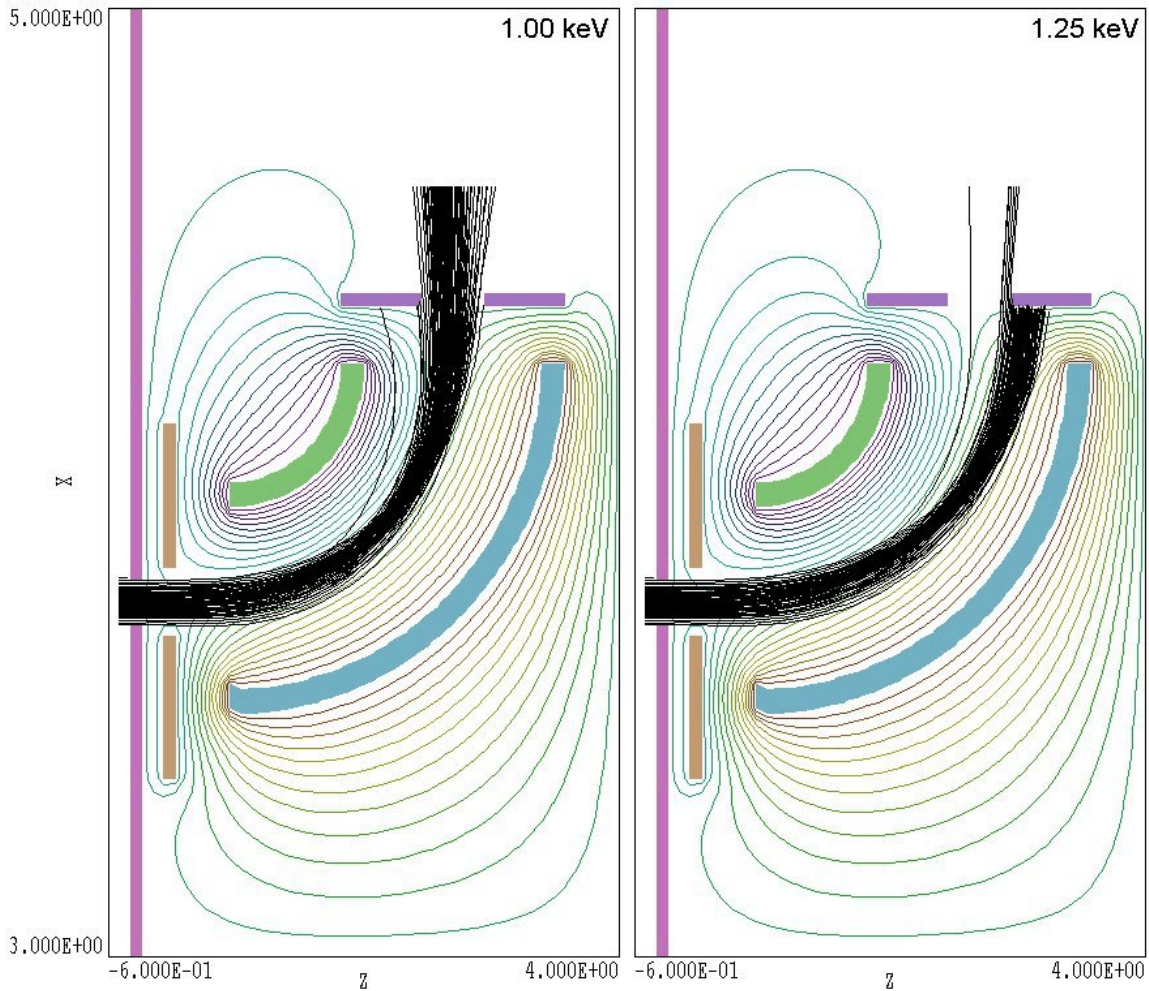


Figure 7.6. Equipotential lines and orbits projected in the plane $y = 0.0$ for a spatially-extended group of particles with a ± 100 eV energy spread for two values of the central energy.

then use the distribution capabilities of **OmniTrak** (Chapter 19) to load the resulting PRT for analysis. Figure 7.5 shows a histogram of the kinetic energy distribution of entering particles. Figure 7.6 shows orbits projected in a plane at $y = 0.0$ for the two central energies.

```
* File PLASMA_ANALYZER.DST
RestMass 12.0
Charge 1.0
Def Circ 0.2 10 40
Offset 0.00 0.00 -0.50
* Change this line to vary the central energy
Energy 1.25E3
TDist
-300.0 0.00012
-275.0 0.00052
-250.0 0.00193
-225.0 0.00633
-200.0 0.01832
-175.0 0.04677
-150.0 0.10540
-125.0 0.20961
-100.0 0.36788
-75.0 0.56978
-50.0 0.77880
-25.0 0.93941
 0.0 1.00000
 25.0 0.93941
 50.0 0.77880
 75.0 0.56978
100.0 0.36788
125.0 0.20961
150.0 0.10540
175.0 0.04677
200.0 0.01832
225.0 0.00633
250.0 0.00193
275.0 0.00052
300.0 0.00012
End
EndFile
```

8. Self-consistent beam forces in the SCharge mode - PARTICLES section

8.1. Beam-generated forces

The space-charge and current of particle beams contribute to the total electric and magnetic fields in the solution volume. In the limit of zero current, the field components generated by the beam are negligible compared to the applied fields and can be neglected. Beam-generated fields play a significant role in the dynamics of high-current beams and may be important even at relatively low currents (*i.e.*, focal limits for low-emittance probe beams). Forces resulting from beam charge and current are discussed in the two texts supplied with **OmniTrak**:

S. Humphries, **Principles of Charged Particle Acceleration** (Wiley, New York, 1986),

S. Humphries, **Charged-particle Beams** (Wiley, New York, 1990)

The ratio of transverse forces associated with the magnetic and electric fields of a beam with axial velocity v and relativistic velocity factor $\beta = v/c$ is approximately equal to

$$\frac{F_m}{F_e} \approx \beta^2.$$

An implication is that magnetic forces are small in ion injectors. For example, a 200 keV proton beam has $v = 6.2 \times 10^6$ m/s or $\beta^2 = 4.3 \times 10^{-4}$. Generally beam-generated magnetic field calculations need be included only in simulations of relativistic electron beams.

A steady-state solution for a charged-particle gun is challenge for numerical simulations. The space-charge and current distributions (and hence the total electric and magnetic fields) depend on the orbits of the model particles. The particle orbits depend, in turn, on the fields. It is impossible to generate a direct boundary-value solution except in systems with simple geometries. Instead, ray-tracing codes like **OmniTrak** use an iterative procedure with the following steps:

- 1) Model particle orbits are calculated in the applied fields.
- 2) Spatial distributions of beam charge-density and current are determined from the orbits.
- 3) Fields are updated to reflect the addition charge and current contributions.
- 4) New orbits are calculated.

Steps 2, 3 and 4 are repeated for several cycles. The procedure usually converges to a self-consistent steady-state solution if the beam contributions to space-charge and current are added gradually.

In this section we shall concentrate on particle generation in the SCharge mode by list input. Chapter 9 covers automatic particle creation from source surfaces to satisfy conditions for space-charge-limited emission. In the *SCHARGE* mode particle data lines in the *PLIST* or *PFILE* structures must contain an addition parameter. The real number gives the model particle current. Every data line must include contains ten entries in the following order:

| | | | | | | | | | |
|------|--------|---------|---|---|---|----|----|----|---|
| Rest | Charge | Kinetic | x | y | z | fx | fy | fz | I |
| mass | | energy | | | | | | | |

Here, I is the current (in A) assigned to a model particle. Note that the current is always a positive number. **OmniTrak** obtains the correct sign of the space-charge and the direction of current from the entry in the *Charge* column. You could use lists with included current in the *TRACK* mode – in this case the program ignores the current entry and carries out a single-particle calculation.

OmniTrak must allocate memory to store the extra mesh quantities to carry out the electric and/or magnetic field solutions. Therefore, a large mesh that fits in memory for single-particle tracking may not work in the space-charge mode. The space-charge density is determined by the following method. A beam is represented by a large number N of model particles. If the beam has total current I_0 , then each model particle carries a current $I = I_0/N$ (in the special case where current is weighted uniformly). A model particle orbit consists of a sequence of positions x_i separated by uniform time step Δt . For each position, **OmniTrak** identifies the occupied element and increases the total element charge by an amount $I\Delta t$. There are two implications for the user:

- The scale length for resolution of variations of the space-charge density equals the local size of elements.
- The space-charge density will be granular if particles traverse several elements per time step.

The granularity of the space-charge density is determined by statistics. Larger numbers of particles give a more continuous function and smoother interpolations of electric field.

8.2. Control of orbit and field updates

When current is assigned to model particles, the electric and/or magnetic field and particle orbits must be updated several times to ensure convergence of the self-consistent solution. Field recalculations are controlled by the commands described in Sect. 4.2. The following list summarizes the important commands:

ResTarget E 5.0E-7

Set a value for the relative accuracy of the iterative electric field solution

MaxCycle E 2500

Set the maximum number of iteration cycles in the electric field solution.

Omega E 1.95

Set the over-relaxation parameter for the electric field solution

ResTarget B 5.0E-7

Set a value for a parameter that controls the accuracy of the 3D magnetic field solution

MaxCycle B 2500

Set the maximum number of iteration cycles for the 3D magnetic field solution.

Omega B 1.95

Set the over-relaxation parameter for the 3D magnetic field solution

The commands introduced in this section control the cycles of field solution and orbit updates.

NCYCLE NCycle

NCYCLE = 12

The integer parameter *NCycle* is the number of recalculation cycles. One cycle consists of calculation of the model particle orbits (accompanied by deposition of space-charge) followed by relaxation iterations to correct the electric field solution. The number of cycles may be small (*NCycle* ~ 5) for low current beams with laminar orbits or large (*NCycle* ~ 20) for complex orbits or intense beams. There are two indicators of convergence: 1) fewer iterations are required to achieve the target residual in the field solutions and 2) the final positions of particles change little from cycle to cycle.

AVG E AvgE

AVG(E) = 0.20

The *AVG* command with the option *E* controls how space charge is apportioned from cycle to cycle. This feature is important for intense beams that strongly affect the electric fields. In this case, if we calculate space-charge only on the basis of orbits in the present cycle, the solution may oscillate between two limits without converging. The *AVG* command allows you to average the predicted space-charge in the present cycle with distributions from previous cycles. Proper use of this feature ensures good convergence even for intense beams. If Q_i^{n-1} is the space-charge of element *i* on the previous iteration cycle and ΔQ_i^n is an increment of charge determined for the present orbit integration, then the element charge changes according to

$$Q_i^n = (1 - \text{AvgE}) Q_i^{n-1} + (\text{AvgE}) \Delta Q_i^n.$$

A value of $\text{AvgE} = 1.0$ means that only the present orbits contribute to the space-charge. A value $\text{AvgE} \ll 1$ means that there is averaging over several cycles. In this limit, you must use a higher value of *NCycle* to ensure convergence. In general pick $\text{NCycle} > 1/\text{AvgE}$. The default value is $\text{AvgE} = 0.30$.

8.3. Application example

The following benchmark example (`SCHARGE_TEST`) illustrates techniques for and the accuracy of space-charge assignment and field recalculation. The example treats the expansion of a one-dimensional, uniform current-density planar beam in vacuum. The simulation is a stringent test because electric fields arise entirely from beam space charge. Figure 8.1 shows the geometry. The solution volume covers an area $1.0 \text{ cm} \times 1.0 \text{ cm}$ in x and y and a length 0.0 to 6.0 cm in z . The mesh is relatively coarse with cubic elements of width 0.1 cm . The Neumann condition applies on all boundaries except the upper boundary in y , which is set to a constant potential $\phi = 0.0 \text{ V}$. The input electric field has $\phi = 0.0$ at all points. In this case, **HiPhi** runs for a minimum number of cycles and then creates a dummy file with mesh information and uniform potential. A portion of the **OmniTrak** script `SCHARGE_TEST.OIN` is shown below. Protons with kinetic energy 100 keV and momentum directed along z are created just inside the boundary at $z = 0.0$ using the *PLIST* input option. Each model particle carries a current $31.0 \times 10^{-4} \text{ A}$. The simulation includes a total of 50 particles at the center of element facets in the x - y plane over the range $x = 0.0$ to 1.0 cm and $y = 0.0$ to 0.5 cm . This setup is equivalent to a uniform sheet beam with infinite extent along x , width 1.0 cm along y , midplane at $y = 0.0$, and line current $J = 31.0 \text{ A/m}$. Because the orbits are simple, we can use a relatively small number of cycles ($NCycle = 8$) and large value of the space-charge averaging factor ($AvgE = 0.9$). The command `EDump Scharge_TestP.HOU` creates an output file of the beam-generated electric field in **HiPhi** format.

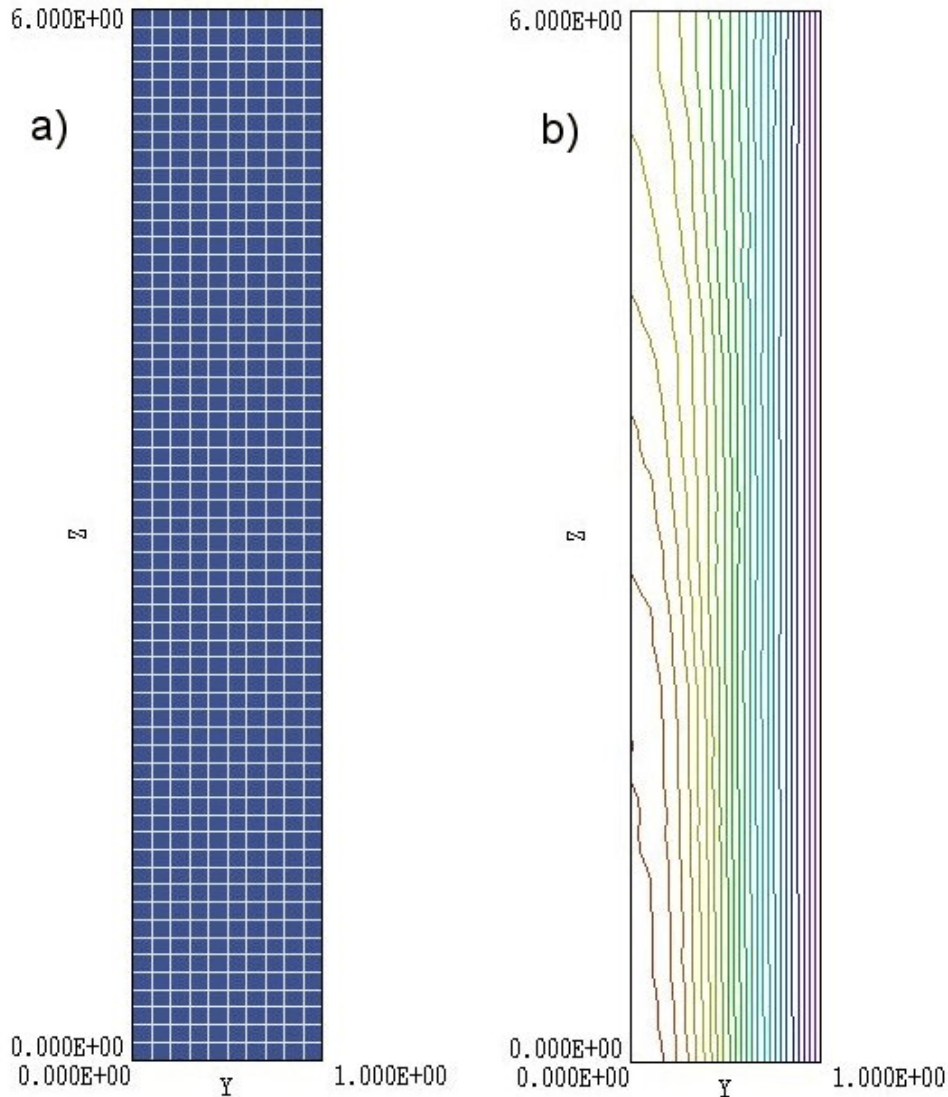


Figure 8.1. Expansion of a planar beam in vacuum. *a)* Mesh projected in y-z plane. *b)* Equipotential contours from 0.0 to 3000 V. Beam waist at the bottom, beam midplane on the left-hand side.

Formulas in the text **Charged Particle Beams** (Sect. 9.1) predict on-axis potential at the beam waist is 3000 V with a potential drop of 1000 V across the beam. The planar generalized perveance is $K_y = 2.000$. The envelope of an ideal sheet beam drifting a distance L should increase by an amount $\Delta y = K_y L^2/2$. Despite the coarse mesh, the on-axis potential at $z = 0.0$ in the simulation is within 1% of the theoretical value. In order to make a comparison of beam expansion, note that the charge enclosed within the outer model particles (starting at $y = 0.45$ cm) is a fraction $0.45/0.50$ times the theoretical envelope charge. Therefore, the final position of the outer orbit should be:

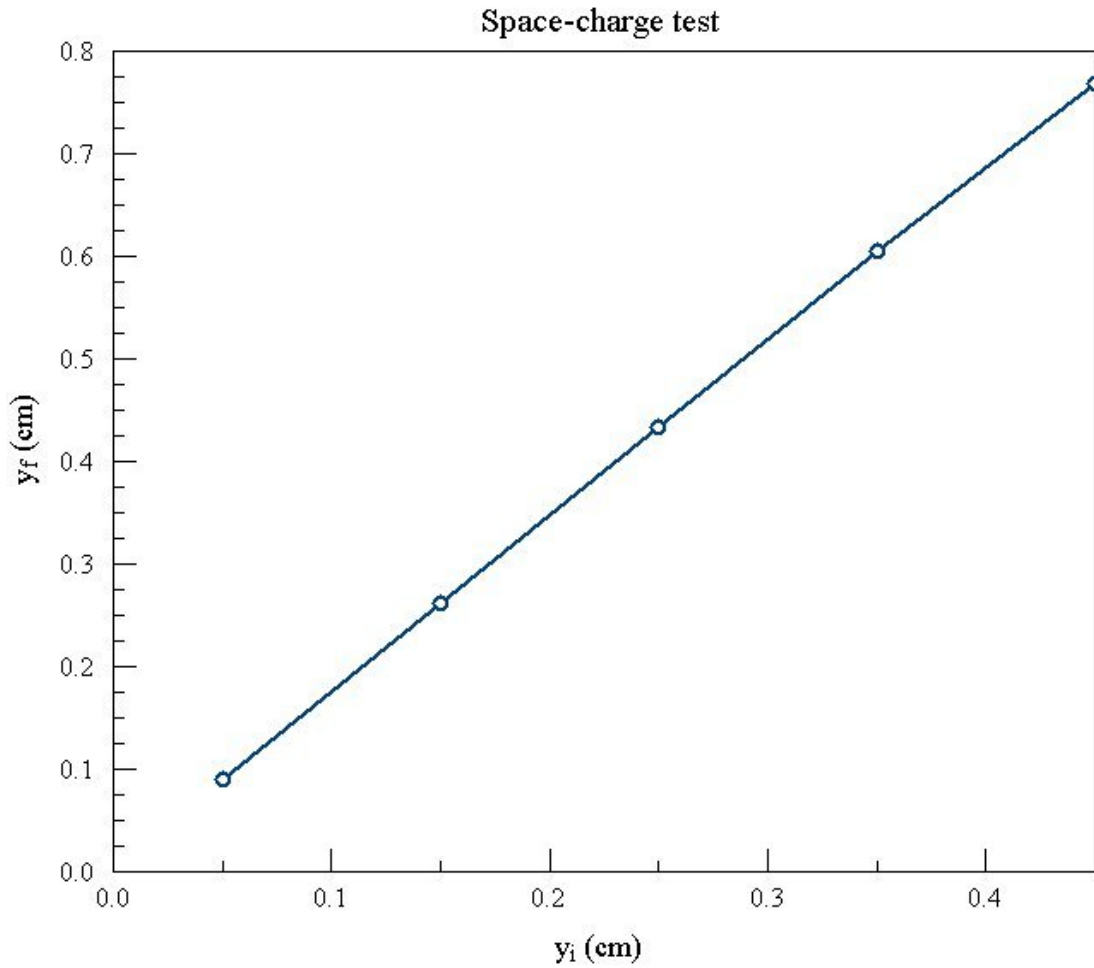


Figure 8.2. Expansion of a sheet beam, final displacement versus initial displacement.

$$y_f = 0.0045 + (0.45/0.50) (0.06)^2/2 = 0.00774 \text{ m} = 0.774 \text{ cm}.$$

Figure 8.2 shows a plot of the final orbit positions in y versus the initial positions. The outer orbits have a final position $y = 0.768 \text{ cm}$, within 0.8% of the theoretical value. The figure also shows that the calculated transverse electric field follows a linear variation.

```

* FILE SChARGE_TEST.OIN
FIELDS
  DUnit = 100.0
  EField3D = SCharge_Test.HOU
END
PARTICLES SCharge
  Dt = 2.0E-10
  TMax = 10.0E-9
  NCycle = 8
  AvgE = 0.90
  PList
* Mass Chrg   Eng      x      y      z      px      py      pz      Current
* =====
  1.0  1.0  1.0E5  -0.45  0.05  1.0E-6  0.00  0.00  1.00  31.0E-4
  1.0  1.0  1.0E5  -0.45  0.15  1.0E-6  0.00  0.00  1.00  31.0E-4
  1.0  1.0  1.0E5  -0.45  0.25  1.0E-6  0.00  0.00  1.00  31.0E-4
  ...
  1.0  1.0  1.0E5   0.45  0.15  1.0E-6  0.00  0.00  1.00  31.0E-4
  1.0  1.0  1.0E5   0.45  0.25  1.0E-6  0.00  0.00  1.00  31.0E-4
  1.0  1.0  1.0E5   0.45  0.35  1.0E-6  0.00  0.00  1.00  31.0E-4
  1.0  1.0  1.0E5   0.45  0.45  1.0E-6  0.00  0.00  1.00  31.0E-4
  End
END
DIAGNOSTICS
  PartList SCharge_Test
  EDump SCharge_TestP.HOU
END
ENDFILE

```

9. Self-consistent Child law emission - PARTICLES section

9.1. Mechanics of the space-charge emission model

If you define one or more *source surfaces* in the *SCharge* mode, **OmniTrak** will automatically launch particles. In this mode, the program generates charged particles near the marked surface and sets particle currents to satisfy conditions consistent with Child's law. Methods to define source surfaces were discussed in Sect. 7.2. In this section, we shall concentrate on the special techniques required for Child-law emission.

After identification of source facets, **OmniTrak** must generate a set of initial particle parameters that define an *emission surface* a distance $DEmit$ from the physical surface. This is because it is not possible to emit particles directly from the source surface. Under the conditions for space-charge-limited emission, particle orbits would not advance because the electric field at the source approaches zero. As an alternative, **OmniTrak** creates particles on a surface that lies a short distance from the physical surface. The potential difference between the physical and emission surfaces is used in conjunction with analytic formulae to assign kinetic energy and current to the particles.

To process a source surface **OmniTrak** sequentially reads the facet list. For each facet the program uses the orientation information to assign node coordinates to ensure that surface normal vectors will point out of the fixed potential element and into the vacuum element (Fig. 9.1). As in the *TRACK* mode, the facet is divided into $N_{Divide} \times N_{Divide}$ sections using the normal coordinates (u, v) of the surface. For example, when $N_{Divide} = 2$ the first model particle on a facet extends over the range $-1.0 < u < 0.0$ and $-1.0 < v < 0.0$. The generation point has the normal coordinates $u_s = -0.5, v_s = -0.5$. The corresponding generation position is given by:

$$x_s = N_0(u_s, v_s) x_0 + N_1(u_s, v_s) x_1 + N_2(u_s, v_s) x_2 + N_3(u_s, v_s) x_3,$$

$$y_s = N_0(u_s, v_s) y_0 + N_1(u_s, v_s) y_1 + N_2(u_s, v_s) y_2 + N_3(u_s, v_s) y_3,$$

$$z_s = N_0(u_s, v_s) z_0 + N_1(u_s, v_s) z_1 + N_2(u_s, v_s) z_2 + N_3(u_s, v_s) z_3,$$

where the shape functions are $N_0 = (1-u)(1-v)/4$, $N_1 = u(1-v)/4$, $N_2 = uv/4$, $N_3 = (1-u)v/4$. The surface area of the facet segment corresponding to the range of normal coordinates is calculated with a 4×4 Gaussian-quadrature integral. The following equations give the surface normal vector \mathbf{n} at the emission location:

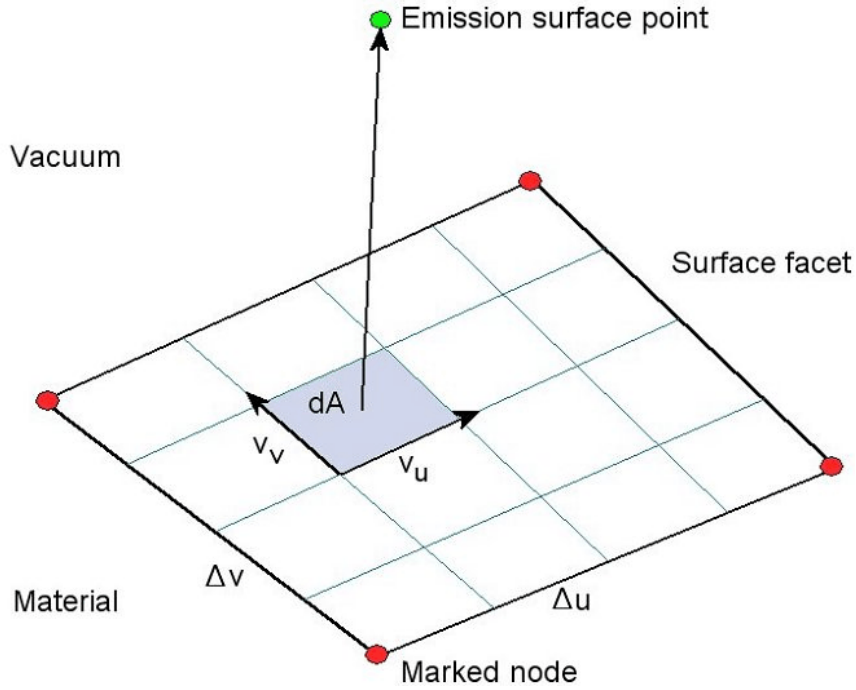


Figure 9.1. Projection of segments of a surface facet to an emission surface.

$$\mathbf{x}_u = \left[\begin{array}{c} \frac{\partial x_s}{\partial u_s} \mathbf{x} + \frac{\partial y_s}{\partial u_s} \mathbf{y} + \frac{\partial z_s}{\partial u_s} \mathbf{z} \\ \frac{\partial x_s}{\partial v_s} \mathbf{x} + \frac{\partial y_s}{\partial v_s} \mathbf{y} + \frac{\partial z_s}{\partial v_s} \mathbf{z} \end{array} \right], \quad \mathbf{x}_v = \left[\begin{array}{c} \frac{\partial x_s}{\partial u_s} \mathbf{x} + \frac{\partial y_s}{\partial u_s} \mathbf{y} + \frac{\partial z_s}{\partial u_s} \mathbf{z} \\ \frac{\partial x_s}{\partial v_s} \mathbf{x} + \frac{\partial y_s}{\partial v_s} \mathbf{y} + \frac{\partial z_s}{\partial v_s} \mathbf{z} \end{array} \right].$$

$$\mathbf{L} = \mathbf{x}_u \times \mathbf{x}_v, \quad \mathbf{n} = \mathbf{L} / |\mathbf{L}|.$$

The particle emission position is given by:

$$x_p = x_s + DEmit \mathbf{n}_x,$$

$$y_p = y_s + DEmit \mathbf{n}_y,$$

$$z_p = z_s + DEmit \mathbf{n}_z.$$

The particle charge q and mass m are set using stored quantities for the corresponding marked region. The program saves the normal vector components to use for setting the initial direction of particles leaving the emission surface.

The space-charge emission model is based on the method described in S. Humphries, J. Comp. Phys. **125**, 488 (1996). This method, extensively tested in **Trak**, ensures correct assignment of space charge in the region between the source and emission surfaces as well as in the bulk of the solution volume. Using the present field solution, the program calculates the values of electrostatic potential at the start points of each model particle. **OmniTrak** employs a second-order least-squares fit procedure that gives high-accuracy even if the emission surface is within one element length of the source. Using the difference in potential between the source and emission points ($\Delta\phi$) the program calculates current density and particle kinetic energy from the equations:

$$j = \frac{4\epsilon_o}{9} \sqrt{\frac{2q}{m} \frac{\Delta\phi^{3/2}}{DEmit^2}},$$

$$T_o = \sqrt{\frac{2q\Delta\phi}{m}}.$$

OmniTrak determines model particle currents from the current density j and the calculated area on the source facet segment. To set the charge in the gap between the source and emission surfaces, the program assigns a particle kinetic energy equal to $4T_o/9$ and sets the momentum components from the surface normal vector so that the orbit is directed back toward the source. A flag is set so that the field interpolation routine returns zero values for the electric field; therefore, the particle travels backward at constant velocity. The resulting charge assignment ensures correct values of potential and normal electric field at the emission surface. When backtracking is complete, the program assigns the full kinetic energy T_o and momentum components directed away from the source and tracks particle orbits through the solution volume.

9.2. Space-charge emission control commands

This section introduces commands that are specific to the control of Child-law emission from source surfaces in the *SCHARGE* mode. The *EMIT* command performs the critical function of defining marked regions and assigning emission surface properties.

EMIT NReg Mass Charge DEmit NDivide

EMIT 4 1.0 1.0 0.05 1

The *EMIT* command in the *SCHARGE* mode is similar to that of the *TRACK* mode – there is one additional parameter. The command specifies region number *NReg* (integer) as a source surface and sets associated emission properties. **OmniTrak** issues an error message if the region does not define a valid source surface (*i.e.*, sets of nodes on the edges of facets between *Material* and *Vacuum* elements). Up to twenty *EMIT* commands may appear in the *PARTICLES* section. The real-number parameter *Mass* is the mass of emitted particles in AMU. If the number 0.0 appears in this position, **OmniTrak** assigns the electron mass. The real number value *Charge* is the particle charge relative to the magnitude of the electron/proton charge. As an example, α particles have *Mass* = 4.0 and *Charge* = 2.0. To specify electrons, set *Mass* = 0.0 and *Charge* = -1.0. The parameter *DEmit* is the distance from the source surface to the emission surface. Specify the quantity in units set by *DUnit*. Generally, *DEmit* should be small so that the emission surface closely conforms to the source. On the other hand, values of *DEmit* smaller than the local element width may result in inaccurate field interpolations. As a rule, set *DEmit* equal to about 1.5 times the local element width along the direction of particle propagation. The final parameter *NDivide* (integer) governs how many particles are emitted per surface facet. It equals to number of divisions along the normal axes of the facet. Figure 9.1 shows a case with *NDivide* = 4. The number of particles per facet equals *NDivide*².

JLIMIT JLimit

JLIMIT = 5.6E4

OmniTrak can model mixed space-charge and source-limited emission. The *JLIMIT* command sets a limit for the previously-defined source region. Therefore, the command should immediately follow the associated *EMIT* command. The real-number value *JLimit* is the source-limited current density (in A/m²). The program calculates the Child-law value at the emission surface and projects it back to the source surface. The program then chooses the smaller of this value or *JLimit*.

DTHETA RegNo DTheta
DTHETA(6) = 2.5

This command assigns an angular spread to particles leaving the emission surface. The *DTHETA* command sets a value for the previously-defined source region. Therefore, the command should immediately follow the associated *EMIT* command. Enter the value *DTheta* in degrees. A polar angle in the range 0.0 to *DTheta* (relative to the surface normal) is randomly assigned with uniform weighting per solid angle. Particles are randomly assigned an azimuthal angle (around the surface normal) uniformly weighted in the range $0.0^\circ \leq \phi \leq 360^\circ$. The angle should be in the range $DTheta \leq 15^\circ$.

SUPPRESS SVal1 SVal2 SVal3 ...
SUPPRESS 0.20 0.30 0.40 0.60 0.80 1.00

To aid solution convergence the current density must be suppressed below the Child-law prediction on the first few iteration cycles. Otherwise, the strong vacuum field values at the source would result in very high currents. **OmniTrak** sets the following default values for the suppression factors:

| NCycle | Suppression value |
|--------|-------------------|
| ===== | |
| 1 | 0.250 |
| 2 | 0.300 |
| 3 | 0.500 |
| 4 | 0.750 |
| 5 | 1.000 |
| 6 | 1.000 |
| 7 | 1.000 |
| 8 | 1.000 |
| 9 | 1.000 |
| 10 | 1.000 |

For difficult solutions you can set you own values. Enter from 1 to 10 real numbers in the range 0.0 to 1.0. Any undefined values default to 1.0. Note that the number of iteration cycles should be in the range $NCycle \geq 5$ for the default values of suppression factors.

EDIRECT [Surface, Field]
EDIRECT = Surface

By default, **OmniTrak** sets the initial momentum of particles at the emission surface parallel to a vector normal to the associated facet of

the source surface (*Surface* option). Alternatively, the code can set the initial direction along a vector parallel to the local electric field (*Field* option). The *Field* option may give a smoother distribution of particle directions when a highly-conformal source surface has a small number of facets. Default: *Surface* mode. **Note: the angular divergence function (*DTHETA*) will not function in the *Field* mode.**

9.3. Application examples

This section reviews three examples included in the library that illustrate **OmniTrak** techniques for space-charge-limited emission. The first (`SC_EMIT1D`) is a benchmark test of the Child law algorithm. It treats one-dimensional space-charge-limited flow. The simulation covers a segment of size 1.0 cm × 1.0 cm in the x - y plane with the Neumann condition applied to boundaries normal to x and y . The volume has length 2.0 cm along the z direction with the boundary at $z = 1.0$ cm set to -1000 V and the boundary at $z = 3.0$ cm set to 0.0 V. We use a moderately-coarse mesh with cubic elements of width 0.10 cm. The marked emission surface is defined with $DEmit = 0.10$ cm and $NDiv = 2$ (4 electrons per facet for a total of 400 electrons). The simulation has 16 relaxation cycles with a charge adjustment factor of $AvgE = 0.40$. The predicted electron current density for the planar gap is 184.4 A/m², giving a total current of 1.844×10^{-2} A over the solution cross section. The equation (**Charged Particle Beams**, Sect. 5.2):

$$\Delta\phi = V_o (z/d)^{4/3} ,$$

implies that the emission surface should assume a potential of 184.2 V relative to the cathode.

The time step for the simulation was chosen so that particles moved at most ½ element per step. Figure 9.3 shows the convergence history. The final value of current was 1.830×10^{-2} A, within 0.76% of the prediction. Accumulated noise from the field interpolations was small — the orbits at the exit were parallel to within ±0.03 milliradian. The final average kinetic energy for the 400 particles was 998.69 ± 0.06 eV, implying energy conservation to within 0.13%.

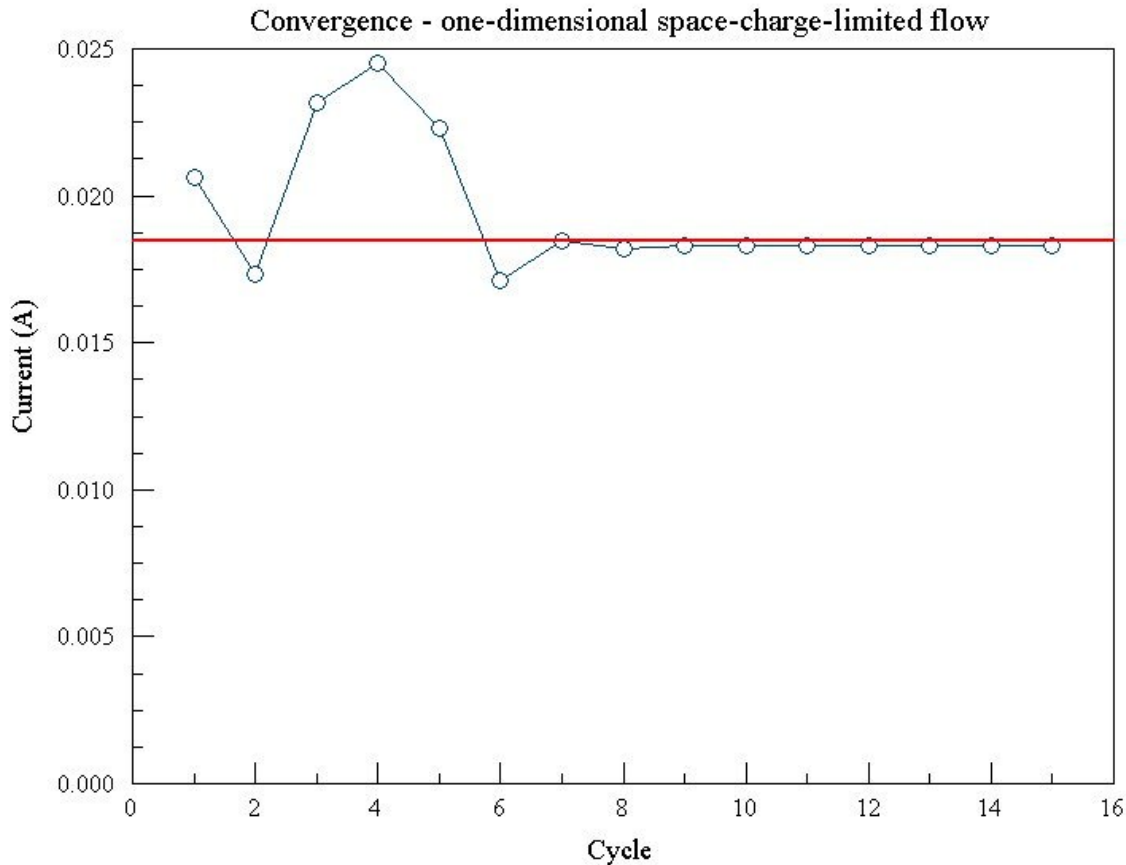


Figure 9.3. Convergence history for one-dimensional space-charge-limited flow — total current versus cycle number. Red line shows the theoretical current value. Example SC_EMIT1D.

In the next example (CL_CYLIN) we proceed to a simulation on a fully-conformal meshes with a curved source surface. The example treats space-charge-limited flow between coaxial electrodes. We can compare the results to analytic calculations reviewed and tabulated in **Charged Particle Beams** as well as to numerical simulations using the **Trak 2D** ray-tracing code. The simulation addresses the space-charge-limited diverging flow of protons between coaxial electrodes. Figure 9.4 shows the geometry and a cross-section of the conformal mesh. The simulation covers one quadrant of a coaxial system with anode radius $r_i = 1.75$ cm and cathode radius $r_o = 5.0$ cm. Symmetry conditions are applied at $x = 0.0$ and $y = 0.0$. The axial extent is 2.0 cm with symmetry boundaries at $z = 0.0$ and $z = 2.0$. Elements of the foundation mesh are cubes with side

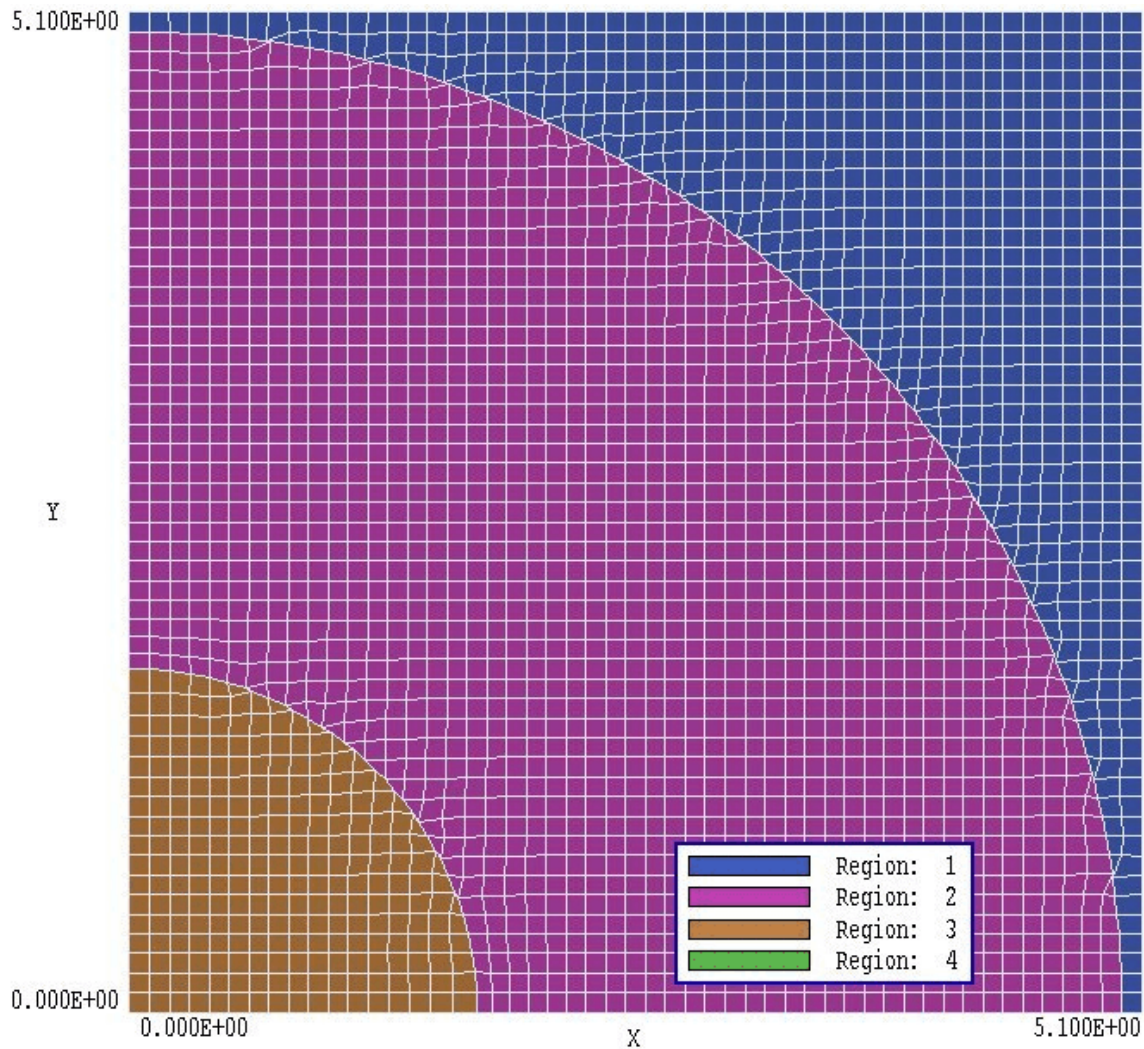


Figure 9.4. Computational mesh — simulation of space-charge-limited flow between concentric cylinders. Dimensions in cm. Example CL_CYLIN.

lengths of 0.10 m. The elements of Region 1 represent a conducting wall at ground potential, Region 2 is the vacuum propagation volume, and the elements of Region 3 comprise the emitter electrode at a potential of 100 kV. Region 4 is the emission surface consisting of nodes on the boundary between regions 2 and 3. The nodes do not appear in the element plot of Fig. 9.4.

The emission surface area for an ideal cylinder of radius 1.75 cm and length 2.0 cm is 5.498 cm². The faceted approximation created by

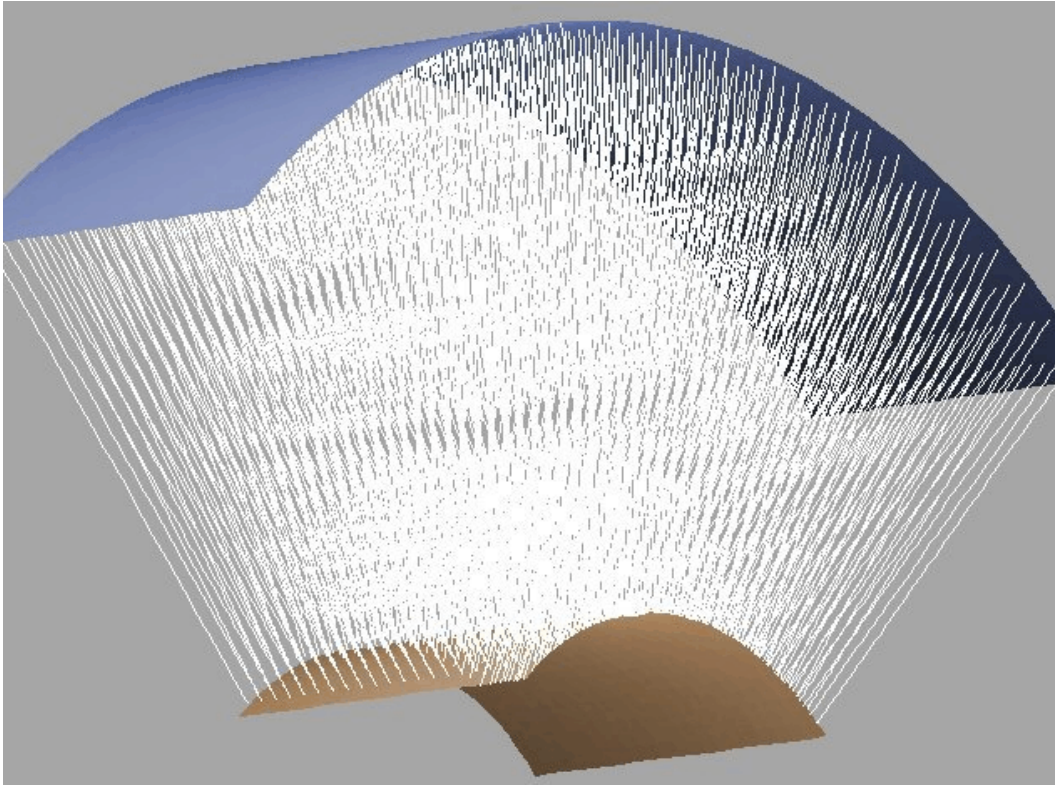


Figure 9.5. Three-dimensional plot of orbits — space-charge-limited flow between coaxial cylinders. Example CL_CYLIN.

MetaMesh has a calculated surface area of 5.497 cm². One model particle is assigned to each facet for a total number of 680 particles. The simulation occupies fifteen iteration cycles with an averaging parameter of $Avg = 0.35$. The theoretical value of total current per axial length is 111.0 A/m. A **Trak** 2D simulation with triangular elements of length 0.05 cm gives a value of 110.8 A/m (-0.16% difference). In comparison, the **OmniTrak** simulation predicts a current of 108.7 A/m, an absolute error of only -2.1%. The emitted current was almost independent of emission surface spacing over the range $DEmit = 0.05$ cm to 0.15 cm. Figure 9.5 shows a three-dimensional plot of calculated orbits.

The final example (ROD_EMIT) provides a comprehensive test of **OmniTrak** capabilities. The simulation addresses electron emission from a rod emitter. We can compare results for the cylindrically-symmetric structure to those generated by the two-dimensional **Trak** code. Figure

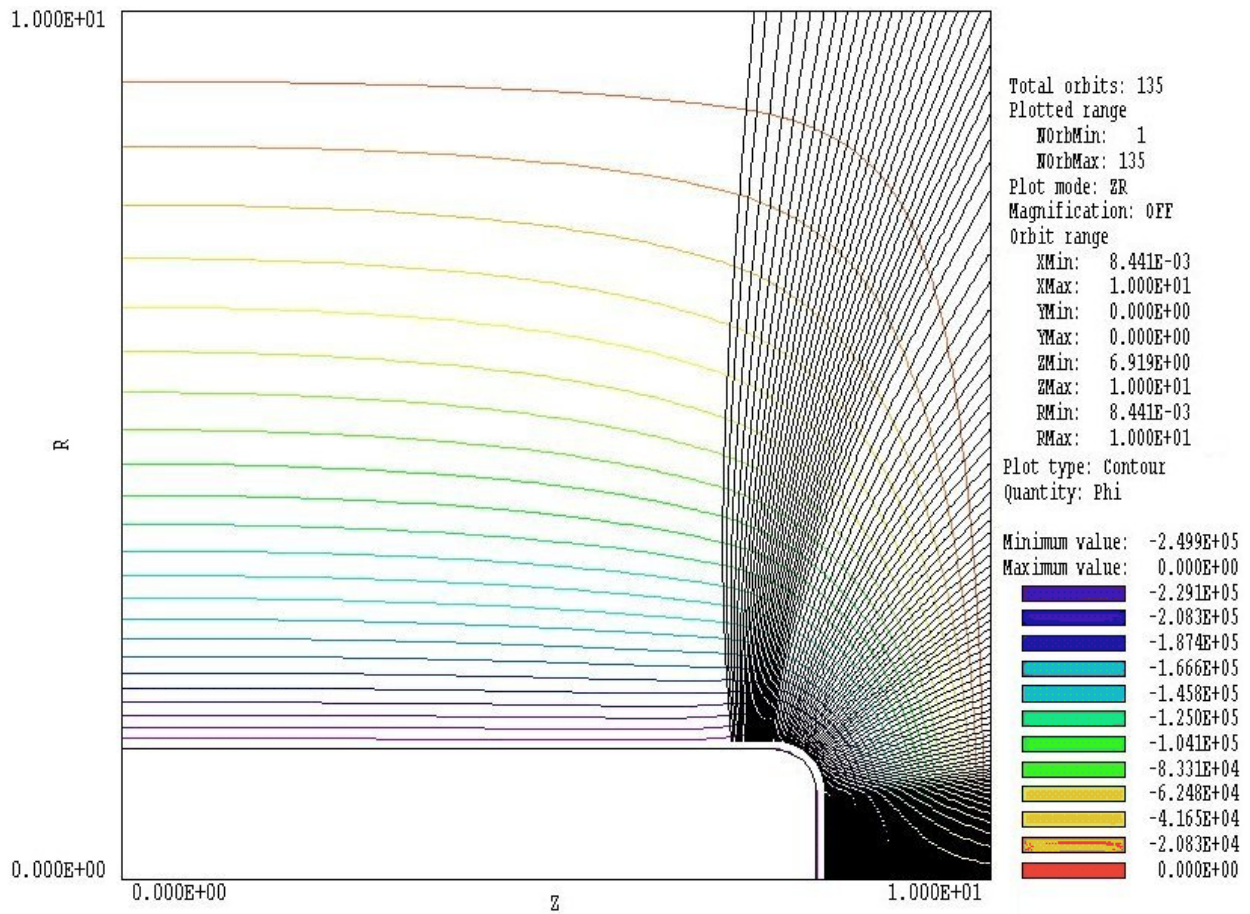


Figure 9.6. Trak simulation — electron emission from a shaped rod

9.6 shows the geometry of the **Trak** calculation along with calculated equipotential contours and electron orbits. Electrons are emitted from a rod of radius 1.5 cm with a filleted edge inside a vacuum chamber with 5.0 cm radius. The applied voltage is -250 kV and the minimum distance to the grounded vacuum chamber is 2.0 cm. The calculation is challenging because there is a strong enhancement of current density on the curved edge of the rod. The **Trak** simulation uses a mesh of conformal triangles with approximate side length 0.05 cm. Effects of beam-generated magnetic fields have not been included. The **Trak** simulation gives a total current of 2.47 kA

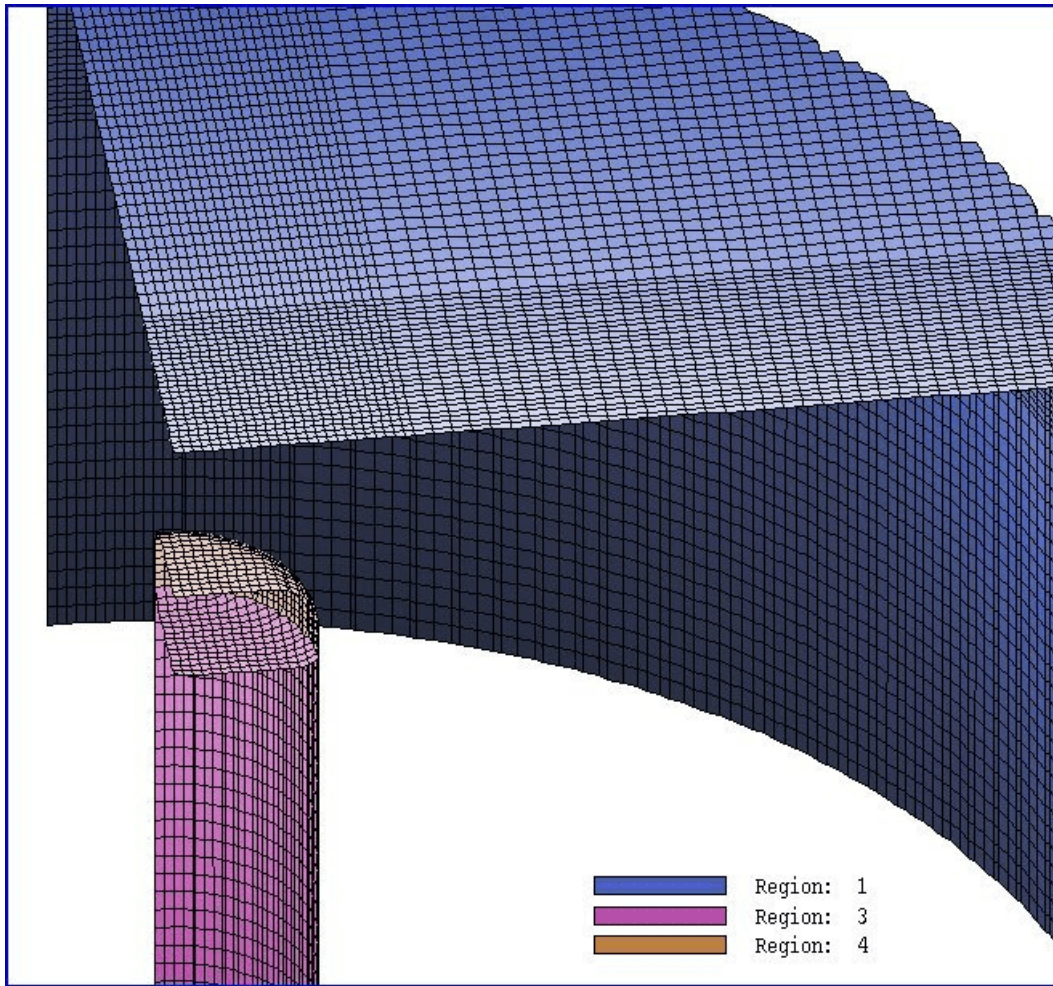


Figure 9.7. Complex emission surface — region surfaces on the mesh for the **OmniTrak** simulation. Example ROD_EMIT.

Figure 9.7 shows the mesh used for the **OmniTrak** simulation. It includes 245760 nodes and 234171 elements. We treat one quadrant with symmetry boundaries at $x = 0.0$ and $y = 0.0$. The variable-resolution mesh has a minimum element length of 0.10 cm in the emission region. Region 1 represents the grounded vacuum chamber and Region 2 represents the vacuum volume. Note that the inner electrode was constructed from two parts with region numbers 3 and 4. The coating operation was applied only to the part corresponding to Region 4 so that the marked emission nodes (Region 5) covered the electrode tip. The simulation used 1 electron per facet for a total of 469 particles.

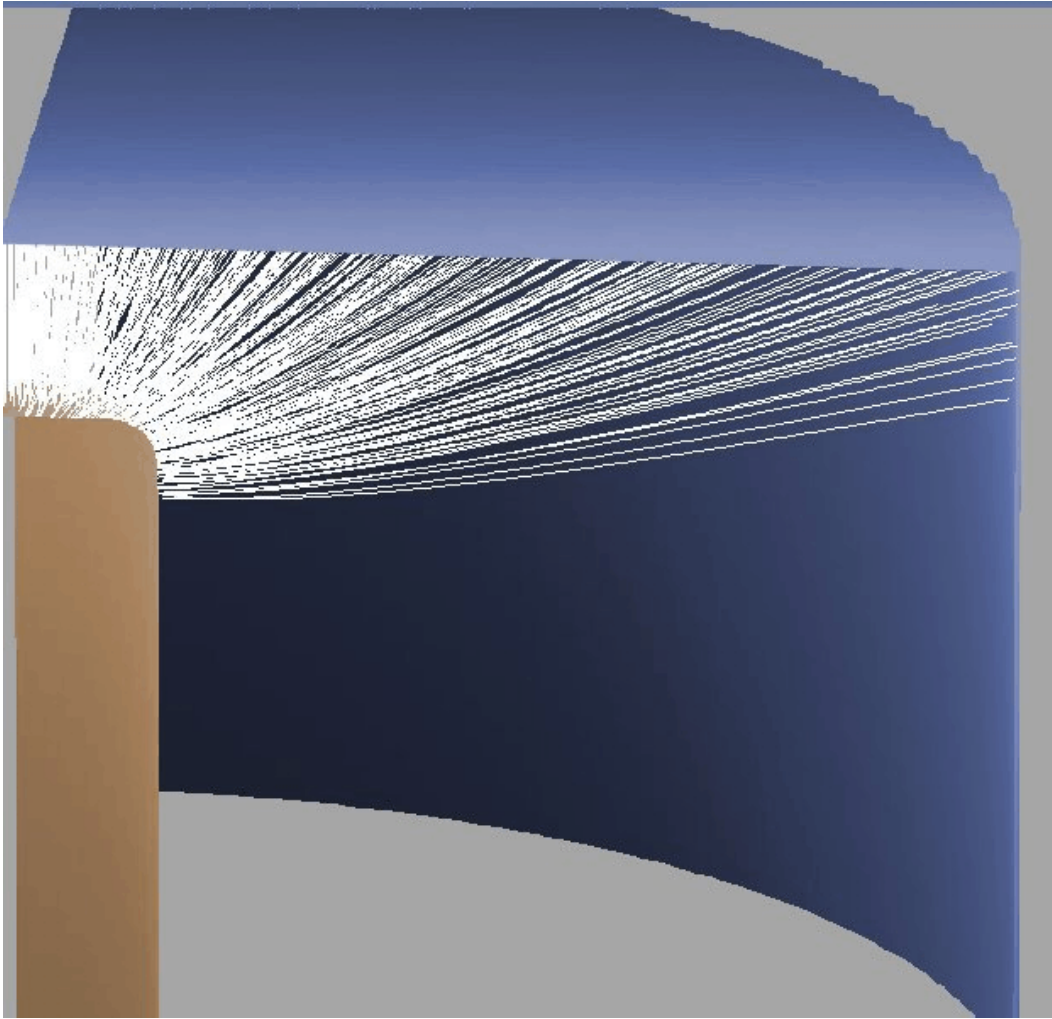


Figure 9.8. Calculated orbits (1/5 shown) for the **OmniTrak** simulation of the complex emission surface. Example ROD_EMIT.

Figure 9.8 shows a 3D plot of 1/5 of the orbits. The irradiation area on the anode is in excellent agreement with Figure 9.6. **OmniTrak** predicts a total current for the quadrant of 594.4 A, equivalent to 2.38 kA for the full cylinder. This value is within 3.6% of the **Trak** prediction. The final orbit positions on the anode are set by the routines in **OmniTrak** that identify the penetration point where a particle enters a material volume and projects final calculated parameters to the location. We can gauge the efficacy of the process from the final electron kinetic energy values. The result is $249.4 \text{ keV} \pm 1.0 \text{ keV}$, implying energy conservation to within $\pm 0.4 \%$.

9.4. Restarting runs with space charge and Child-law emission

You can restart **OmniTrak** runs that include space-charge and beam-generated magnetic fields. One application is to add more cycles of orbit tracking and field recalculation to improve accuracy. An important application is the modeling of relativistic electron guns with space-charge-limited emission. **OmniTrak** runs much slower in the *RELBEAM* mode. It is more efficiency to make a preliminary *SCHARGE* run with no beam-generated magnetic field and a large number of cycles to seek an electric field solution consistent with Child-law emission. You can then run a second solution with small number of cycles to add the effects beam-generated magnetic fields. In most practical guns, beam-generated magnetic fields may have a significant effect on downstream transport but have only a small effect on the emitted current.

As an example, suppose you have a proton gun solution from an *SCHARGE* mode simulation (RUN01) with 12 cycles. The run has created the electric field file RUN01P.HOU that includes the effects of space-charge and has values of potential that are approximately consistent with Child-law emission. The goal is to set up a second simulation (RUN02) that advances the solution for 10 extra cycles. The script RUN02.OIN should be the same as RUN01.OIN with the following two exceptions:

- The *EFILE* command should load the modified electric field file RUN01P.HOU.
- You must include the command `SUPPRESS 1.0 1.0 1.0 1.0 1.0` to deactivate current suppression on initial cycles.

In the case of a relativistic electron beam, the preliminary run should be performed in the *SCHARGE* mode and the second in the *RELBEAM* mode.

Note: **OmniTrak** always uses the full calculated beam charge and current on the first cycle and applies averaging set by the *AVG E* and *AVG B* commands on subsequent cycles. Therefore, a second run will restore the space-charge and current arrays to their approximate state at the end of the first run and continue with charge and current averaging in subsequent cycles. The implication is that you can use any desired values of the averaging factors in the second run.

10. Beam-generated magnetic fields - PARTICLES section

10.1. Modeling relativistic beams

The generation of three-dimensional simulations of high-current relativistic electron injectors is the major challenge facing gun design codes. The capability is essential for the design of non-circular guns (such as sheet beam injectors) and for studies of perturbations in circular guns. **OmniTrak** is the only existing gun-design code that can handle the complete problem with moderate setup time and reasonable run time. The code also includes the effects of image-currents generated in surrounding conductive structures by pulsed beams.

This section gives a brief review of physical aspects of relativistic beam transport and the methods used in **OmniTrak** to calculate magnetic fields. You can find detailed information on relativistic beam physics in the text S. Humphries, **Charged Particle Beams** (Wiley, New York, 1990) available for download in PDF form at <http://www.fieldp.com/cpb/cpb.html>. To begin, consider a circular paraxial electron beam traveling along the z axis in free space. The term *paraxial* implies that 1) electrons have about the same axial velocity ($v_z \approx \beta c$) and 2) orbits make small angles with respect to the axis:

$$z' = dr/dz \ll 1. \quad (10.1)$$

Equation (10.1) implies that changes in the beam radius r_0 take place over axial distances much greater than r_0 . Therefore local beam-generated fields are approximately equal to those of an infinite-length beam.

Suppose the beam carries current I and that the space-charge density is a function of radius, $n(r)$. The electric and magnetic fields created by the beam can be determined from Poisson's equation and Ampere's law:

$$E_z \cong 0, \quad (10.2)$$

$$E_{\perp} = E_r(r) = -\frac{e}{2\pi\epsilon_0 r} \int_0^r 2\pi r' dr' n(r'), \quad (10.3)$$

$$B_z \cong 0, \quad (10.4)$$

$$B_{\perp} = B_{\theta}(r) = -\frac{e\beta c\mu_0}{2\pi r} \int_0^r 2\pi r' dr' n(r'). \quad (10.5)$$

Equations (10.3) and (10.5) imply that the transverse electric and magnetic forces acting on individual electrons are related by

$$F_{\perp}(\text{magnetic}) = -\beta^2 F_{\perp}(\text{electric}). \quad (10.6)$$

The quantity β is small compared to unity for non-relativistic beams; therefore, the magnetic force can usually be neglected. In contrast, the repulsive electric force and attractive magnetic force are almost balanced for highly relativistic beams ($\beta \cong 1$).

The transverse forces of a planar beam also satisfy Eq. (10.6). In fact, the relationship holds for paraxial beams with any shape and with any transverse distribution of space-charge. The equation also holds if we include field contributions of perfectly-conducting boundaries of any shape whose dimensions change gradually in the axial direction. We can prove the result by 1) making a Lorentz transformation by velocity $-\beta c$ to the average rest frame of the beam, 2) calculating the electrostatic fields resulting from the stationary distribution of charge, and 3) calculating transformed electric and magnetic field values in the laboratory frame. This derivation also shows that the criteria underlying paraxial beam motion and the definition of gradual changes in boundary conditions is less stringent than that implied by Eq. (10.1). If D is the transverse scale size of the system and L is the axial distance for a significant change in the beam or boundary dimensions, then Eq. (10.6) holds if

$$\frac{R}{\gamma L} \ll 1, \quad (10.7)$$

where

$$\gamma = \frac{1}{\sqrt{1 - \beta^2}} . \quad (10.8)$$

The quantity γ is related to the rest energy m_0 and kinetic energy T of the particle by:

$$\gamma = 1 + \frac{T}{m_0 c^2} . \quad (10.9)$$

Equations (10.1) through (10.9) imply a strategy (that we shall refer to as the *relativistic mode*) to avoid time-consuming calculations of three-dimensional magnetic fields for relativistic beams. The total transverse force is related to the electric force by:

$$\begin{aligned} F_{\perp}(\text{total}) &= F_{\perp}(\text{electric}) - F_{\perp}(\text{magnetic}) = \\ (1 - \beta^2) F_{\perp}(\text{electric}) &= \frac{F_{\perp}(\text{electric})}{\gamma^2} . \end{aligned} \quad (10.10)$$

We simply calculate the electric-field force and then divide by the square of the particle γ to include the effect of the magnetic field. The relativistic mode holds under the following conditions:

- Transverse electric fields arise almost entirely from the presence of the beam space and surrounding conducting boundaries.
- Beam particles move predominantly in one direction and have approximately the same axial velocity $v_z \cong \beta c$
- The axial scale length for changes in the transverse dimensions of the beam and surrounding boundaries satisfies Eq. (10.7).

The following section discusses applications of the relativistic mode and its implementation in **OmniTrak**.

The relativistic mode does not apply in the acceleration gap of an electron gun. In this case there are strong transverse applied electric fields and beam properties change significantly over a short axial distance. Gun simulations require a complete three-dimensional calculation of the magnetic field. **OmniTrak** utilizes the *reduced potential* method, the same solution technique used in **Magnum**. In contrast to a direct solution for the vector potential, the reduced potential method offers highly reduced storage requirements and computation times. Furthermore, it is relatively easy to include contributions from beam currents on arbitrary conformal meshes. The key is to divide the magnetic field intensity into two components:

$$\mathbf{H} = \mathbf{H}_s + \mathbf{H}_m . \quad (10.11)$$

The component \mathbf{H}_s arises from the applied and beam currents while \mathbf{H}_m is the component created by currents in materials. The components individually satisfy the equations:

$$\nabla \times \mathbf{H}_s = \mathbf{j}_s , \quad (10.12)$$

$$\nabla \times \mathbf{H}_m = \mathbf{0} . \quad (10.13)$$

Given the spatial distribution of beam current \mathbf{j}_s , the component \mathbf{H}_s can be determined directly by a Biot-Savart integral or other approximation at any point in space without recourse to finite-element methods. Equation (10.13) implies that the material component \mathbf{H}_m can be expressed as the gradient of a scalar potential:

$$\mathbf{H}_m = -\nabla\phi_m , \quad (10.14)$$

where ϕ_m is called the *reduced potential*. The equation

$$\nabla \cdot \mathbf{B} = 0 , \quad (10.15)$$

implies that

$$\nabla \cdot [\mu (\mathbf{H}_s - \nabla\phi_m)] = 0, \quad (10.16)$$

or

$$\nabla \cdot (\mu \nabla\phi_m) = \nabla \cdot (\mu \mathbf{H}_s). \quad (10.17)$$

Equation (10.17) has the form of the Poisson equation with a source term on the right-hand side. Because we know the spatial variations of μ and \mathbf{H}_s (from the integral over applied and beam currents), the source term is a known quantity. The magnetic field acting on particles through the Lorentz force is given by

$$\mathbf{B} = \mu (\mathbf{H}_s - \nabla\phi_m). \quad (10.18)$$

10.2. Using the relativistic mode

Application of the relativistic mode is always preferable to a full calculation of the beam-generated magnetic fields. Run times are significantly shorter. To review, the system must satisfy the following criteria:

- 1) The electric field must result solely from the beam space-charge and from charges induced by the beam in surrounding conducting structures. In general, the relativistic mode is useful for transport rather than acceleration solutions.
- 2) Particles in the beam must have about the same energy.
- 3) Applied magnetic fields may be included and the main axis of the beam need not be a straight line. The relativistic mode could be employed, for example, to simulate the dynamics of a high-intensity beam in a bending magnet.
- 4) Orbits of individual particles must make small angles with respect to the axis (*i.e.*, paraxial motion). Equivalently, the axial kinetic energy should be much larger than the transverse energy.

5) The axial distance L for changes in the beam direction, changes in the envelope width, or changes in the transport tube geometry must satisfy the inequality $L \gg \gamma D$. Here, D is a characteristic transverse dimension of the beam and/or transport tube.

The conditions are satisfied in most practical relativistic beam transport systems.

You can apply the relativistic mode approximation to **OmniTrak** calculations in the *PARTICLES SCHARGE* mode. Particle input must be through the *PLIST* or *PFILE* commands. Input through emission surfaces would require an applied accelerating electric field that would violate the first condition listed above. Note that the list should contain only a single particle species. In most practical cases the particles are electrons. To invoke the relativistic mode, simply include the command *RELMODE* in the *PARTICLES* section. The section then has the general form:

```
PARTICLES Scharge
    . . .
    PLIST
    . . .
    END
    . . .
    RELMODE
    . . .
    END
```

The command has the format

RELMODE

General application of the relativistic mode. Divide all Cartesian components of the electric force by γ^2 for the calculation of particle dynamics. The quantity γ is the local relativistic energy factor of the particle.

During particle tracking **OmniTrak** calculates forces acting on the particles according to:

$$F_x = \frac{qE_x}{\gamma^2}, \quad F_y = \frac{qE_y}{\gamma^2}, \quad F_z = \frac{qE_z}{\gamma^2}. \quad (10.19)$$

This type of calculation could apply, for instance, to a high-current relativistic beam in a bending magnet inside an approximately uniform transport tube. The calculation is valid if the beam current is well below the propagation limit in the tube set by space-charge effects. In other words, the space charge potential ϕ at positions inside the beam is much smaller than the average kinetic energy expressed in eV:

$$\phi \ll T_e / q. \quad (10.20)$$

Therefore, values of β and γ for the particle do not change significantly over the length of the simulation.

You can use an alternate form of the *RELMODE* command to apply the relativistic mode to beams near the space-charge limit that move nearly parallel to a Cartesian axis.

RELMODE Axis

RELMODE Z

Apply the relativistic mode only to electric field components transverse to the beam axis. The character parameter (*X*, *Y* or *Z*) specifies the direction of beam motion.

For example in response to the command *RELMODE Z* command **OmniTrak** calculates forces according to

$$F_x = \frac{qE_x}{\gamma^2}, \quad F_y = \frac{qE_y}{\gamma^2}, \quad F_z = qE_z. \quad (10.21)$$

The set of equations could apply, for example, to a beam that moves along the *z* direction from a narrow to a broad transport tube. In this case the expressions of Eq. 10.21 would give the relativistically-correct transverse forces and also yield the correct reduction in beam kinetic energy associated with the increase in the magnitude of the space-charge potential.

The example *ELLIPLENS* illustrates an application of the relativistic mode. The goal is to transform a circular high-perveance electron beam into a stretched elliptical beam. The approach is to use an elongated three-dimensional solenoid lens to apply unequal forces along two normal axes.

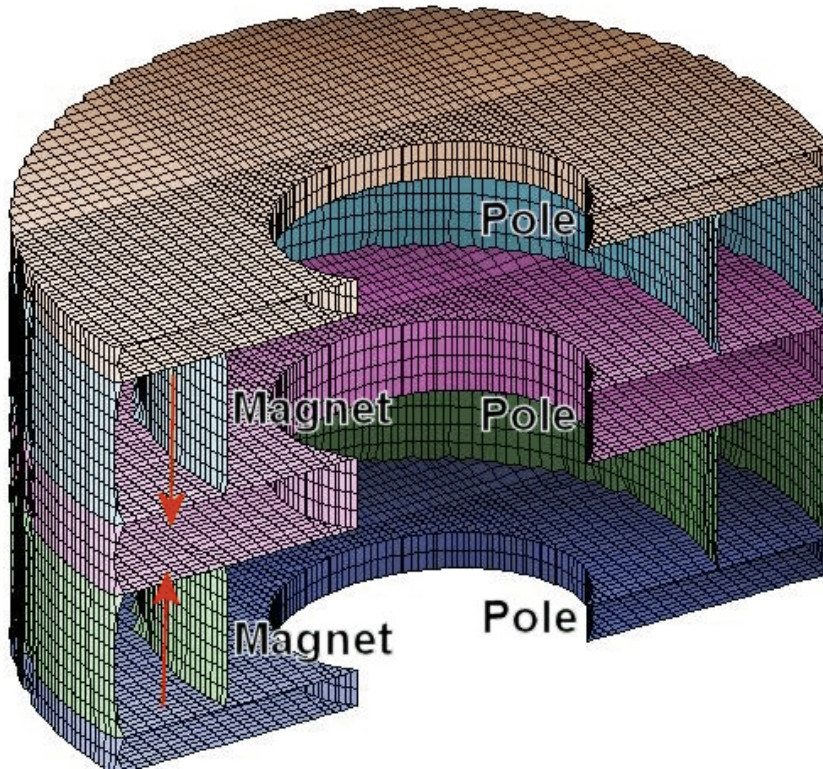


Figure 10.1. Mesh representation of the elliptical solenoid lens

Figure 10.1 shows the geometry (ELLIPLENS.B.MIN). The lens consists of five components: 1) a soft iron pole of length 0.5 cm, 2) an annular permanent magnet magnetized in the $+z$ direction with length 2.0 cm, 3) a 1 cm thick pole, 4) another magnet magnetized along $-z$ and 5) a final pole of thickness 0.5 cm. The dual magnet configuration minimizes beam rotation. The permanent magnets have an outer radius of 5.00 cm and an inner radius of 3.50 cm. The outer boundary of the poles is circular with radius 5.0 cm. The inner boundary is elliptical with $R_x = 2.02$ cm and $R_y = 1.60$ cm, giving stronger focusing in the y direction. Note that the full circular lens was modeled. Because of beam rotation in the solenoid lens, it is not possible to apply bilateral or quadrilateral symmetry.

An electron beam with kinetic energy 120 keV and current 20 A starts from a waist with radius 0.6 cm at a position 8.0 cm upstream from the lens midplane. The *CIRCBEAM* tool of **OmniTrak** was used to create the

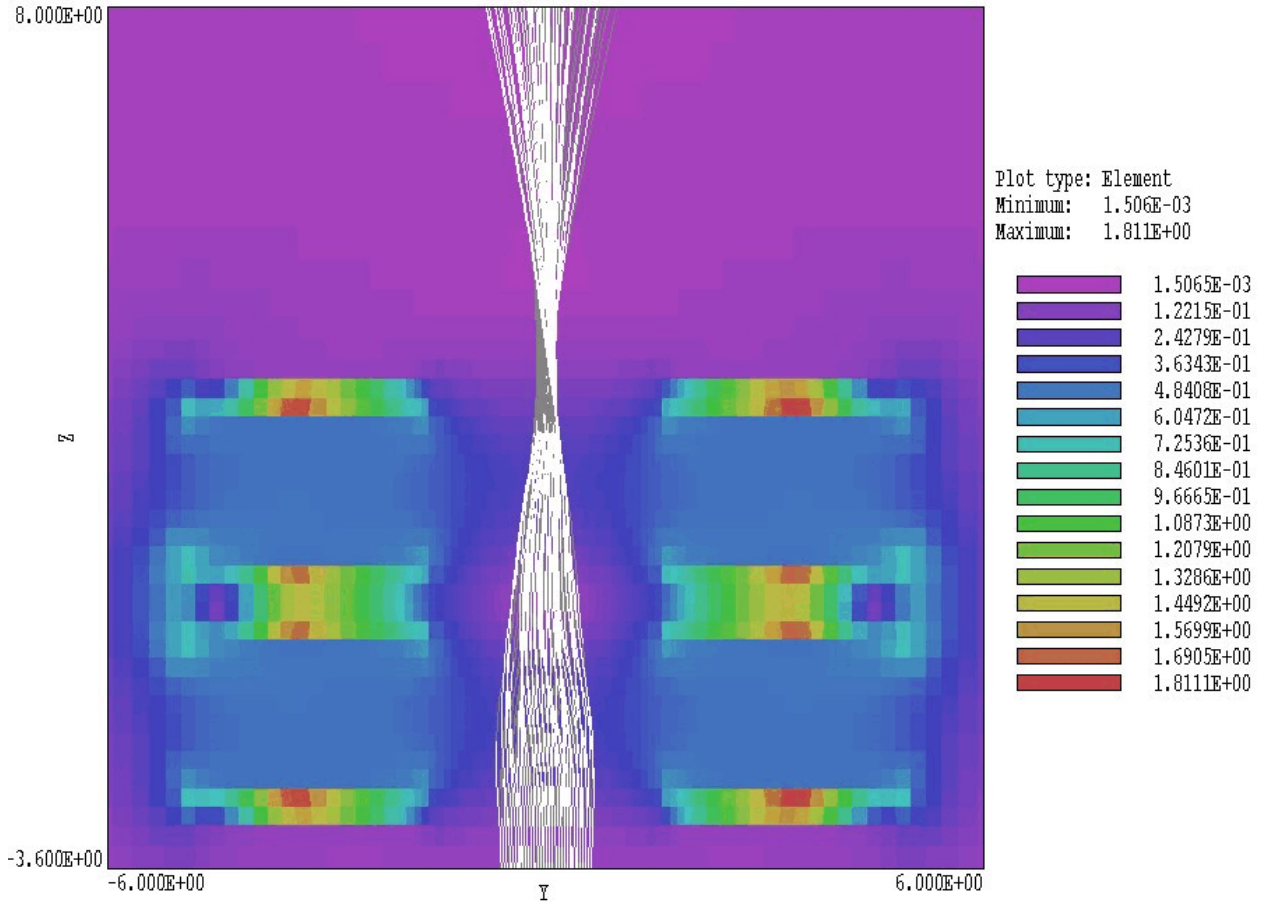


Figure 10.2. Electron orbits projected to the plane $x = 0.0$ in example ELLIPLENS. Light orbit vectors are in front of the projection plane and dark vectors are behind the plane. Color coding shows $|B|$.

initial distribution with 239 model particles (ELENSINPUT.PRT). The simulation requires an electric field mesh for the calculation of space-charge fields. The mesh (ELLIPLENSE.MIN) defines a long cylindrical vacuum chamber of radius 1.5 cm. The **OmniTrak** input file ELLIPLENS.OIN is listed below. Figure 10.3 shows a cross section of the mesh and a projection of electron orbits to a plane normal to x . Figure 10.4 shows the transverse beam distribution at the waist point in y just past the lens exit ($z = 3.13$ cm). The distribution has a slight rotation and an approximately elliptical cross section. The half-width in x' is 0.6 cm and half-width in y' is only 0.06 cm.

Table 10.1. Contents of the file ELLIPLENS.OIN

```
GLOBAL
  DUnit = 100.0
  EField3D = ELLIPLENSE.HOU
  BField3D = ELLIPLENSB.GOU
  NCheck(E) = 25
  Omega(E) = 1.95
  MaxCycle(E) = 250
  ResTarget(E) = 5.0E-7
  Boundary = (-1.5, -1.,5 -8.0) (1.5,
1.5, 12.0)
END
PARTICLES SCharge
  Dt = 1.5E-11
  TMax = 2.0E-9
  NCycle = 20
  Avg(E) = 0.35
  PFile ELENSINPUT
  RelMode
  Vacuum(E) = 2
  Material(E) = 1
END
DIAGNOSTICS
  PartList
  EDump ELLIPLENSEP.HOU
  PartFile ELLIPLENS
END
ENDFILE
```

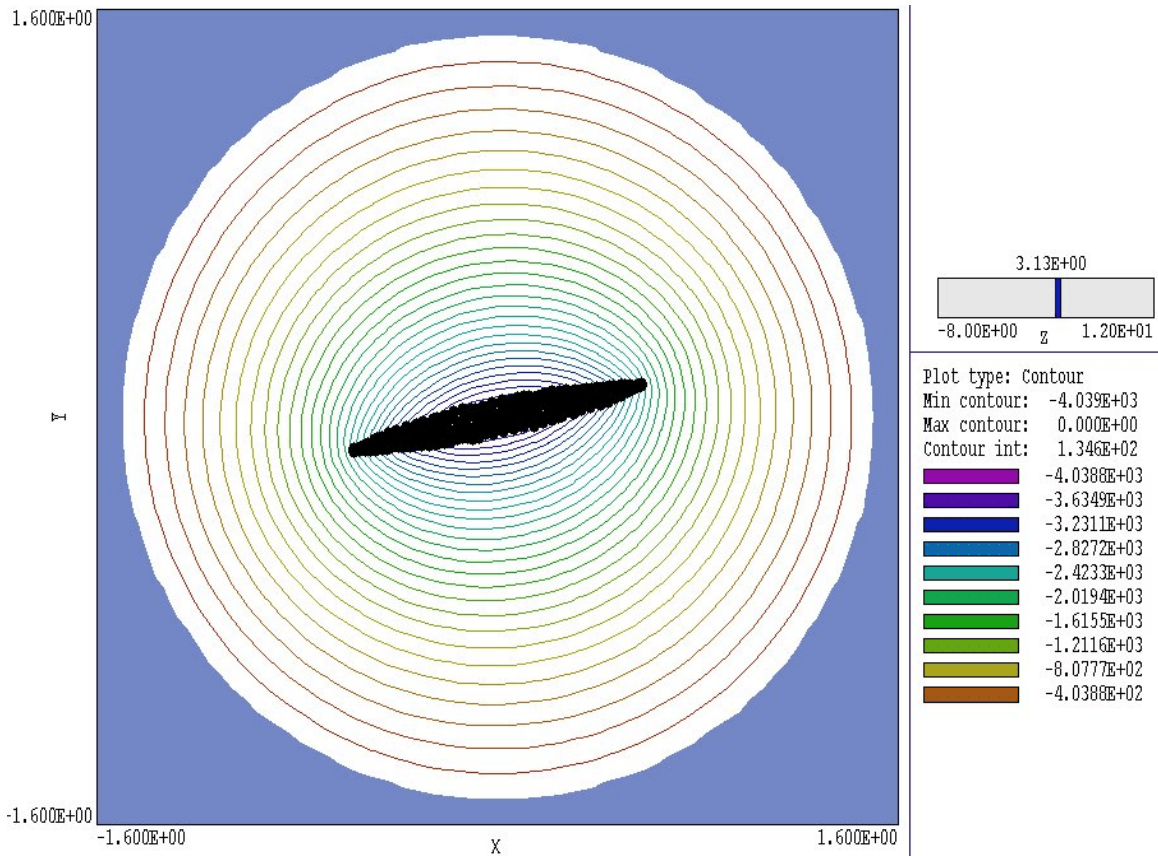


Figure 10.3. Orbit projections in the plane $z = 3.13$ cm for the example ELLIPLENS. Plot includes electrostatic equipotential lines. Blue area is the grounded transport tube.

10.3. Physical basis of the RELBEAM tracking mode

OmniTrak performs a full three-dimensional calculation of beam-generated magnetic fields and forces in the *PARTICLES RELBEAM* mode. The calculation is performed on the same spatial mesh used for the applied and beam-generated electric fields. As discussed in Section 10.1, the calculation is a two-step procedure:

- 1) evaluation of applied field components \mathbf{H}_s resulting from the presence of beam current, and
- 2) determination of the reduced magnetic potential ϕ_m using source functions determined from \mathbf{H}_s .

Following Eq. 10.18, the magnetic flux density \mathbf{B} that enters the particle equation of motion consists of contributions from the beam current ($\mu_0 \mathbf{H}_s$) and the currents in surrounding materials ($-\mu_0 \nabla \phi$). Here we use the factor

μ_0 because field interpolations for beam dynamics are performed in vacuum/air regions.

The reduced-potential approach is effective, but the correct implementation can be challenging. Before discussing the commands of the *RELBEAM* mode, it is useful to review techniques of the magnetic field calculations.

10.4. Applied field calculations in the RELBEAM tracking mode

In the **Magnum** program of the **AMaze** series, the applied field \mathbf{H}_s is calculated by a Biot-Savart integral over the current elements that constitute drive coils. The contribution from each current element must be evaluated at every node of the mesh. There are two reasons why this approach is practical in **Magnum**:

- The applied field calculation is performed only once at the beginning of the solution.
- Drive coils can usually be represented with a moderate number of elements (on the order of a few thousand).

The situation is different in **OmniTrak**. In this case there may be hundreds of particle orbits, each composed of hundreds of integration steps. The applied field calculation would be overwhelming if we associated each integration step with a current element. Adding to the difficulty, the applied field calculation must be performed at the beginning of each iteration cycle. Typically, a run requires 12-20 cycles for convergence. Figure 10.4 illustrates another problem with the use of a literal Biot-Savart integral to calculate the applied field. The orbit calculation gives only the current elements in the acceleration gap. For the purpose of the applied field calculation, each model particle is connected to drive current filaments, one that extends from $-\infty$ to the orbit start point and the other from the orbit end point to $+\infty$. If we do not include the contribution of the drive currents the calculated applied field values near the start and end points will be about 50% too low.

The problems arise because the Biot-Savart approach represents considerably more work than is warranted by simple magnetic field distribution created by the beam. In a practical relativistic gun, electrons travel from the cathode to the extraction aperture along smooth traces. We can apply two simplifications that maintain good accuracy while substantially reducing the amount of work:

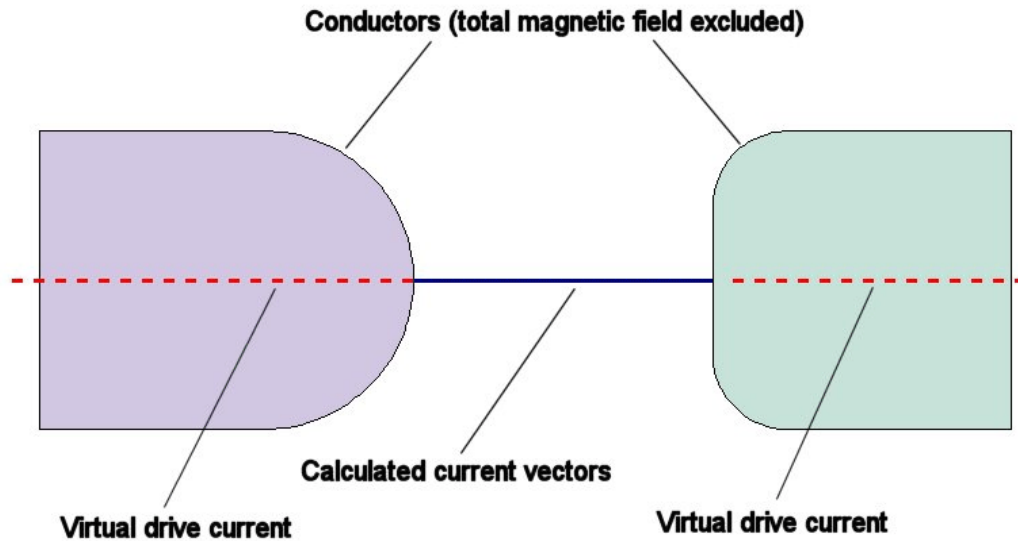


Figure 10.4. Virtual drive currents are required to calculate the correct beam-generated applied field in the vacuum region between conductors.

- Although a beam orbit may be composed of hundreds of small integration steps, a moderate number of longer current element vectors (5-10) gives a good estimate for the magnetic field.
- The Biot-Savart procedure involves an axial integration over the length of each orbit for every point in the mesh. If the traces are relatively smooth, we can avoid the axial integration by approximating local and far fields by those of a current cylinder of infinite length.

OmniTrak uses the following procedure (illustrated in Fig. 10.5) to calculate applied fields created by a model particles.

- 1) Using calculated orbit points, the program resolves a particle trace into a moderate number of vectors of approximate length $LElem$. **OmniTrak** sets the end point of a vector when the distance from the start point exceeds $LElem$. The end point becomes the start point for the next vector.
- 2) When a model particle orbit is complete, **OmniTrak** makes a loop through the nodes of the electric field mesh. For each node, the

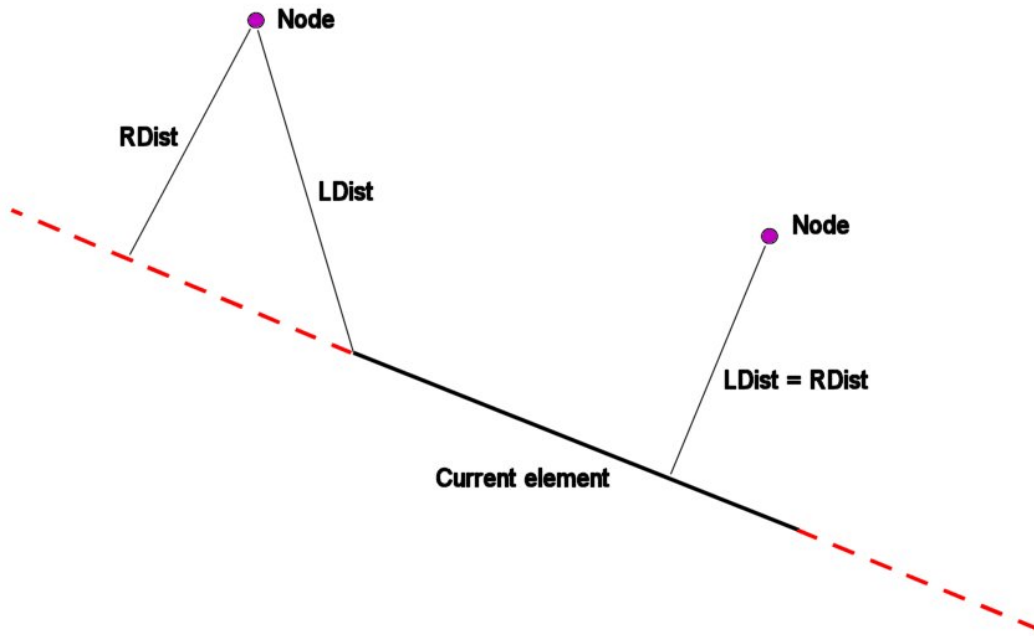


Figure 10.5. Treatment of a current element as a section of an infinite-length cylinder.

program identifies the closest particle vector corresponding to the minimum value of $LDist$ in Figure 10.5. The code also determines the distance from the node to the extended vector ($RDist$). If \mathbf{u}_e is a unit vector along the model particle vector and \mathbf{u}_r is a unit vector from the extended line vector to the node, then the contribution of a model particle carrying current I at the node is approximately,

$$d\mathbf{H}_s = (\mathbf{u}_e \times \mathbf{u}_r) \frac{I}{2\pi R_{dist}} \quad (R_{dist} > R_{elem}),$$

$$d\mathbf{H}_s = (\mathbf{u}_e \times \mathbf{u}_r) \frac{I R_{dist}}{2\pi R_{elem}^2} \quad (R_{dist} \leq R_{elem}).$$
(10.21)

An advantage of the algorithm is that the applied fields are calculated as through the current extended an infinite distance in the upstream direction from the start point and in the downstream direction from the end point (Fig. 10.5). In other words, drive currents are automatically included in the applied field calculation. Note that the assignment of a radius $RElem$ to the element gives transverse smoothing of the calculated field and prevents a diverging field magnitude.

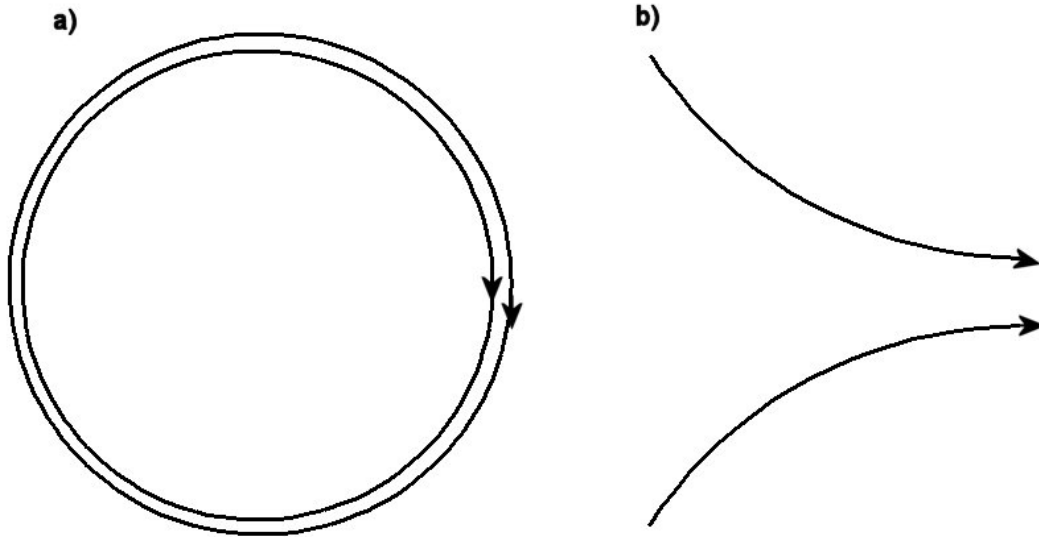


Figure 10.6. Accuracy of the infinite cylinder approximation is poor when all particles of a beam have the same inflection (a) and good when beam particles have opposite inflection (b).

It is important to realize the limitations of the method. First, although the approach saves work the magnetic field calculation still adds a significant burden to the program. The requirement to update the applied field at all points in the mesh for each orbit slows the tracking procedure. Furthermore, in cases where there are material contributions to fields the program must perform a relaxation solution for ϕ_m in each cycle. A second limitation is that the method may not be accurate in some cases. Figure 10.6a shows a worst-case scenario for a beam moving in a circle. The infinite-cylinder approximation does not correctly predict that the field is stronger inside the beam. The method does give good results for the more practical case of Fig. 10.6b. Here field gradients cancel because orbits on opposite sides of the beam have opposing inflections.

10.5. Material fields in the RELBEAM tracking mode

OmniTrak simulations may include a static magnetic field such as a solenoid lens or bending magnet. These global fields (which may incorporate ferromagnetic materials, drive coils or permanent magnets) are usually unaffected by the presence of a beam inside a conducting transport tube. Therefore **OmniTrak** makes no modifications to magnetic fields loaded with the *BFIELD3D* command. The implication is that the size of the three-dimensional applied field mesh and the number of drive current elements has little effect on the **OmniTrak** run time.

Image charge and return current may be induced in conductors by the presence of a pulsed beam. The fields created by image charge and return current may, in turn, influence the beam dynamics. To find the total magnetic field we must perform a self-consistent calculation for ϕ_m . Conducting regions that have the fixed-potential property for the electric field solution often also represent volumes of that exclude pulsed magnetic fields. For this reasons the calculation of beam-generated magnetic fields is carried out using the nodes and region divisions of the electric field mesh.

Generally a reduced-potential calculation is warranted if the beam pulse length is shorter than the penetration time for magnetic fields into surrounding conductors. In this limit the magnetic fields of the beam are excluded and the attendant compression of field lines may result in a force on the beam. In contrast, the return current of steady-state or long-pulse beams spreads over the largest possible area of conducting structures. In this case the flow of return current along a transport tube usually does not create a force on a displaced beam and a calculation of only the applied field created by the beam current is sufficient.

It is important to note that **OmniTrak** calculates material contributions only for perfect conductors or good conductors in the presence of a short-pulse beam. The program does not address intermediate regimes of magnetic diffusion. To set up the calculation you must assign appropriate material properties to regions of the electric field mesh that represent vacuum ($\mu_r = 1.0$) or conductors ($\mu_r \ll 1.0 \cong 1.0 \times 10^{-6}$). The outer boundary of the electric field mesh automatically assumes the condition of a perfect conductor (*i.e.*, magnetic field lines parallel to the boundary). Therefore you must ensure that the boundary is far enough away so it does not affect the dynamics of the beam. This condition would normally be satisfied on the mesh if the free boundaries for the electric field calculation (Neumann condition with $E_{\parallel} = 0.0$) have a small effect on the electric fields near the beam. Given the material properties **OmniTrak** calculates the reduced potential ϕ_m using the beam-generated magnetic field \mathbf{H}_s to define source terms.

An interesting feature of the reduced-potential calculation is that it gives the correct current distribution on conducting surfaces independent of details of the applied field calculation. For example, Fig. 10.7 shows a simulation of off-axis beam generation from an extended cathode stalk in a coaxial structure. The 150 A beam is generated from a cathode of radius 0.5 cm displaced 0.7 cm from the axis and accelerated across a gap with applied voltage of 1.0 MV. The figure shows electron orbits with color-coding by $|\mathbf{B}|$. For reference, the envelope field for a cylindrical beam

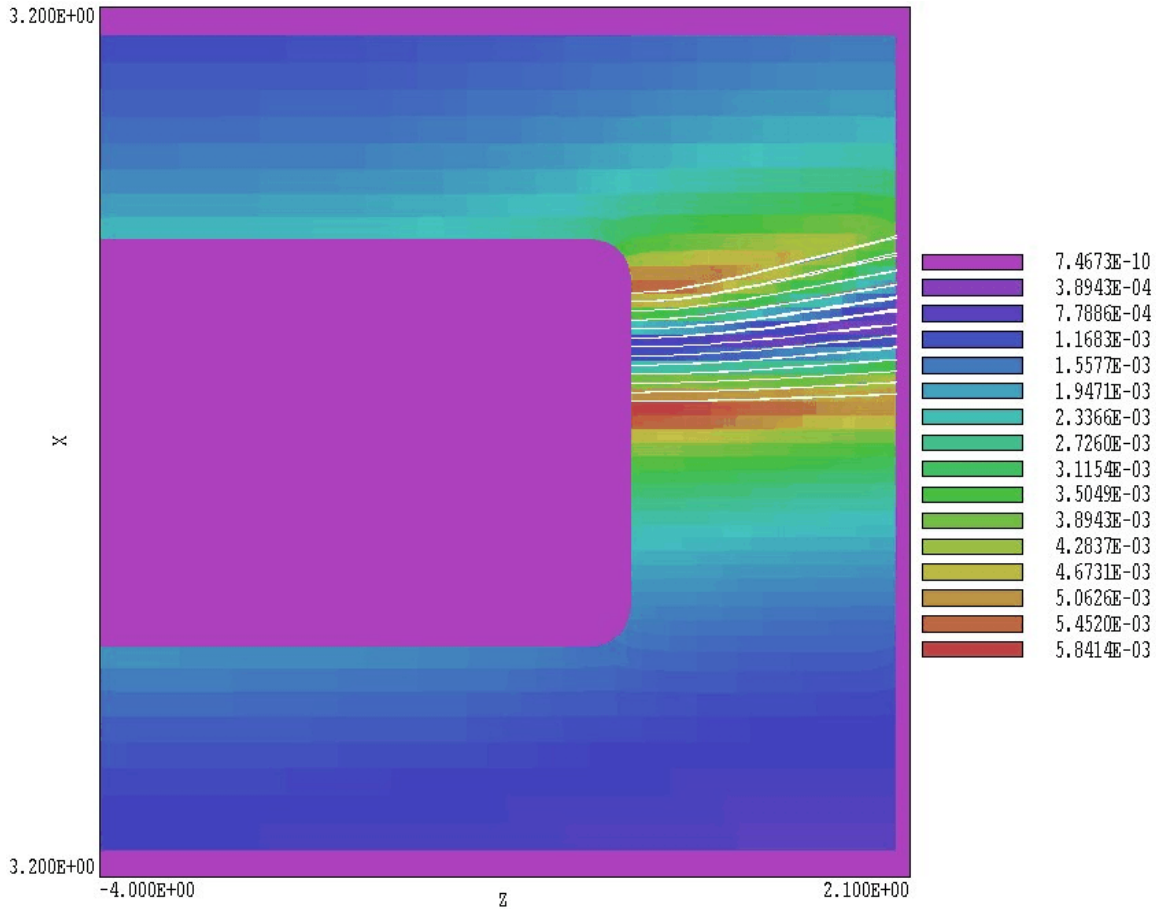


Figure 10.7. Example BBCOAX. Injection of an off-axis beam from a coaxial cathode support. Support radius: 1.6 cm. Wall radius: 3.0 cm. Cathode radius: 1.5 cm. Beam current: 150 A. Applied voltage: 1.0 MV. The plot is a longitudinal section in the plane $y = 0.0$ with electron orbits and color-coding by $|\mathbf{B}|$.

would be $B_\theta = 6.0 \times 10^{-3}$ tesla. Figure 10.8 shows the magnitude and vector lines of \mathbf{B} in two planes normal to z that cut the cathode stalk: *a*) just upstream from the cathode surface and *b*) 3.9 cm upstream from the cathode. Although the total magnetic field near the cathode is skewed, the total field at the distant location is almost azimuthally symmetric, even though the applied field at both locations is almost the same. The peak field in the figure is 2.11×10^{-3} tesla compared to a predicted field value of 2.00×10^{-3} tesla for an ideal, infinite-length coaxial tube. Because of this feature, if it not necessary to be concerned with details of the applied field calculation in regions with conductors upstream and downstream from the beam propagation volume.

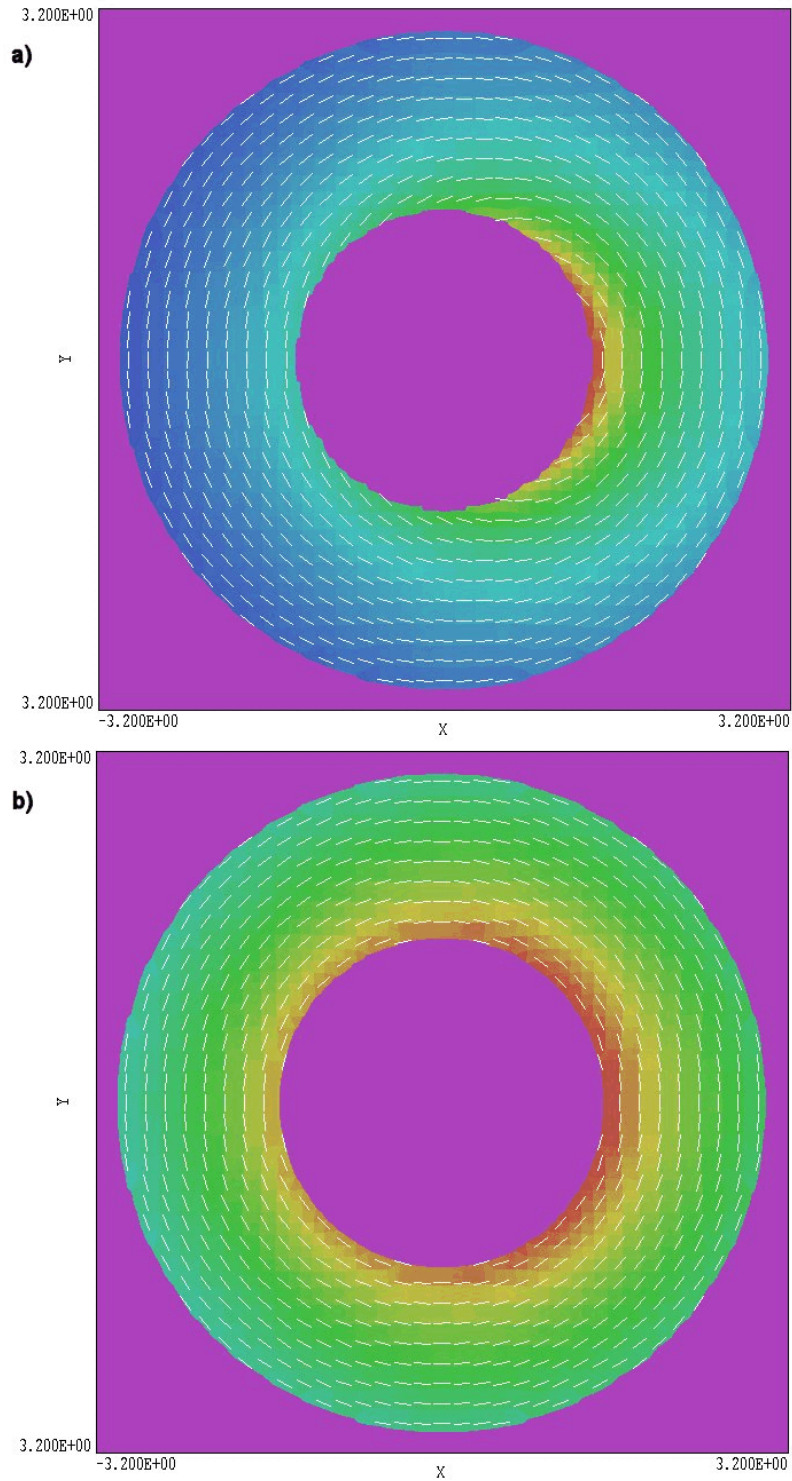


Figure 10.8. Magnetic field in planes normal to z for the BBCOAX example. a) Near the cathode surface – the peak field (red) is 3.76×10^{-3} tesla. b) 3.9 cm upstream from the cathode – the peak is 2.11×10^{-3} tesla.

10.6. Symmetry conditions in the RELBEAM tracking mode

You can substantially reduce **OmniTrak** run times by applying symmetry conditions whenever possible. For example in the calculation of Figure 10.7 we could have defined the plane $y = 0.0$ as a symmetry boundary (*bilateral symmetry*). If the cathode were not displaced we could apply *quadrilateral symmetry* about the planes $x = 0.0$ and $y = 0.0$. A symmetry boundary can be defined under the following conditions:

1) Electric and magnetic fields are symmetric about the boundary. The two possibilities for the electric field are that the field is normal to the boundary (fixed-potential Dirichlet condition, often $\phi_e = 0.0$) or parallel to the boundary (Neuman boundary with $E_{\perp} = 0.0$). The Neuman condition applies on any unspecified boundaries in a finite-element electrostatic calculation. We shall discuss boundary conditions for the magnetic field in this section.

2) Particle orbits either do not cross the boundary or have reflection symmetry. In the latter case each particle that penetrates the boundary is replaced by another particle that has identical properties except that the normal velocity is reversed. Particle reflection boundaries are defined by the *REFLECTPLANE* command.

Note that particle symmetry was not applied in the solenoid lens calculation of Sect. 10.2 because the lens induced a net rotation of the beam.

To apply symmetry in the calculation of the beam-generated magnetic field we must ensure that the contributions from both the beam and materials are individually symmetric. To illustrate, Fig. 10.9 shows a case where a beam and surrounding structures have symmetry about the plane $x = 0.0$. We shall perform the calculation in the region $x > 0.0$. The condition on the total magnetic field at the plane is

$$B_{\parallel} = 0. \quad (10.22)$$

The applied field contribution generated by the model particle current vectors will satisfy the condition of Eq. (10.22) if we create a set of virtual vectors at the mirror position ($x = -x, y = y, z = z$). Although **OmniTrak** must sum over the additional vectors, the total run time is significantly reduced because both the number of model particles and the number of nodes in the electric field mesh are halved.

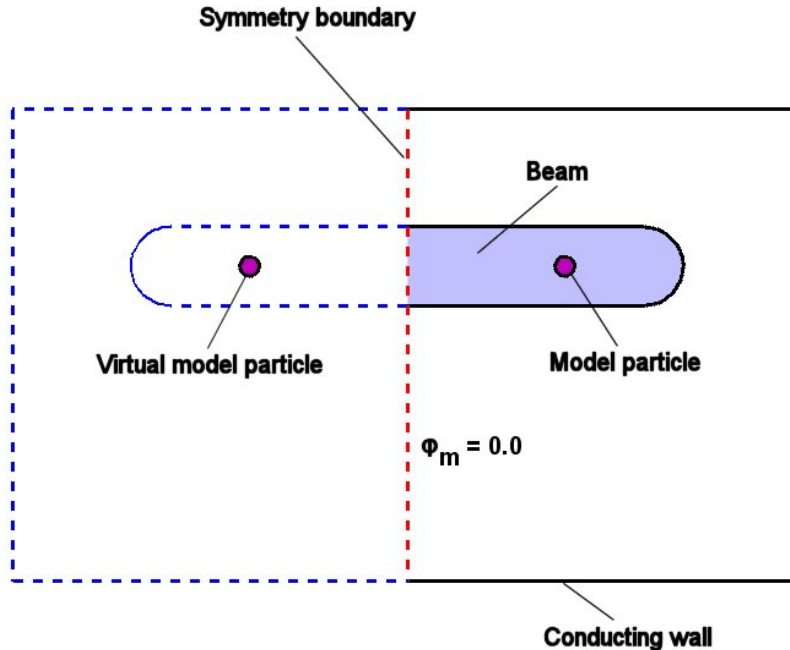


Figure 10.9. Bilateral symmetry boundary for the calculation of beam-generated magnetic fields.

We must also ensure that material field contributions are symmetric by applying an appropriate boundary condition in the reduced potential calculation. In this case, all nodes on the boundary at $x = 0.0$ are set to the fixed-potential condition $\phi_m = 0.0$. This condition guarantees that the material components satisfy Eq. (10.22). Both the generation of vectors and the definition of fixed-potential planes is performed automatically by **OmniTrak** in response to the *BBSYMPPLANE* command.

10.7. Commands for the RELBEAM mode

The command *PARTICLES RELBEAM* signals that **OmniTrak** should prepare for calculations of three-dimensional beam-generated magnetic fields. The program allocates additional space on the electric field mesh to store applied magnetic field values. If the calculation involves material contributions, **OmniTrak** must also store values of the reduced potential and coupling coefficients involved in its calculation. The operations are performed in dynamic memory and the program issues an error message if your computer does not have sufficient RAM. You must decide whether material contributions should be included. The effects of conductors are generally not important if the beam is continuous or has a pulse length long compared to the magnetic penetration time of surrounding conductors. Material effects can also be omitted if conducting boundaries are distant from the beam.

The *EMIT*, *JLIMIT* and *DTHETA* commands have the same function as in the *SCHARGE* mode. In addition, the following commands may appear in the *PARTICLES* section of the input script under the *RELBEAM* option. The first two commands control the calculation of applied fields from orbit vectors in both modes.

BBELEM LElem RElem
BBELEM 1.0 0.05

The parameters in this command control the applied field calculation described in Sect. 10.4. The real-number quantity *LElem* is the minimum length of current-element vectors stored during an orbit integration. Enter the value in units set by *DUnit*. To minimize the calculation time set *LElem* equal to the maximum value that will give a good representation of the orbits. For a gun with approximately laminar orbits a value of *LElem* that gives 5-10 vectors would be sufficient. The second parameter *RElem* is the effective radius of the current element vector (Eq. 10.21). Enter the value in units set by *DUnit*. Non-zero values of *RElem* prevent divergence of field magnitude and give transverse smoothing over the model-particle distribution. Pick *RElem* equal to or greater than the average transverse spacing between model particles. Note that a large value of *RElem* could result in excessive smoothing on the envelope of a sharp-edge beam.

BBSYMPPLANE Axis
BBSYMPPLANE X

This command ensures that the calculation of the applied and material fields created by beam current has bilateral symmetric about a given plane. The string parameter [*X*, *Y*, *Z*] specifies the axis normal to the symmetry plane. The program locates the symmetry plane on the lower side of the solution volume along the normal axis. For example, the command *BBSYMPPLANE X* defines a symmetry plane located at $x = XMinE$. For the reduced potential calculation **OmniTrak** sets all nodes on this boundary to the fixed condition with $\phi_m = 0.0$. This action guarantees that the component of magnetic field created by materials is normal to the plane. The program also creates a mirror set of current-element vectors for each particle orbit. The mirror vectors have start and end points (x' , y' , z') related to those of the orbit (x , y , z) by [$x' = -x$, $y' = y$, $z' = z$]. The contribution of the mirror vectors ensures that the applied component of magnetic field is normal to the boundary.

BBSYMPPLANE Axis1 Position1

BBSYMPPLANE Axis2 Position2

BBSYMPPLANE X 0.00

BBSYMPPLANE Y 0.00

You can define a solution with quadrilateral symmetry by including two instances of the *BBSYMPPLANE* command. The example shown above specifies that the beam distribution has symmetry about both the planes $x = XMinE$ and $y = YMinE$. In response to the two commands, **OmniTrak** sets nodes on the two boundaries equal to the fixed-potential condition $\phi_m = 0.0$. Furthermore, the program creates three sets of mirror vectors for each particle orbit with start and end coordinates related by $[x' = -x, y' = y, z' = z]$, $[x' = x, y' = -y, z' = z]$ and $[x' = -x, y' = -y, z' = z]$.

The remaining commands are used only when material contributions are included. In this case, the program must calculate the reduced potential ϕ_m using finite-element methods. The commands separate into two groups: 1) control of the relaxation solution for the reduced potential, and 2) definition of material properties. The following commands to control the relaxation solution are similar to those used in the calculation of the self-consistent electrostatic potential:

RESTARTGET B 5.0E-7

Set a value for a parameter that controls the accuracy of the iterative reduced potential solution

MAXCYCLE B 2500

Set the maximum number of iteration cycles in the magnetic field solution.

OMEGA B 1.95

Set the over-relaxation parameter for the magnetic field solution. If this command does not appear, the program uses default values determined from the Chebyshev prescription.

AVG B 0.20

The *AVG* command with the option *B* controls how beam contributions

to the applied field \mathbf{H}_s are apportioned over field/particle recalculation cycles. The *AVG* command allows you to average the predicted field contributions in the present cycle with the distributions from previous cycles. If H_i^{n-1} is a field component at node i on the previous iteration cycle and ΔH_i^n is an increment from the present orbit integration, then the node field changes according to

$$H_i^n = (1 - \text{Avg}B) H_i^{n-1} + (\text{Avg}B) \Delta H_i^n.$$

As a rule set *AvgB* equal to the parameter *AvgE* used for the beam space-charge calculation.

The following command defines magnetic properties of materials in the electric field mesh and also signals that **OmniTrak** should allocate memory and perform a calculation of the reduced potential.

MU NReg MuR

MU(5) = 1.0E-6

This command defines the relative magnetic permeability of regions in the electric field mesh for the calculation of beam-generated magnetic fields. In most simulations only two values are used: $\mu_r = 1.0$ is assigned to air or vacuum elements and $\mu_r \ll 1 \approx 1.0 \times 10^{-6}$ is assigned to ideal conductors. Note that these assignments do not affect the orbit tracking stop conditions set by the *VACUUM E* and *MATERIAL E* commands. The appearance of any *MU* command signals that a reduced-potential calculation should be performed and that the magnetic flux density should be calculated as $\mathbf{B} = \mu_0(\mathbf{H}_s - \nabla\phi)$. Otherwise, only the applied field calculation is performed and the magnetic flux density is determined as $\mathbf{B} = \mu_0\mathbf{H}_s$.

Note that the *dual potential* calculation applied in **Magnum** is not used in **OmniTrak**. The quantity ψ_m is used only to determine accurate values of \mathbf{B} inside materials with $\mu_r \gg 1.0$. The reduced potential gives accurate

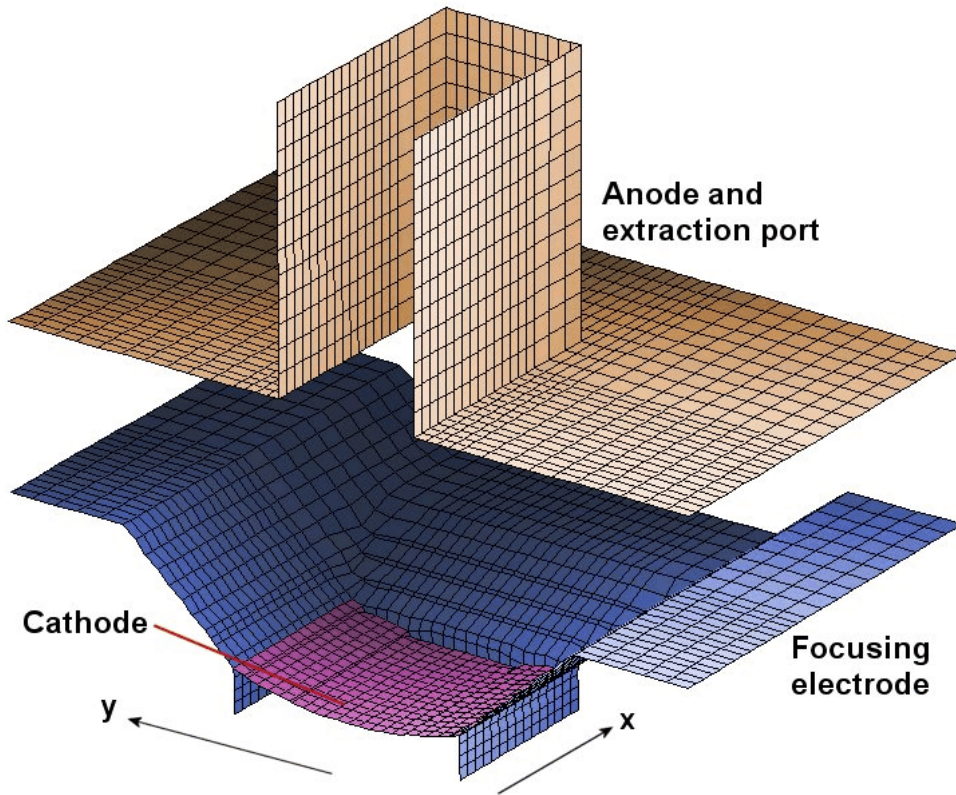


Figure 10.10. Electric field mesh for the example PLANAR.

values of \mathbf{B} in regions with $\mu_r \leq 1.0$, independent of the dual potential calculation. The typical calculation of beam-generated fields in **OmniTrak** does not include ferromagnetic materials. Even if such materials are present, the solution would be non-physical if orbits were inside.

10.8. Application example - planar beam gun

The example *PLANAR* described in this section illustrates many of the techniques to handle relativistic electron beam guns in the *RELBEAM* mode. The planar gun shown in Fig. 10.10 produces a sheet electron beam. The beam moves in the z -direction, and the gun has symmetry about the plane $x = 0.0$. The cathode has a full-width of 1.50" along x and 1.00" along y . The cathode is curved (1.50" radius) so that the beam converges in y for extraction through an anode aperture with full width 2.00" in x and 0.60" in y . The mesh is fairly coarse to ensure a short run

time. The focusing electrode geometry is an initial guess and is clearly not optimal. The flat faces at the x and y boundaries meet the cathode at the Pierce angle of 22.5° . The boundary at $x = 0.75''$, which does not follow the curvature of the cathode surface, will cause distortion of the beam distribution on the periphery. The gun voltage is 200 kV. We assume that the beam has a short pulse length ($5 \mu\text{s}$) so that conducting materials exclude the beam-generated magnetic field.

Table 10.2. OmniTrak input file for the PLANAR example

```
01: FIELDS
02:   DUnit = 39.37
03:   EField3D = PLANAR.HOU 2.0
04:   NCheck(E) = 10
05:   NCheck(B) = 10
06:   MaxCycle(E) = 250
07:   MaxCycle(B) = 250
08: END
09: PARTICLES RelBeam
10:   NCycle = 10
11:   MarkReg(5) = (0.0, -1.0, 0.110, 1)
12:   Vacuum(E) = 1
13:   Material(E) = 2
14:   Material(E) = 3
15:   Material(E) = 4
17:   Avg(E) = 0.40
18:   Avg(B) = 0.40
19:   Dt = 5.0E-12
20:   BBElem = (0.40, 0.05)
21:   ReflectPlane(XDn) = 0.0
22:   DiagPlane(ZUp) = 1.60
23:   Mu(1) = 1.0
24:   Mu(2) = 1.0E-6
25:   Mu(3) = 1.0E-6
26:   Mu(4) = 1.0E-6
27:   BBSymPlane X
28:   EDirect = Field
29: END
30: DIAGNOSTICS
31:   EDump PLANAR02E.HOU
32:   BBDump PLANAR02BB.GOU
33: END
34: ENDFILE
```

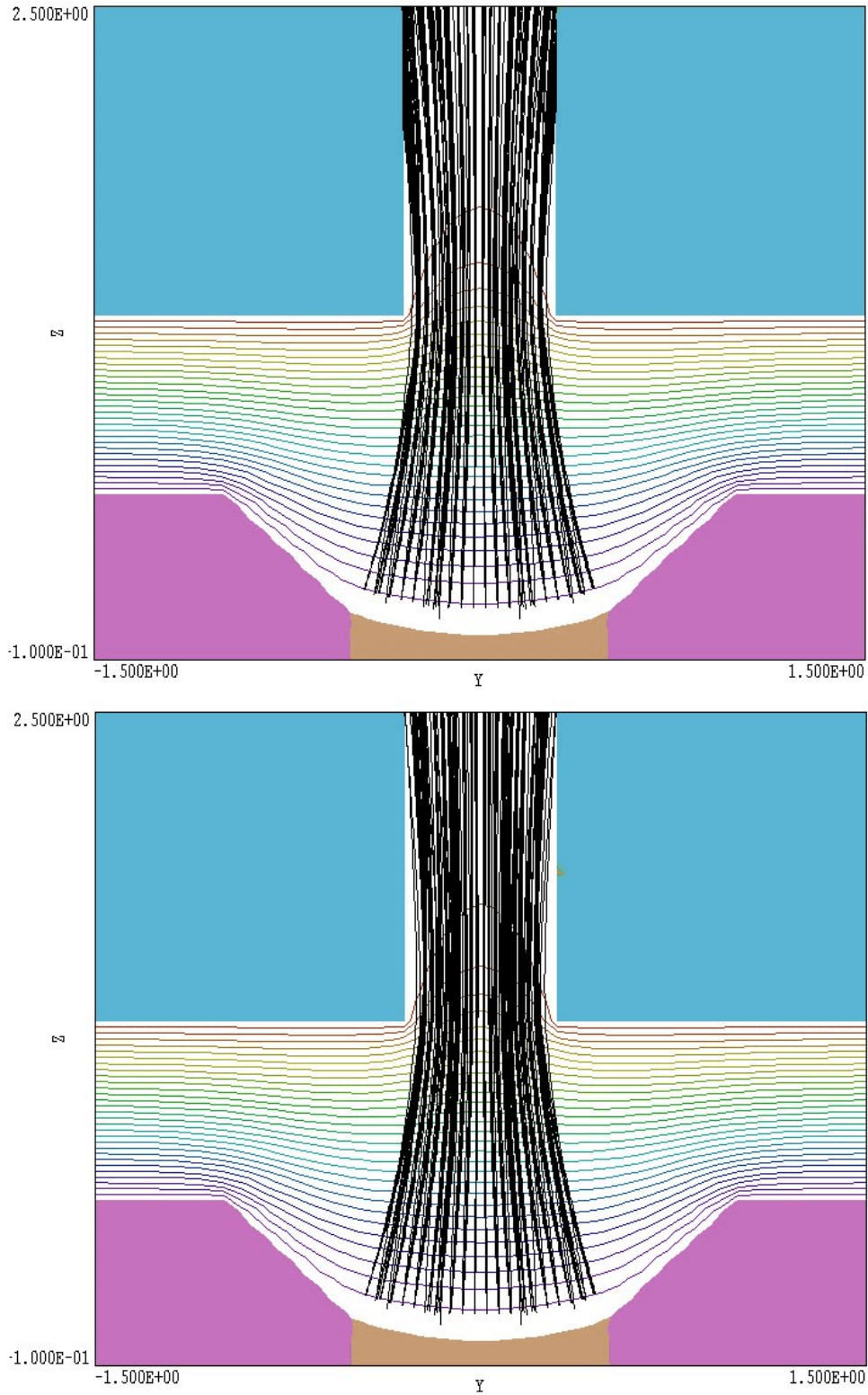



Figure 10.11. PLANAR example, projected particle orbits and equipotential lines in the plane $x = 0.0$. Top: No beam-generated magnetic field. Bottom: With self-consistent beam-generated field.

Two **OmniTrak** input files are supplied: PLANAR01.OIN has no beam-generated magnetic field while PLANAR02.OIN models all effects. Table 10.2 lists the contents of PLANAR02.OIN with added line numbers. The *FIELDS* section includes two commands (Lines 05 and 07) to control the additional relaxation solution for the reduced potential ϕ_m . Many of the commands in the *PARTICLES* section control the electric field solution and Child law emission from the cathode. We shall concentrate on new commands that appear in the *RELBEAM* mode. The *BBELEM* command (Line 20) sets $LElem = 0.4$ " and $RElem = 0.05$ ". The choice of $LElem$ gives 5 or more current vectors over the 2.50" track length of the model particles, while the choice of $RElem$ sets the effective diameter of the current elements approximately equal to the inter-particle spacing. The *REFLECTPLANE* statement (Line 21) guarantees particle orbit symmetry about the plane $x = 0.0$. The *DIAGPLANE* command (Line 22) instructs the program to record orbit parameters near the waist point at $z = 1.60$ " while continuing orbits to the axial boundary at $z = 2.50$ ". The statements in Lines 23 through 26 set values of relative magnetic permeability for the reduced potential calculation in regions with non-zero volume. The cathode, focusing electrode and anode are defined as ideal conductors. In response to the *BBSYMPPLANE* command in Line 27, **OmniTrak** sets all nodes in the plane $x = 0.0$ to the fixed potential condition $\phi_m = 0.0$ for the reduced-potential calculation. The properties of the nodes for the electric field calculation (Neumann condition) are unaffected. The final statement of interest is *EDIRECT = EFIELD* in Line 28. The coarse resolution of the curved cathode leads to some small surface facets with skewed normal vectors. In this case, it was more accurate to launch particles at the Child-law emission surface parallel to the local direction of the electric field.

The emitted current was 62.67 A. The total solution time was 500 s on a 2.4 GHz Pentium IV computer. Figure 10.11 shows projected particle orbits and equipotential lines for the electric field solution in the plane $x = 0.0$. The solution on the top has no beam-generated magnetic field while self-consistent magnetic fields are included in the solution on the bottom. The addition of magnetic confining forces allows the full beam to propagate through the exit aperture. Finally Figure 10.12 shows particle orbit intersections and elements color-coded by $|\mathbf{B}|$ in the waist plane ($z = 1.60$ "). For comparison, the predicted magnetic field at the edge of a sheet beam with 62.67 A distributed over 0.75" ($J = 3290$ A/m) is 2.073×10^{-3} tesla. Note the distortion of the beam distribution on the edge caused by the lip on the focusing electrode.

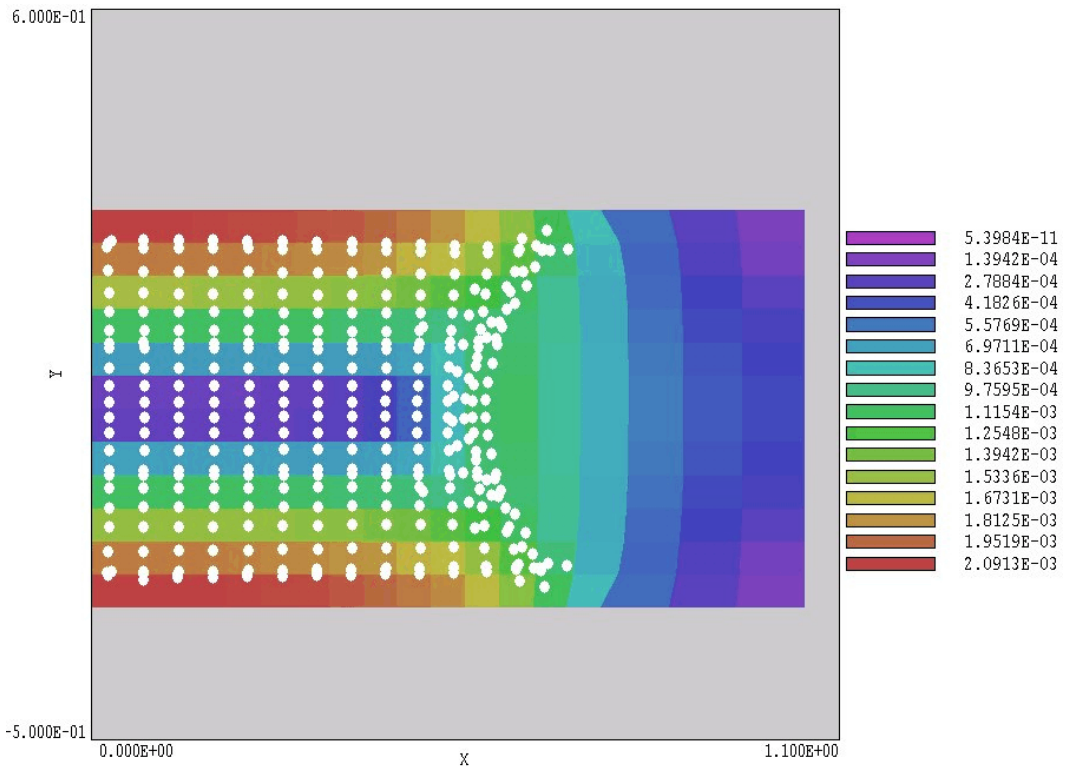


Figure 10.12. Orbit intersections with and $|B|$ in the plane $z = 1.60$ " for the PLANAR example.

11. Self-consistent field emission - PARTICLES section

In the *FEMIT* particle tracking mode, **OmniTrak** can model electron field emission. As in the *TRACK*, *SCHARGE* and *RELBEAM* modes, the program can automatically generate particles over marked source surfaces. The calculations include self-consistent effects of space charge, and you can add additional electrons or ions using the *PLIST* or *PFILE* commands. There are three differences from the *SCHARGE* mode:

- Only electrons can be created on source surfaces.
- Because the electric field always has a non-zero value on the source surface, it is not necessary to create an emission surface. Electrons are generated directly adjacent to the source facets.
- Electron current is assigned according to the Fowler-Nordheim equation rather than the Child-law algorithm.

OmniTrak uses the Fowler-Nordheim functions tabulated in A. Modinos, **Emission Spectroscopy** (Plenum Press, New York, 1984), p. 12. If the quantity E is the local electric field (including space-charge contributions) at a source facet with work function ϕ , then current density is given by

$$j_{FE} = 1.537 \times 10^{-6} \frac{e^{\Gamma} E^2}{\phi \nu(\chi)^2} \quad (A/m^2), \quad (11.1)$$

where

$$\Gamma = -6.83 \times 10^9 \frac{\phi^{3/2} \nu(\chi)}{E}, \quad (11.2)$$

and

$$\chi = 3.79 \times 10^{-5} \frac{\sqrt{E}}{\phi}. \quad (11.3)$$

In the equations E is expressed in V/m and ϕ in eV. Table 11.1 lists values for the functions $v(\chi)$ and $t(\chi)$. **OmniTrak** uses a cubic spline interpolation to find intermediate values.

| Table 11.1. Fowler-Nordheim functions | | |
|--|-----------|-----------|
| χ | $v(\chi)$ | $t(\chi)$ |
| 0.0000 | 1.0000 | 1.0000 |
| 0.0500 | 0.9948 | 1.0011 |
| 0.1000 | 0.9817 | 1.0036 |
| 0.1500 | 0.9622 | 1.0070 |
| 0.2000 | 0.9370 | 1.0111 |
| 0.2500 | 0.9068 | 1.0157 |
| 0.3000 | 0.8718 | 1.0207 |
| 0.3500 | 0.8323 | 1.0262 |
| 0.4000 | 0.7888 | 1.0319 |
| 0.4500 | 0.7413 | 1.0378 |
| 0.5000 | 0.6900 | 1.0439 |
| 0.5500 | 0.6351 | 1.0502 |
| 0.6000 | 0.5768 | 1.0565 |
| 0.6500 | 0.5132 | 1.0631 |
| 0.7000 | 0.4504 | 1.0697 |
| 0.7500 | 0.3825 | 1.0765 |
| 0.8000 | 0.3117 | 1.0832 |
| 0.8500 | 0.2379 | 1.0900 |
| 0.9000 | 0.1613 | 1.0969 |
| 0.9500 | 0.0820 | 1.1037 |
| 1.0000 | 0.0000 | 1.1107 |

The set of allowed commands is similar to that for the *SCHARGE* mode. The one difference is that alternate parameters should appear in the *EMIT* command.

EMIT WorkFunc NDivide [Beta]
EMIT 4 3.56 2 1000.0

This command identifies region number *RegNo* (integer) as a source surface and sets associated emission properties. **OmniTrak** issues an error message if the region does not define a valid source surface (*i.e.*, sets of nodes on the edges of facets between fixed-potential *Material* elements and *Vacuum* elements). At least one *EMIT* command is required under the *FEMIT* option. Up to twenty *EMIT* commands may appear in the *PARTICLES* section. Because only electrons are allowed, it is not necessary to specify mass and charge of emitted particles. The real number parameter *WorkFunc* is the work function ϕ in eV (electron volts). The integer parameter *NDivide* governs how many model electrons are created per surface facet. The quantity equals the number of divisions along the normal axes of the facet. The number of particles per facet equals $NDivide^2$. The optional parameter β is a field enhancement factor that may be useful (for example) to simulate emission from carbon nanotube assemblies. If E_{loc} is the local electric field at the facet, then the quantity $E = \beta E_{loc}$ is used in evaluating Eqs. 11.1 through 11.3.

SUPPRESS SVal1 SVal2 SVal3 ...
SUPPRESS 0.20 0.30 0.40 0.60 0.80 1.00

Space-charge effects are usually small in field emission problems so the role of the suppression factors discussed in Chapter 10 is not as critical. The default values in the *FEMIT* mode are

| NCycle | Supression value |
|--------|------------------|
| ===== | |
| 1 | 0.500 |
| 2 | 0.750 |
| 3 | 1.000 |
| 4 | 1.000 |
| ... | |

Note that the number of iteration cycles should be in the range $NCycle \geq 3$ for the default values of suppression factors.

12. Space-charge-limited emission from plasma surfaces - PARTICLES section

(In preparation)

13. Electron secondary emission - PARTICLES section

13.1. Theoretical background

OmniTrak can represent secondary-electron emission processes to model electro-optical devices and collectors for high-power vacuum tubes. This capability applies only to electrons. Secondary emission calculations can be performed in the *TRACK*, *SCHARGE*, *RELBEAM* and *FEMIT* modes. In the latter modes the space-charge of the emitted electrons is added to the electric field recalculation.

A single model electron can represent a multi-generational set (incident electron plus one or more secondaries generated in sequential collisions). Regions with elements corresponding to secondary emitters are defined with the *SECONDARY* command. When an electron enters a secondary element, **OmniTrak** calculates the secondary emission coefficient δ from the energy E of the incident electron and the angle α between the orbit vector and a vector normal to the material surface. A new orbit to represent secondaries is started at a position in the vacuum space near the surface entrance point. For tracking modes with space-charge, the secondary is assigned a current equal to that of the incident electron multiplied by δ . The kinetic energy of the emitted electron follows a Maxwell distribution with $T_e = 2.0$ eV. The direction of emission is normal to the surface.

OmniTrak determines the secondary emission coefficient from a parametric model based on work by Jonker¹ and Vaughn². The model involves the angle α between the direction of the incident electron and the surface-normal vector. The defining function for the angular dependence of δ is

$$F(\alpha) = \frac{1}{\sqrt{\cos(\alpha)}} . \quad (13.1)$$

The maximum value of secondary coefficient and the corresponding incident electron energy are given by

$$\begin{aligned}\delta_m &= \delta_{mo} F(\alpha), \\ E_m &= E_{mo} F(\alpha),\end{aligned}\tag{13.2}$$

where δ_{mo} and E_{mo} are the values at normal incidence. Table 13.1 lists these quantities for a variety of materials. The secondary emission coefficient as a function of the angle α and kinetic energy E of the incident electron is given approximately as

$$\delta(\alpha, E) \approx \delta_m(\alpha) [f e^{(1-f)}]^a ,\tag{13.3}$$

where $f = E/E_m(\alpha)$. The parameter a has the value 0.62 for $f < 1$ and 0.25 for $f \geq 1$. To prevent infinite values of δ , the code takes $\alpha = 80^\circ$ for orbits with incident angles that exceed 80° . Reference 3 (included with the package) gives a detailed description of numerical methods used to treat secondary emission in the **Trak** code.

-
1. J.L.H. Jonker, Phillips Research Reports **6**, 372 (1951), Phillips Research Reports **7**, 1 (1952), Phillips Research Reports **12**, 249 (1957).
 2. R.M. Vaughn, IEEE Trans. Electron Devices **ED-36**, 1963 (1989) and IEEE. Trans. Electron Devices **ED-40**, 830 (1993).
 3. S. Humphries, N. Dione and J. Petillo, *Secondary-electron emission modeling on a conformal mesh*, **Proc. WorkShop on RF Superconductivity**, Santa Fe, 1999.
-

Table 13.1. Secondary emission parameters for elements

| Element | δ_{mo} | E_{mo} (eV) | Element | δ_{mo} | E_{mo} (eV) |
|--------------|---------------|---------------|---------|---------------|---------------|
| Ag | 1.5 | 800 | Li | 0.5 | 85 |
| Al | 1.0 | 300 | Mg | 0.95 | 300 |
| Au | 1.4 | 800 | Mo | 1.25 | 375 |
| B | 1.2 | 150 | Na | 0.82 | 300 |
| Ba | 0.8 | 400 | Nb | 1.2 | 375 |
| Bi | 1.2 | 550 | Ni | 1.3 | 550 |
| Be | 0.5 | 200 | Pb | 1.1 | 500 |
| C (diamond) | 2.8 | 750 | Pd | >1.3 | >250 |
| C (graphite) | 1.0 | 300 | Pt | 1.8 | 700 |
| C (soot) | 0.45 | 500 | Rb | 0.9 | 350 |
| Cd | 1.1 | 450 | Sb | 1.3 | 600 |
| Co | 1.2 | 600 | Si | 1.1 | 250 |
| Cs | 0.7 | 400 | Sn | 1.35 | 500 |
| Cu | 1.3 | 600 | Ta | 1.3 | 600 |
| Fe | 1.3 | 400 | Th | 1.1 | 800 |
| Ga | 1.55 | 500 | Ti | 0.9 | 280 |
| Ge | 1.15 | 500 | Tl | 1.7 | 650 |
| Hg | 13 | 600 | W | 1.4 | 650 |
| K | 0.7 | 200 | Zr | 1.1 | 350 |

Adapted from D.R. Lide, ed., **Handbook of Chemistry and Physics, 74th Edition** (CRC Press, Boca Raton, 1993), 12-107

Table 13.2. Secondary emission parameters for compounds

| Element | δ_{mo} | E_{mo} (eV) | Element | δ_{mo} | E_{mo} (eV) |
|-----------------------|---------------|---------------|--|---------------|---------------|
| Alkali halides | | | Oxides | | |
| CsCl | 6.5 | | Ag ₂ O | 1.0 | 2-9 |
| KBr (crystal) | 14 | 1800 | Al ₂ O ₃ (layer) | 2.3-4.8 | 400 |
| KCl (crystal) | 12 | 1600 | BaO (layer) | 3.4 | 2000 |
| KCl (layer) | 7.5 | 1200 | BeO | 2.2 | 500 |
| KI (crystal) | 10 | 1600 | CaO | 1.2 | 400 |
| KI (layer) | 5.6 | | Cu ₂ O | 20-25 | 1500 |
| LiF (crystal) | 8.5 | | MgO (crystal) | 3-15 | 400-1500 |
| LiF (layer) | 5.6 | 700 | MgO (layer) | 1.2 | |
| NaBr (crystal) | 24 | 1800 | MoO ₂ | 2.1-4 | 400 |
| NaBr (layer) | 6.3 | | SiO ₂ (quartz) | 3.2 | 640 |
| NaCl (crystal) | 14 | 1200 | SnO ₂ | | |
| NaCl (layer) | 6.8 | 600 | Sulfides | | |
| NaF (crystal) | 14 | 1200 | MoS ₂ | 1.1 | |
| NaF (layer) | 5.7 | | PbS | 1.2 | 500 |
| NaI (crystal) | 19 | 1300 | WS ₂ | 1.0 | |
| NaI (layer) | 5.5 | | ZnS | 1.8 | 350 |
| RbCl (layer) | 5.8 | | Others | | |
| | | | BaF ₂ (layer) | 4.5 | |
| | | | CaF ₂ (layer) | 3.2 | |
| | | | BiCs ₃ | 6 | 1000 |
| | | | BiCs | 1.9 | 1000 |
| | | | GeCs | 7 | 700 |
| | | | Rb ₃ Sb | 7.1 | 450 |
| | | | SbCs ₃ | 6 | 700 |
| | | | Mica | 2.4 | 350 |
| | | | Glasses | 2-3 | 300-450 |

Adapted from D.R. Lide, ed., **Handbook of Chemistry and Physics, 74th Edition** (CRC Press, Boca Raton, 1993), 12-107

13.2. Control commands

The following commands may appear in the *TRACK*, *SCHARGE*, *SCEMIT* and *FEMIT* sections.

SECONDARY [E,B] NReg DeltaMax0 EngMax0 **SECONDARY (E,5): 2.45 320.0**

This command designates region NReg in the electric or magnetic field meshes as a SECONDARY type and assigns emission parameters. The real-number quantity *DeltaMax0* (real) is the maximum value of the secondary emission coefficient for normal incidence. The quantity *EngMax0* (real) is the kinetic energy of the incident electron (in eV) at which the maximum occurs.

SECONDPARAM KECutOff MinFact **SECONDPARAM 1.0 2.0E-3**

This command sets global parameters that control the termination of multi-generational orbits to prevent infinite loops. *KECutOff* is a cutoff value (in eV) for kinetic energy. The orbit terminates if the energy of the incident electron falls below this value. The default is *KECutOff* = 2.5 eV. During a multi-generational orbit calculation **OmniTrak** maintains a quantity *MultFact* equal to the effective number of electrons in a generation per incident electron. This quantity equals the product of secondary emission coefficients for collisions with secondary materials, $MultFact(N) = \delta_1 \delta_2 \delta_3 \dots \delta_N$. An orbit is terminated if the multiplication factor drops below the value *MinFact*. The default is $MinFact = 1.0 \times 10^{-4}$.

SECONDLIST

This command initiates a detailed listing of secondary emission processes during an orbit integral. In runs with *NCycle* > 1, the listing occurs only on the final cycle.

Finally, additional information on particle multiplication is contained in data written in response to the *REGLIST* command in the *DIAGNOSTICS* section.

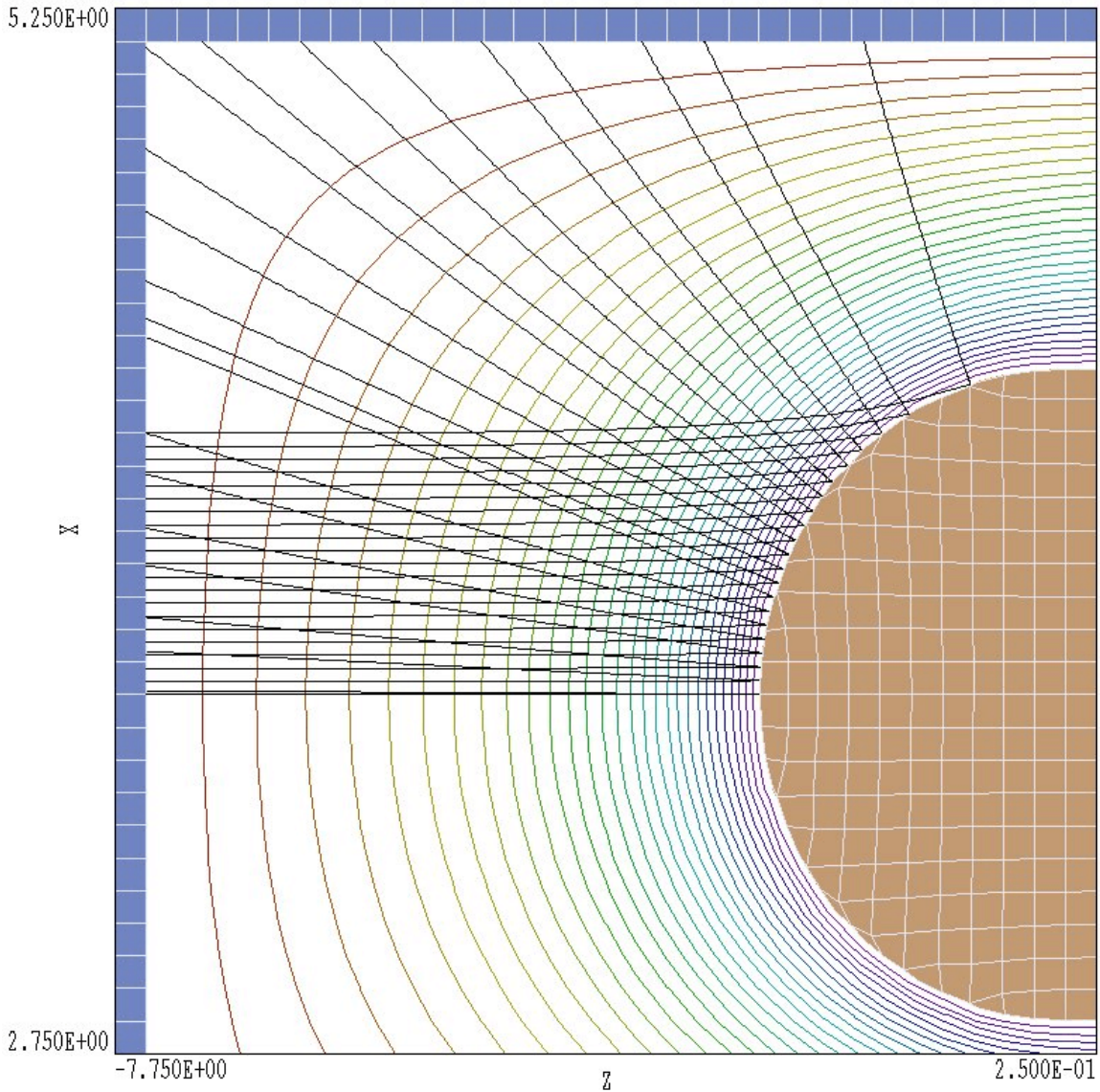


Figure 13.1. Orbits and electronstatic equipotentials for the example SECONDTEST in the plane $y = 0.0$.

13.3. Benchmark test

The example SECONDTEST illustrates the mechanism for treating secondary emission. The file SECONDTEST.OIN is listed in Table 13.3. Figure 13.1 shows the geometry. Twenty-one electrons with 300 eV kinetic energy are launched from the grounded left-hand boundary (Region 1). They strike a spherical electrode (Region 3) biased to a potential of -100 V. The electrode is coated with aluminum oxide and has secondary emission properties $\delta_{mo} \approx 3.5$ and $E_{mo} = 400$ eV. For each orbit, **OmniTrak** performs the following operations: 1) track the incident

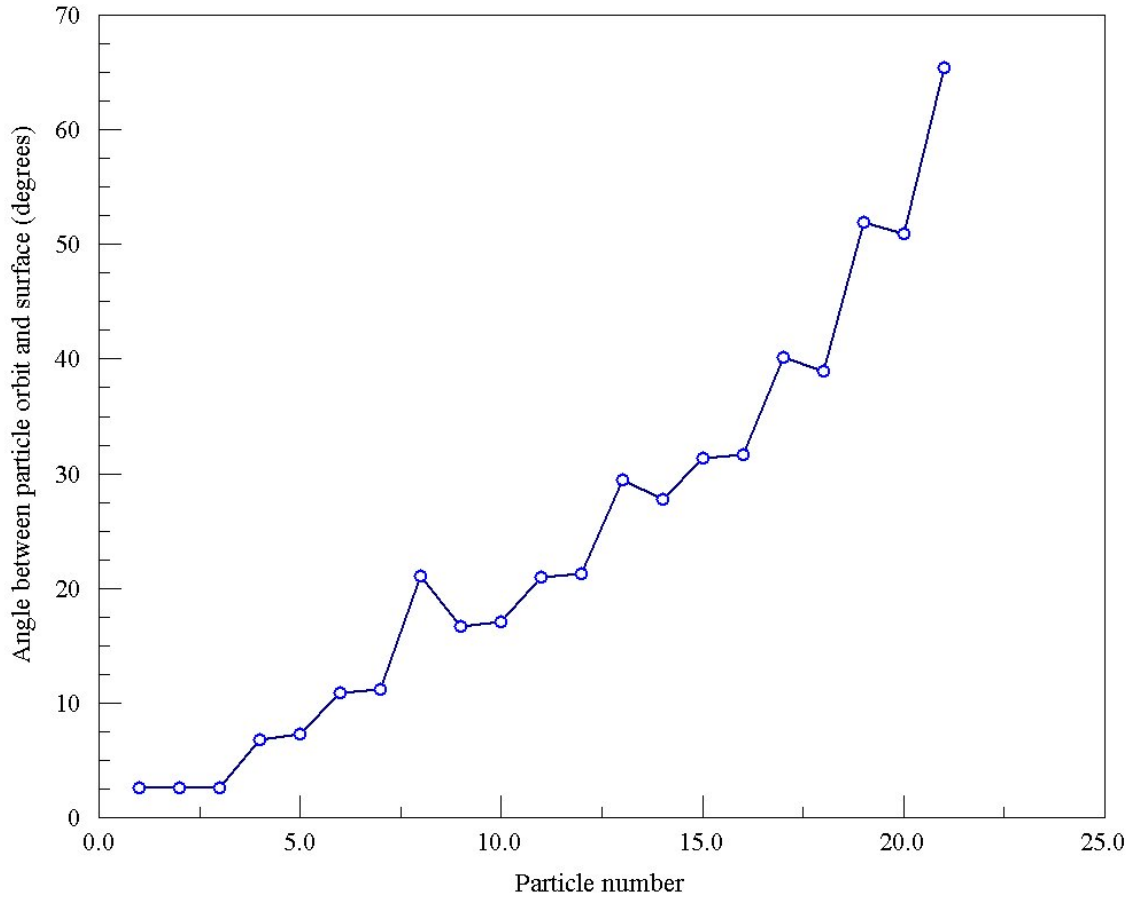


Figure 13.2. Calculated angle between the particle orbit and a vector normal to the spherical surface as a function of particle number in example SECONDTEST.

particle until it enters a secondary element, 2) calculate the point where the electron penetrates the surface and find the angle between the orbit and surface-normal vectors, 3) find the secondary emission coefficient δ according to Eqs. (13.1) through (13.3), 4) restart an electron near the surface penetration point with low kinetic energy and a weight adjusted by δ , 5) track the secondary-electron orbit until penetrates an element of the material wall, 6) calculate the penetration point on the wall and 7) find interpolated final parameters. In order to ensure success in the surface location operations, the run uses a relatively short time step of 0.5×10^{-10} s. An electron moves a distance of about 0.42 mm in a step, small compared to the element width of approximately 2.5 mm. The following entry in SECONDTEST.LST shows that all 42 surface identification operations were successful.

Status of surface indentification operations at orbit termination
 Total number of operations: 42
 Fraction of successful operations: 1.000

Inspection of Figure 13.1 shows that orbit 1 strikes the sphere at approximately normal incidence while orbit 21 strikes at grazing incidence. Figure 13.2 shows a plot of the calculated surface angle α versus particle number. Variations in the plot result from the fact that the surface is faceted rather than spherical. In response to the *REGLIST* command in the *DIAGNOSTICS* section, **OmniTrak** makes the following entry in the list file:

```

ELECTRICAL MESH
Region      Number      Number absorbed      Total current
number      of hits
=====
1           21           72.556               0.0000E+00
3           21          -51.556               0.0000E+00

```

For 21.000 electrons incident on the sphere, a number 72.556 are absorbed in the wall. Therefore, the average second emission coefficient is $\delta_{\text{avg}} = 3.455$. Finally, in response to the *MULTFACT* command the code creates the following table:

| Part No | MultFact | NStrike | Time | Distance |
|---------|------------|---------|------------|------------|
| 1 | 3.1101E+00 | 2 | 1.6167E-08 | 9.9855E-02 |
| 2 | 3.1117E+00 | 2 | 1.6471E-08 | 9.9942E-02 |
| 3 | 3.1011E+00 | 2 | 1.6405E-08 | 1.0050E-01 |
| 4 | 3.1158E+00 | 2 | 1.6737E-08 | 1.0092E-01 |
| 5 | 3.1211E+00 | 2 | 1.6568E-08 | 1.0143E-01 |
| 6 | 3.1486E+00 | 2 | 1.6642E-08 | 1.0222E-01 |
| 7 | 3.1454E+00 | 2 | 1.7099E-08 | 1.0334E-01 |
| 8 | 3.2482E+00 | 2 | 1.7704E-08 | 1.0567E-01 |
| 9 | 3.1908E+00 | 2 | 1.7380E-08 | 1.0705E-01 |
| 10 | 3.2052E+00 | 2 | 1.7614E-08 | 1.0829E-01 |
| 11 | 3.2562E+00 | 2 | 1.7956E-08 | 1.1088E-01 |
| 12 | 3.2515E+00 | 2 | 1.8872E-08 | 1.1379E-01 |
| 13 | 3.4030E+00 | 2 | 1.9470E-08 | 1.1872E-01 |
| 14 | 3.3621E+00 | 2 | 1.8908E-08 | 1.1558E-01 |
| 15 | 3.4485E+00 | 2 | 1.7484E-08 | 1.1013E-01 |
| 16 | 3.4574E+00 | 2 | 1.6986E-08 | 1.0661E-01 |
| 17 | 3.6917E+00 | 2 | 1.5843E-08 | 1.0151E-01 |
| 18 | 3.6581E+00 | 2 | 1.5429E-08 | 9.9379E-02 |
| 19 | 4.1991E+00 | 2 | 1.4751E-08 | 9.6011E-02 |
| 20 | 4.1468E+00 | 2 | 1.4479E-08 | 9.5420E-02 |
| 21 | 5.1838E+00 | 2 | 1.4391E-08 | 9.5455E-02 |

For the single-event simulation, the quantity *MultFact* equals the secondary emission coefficient on the surface of Region 3. The value of δ increases with decreasing angle of incidence.

Table 13.3. File SECONDTEST.OIN

```

FIELDS
  DUnit = 100.0
  EField3D = SECONDTEST.HOU
END
PARTICLES Track
  Material(E) = 1
  Vacuum(E) = 2
  Secondary(E) = 3 (3.5, 400.0)
  OrbInfo
  SecondList
  Dt = 0.5E-10
  Plist
* Mass Chrg Eng  x  y  z  px  py  pz
* =====
  0.0 -1.0 300.0  0.00  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  0.10  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  0.20  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  0.30  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  0.40  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  0.50  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  0.60  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  0.70  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  0.80  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  0.90  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  1.00  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  1.10  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  1.20  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  1.30  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  1.40  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  1.50  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  1.60  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  1.70  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  1.80  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  1.90  0.001 -7.4999  0.0  0.0  1.0
  0.0 -1.0 300.0  2.00  0.001 -7.4999  0.0  0.0  1.0
  End
END
DIAGNOSTICS
  PartList
  RegList
  MultFact
END
ENDFILE

```

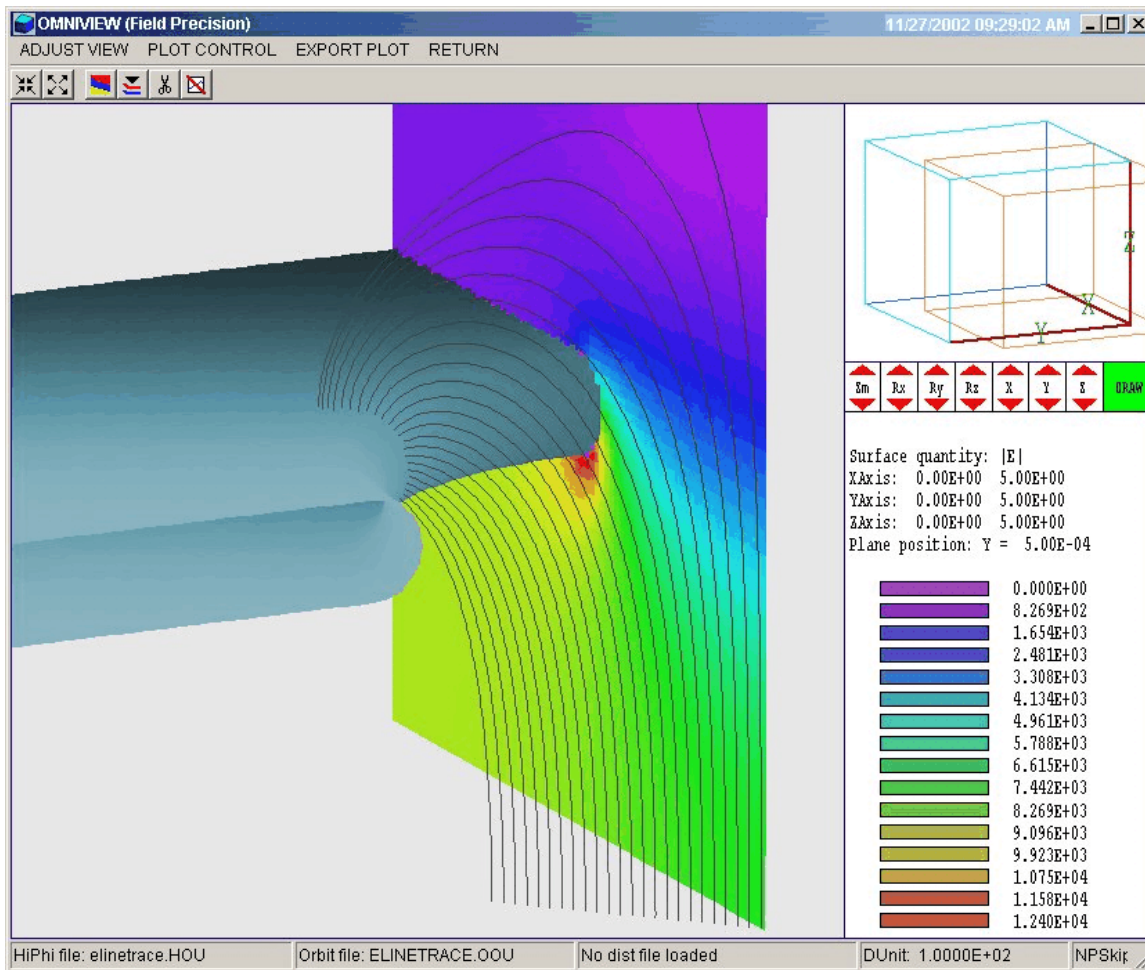



Figure 14.1. Three-dimensional plot of electric field lines near a circular aperture.

14. Tracing electric and magnetic field lines - PARTICLES section

We have added a mode to the *PARTICLES* section of **OmniTrak** for precision three-dimensional tracing of lines of electric field (**E**) and magnetic flux density (**B**). In this case the section begins with the *PARTICLES* command followed by the keyword *FLINE*.

Calculated coordinates of points along field lines are recorded in the plot file (TOU) in a format similar to that used for particle orbits. Therefore, you can use all the two- and three-dimensional plotting capabilities of

OmniView to superimpose field lines on the standard field information. Besides the expanded plotting capabilities, there are technical motivations for field-line tracing. For example, to calculate ion drift motion in a resistive solution it may be important to know the exact position of a field line termination. In a high-voltage system with magnet insulation it is possible to find detailed shapes of magnetic field lines near structures with high electric field stress.

14.1. Commands for field line tracing

OmniTrak can trace up to 5000 field lines in a run. The initiation of lines can be accomplished with a list or source surfaces. Regarding lists, field-line information lines can be inserted directly into the input command file with the *FList* command or it can be read from an independent file with the *FFile* command. For both the *FList* and *FFile* commands, **OmniTrak** expects a series of text file lines consisting of three real numbers that give the coordinates of the start point and an optional string that specifies the direction of the trace. The *FList* section has the format

```
FLIST
  <Field Line 1>
  <Field Line 2>
  ...
  <Field Line NPart>
END
```

Real numbers can be expressed in any valid floating point form. Input follows the standard Field Precision free format using the delimiters listed in Sect. 3.2. Each line has the following components.

```
x   y   z   [Polarity]
```

Enter the coordinates in units set by *DUnit*. The optional polarity string may have the values *NEG* and *POS*. In the *POS* option (the default) the spatial integration proceeds away from the starting point in the direction of positive electric or magnetic field. Note that line entries can be separated by any of the valid delimiters listed in Sect. 3.2.

FLIST

This command signals that a series of lines containing field starting points will follow in the **OmniTrak** script. The list may contain blank and comment lines and must terminate with the *End* command. Comment lines are marked by the asterisk (*) symbol. The *FList*

sequence can appear anywhere in the *Particles* section.

FFILE FPrefix

FFILE HVPulser.FLD

This command specifies that **OmniTrak** should read particle parameter lines from an external file with a name of the form *FPrefix.FLD*. The file prefix may contain from 1 to 20 characters. The file must be available in the working directory. A field line file may consist of blank lines, comments and from 1 to 5000 data lines. The file must terminate with the *End* command.

```
<Field Line 1>  
<Field Line 2>  
...  
<Field Line NPart>  
END
```

The following command initiates field lines from one or more source surfaces.

EMIT NReg NDivide [Polarity]

EMIT = 5 (2, NEG)

The *EMIT* command in the *FLINE* mode has a function similar to that in the *TRACK* mode. **OmniTrak** searches for source facets that have four nodes marked with region number *NReg*. The facets must separate elements with the fixed-potential *Material* designation and the *Vacuum* designation. In other words, the source should be on the surface of an electrode adjacent to a vacuum or dielectric region. The integer parameter *NDivide* governs the number of field lines created per facet ($NDivide^2$). The string parameter *Polarity* (*NEG* or *POS*) governs the tracing direction (default: *POS*).

There are two additional commands that control the tracing process.

DS 0.010

Set the length step for spatial integrations. Enter the value of *Ds* in units set by *DUnit*. If the *DS* command does not appear, **OmniTrak** sets a default approximately equal to the smallest element length along *x*, *y* or *z*.

SMAX 10.0

Set a maximum value for the length of a field line. If the command does not appear, **OmniTrak** sets a default value equal to twice the length of the longest side of the solution volume. The default prevents an infinite code loop for calculations of magnetic field lines.

14.2. Conventions for field line tracing

This section lists some considerations for using the *FLINE* option.

- Starting and stopping conventions are the same as those for particle integrations. Field lines must start in a region marked *Vacuum*. They terminate if they enter a *Material* region. By default, **OmniTrak** sets regions with fixed potential, $\epsilon_r \neq 1$ and $\mu_r \neq 1$ as *Material*. For field line tracking into dielectric and ferromagnetic materials, you should actively set the region properties using the `VACUUM` command.
- Precision stopping conditions can be used in field line tracking. Stopping and diagnostic planes function the same way they do in particle tracking. Reflection planes are not recognized. **OmniTrak** performs interpolations to project the end points of field lines to stopping/diagnostic planes or to the surface of the solution volume. Furthermore field lines are projected back to entrance point when they enter a *Material* region.
- In a single run you can load either a three-dimensional electric or magnetic field solution. **OmniTrak** issues an error message if both solution types loaded.
- If the script contains the *LISTON* command, **OmniTrak** records a set of values for electric or magnetic field along each field line. The listing information depends on which type of field solution has been loaded. The *LISTYPE* command is inactive.
- In two-dimensional magnetic calculations, contours of vector potential A_z or stream function rA_θ lie along magnetic field lines and are separated by intervals of equal magnetic flux. In contrast, the absolute distance between field lines in the three-dimensional plots of **OmniView** do not have a special meaning – you can start field lines at any positions with the *FLIST*, *FFILE* and *EMIT* commands. Nonetheless, you can get a sense of changes in electric or magnetic field magnitude by observing the *relative* distance between two lines. The field strength is inversely proportional to the line spacing.

- In particle tracking four quantities are recorded on each line of the plot file (TOU): *Time*, *x*, *y* and *z*. Because elapsed time has no meaning in a field line calculation, *OmniTrak* records the following quantities in the plot file: *Distance*, *x*, *y* and *z*. The quantity *Distance* is the line length measured from the starting point.
- The command *FLINELIST* in the *DIAGNOSTICS* section creates a list of the initial and final positions of field lines. **OmniTrak** records the quantity $\int \mathbf{E} \cdot d\mathbf{l}$ for electric field traces. The quantity equals the change in voltage along the line. If you start a line close to one fixed-potential region and it terminates on another, then the voltage drop should be close to the difference in applied potential. Lower the value of *Ds* to increase the accuracy of the solution.

15. DIAGNOSTICS section

OmniTrak records information on the progress of the run in the listing file `RunName.OLS`. After computing particle orbits, the program can perform several diagnostics in response to commands in the input file. These are contained in the *DIAGNOSTICS* section that follows the *PARTICLES* section. Diagnostic commands are contained between the lines *DIAGNOSTICS* and *END*. After the *DIAGNOSTICS* section (or if there is no section), **OmniTrak** continues to the *ENDFILE* command, closes all files, and terminates operation.

15.1. Field diagnostics

OmniTrak can write field quantities to the listing file and can also generate standard files of modified field solutions. It is important to note that the field values reported correspond to field solutions that may have been changed during the **OmniTrak** run. Field modifications may result from the following causes:

- Global scaling of the field values when *EMult3D*, *BMult3D* or *BMult2D* do not equal unity in the *EFIELD3D*, *BFIELD2D* and *BFIELD3D* commands.
- Modifications of individual region potentials in the electric field solution (*CHANGE POT* command).
- Coordinate shifts (*SHIFT* command) and rotations of two-dimensional magnetic field solutions (*ROTB2* command).
- Beam space-charge contributions to the three-dimensional electric field solution.
- Beam current contributions to the three-dimensional magnetic field solution

If a field solution changes for any reason during the **OmniTrak** run, you should instruct the program to write a modified output file for plotting in **OmniView**. If you load the original, unchanged field into **OmniView**, the mesh structures and field values will not be appropriate for the particle orbits. Use the *EDUMP* and *BDUMP* commands to record modified field distributions.

The *EPOINT* and *ESCAN* commands write electric field values to the listing file. Calculations are performed on the final state of the electric field solution using the same routines applied in the orbit integration. The commands are useful for the following applications:

- Check the validity of field interpolations in critical areas.
- Find values of ϕ and \mathbf{E} that include contributions from beam space charge.
- Confirm that shift and scaling operations have produced the correct field values.

EPOINT Xp Yp Zp

EPOINT = (0.567, 12.27, 18.5)

In response to this command, **OmniTrak** records values of the electrostatic potential, electric field components and local region number at the spatial point (x_p, y_p, z_p) . Enter the real-number coordinates in units set by *DUnit*.

ESCAN Xs Ys Zs Xe Ye Ze

ESCAN (0.0, 0.0, 0.0) (0.0, 0.0, 25.0)

In response to this command, **OmniTrak** writes a series of data lines containing the electrostatic potential, electric field components and the local region number along a line from (x_s, y_s, z_s) to (x_e, y_e, z_e) . Enter coordinates in units set by *DUnit*. The scan lines need not be parallel to the coordinate axes. The number of intervals equals the parameter *NScan*. The code records values at $(NScan+1)$ points. Zero values are recorded if the scan line passes through an electrode or extends outside the electrostatic solution boundary.

BPOINT Xp Yp Zp

BPOINT = (0.567, 12.27, 18.5)

In response to this command, **OmniTrak** records values of components of the total magnetic flux density and local region number at the spatial point (x_p, y_p, z_p) . Enter the real-number coordinates in units set by *DUnit*. The command works if either a two-dimensional or three-dimensional magnetic field solution has been loaded. Uniform field components are included. This command is particularly useful if

you want to confirm the results of *SHIFT* and *ROTB2* operations on a two-dimensional magnetic field solution. An example of a file listing is shown below.

```

--- Point Field Analysis ---
Position: [ 0.0000E+00, 0.0000E+00, 0.0000E+00 ]
Region number: 1
  Bx: 3.0933E-02 (tesla)
  By: 0.0000E+00 (tesla)
  Bz: 7.7133E-01 (tesla)
  |B|: 7.7195E-01 (tesla)
Number of data points: 12

```

BSCAN Xs Ys Zs Xe Ye Ze
BSCAN (0.0, 0.0, 0.0) (0.0, 0.0, 25.0)

In response to this command, **OmniTrak** writes a series of data lines containing components of the total magnetic flux density and the local region number along a line from (x_s, y_s, z_s) to (x_e, y_e, z_e) . Enter coordinates in units set by *DUnit*. The scan lines need not be parallel to the coordinate axes. The number of intervals equals the parameter *NScan*. The code records values at $(NScan+1)$ points. Zero values are recorded if the scan line extends outside the magnetostatic solution boundary. The command works if either a two-dimensional or three-dimensional magnetic field solution has been loaded. Uniform field components are included.. An portion of a listing is shown below.

```

----- Field scan between points -----
XStart: 0.0000E+00 YStart: 0.0000E+00 ZStart: -7.0000E+00
XEnd: 0.0000E+00 YEnd: 0.0000E+00 ZEnd: 7.0000E+00

      X           Y           Z           Bx           By           Bz           |B|           NReg
-----
0.0000E+00 0.0000E+00 -7.0000E+00 1.2559E-01 0.0000E+00 6.2331E-01 6.3584E-01 1
0.0000E+00 0.0000E+00 -6.7200E+00 1.2683E-01 0.0000E+00 6.9604E-01 7.0750E-01 1
0.0000E+00 0.0000E+00 -6.4400E+00 1.2576E-01 0.0000E+00 7.6926E-01 7.7947E-01 1
0.0000E+00 0.0000E+00 -6.1600E+00 1.2122E-01 0.0000E+00 8.4164E-01 8.5032E-01 1

```

BBPOINT Xp Yp Zp
BBPOINT = (0.567, 12.27, 18.5)

In response to this command, **OmniTrak** records values of components of beam-generated magnetic flux density and local region number at the spatial point (x_p, y_p, z_p) . Enter the real-number

coordinates in units set by *DUnit*. The command works only for calculations in the RELBEAM mode.

BBSCAN Xs Ys Zs Xe Ye Ze

BSCAN (0.0, 0.0, 0.0) (0.0, 0.0, 25.0)

In response to this command, **OmniTrak** writes a series of data lines containing components of the beam-generated magnetic flux density and the local region number along a line from (x_s, y_s, z_s) to (x_e, y_e, z_e) . Enter coordinates in units set by *DUnit*. The scan lines need not be parallel to the coordinate axes. The number of intervals equals the parameter *NScan*. The code records values at $(NScan+1)$ points. Zero values are recorded if the scan line extends outside the magnetostatic solution boundary. The command works only in the RELBEAM mode.

NSCAN NScan

NSCAN 150

This command controls the number of intervals in *ESCAN* and *BSCAN* listings. The default value is $NScan = 25$.

EDUMP FileName [Text]

EDUMP = KlyMod.HOU

In response to this command, **OmniTrak** records a file of electrostatic potential values reflecting conditions at the end of the run. The file has a name *FileName* and is created in the current directory. Be sure that the filename has the suffix HOU. The file is in standard **HiPhi** format. It can be inspected with **PhiView** or used as input to subsequent **OmniTrak** runs. The command functions only when a three-dimensional electrostatic solution has been loaded. Add the keyword *TEXT* to create the file in text format for transfer to your own analysis programs. In this case, the file is not compatible with **PhiView**.

BDUMP FileName [Text]

BDUMP = SolLens.GOU

In response to this command, **OmniTrak** records a file of magnetic field quantities reflecting the state of the solution at the end of the run. Be sure that the filename has the suffix MOU. The values include the effects of scaling and shifts. The file has the name *FileName* and is created in the current directory. This file is in standard **Magnum** format. It can be inspected with **MagView** or used as input to

subsequent **OmniTrak** runs. The command functions only when a three-dimensional magnetic field solution has been loaded. Note that **OmniTrak** does not load or save the dual potential ψ (which gives the field inside ferromagnetic materials) because charged-particle orbits inside iron regions are non-physical. Add the keyword *TEXT* to create the file in text format for transfer to your own analysis programs. In this case, the file is not compatible with **MagView**.

BBDUMP FileName

BBDUMP = RBInjectorP.GOU

In response to this command, **OmniTrak** records a file of beam-generated magnetic field quantities. Be sure that the filename has the suffix MOU. The file is in standard **Magnum** format and lists values at the nodes of the electric field mesh. The command functions only for calculations in the *RELBEAM* mode. The output file can be inspected with **MagView** or used as input to subsequent **OmniTrak** runs. Add the keyword *TEXT* to create the file in text format for transfer to your own analysis programs. In this case, the file is not compatible with **MagView**.

15.2. Particle diagnostics

There are three commands that control records of final particle orbit parameters, *PARTLIST*, *PARTFILE* and *PARTDIST*. A third command, *REFERENCE*, sets up a reference position and direction for the *PARTDIST* command.

PARTLIST

In response to this command, **OmniTrak** writes ordered tables of the initial and final particle positions and momenta to the listing file. The information can be used directly in spreadsheets, plotting programs, and mathematical analysis programs. The final particle information is calculated at the termination position of the orbits unless diagnostic planes have been defined

PARTFILE FPrefix

PARTFILE RFGOutDist

OmniTrak creates a file with the name *FPrefix.PRT* in the current directory. This file contains parameters evaluated at a diagnostic plane or at the orbit stopping point in the standard line

format discussed in Sect. 4.1. This file can be used as the input target file for the *PFile* command in a succeeding **OmniTrak** run.

PARTDIST

In response to this command, **OmniTrak** analyzes particle parameters (either at the diagnostic or stopping points) and calculates beam quantities like angular spread, spatial spread and emittance. These calculations require a reference direction and origin that can be set with the *REFERENCE* command. The calculations are identical to those performed in response to the *Beam analysis* command in the *Distribution* menu of **OmniView**. Chapter 20 gives a detailed description of the calculations and listings.

REFERENCE RAxis [HOrig VOrig]

REFERENCE Z 0.5 0.5

You can use this command to set a reference direction for beam distribution calculations initiated by *PARTDIST*. The string parameter *RAxis* sets the beam axis: the choices are *X*, *Y* or *Z*. The optional real-number quantities *HOrig* and *VOrig* give the location of the origin in the plane normal to the beam axis. The quantities are ordered following the right-hand rule. For example, if *RAxis* = *Y*, then *HOrig* = *ZOrig* and *VOrig* = *XOrig*. If origin quantities are not included, then **OmniTrak** uses the average beam position in the normal plane (*i.e.*, *ZOrig* = $\langle Z \rangle$ and *XOrig* = $\langle X \rangle$). A current-weighted average is used when model particles carry current. If no *REFERENCE* command appears, **OmniTrak** uses default settings of *RAxis* = *Z*, *XOrig* = 0.0 and *YOrig* = 0.0 for *PARTDIST* calculations.

15.3. Particle termination conditions

REGLIST

In response to this command, **OmniTrak** writes a list of orbit termination conditions organized by region. Quantities listed include the total number of model particles that strike *Material* or *Secondary* regions, the total number of particles absorbed and the incident current. For a secondary material the number of electrons absorbed equals the number incident minus the number emitted. The quantity need not have an integer value and may have a negative value for $\delta > 1$. In the *TRACK* mode the absorbed current is zero.

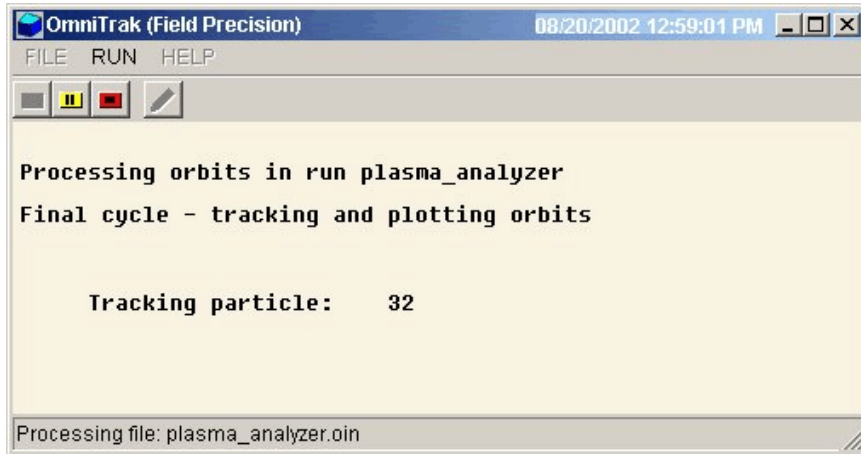


Figure 16.1. OmniTrak running in a window – interactive mode.

16. Running OmniTrak

16.1. Interactive mode

The program `OmniTrak.EXE` can run interactively in a window. In this mode you can perform several solutions in a session and temporarily leave the program to work on other tasks. You can launch the program from **AMaze** or create your own shortcuts. Figure 16.1 shows the program window.

The program has three popup menus: *File*, *Run* and *Help*. The following commands appear in the *File* menu.

Edit input file

Edit listing file

Edit file

The commands call up the internal editor to inspect or to modify text input and output files for the solution program. With the *Edit input file* command you can work on files with names of the form `FPREFIX.OIN`. With the *Edit listing file* command you can pick files with names of the form `FPREFIX.OLS`. The *Edit file* command shows all available files. Choosing a file from an alternate directory does not change the working directory of the program.

The *Run* menu has three commands.

Start run

Pick an input file with a name of the form `FPREFIX.OIN` to start a solution. The working directory changes if you pick a file from an alternate directory. The run begins if all required files are available in the working directory. **OmniTrak** writes information to the screen during extended operations such as orbit tracking and field solution updates.

Pause run

The intensive calculations of **OmniTrak** demand the full resources of your computer, causing other tasks to run slowly. If you need to perform critical work, you can pause the solution program during the relaxation process and restart it later without loss of data.

Stop run

This command terminates the program during the relaxation process and saves available information.

The *Help* menu has a single command.

Instructions

Displays the HTML help file using your default browser. In order for this command to operate correctly, the file `OmniTrak.HTML` must be in the same directory as `OmniTrak.EXE`.

16.2 Command line operation

Batch file control is a useful option for running large technical programs like **OmniTrak**. You can prepare scripts to organize complex operations. The sequenced programs run automatically in the background or overnight.

To make a single **OmniTrak** simulation in the background, go to the Command Prompt from Windows and log to the data directory that contains the required input files. For example, suppose the input files `SWITCH.HOU` and `SWITCH.OIN` are stored in `\AMAZE\BUFFER`

and that the program OMNITRAK . EXE is in the directory \AMAZE.
From \AMAZE\BUFFER type

```
..\OmniTrak SWITCH <Enter>
```

The program runs silently, writing detailed information in the listing file SWITCH . TLS and the plot file SWITCH . OOU. During lengthy runs you can perform other tasks in Windows. Note that the response of the computer will be considerably slower because **AMaze** programs seek to use the full power of the CPU.

16.3. Batch files

The main function of the command mode is autonomous operation under batch file control. As an example, assume you have prepared the input files SWITCH01 . MIN , . . . , SWITCH08 . MIN, SWITCH01 . HIN , . . . , SWITCH08 . HIN and SWITCH08 . OIN , . . . , SWITCH08 . OIN in the directory \AMAZE\BUFFER.. Next you create the following batch file SW_RUN . BAT in the data directory using a text editor.

```
ECHO OFF
ECHO Main switch data run
START ..\METAMESH.EXE SWITCH01
START ..\HIPHI.EXE SWITCH01
START ..\OMNITRAK.EXE SWITCH01
START ..\METAMESH.EXE SWITCH02
START ..\HIPHI.EXE SWITCH02
START ..\OMNITRAK.EXE SWITCH02
. . .
START ..\METAMESH.EXE SWITCH08
START ..\HIPHI.EXE SWITCH08
START ..\OMNITRAK.EXE SWITCH08
```

Type

```
SW_RUN <Enter>
```

to generate all solutions without the need for further keyboard input.

16.4. Introduction to GCon

Microsoft has released over thirty versions of its 32-bit operating system since Windows 95. There is considerable inconsistency in DOS emulation between versions. For example, Windows 2000 runs a set of programs in the proper sequential order for any valid batch format. Windows 95 starts all programs simultaneously unless you use the command form:

```
START /WAIT ProgName
```

Windows ME does not recognize the `START` command at all. And so on.... To ensure consistent batch file operation we supply the utility program **GCon** with all our software. To avoid problems we advise running batch scripts from **GCon** rather than from the Command Prompt. An additional benefit is that **GCon** control scripts can be used directly in Linux.

GCon handles the following subset of DOS commands. Additional capabilities are described in the file `GCON.HTML`.

CD

Function: Displays the name of or changes the current directory.

Example: `CD C:\AMaze\Buffer`

COPY ... TO ...

Function: Copies a file to another location.

Example: `COPY Diode.min TO C:\AMaze\Diode.min`

ERASE

Function: Deletes a file.

Example: `ERASE diode.mls`

REM

Function: Displays the line, no action taken

MOVE ... TO ...

Function: Moves a file from one directory to another directory.

Example: `MOVE Diode.mou TO C:\AMaze\Diode.mou`

RENAME

Function: Renames a file or files.

Example: RENAME diode.mou diode_compare.mou

RUN

Function: Runs a specified program or command.

Example: RUN C:\tricomp\mesh diode

As an example, the following script creates a mesh and an electrostatic solution using the input files DIODE.MIN and DIODE.HIN in the directory C:\EStudies\Inputs and stores the results in the directory C:\EStudies\Outputs

```
REM Sample GCon script
CD C:\EStudies\Inputs
RUN C:\AMAZE\METAMESH Diode
RUN C:\AMAZE\OmniTrak Diode
MOVE Diode.mou TO C:\EStudies\Outputs\Diode.mou
```

16.5. Running GCon

GCon is simple to run in the interactive mode. Start the program and click on the command *Run GCon script*. The dialog shows all files with the suffix GCN. The working directory is changed if you change directories in the dialog. Pick a batch file and click *OK*. The program executes the commands in the file in sequence and displays the current status. The *Stop on error* command toggles a switch that determines whether the program will stop the batch process if there is an error in one of the batch commands. The command *Instructions* in the *Help* menu displays a help file in your default browser. The function requires that the file FPBATCH.HTML is in the same directory as FPBATCH.EXE.

You can also run **GCon** from the command prompt with a line of the following form:

```
ProgPath/GCON ScriptPath/SPREFIX <ENTER>
```

where *ProgPath* is the path to GCON.EXE if it is not in the current directory, *Script path* is the path to the **GCon** script SPREFIX.GCN.

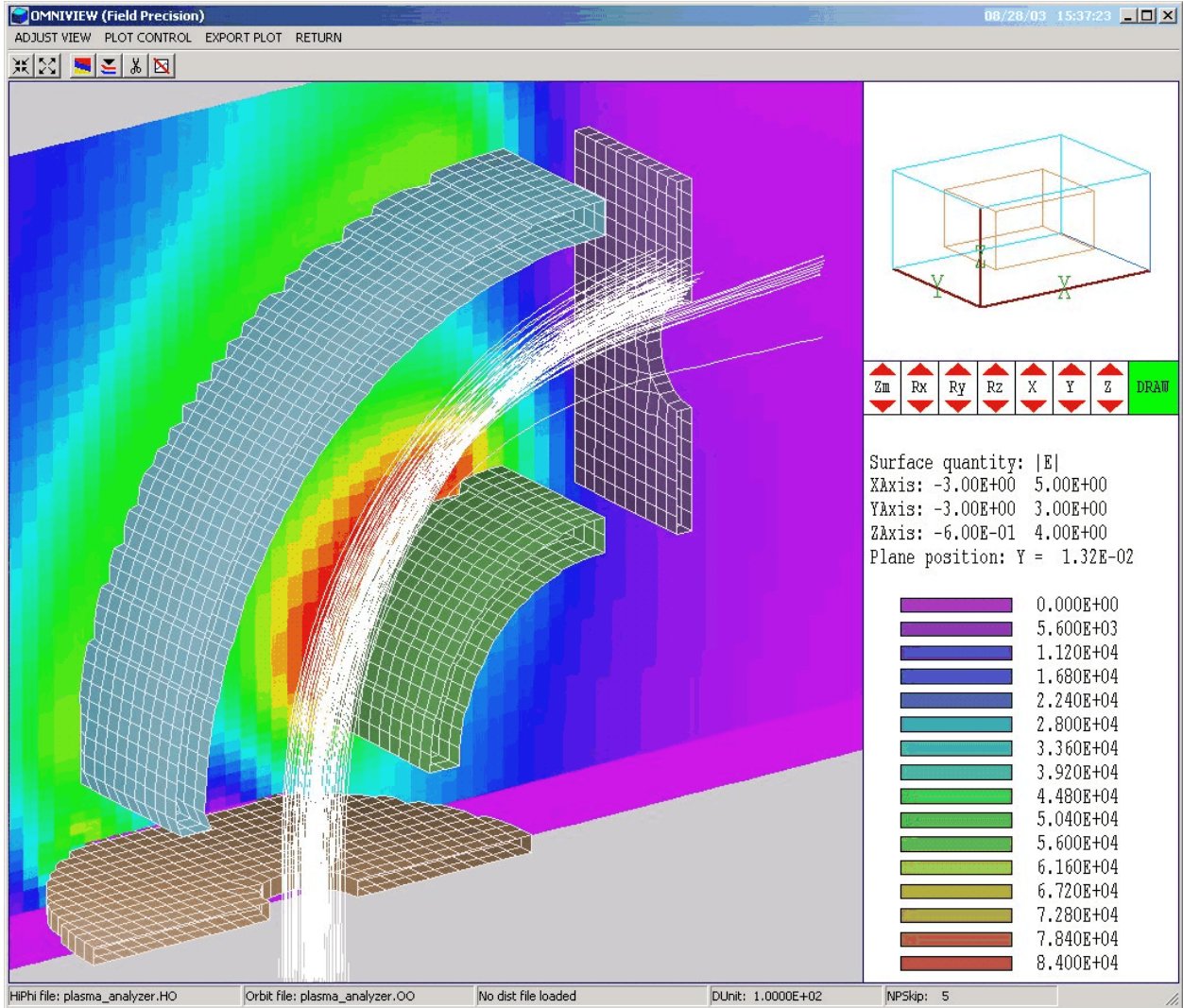


Figure 17.1. Working environment of the **OmniView** postprocessor – 3D orbit plot with region surfaces in the electrostatic mesh.

17. OmniView – file operations

The **OmniView** post-processor has two functions: 1) generation of 2D and 3D plots showing particle orbits and field quantities, and 2) analysis and plotting of particle distributions using information in PRT files. The program has the following popup menus: *File operations*, *2D plots*, *3D plots*, *Distributions* and *Help*. When the program starts, only the *File* and *Help* menus are active. You must load one or more field data files in order to create plots or to perform analyses. This section reviews options in the *File operations* menu.

Load HiPhi solution file

With this command you can load mesh and field information from any **HiPhi** electrostatic field solution. For meaningful plots, you must ensure that the field solution matches the superimposed orbits. The dialog shows a list of solution files with names of the form `FPREFIX.HOU`. Changing the directory in the dialog changes the program working directory. Pick an available file and click *OK*. The program loads the solution, updates the status bar and shows reference information on the file. Click *OK* to continue. If data retrieval is successful, the *2D plot* and *3D plot* menus become active. A field file must be loaded to create plots. The program can accommodate information from an electrostatic or a magnetostatic solution, but not both simultaneously.

Load Magnum solution file

This command loads mesh and field information from any **Magnum** magnetostatic field solution. The dialog shows a list of solution files with names of the form `FPREFIX.HOU`. Changing the directory in the dialog changes the program working directory. Pick an available file and click *OK*. The program loads the solution, updates the status bar and shows reference information on the file. Click *OK* to continue. If data retrieval is successful, the *2D plot* and *3D plot* menus become active.

Load OmniTrak plot file

This command loads orbit trace information created during an **OmniTrak** run. When this information is available, you can superimpose orbits on field plots. A field plot must be loaded to activate the the *2D plot* and *3D plot* menus. For meaningful plots, you must ensure that orbits match the loaded field solution. The dialog shows a list of **OmniTrak** plot files with names of the form `FPREFIX.OOU`. Changing the directory in the dialog changes the program working directory. Pick an available file and click *OK*. The program loads the solution, updates the status bar and displays file information. Click *OK* to continue.

Load particle file

This command loads a file of particle parameters (`FPrefix.PRT`) for analysis in the *Distribution* menu. The program can handle any file in the standard format described in Section 7.1. Possibilities include

input files created by **TDist** and output files created by either **OmniTrak** or the two-dimensional **Trak** program.

Edit file

Use this command to view or to modify any text file with the internal program editor.

The *Help* menu shows program information and contains the following command:

Instructions

Displays the HTML help file using your default browser. In order for this command to operate correctly, the file `OMNITRAK.HTML` must be in the same directory as `OMNIVIEW.EXE`.

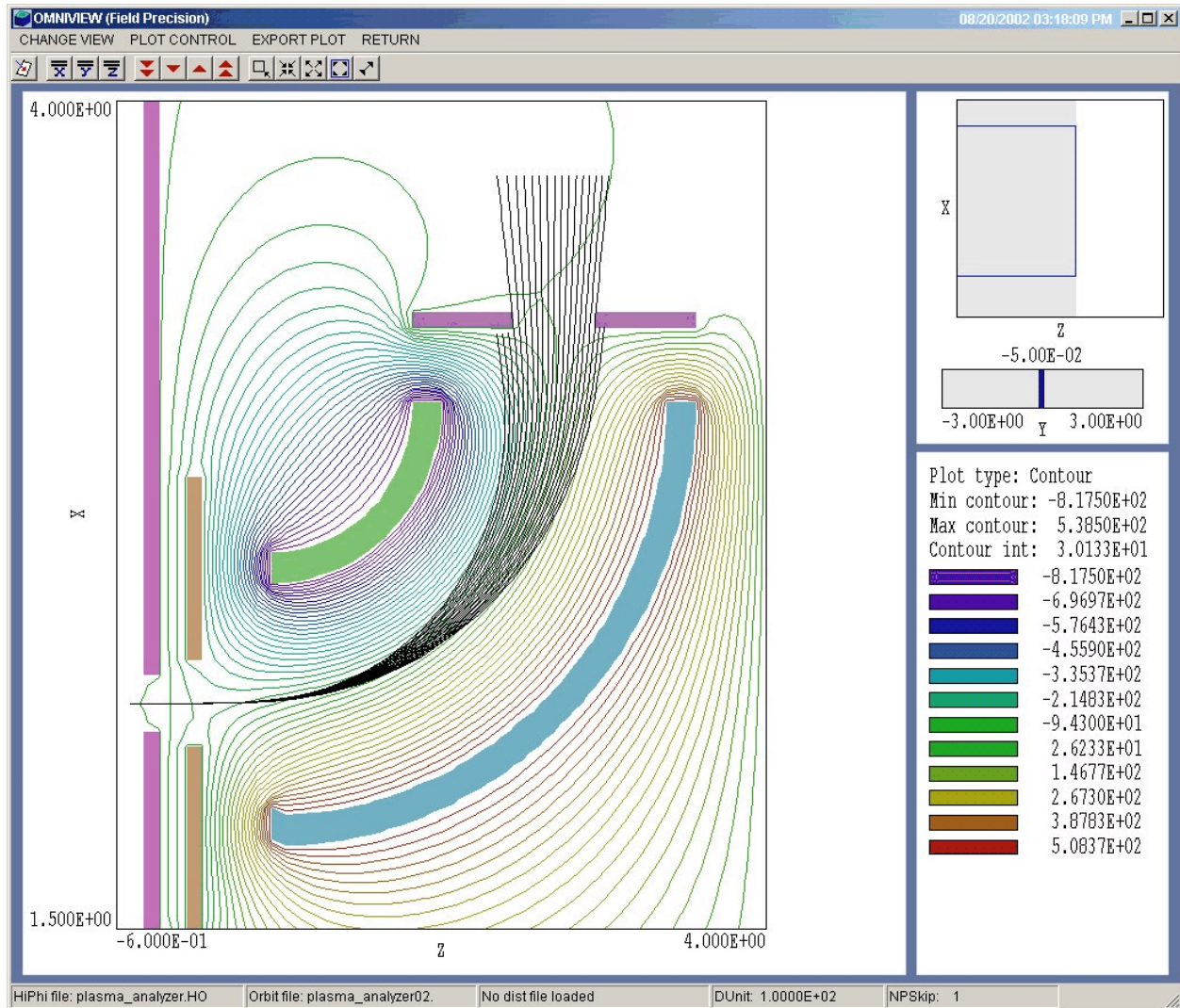


Figure 18.1. Working environment of the **OmniView** postprocessor – 2D projected orbit plot with electrostatic equipotentials

18. OmniView – 2D plots

You can enter the *2D plot* menu if an electrostatic or magnetostatic field file has been loaded. Field plots use many of the same controls as slice plots in **PhiView** or **MagView**. Orbits can be superimposed on the field plot if an orbit file has been loaded.

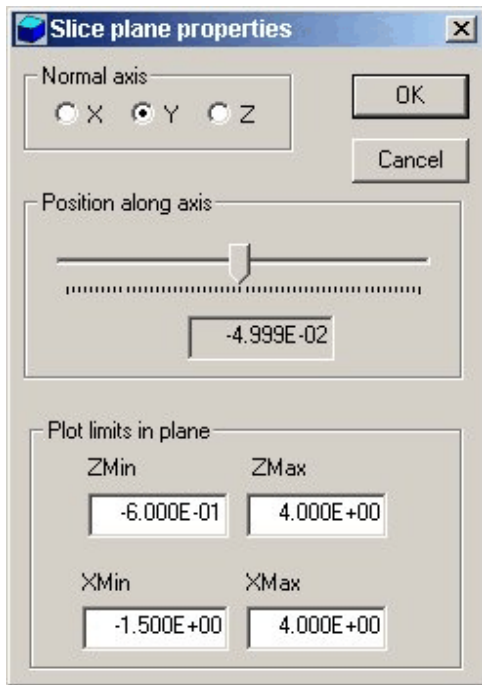


Figure 18.2. Dialog to set properties of the two-dimensional slice plane.

18. 1. Setting the view

Two-dimensional plots show the variation of field quantities over a plane normal to one of the Cartesian axes. The plot incorporates the structure of the mesh projected to a plane. This structure may be quite complex for a conformal mesh; therefore, two-dimensional plots require some computational effort. To facilitate the process, slices views are constructed at discrete locations along the normal axis corresponding roughly to the planes of the **MetaMesh** foundation mesh. It is important to note that orbit traces in plots are projections to the slice plane, independent of their location along the normal axis. Portions of the orbit in back of the slice plane are plotted in light gray and the portions in front of the slice plane (i.e., between the plane and viewer) are plotted in black. You can use the *Set particle filter* command to restrict orbit vectors to the vicinity of the plot plane.

The *Change view* popup menu contains commands to set the slice plane and to adjust the dimensions of the plot.

Set slice plane properties

This command calls up the dialog shown in Fig. 18.2.. You can change the normal axis, change the position along the normal axis, and set plot limits in the normal plane.

Slice normal to X**Slice normal to Y****Slice normal to Z**

Quick commands to change the normal axis.

Jump forward**Step forward****Step backward****Jump backward**

Move along the slice axis by small or large steps. The small step is approximately one layer of the foundation mesh and the large step is 5 layers. The term *forward* implies motion toward higher indices of the normal axis. The slider bar in the orientation area to the right of the plot (Fig. 18.1) shows the current location.

Zoom window

As an alternative to the entries in the *Set slice plane* dialog, you can interactively change plot limits in the normal plane using the mouse. Choose the command and move the mouse pointer into the plot area. The status bar enters coordinate mode. It shows the current mouse position in the plot and the snap mode. Use the left button to pick one corner and then move the mouse to create a view box. Click the left button again, and the plot regenerates. On any coordinate operation, press the *FI* key if want to enter values from the keyboard. Note that the normal plane box in the orientation area to the right of the plot (Fig. 18.1) shows the dimensions of the slice plane and the outline of the current zoomed view.

Zoom in

Enlarge the plot about the current view center.

Expand view

Expand the plot about the current view center.

Global view

Enlarge the plot boundaries to show the entire normal plane.

18.2. Setting plot properties

Pan

When the plot is zoomed, you can use this operation to shift the current view center. Use the mouse to define relative starting and ending points for the shift.

The commands in the *Plot control* popup menu are used to set plot style options.

Set field plot style

This command brings up a dialog to change properties of the field plot. Four plot types are available. The *Region plot* style is a cross-section view of the mesh elements color-coded by region. In contrast to the logical plane plot of **MetaMesh**, **OmniView** attempts to resolve the exact mesh structure in the plane. The *Contour plot* type (available only for electrostatic solutions) shows equipotential lines. The *Element* plot type shows element cross-sections color-coded by $|\mathbf{E}|$ or $|\mathbf{B}|$. There are two choices in the *Field plot options* group of the dialog. When Region shading is checked the program adds shading to contour plots to show difference dielectric regions. When Element outline is checked **OmniView** shows element boundaries in element plots. In contour plots the boundaries are plotted only inside fixed-potential regions. The radio buttons in Orbit plot style control trajectory plotting. The choices are projection of the full trajectory to the slice plane or a plot of the discrete intersection points. The latter is useful if you want to view the beam profile at different points along the main beam axis.

Plot limits

In the autoscale mode, **OmniView** chooses defaults for the minimum and maximum values of ϕ , $|\mathbf{E}|$ or $|\mathbf{B}|$. Deactivate *Autoscale* to set the values manually.

Number of contour lines

Change the number of lines for contour plots.

Set particle filter

This command brings up a dialog (Fig. 18.3) where you can restrict the spatial limits for orbit plotting. You can use the feature to improve

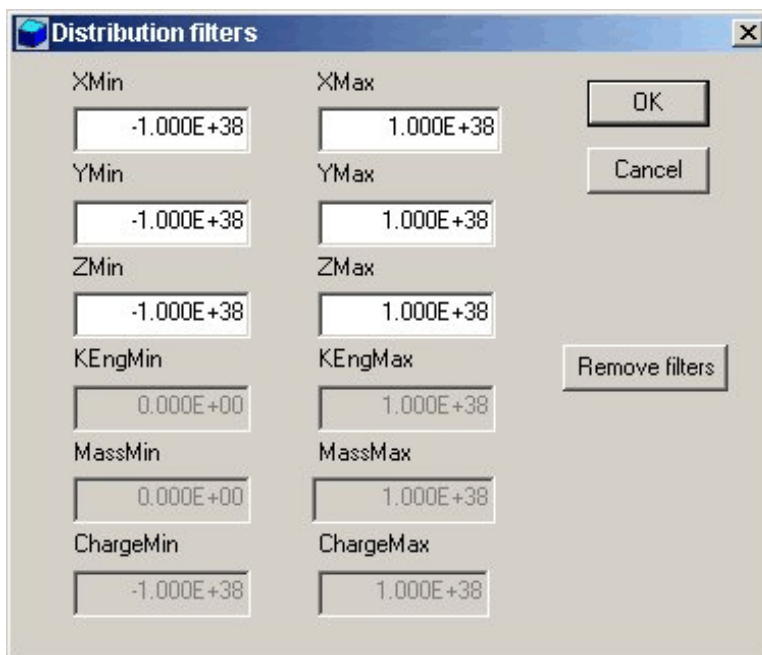


Figure 18.3. Particle filter dialog.

the appearance of two-dimensional plots. One application is to restrict plotted orbits to a region along the normal axis near the plot plane. Note that only the spatial coordinate entries are active in the dialog. The other filter quantities can be used in the *Distribution* menu.

Set NPSkip

The plot refresh may be slow when a run includes a large number of particles. Furthermore, the plot may be confusing. You can set the value of the integer parameter *NPSkip* to control the fraction of plotted orbits. All orbits are included when *NPSkip* = 1; one out of ten orbits is included when *NPSkip* = 10. The default value is *NPSkip* = 5. Note that the current value of *NPSkip* is displayed in the status bar.

Set particle range

By default **OmniView** plots the full range of particle orbits (subject to the value of *NSkip*). Use this command to plot a specific range. Supply integer values for *NPMin* and *NPMax*. You can use information in the listing file (OLS) to identify the index numbers of different classes of particles.

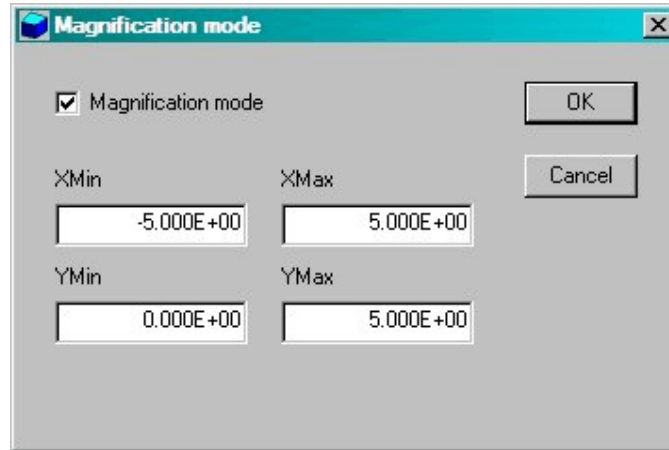


Figure 18.4. Magnification mode dialog.

Toggle snap mode

Mouse coordinates for commands such as *Zoom window*, *Pan*, and *Scan in slice* can be entered in two modes. In the normal mode, the returned position corresponds to the mouse position on the screen. In the snap mode, the program picks a point at an even interval close to the mouse position. The returned point depends on the value of the parameter *DSnap*. For example if $DSnap = 0.1$ and the mouse is at position (6.2345,-5.6113), the returned position is (6.2000,-5.6000). The status bar displays the actual or snapped position of the mouse.

Set snap distance

Change the value of *DSnap* from the default value determined by the program. The current value is displayed in the status bar.

Toggle grid

A set of dashed grid lines can be superimposed on slice plots. **PhiView** automatically chooses intervals and positions so that the lines occur at convenient values along the horizontal and vertical directions (for example, 0.01 rather than 0.01153). Listings of the grid intervals are included in the axis labels. Grids corresponding to the normal plane axes are plotted as solid lines.

Magnification mode

The default mode in **OmniView** two-dimensional plots is to preserve true scaling. In many charged-particle calculations, the beam width is

much smaller than the simulation length. In this case, you can change the scaling to magnify the plot in the horizontal or vertical direction. Figure 18.4 shows the control dialog. To activate the magnification mode, check the box and fill in values for the horizontal and vertical limits. The resulting plot fills the full view window. Uncheck the box to return to true scaling.

18.3. Exporting plots

The commands of the *Export plot* menu are used to generate hardcopy or to create plot files.

Default printer

With this command, an **OmniView** plot can be ported to any installed Windows printer (including network printers, postscript drivers, PDF drivers...). You can generate colored plots if you have a color printer. Note that the current screen plot is sent to the default Windows printer. If necessary, change the default using the *Settings* command of Windows before issuing the command.

Plot file (EPS)

Use this command to make a plot file of the current screen plot in Encapsulated PostScript format. Supply a file prefix in the dialog box. The plot file will be created in the current directory with the name `FPREFIX.EPS`.

Plot file (BMP)

Use this command to make a plot file of the current screen plot in Windows Bitmap format. Supply a file prefix in the dialog box. The plot file will be created in the current directory with the name `FPREFIX.BMP`.

Plot file (PNG)

Use this command to make a plot file of the current screen plot in Portable Network Graphics format. Supply a file prefix in the dialog box. The plot file will be created in the current directory with the name `FPREFIX.PNG`.

Copy to clipboard

The current plot is copied to the clipboard in Windows Metafile format. You can then paste the image into graphics software.

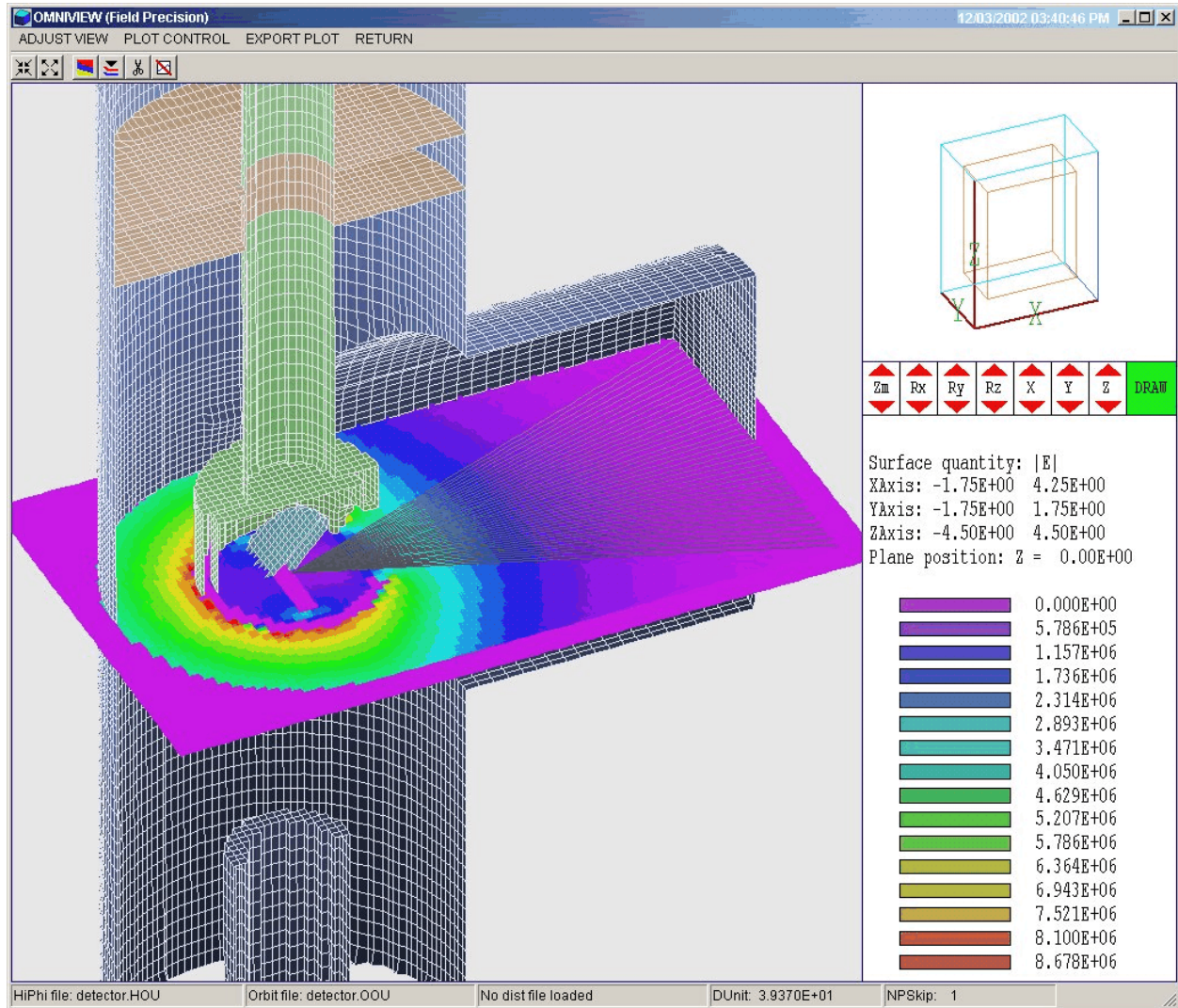


Figure 19.1. Three-dimensional plots in OmniView

19. OmniView – 3D plots

OmniView can create three-dimensional views of the solution space. Plots may include three types of information: 1) boundaries of regions in the currently-loaded electric or magnetic field mesh, 2) computed field quantities ($|\mathbf{E}|$ or $|\mathbf{B}|$) in a slice plane normal to one of the Cartesian axes and 3) particle orbits. Three-dimensional plots are created from the conformal mesh and preserve true spatial scaling.

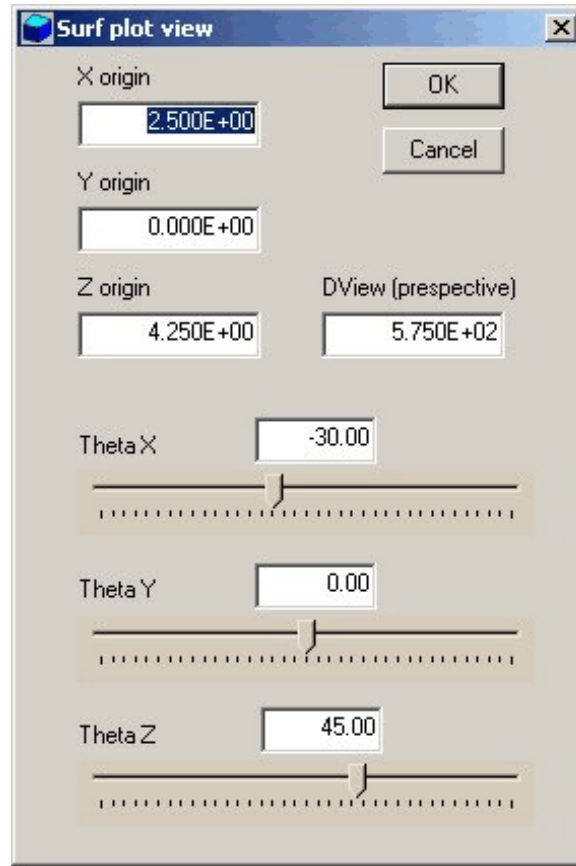


Figure 19.2. Set surface view dialog

The three-dimensional display of the current plot style is controlled through the orientation window and control region shown in Figure 19.1. The orientation window (upper right-hand corner) shows the Cartesian axes (red lines) and boundaries of the current solution volume (blue lines). The orange box shows the approximate size and location of the current view. The control region is directly below the orientation window. When you move the mouse pointer into the control region the cursor changes to a directional symbol. Left-click on the red up/down arrows to shift the view. Hold down the left mouse button for a continuous change. The seven boxes have the following functions: 1) zoom in or expand the view, 2) rotation about the x axis, 3) rotation about the y axis, 4) rotation about the z axis, 5) translation along the x axis, 6) translation along the y axis and 7) translation along the z axis. The changes are reflected in the orientation window. Left-click on the green *DRAW* box or right-click to regenerate the plot. To set specific view parameters, you can use the Set surface view command in the Adjust view popup menu. The command calls up the dialog of Figure 19.2. The quantity *DView* controls the distance from the

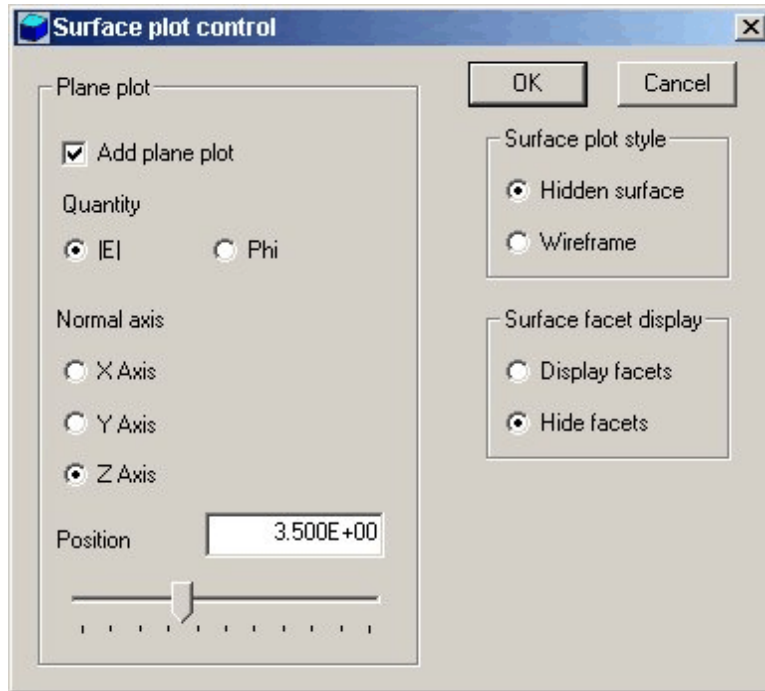


Figure 19.3. Three-dimensional plot style dialog

average position of displayed elements to the viewpoint. Decrease *DView* to increase the degree of perspective. Note that the plot may be invalid if you choose a small value of *DView* that places the viewpoint within the plotted objects.

The commands of the *Plot control* popup menu control the appearance of the plot.

Plot style

This command brings up the dialog of Fig. 19.3 that controls the appearance of the plot. Three types of information can be included: 1) region boundaries showing the surfaces of objects in the solution space, 2) field quantities ($|\mathbf{E}|$, ϕ or $|\mathbf{B}|$) in a plane normal to one of the axes, and 3) particle orbits. The display of regions is controlled by the *Displayed regions* command described below. The commands of the *plane plot* group control the display of field quantities. Use the *Normal axis* and *Position* controls to set the orientation and position for the displayed plane. The commands of the *Surface plot style* and *Surface plot display* groups control the appearance of the three-dimensional region boundaries.

Displayed regions

The command brings up a dialog where you can pick the region boundaries that will be included in the plot. Region numbers correspond to those defined in **MetaMesh**.

Set cut planes

In a hidden surface plot, internal details may be obscured by surrounding parts or regions. This command brings up a dialog that allows you to adjust the displayed portion of region surfaces along the x , y and z axes. **OmniView** does not display facets that lie outside the limits. With this feature you can create cutaway views. By default the cut limits equal the dimensions of the solution volume (all facets included).

Set particle filter

This command brings up a dialog (Fig. 17.3) that enables you to restrict the spatial limits for orbit plotting. You can use the feature to improve the appearance of three-dimensional plots.

Set NPSkip

The plot refresh may be low when a run includes a large number of particles. Furthermore, the plot may be confusing. You can set the value of the integer parameter *NPSkip* to control the fraction of plotted orbits. All orbits are included when *NPSkip* = 1; one out of ten orbits is included when *NPSkip* = 10. The default value is *NPSkip* = 5. Note that the current value of *NPSkip* is displayed in the status bar.

Set particle range

By default **OmniView** plots the full range of particle orbits (subject to the value of *NSkip*). Use this command to plot a specific range. Supply integer values for *NPMin* and *NPMax*. You can use information in the listing file (OLS) to identify the index numbers of different classes of particles.

Finally, the commands of the *Export plot* menu are identical to those for two-dimensional plots (Chapter 18) and distribution plots (Chapter 20).

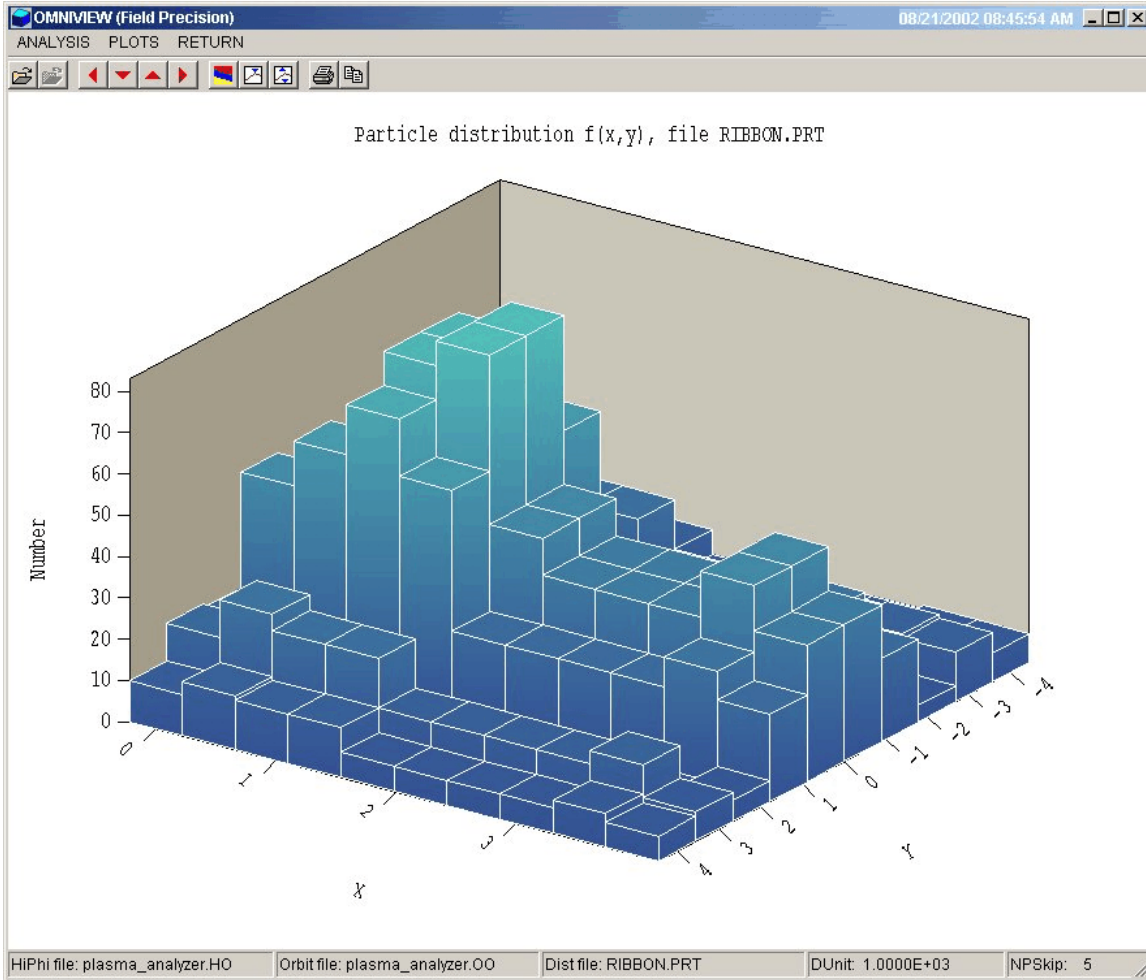


Figure 20.1. OmniView working environment – Distribution menu

20. OmniView – Distribution plots and analyses

The *Distribution* menu becomes active when you load a particle file (FPrefix.PRT) using the *Load particle file* command in the *File* menu. Two popup menus are displayed when you enter the *Distribution* menu: *Analysis* and *Plots*.

20. 1. Analysis commands

Analysis commands control calculation of numerical data from values in the particle file. It is important to note that particle positions in PRT files may be entered in any convenient units (*i.e.*, centimeters, inches, microns, ...). The values are converted to meters when the file is loaded into **OmniTrak** according to the present value of the *DUnit* parameter. In turn, **OmniTrak** uses the same conversion factor to write positions to output PRT files. The unit conversion factor must be known for RMS emittance calculations and phase-space plots in **OmniView**. Therefore, **OmniTrak**, **TDist 2.0** and **Trak 6.0** insert a comment line of the form

```
* DUnit: 1.00000E+02
```

when creating a PRT file. **OmniView** checks for the comment when a file is loaded and uses the value of *DUnit* if the information is available. You can add a *DUnit* comment line to PRT files that you construct with a spreadsheet. Alternatively, you can set the value using the *Set DUnit* command in the *Analysis* menu. Note that the current value of *DUnit* is displayed in the status bar.

Beam analysis

In response to this command, **OmniTrak** performs a statistical analysis of the particles and records several beam characterization quantities. The program uses current values of the reference axis and unit conversion factor. You can modify settings with the *Set reference axis* and *Set DUnit* commands. The program displays a summary of information on the screen and will record the complete analysis in a data file (see the *Open data file* command). Section 20.3 summarizes equations used in the calculations.

| Table 20.1. Reference axis orientation | | |
|--|----------------------------|--------------------------|
| Reference axis | Normal plane Horizontal | Normal plane Vertical |
| X | Y | Z |
| Y | Z | X |
| Z | X | Y |

Set particle filter

This command calls up the dialog of Fig. 18.3. The difference is that all options are active. In addition to position along the three axes, you can filter particles by kinetic energy, mass and charge. The last two options are useful only if the simulation contains multiple particle species.

Set reference axis

Beam quantities are calculated relative to a beam axis. The choices are the X , Y or Z axes of the solution coordinate system. The quantities *HOrig* and *VOrig* give the position of the beam axis in the horizontal and vertical directions of the normal plane. Table 20.1 lists the association of the horizontal and vertical directions with coordinate directions. The convention follows the right-hand and is applied in several of the distribution plots.

Set DUnit

Use this command to set a value for the conversion factor that will be used in the calculation of RMS emittance. The parameter *DUnit* gives the conversion between length units used in the PRT file to meters. For example, if positions in the PRT file are given in inches, set $DUnit = 39.37$.

Write PRT file

This command writes a PRT file that contains data lines only for particles that satisfy the current filter condition. As an application example, consider a model of an electron extractor where some particles strike the anode and others continue through an aperture. To analyze only the extracted electrons, you could set a filter on z_{\min} . If you then created a PRT file with this command, it would contain only propagating electrons. The modified file could then be used as input for a simulation of the transport region.

Open data file

OmniTrak will record the results of beam calculations in text format if you open a data file. Supply a complete file name in the dialog. The program will write results of multiple analyses to the file until it is closed.

Close data file

Close the current data file.

20. 2. Plot commands

OmniTrak can create presentation-quality plots based on the parameters in particle files. You can use this feature to display characteristics of beam distributions generated by **OmniTrak** and to check the validity of input distributions created by **TDist**. Note that only particles that pass the current filter are included in the plots.

The program supports the following plot types:

Spatial projections: particle positions projected to a normal plane. The options are x - y , y - z and z - x . For example, an x - y plot shows the position of a particle in the plane normal to z regardless of the particle position in z . You could set a filter to restrict plotted orbits to a range in z . The program aborts the plot and displays an error message if the range of values along either the horizontal or vertical axis equals zero.

Phase-space plots: phase-space distributions relative to the reference axis. The options are x - x' , y - y' and z - z' . To illustrate the meaning of quantities, suppose the reference axis is Z and we choose a plot of x - x' . The abscissa of the plot is $(x - x_{\text{orig}})$ expressed in meters. The ordinate is the quantity $x' = \tan^{-1}(p_x/p_z)$ expressed in radians. This plot type is useful only if particles have orbits that are paraxial with respect to the z axis (*i.e.*, the distribution constitutes a beam moving in $\pm z$ where orbits have small angles with respect to the z axis). A plot is not generated if you pick the option z - z' when the reference axis is set to Z .

1D bins: histograms of number of particles or particle current organized by a quantity. The following plot quantities are available for all PRT files: $f(x)$, $f(y)$, $f(z)$ and $f(T)$. For example, the option $f(T)$ shows the distribution of computational particles in terms of kinetic energy. The option $f(x)$ divides particles into bins according to their position in x , regardless their position in y and z . You can limit the range of particles in either x , y or z using a filter. The resulting histogram shows the number of model particles in each bin along x . The following options are available only when the PRT file contains information on the model particle current: $Curr(x)$, $Curr(y)$, $Curr(z)$ and $Curr(T)$. In this case, the histogram shows the relative distribution

of current in position or kinetic energy. The quality of the plot depends on the number of simulation particles and the number of bins. The height of a bin varies statistically approximately a fraction $1/N_b^{1/2}$, where N_b is the number of particles in the bin. You can change the number of bins using the *Set number of bins* command. You can also manually set limits on the abscissa using *the Set limits* command.

2D bins: histograms of the number of model particles or the current sorted according to position in a normal plane. The following options are available for all PRT files: $f(x,y)$, $f(y,z)$ and $f(z,x)$. For example, in the $f(x,y)$ option particles are assigned to a two-dimensional array of bins according to their positions in x and y , regardless of the position in z . You can use particle filters to restrict plotted particles to a region of space. The following options are available if the PRT file contains information on the particle current: $Curr(x,y)$, $Curr(y,z)$ and $Curr(z,x)$. You can change the number of bins using the *Set number of bins* command. You can also manually set plot limits using *the Set limits* command.

3D scatter: three-dimensional plot of particle positions, (x, y, z) . Only particles that pass the present filter are included. When this plot is active, you can shift the viewpoint using the *Rotation* and *Elevation* commands.

In addition to the standard hardcopy options, the following commands appear in the *Plots* popup menu of the *Distributions* menu.

Set plot type

Pick the plot type from the list described above. **OmniView** picks a default plot option valid for the current plot type.

Plot quantity

Pick a plotted quantity from one of the displayed options. **OmniView** displays only options that are valid for the current plot type.

Plot limits

Manually set limits on the plot. The number of quantities to supply depends on the plot type.

Set number of bins

Set the number of bins for 1D and 2D histograms.

Rotation (Up)

Rotation (Down)

These commands function only when the current plot type is *Scatter 3D*. Rotate the plot in azimuth.

Elevation (Up)

Elevation (Down)

These commands function only when the current plot type is *Scatter 3D*. Shift the view (polar angle)

20. 3. Beam analysis equations

The quantities used in beam analyses depend on the choice of reference axis. Table 20.1 lists the association between Cartesian coordinates and the horizontal (h) and vertical (v) directions. In the following equations, the subscript n refers to a model particle and the quantity n_p is the total number of particles. Particle n has position (h_n, v_n) relative to the reference axis and angles h_n' and v_n' given by

$$h_n' = \tan^{-1} (p_{hn} / p_{rn}) , \quad v_n' = \tan^{-1} (p_{vn} / p_{rn}) .$$

In the equation, p_{hn} and p_{vn} are the dimensionless momenta in the horizontal and vertical directions while p_{rn} is the momentum along the reference axis. The first group of equations applies when the PRT file does not contain current information:

Average displacement from the reference axis

$$\bar{h} = \frac{\sum_{n=1}^{n_p} (h_n - h_{orig})}{n_p}, \quad \bar{v} = \frac{\sum_{n=1}^{n_p} (v_n - v_{orig})}{n_p}.$$

Average angle with respect to the reference axis

$$\bar{h}' = \frac{\sum_{n=1}^{n_p} h'_n}{n_p}, \quad \bar{v}' = \frac{\sum_{n=1}^{n_p} v'_n}{n_p}.$$

Average kinetic energy

$$\bar{T} = \frac{\sum_{n=1}^{n_p} T_n}{n_p}.$$

RMS displacements from the reference axis

$$h_{rms} = \sqrt{\frac{\sum_{n=1}^{n_p} (h_n - h_{orig})^2}{n_p}}, \quad v_{rms} = \sqrt{\frac{\sum_{n=1}^{n_p} (v_n - v_{orig})^2}{n_p}}.$$

RMS angles with respect to the reference axis

$$h'_{rms} = \sqrt{\frac{\sum_{n=1}^{n_p} (h'_n)^2}{n_p}}, \quad v'_{rms} = \sqrt{\frac{\sum_{n=1}^{n_p} (v'_n)^2}{n_p}}.$$

The RMS emittance in the horizontal direction is

$$\epsilon_h = 4 \sqrt{\overline{h^2} \overline{h'^2} - (\overline{h h'})^2},$$

where

$$\overline{h^2} = \frac{\sum_{n=1}^{n_p} (h_n - h_{orig})^2}{n_p},$$

$$\overline{h'^2} = \frac{\sum_{n=1}^{n_p} (h'_n)^2}{n_p},$$

$$\overline{h h'} = \frac{\sum_{n=1}^{n_p} (h_n h'_n)}{n_p}.$$

Similar equations hold for the vertical direction. The spread in kinetic energy is given by

$$\delta T = \sqrt{\frac{\sum_{n=1}^{n_p} (T_n - \bar{T})^2}{n_p}}$$

OmniView takes current-weighted averages if the particle current is included in the PRT file. The total current equals:

$$I_o = \sum_{n=1}^{n_p} I_n.$$

Average displacement from the reference axis

$$\bar{h} = \frac{\sum_{n=1}^{n_p} (h_n - h_{orig}) I_n}{I_o}, \quad \bar{v} = \frac{\sum_{n=1}^{n_p} (v_n - v_{orig}) I_n}{I_o}.$$

Average angle with respect to the reference axis

$$\bar{h}' = \frac{\sum_{n=1}^{n_p} h_n' I_n}{I_o}, \quad \bar{v}' = \frac{\sum_{n=1}^{n_p} v_n' I_n}{I_o}.$$

Average kinetic energy

$$\bar{T} = \frac{\sum_{n=1}^{n_p} T_n I_n}{I_o}.$$

RMS displacements from the reference axis

$$h_{rms} = \sqrt{\frac{\sum_{n=1}^{n_p} (h_n - h_{orig})^2 I_n}{I_o}}, \quad v_{rms} = \sqrt{\frac{\sum_{n=1}^{n_p} (v_n - v_{orig})^2 I_n}{I_o}}.$$

RMS angles with respect to the reference axis

$$h'_{rms} = \sqrt{\frac{\sum_{n=1}^{n_p} (h_n')^2 I_n}{I_o}}, \quad v'_{rms} = \sqrt{\frac{\sum_{n=1}^{n_p} (v_n')^2 I_n}{I_o}}.$$

The averages for computing the RMS emittance in the horizontal direction are

$$\overline{h^2} = \frac{\sum_{n=1}^{n_p} (h_n - h_{orig})^2 I_n}{I_o} ,$$

$$\overline{h'^2} = \frac{\sum_{n=1}^{n_p} (h'_n)^2 I_n}{I_o} ,$$

$$\overline{h h'} = \frac{\sum_{n=1}^{n_p} (h_n h'_n) I_n}{I_o} .$$

Spread in kinetic energy

$$\delta T = \sqrt{\frac{\sum_{n=1}^{n_p} (T_n - \bar{T})^2 I_n}{I_o}}$$

Chapter 21. OmniTrak output files

OmniTrak can create a variety of output files. The listing file (RUNNAME.OLS) contains a wealth of information in text format. We shall not review the structure of this file because: 1) the contents depend on the type of run and the commands appearing in the *DIAGNOSTICS* section and 2) the labels and format of entries make them largely self-explanatory. **OmniTrak** can generate two types of binary files:

- Electric field files are created in response to the *EDUMP* command. They have a name of the form FPREFIX.HOU and are in the standard **HiPhi** format. You can load the files into **OmniView** or **PhiView** or use them as input for subsequent **OmniTrak** runs.
- Files of beam-generated magnetic field may be created in *RELBEAM* simulations in response to the *BBDUMP* command. They have a name of the form FPREFIX.GOU and are in the standard **Magnum** format. You can load the files into **OmniView** or **MagView** or use them as input for subsequent **OmniTrak** runs.

Information of the binary file format can be useful if you want to write programs that interface directly with **OmniTrak** data.

The **HiPhi** manual describes the structure of electric-field files in detail. Table 21.1 shows the output code that generates the file. In the header, the quantities I_{\max} , J_{\max} and K_{\max} (4-byte integers) are the maximum mesh index values along x , y and z . The quantity *DUnit* (4-byte real) is the conversion factor between mesh dimension units and meters. The quantity *NCQuant* (1-byte integer) equals the number of stored quantities for the solution program. In **HiPhi**, *NCQuant* = 1 and the recorded node quantity is the electrostatic potential ϕ at the node. The name of the recorded quantity is stored as a string of length 12. The next section of the file consists of information for each node. The quantities *RegNo* (the region number of the node) and *RegUp* (the region number of the adjacent element in the direction of increasing I , J and K) are 1-byte integers. The coordinates (X , Y and Z) and the stored quantity (ϕ) are 4-byte real numbers. The final step is to record region information. The quantity *NRQuant* (the number of physical quantities stored per region) equals 3 in

Table 21.1. Code to create the electric field output file

```
! --- Mesh size parameters
WRITE (OutField) IMax,JMax,KMax,DUnit,NCQuant
DO N=1,NCQuant
  WRITE (OutField) CQuantName(N)
END DO
! --- Node and element properties
DO K=0,KMax
  DO J=0,JMax
    DO I=0,IMax
      MC = M(I,J,K)
      WRITE (OutField) &
        C(MC).RegNo,C(MC).RegUp,C(MC).x,C(MC).y,C(MC).z, &
        REAL(C(MC).Phi)
    END DO
  END DO
END DO
! --- Region properties
WRITE (OutField) NRQuant
DO N=1,NRQuant
  WRITE (OutField) RQuantName(N)
END DO
WRITE (OutField) NRegMax
DO N=1,NRegMax
  IF (Reg(N).Fixed) THEN
    NFix = 1
  ELSE
    NFix = 0
  ENDIF
  WRITE (OutField) NFix,Reg(N).Epsi,Reg(N).Rho,Reg(N).Pot
END DO
```

HiPhi. The quantities are the relative dielectric constant ϵ_r , the space-charge density ρ in coulombs/m³, and the value of ϕ for fixed-potential regions. The region quantity names (*RQuantName*) are strings of length 12. The variable *NFix* is a 1-byte integer and the region quantities are 4-byte real numbers.

Table 21.2. Code to create the beam-generated magnetic field output file

```
! --- Header information
WRITE (OutField) IMax,JMax,KMax,DUnit,NCQuant
  DO N=1,NCQuant
    WRITE (OutField) CQuantName(N)
  END DO
! --- Node and element properties
DO K=0,KMax
  DO J=0,JMax
    DO I=0,IMax
      MC = M(I,J,K)
      WRITE (OutField) &
        C(MC).RegNo,C(MC).RegUp,C(MC).x,C(MC).y,C(MC).z, &
        C(MC).Phi,C(MC).H.x,C(MC).H.y,C(MC).H.z,C(MC).Psi
    END DO
  END DO
END DO
! Region properties
WRITE (OutField) NRQuant
DO N=1,NRQuant
  WRITE (OutField) RQuantName(N)
END DO
WRITE (OutField) NRegMax
DO N=1,NRegMax
  IF (Reg(N).Fixed) THEN
    NFix = 1
  ELSE
    NFix = 0
  ENDIF
  WRITE (OutField) NFix,Reg(N).Mu,Reg(N).Pot
END DO
```

Table 21.2 shows the code that generates the beam-generated magnetic field file. The **Magnum** manual describes the file format in detail. The structure is similar to the electric field file with the following exceptions:

- The quantity $NCQuant = 5$ and the recorded node quantities are the reduced potential (ϕ), the components of the applied magnetic field intensity (H_{sx} , H_{sy} and H_{sz}) and the dual potential (ψ).
- The names of the five quantities are recorded as strings of length 12.

- Values of the five quantities are recorded quantities at each node.
- The recorded region quantities are the relative magnetic permeability μ_r for standard materials and the value of the reduced and dual potentials for fixed regions.

Normally **OmniTrak** generates orbit vector files (RUNNAME .OOU) in text format for all runs. The files They contain information on the orbit traces in three-dimensional space. It is relatively easy to access this information for your own plotting and analysis programs. Table 21.3 shows an extract from the file.. The first section is a free-format header. In the header, the first line is a generic title. The second line contains the number of orbits recorded (*NPart*). Lines 3-8 give the dimensions of the solution space, set either by the *BOUNDARY* command or automatically from an analysis of the field solution boundaries. The final quantity is the unit conversion factor used in the **OmniTrak** run. Note that all real number are in E14.6 format

The header is follow by *NPart* sections in fixed format that describe the particle orbit. The first line is a separator for readability. The second line consists of the string *PARTICLE* followed by the particle number in I5 format. Subsequent line list orbit data in 4E14.6 format. Each line contains the following quantities that define one point on the orbit:

t (elapsed time in seconds)
x, y, z (the point position in meters)

The number of lines per particle varies, depending on the length of the orbit and the time step. The following line marks the end of an orbit:

```
-1.000000E+00 -1.000000E+00 -1.000000E+00 -1.000000E+00
```

Data lines in the plot file for an *FLINE* simulation have the following entries:

d (total trace length in meters)
x, y, z (the point position in meters)

Table 21.3. OmniTrak plot file extract

OmniTrak Plot File (Field Precision)

NPart: 694
XMin: -5.000000E-02
YMin: -5.000000E-02
ZMin: -5.000000E-02
XMax: 5.000000E-02
YMax: 5.000000E-02
ZMax: 5.000000E-02
DUnit: 1.000000E+02

Particle 1
0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
1.000000E-03 -5.774023E-04 -5.773241E-04 -5.773245E-04
2.000000E-03 -1.154818E-03 -1.154641E-03 -1.154642E-03
...
9.000000E-02 -4.642266E-02 -4.998765E-02 -4.875288E-02
9.001238E-02 -4.642213E-02 -5.000000E-02 -4.875235E-02
-1.000000E+00 -1.000000E+00 -1.000000E+00 -1.000000E+00

Particle 2
0.000000E+00 -2.000000E-03 0.000000E+00 0.000000E+00
1.000000E-03 -2.631299E-03 -5.483889E-04 -5.483893E-04
2.000000E-03 -3.264534E-03 -1.095661E-03 -1.095662E-03
...

Particle 694
0.000000E+00 3.129768E-02 1.876559E-02 1.503090E-02
1.000000E-03 3.223628E-02 1.907500E-02 1.518355E-02
...
1.900000E-02 4.971386E-02 2.266348E-02 1.674594E-02
1.928615E-02 5.000000E-02 2.266521E-02 1.674741E-02
-1.000000E+00 -1.000000E+00 -1.000000E+00 -1.000000E+00

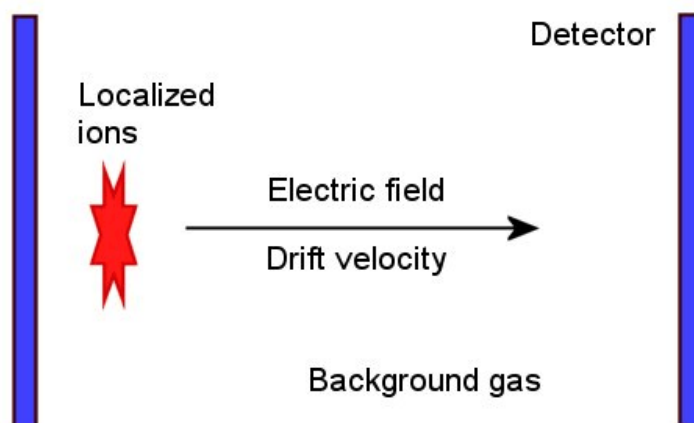


Figure 22.1. Principle of the ion-mobility time-of-flight spectrometer.

Chapter 22. Features for ion mobility spectrometry

OmniTrak includes useful features for the design of time-of-flight ion-mobility spectrometers. Figure 22.1 shows the basic principle. The material to be analyzed is ionized in a gas background (typically air at atmospheric pressure). A electric field is applied and the ions drift with velocity vector:

$$\mathbf{v} = \mu \mathbf{E},$$

where \mathbf{E} is the electric field vector (in V/m) and μ is the mobility (in $\text{m}^2/\text{V}\cdot\text{s}$). Species with different mobilities separate in transit. With a known electric field distribution, the time-of-flight to a detector can be used to infer μ .

Because the drifting ions follow lines of \mathbf{E} , the electric-field tracking capabilities of **OmniTrak** (Chapter 14) are ideal for this application. The program can calculate and plot precise ion trajectories in complex three-dimensional geometries. **OmniTrak** can also compute the ion time-of-flight for a given mobility. The time increment Δt for an ion to move a distance Δs is

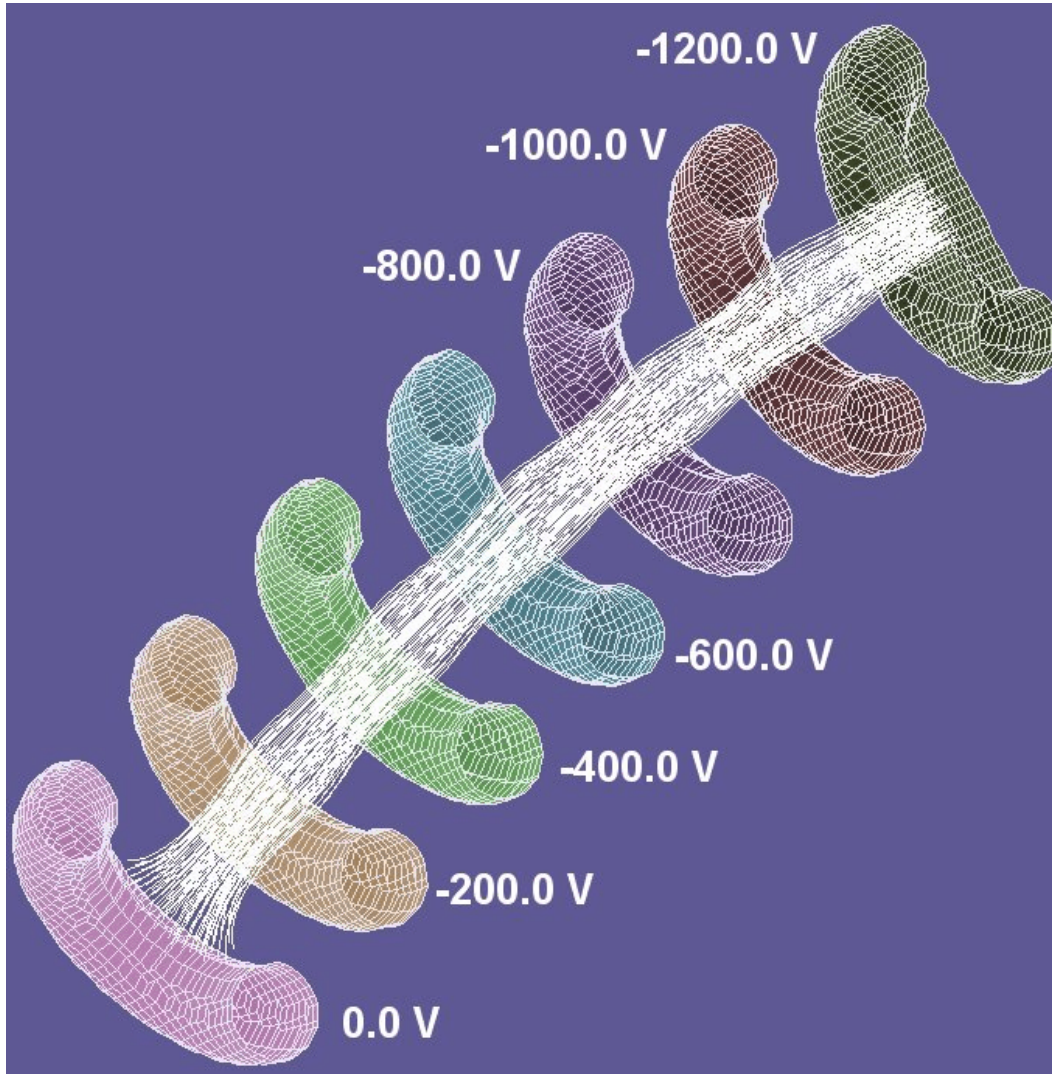


Figure 22.2. Benchmark calculation of a curved ion-mobility spectrometer showing a cutaway view of electrodes to create an accelerating electric field and selected ion orbits for an initial plasma radius of 0.4 cm.

$$\Delta t = \frac{\Delta s}{|v|} = \frac{\Delta s}{\mu |E|} .$$

The time-of-flight from point A to point B is therefore

$$t_{AB} = \int_A^B \frac{ds}{\mu |E|} .$$

Table 22.1. File MOBILITY_DEMO01.OIN

```
FIELDS
  DUnit = 100.0
  EField3D = MOBILITY_DEMO.HOU
END
PARTICLES FLine
  FList
    -0.49 0.00 0.00
    -0.45 0.00 0.00
    ...
    0.45 0.00 0.00
    0.49 0.00 0.00
  End
  Ds = 0.05
END
DIAGNOSTICS
  FLineList
END
ENDFILE
```

While tracing a field line **OmniTrak** computes the following integral to yield the *normalized transit time*:

$$\tau_{AB} = \mu t_{AB} = \int_A^B \frac{ds}{|E|} .$$

The quantity τ (in m^2/V) depends only the nature of the electric field solution. The properties of the background gas and ions are contained in μ . Therefore, a knowledge of the normalized transit time distribution completely characterizes the performance of a detector for any fill gas or ion species.

To illustrate the procedure consider the geometry of Figure 22.2. A set of six rings and a detector generates an electric field pointing approximately in the z direction. The inner diameter of the rings is 1.0 cm and the axial spacing is about 1.0 cm. The voltage drop between rings is -200 V. The assembly curves over an arc of 18.9° giving a 1 cm displacement of the detector in the x direction to ensure that there is no optical line-of-sight from the entrance aperture. The curvature introduces differences in τ that affect the resolution of the detector. In the first calculation we trace ion trajectories (electric field lines) that start in the entrance plane at different

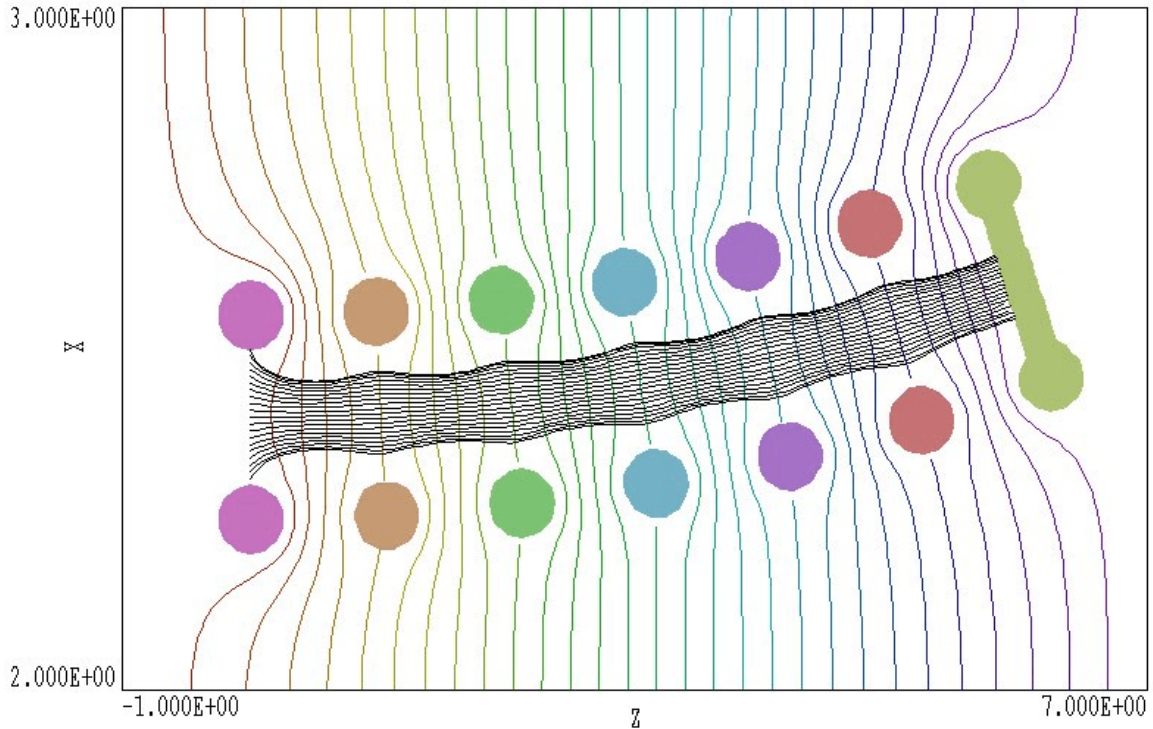


Figure 22.3. View of equipotential lines and ion orbits in the plane $y = 0$ for run MOBILITY_DEMO01.

values of x (displacement in the direction of curvature). Table 22.1 shows the **OmniTrak** script to control the run. Figure 22.3 shows equipotential contours and electric field lines in the plane $y = 0.0$. The field distribution near the entrance aperture has the favorable property that trajectories are compressed and all ions are collected. There is clearly a difference in the physical length of the trajectories. Figure 22.4 shows a plot of τ versus the start position in x . The ions that start at the bottom of Fig. 22.3 follow longer paths and pass through a region with lower $|\mathbf{E}|$. Therefore they have larger transit times. Ions that start at the top of Fig. 22.3 require additional time to pass through the low-field region near the entrance.

In a second run (MOBILITY_DEMO02) the *Circular Beam Tool* of **OmniTrak** was used to create a large number of ions (239) distributed uniformly over a circle of radius 0.4 cm in the entrance plane. A selection of ion trajectories is shown in Fig. 22.2. In response to the *FLINELIST* command **OmniTrak** records a statistical analysis of normalized transit time and field-line length in the listing file (Table 22.2) The average normalized transit time is $\tau_{\text{avg}} = 3.14 \times 10^{-6} \text{ m}^2/\text{V}$. If the ion mobility is

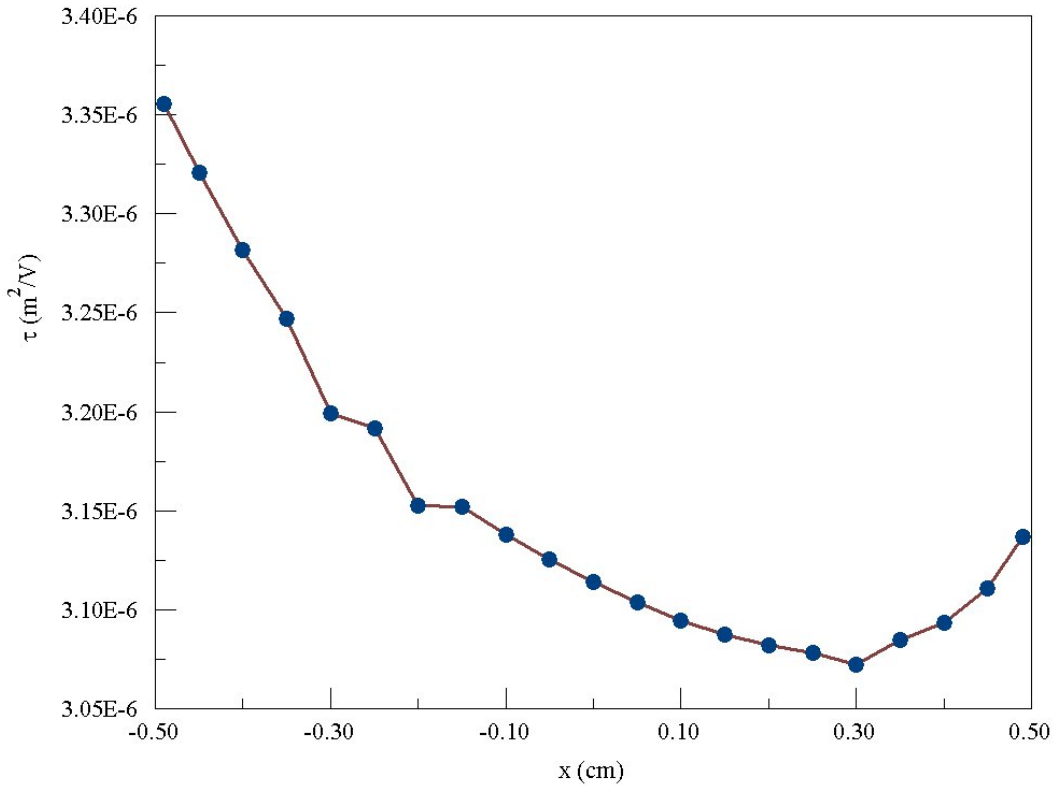


Figure 22.4. Plot of normalized transit time as a function of initial position in x for ions initiated on the line $y = 0.0, z = 0.0$.

$1.36 \times 10^{-4} \text{ m}^2/\text{V}\cdot\text{s}$, the predicted average transit time is 23.1 ms. The dispersion in normalized transit time is $\Delta\tau = 4.69 \times 10^{-8} \text{ m}^2/\text{V}$. Therefore, we expect that the field-magnitude and trajectory variations limit the detector resolution to about $R \geq \Delta\tau/\tau_{\text{avg}} = 1.5\%$.

For reference, the absolute value of mobility μ is related to the reduced mobility μ_0 (defined at standard temperature and pressure) by:

$$\mu = \mu_0 \left(\frac{760}{P} \right) \left(\frac{T + 273.15}{273.15} \right).$$

In the equation P is the pressure in mm of mercury and T is the temperature in $^{\circ}\text{C}$.

Table 22.2. Statistical analysis listing, MOBILITY_DEMO02

Number of values: 239
Normalized transit time ($\text{Tau} = \text{Int}(ds/|E|)$)
Tau (average): 3.13952E-06 (m²/V)
Tau (stddev): 4.68861E-08 (m²/V)
Tau (minimum): 3.07809E-06 (m²/V)
Tau (maximum): 3.26267E-06 (m²/V)
Field line length
D (average): 6.01877E-02 (m)
D (stddev): 4.30777E-04 (m)
D (minimum): 5.96335E-02 (m)
D (maximum): 6.13396E-02 (m)

Appendix 01

Fine-tuning cathode surfaces in MetaMesh

The techniques used in **MetaMesh** produce the best mesh possible within the limits of the user-specified foundation mesh and objects. In cases where the main interest is volumetric fields, small imperfections in the shape of surfaces have a negligible effect. On the other hand, facet errors can significantly degrade results in applications where surface fields are critical. One such application is the creation of cathode emission surfaces for space-charge-limited emission in OmniTrak. Here the facets that constitute the emission surface must have the correct shapes and orientations. Invalid facets lead to poor electric field interpolations that in turn give misdirected electrons with inaccurate assigned currents.

We shall consider an example of a complex assembly consisting of a cylindrical-surface cathode with focusing electrodes to generate a converging sheet beam. To minimize computation time, we treat one quadrant. Figure 1 illustrates the geometry and the complete script is included at the end. The concave cathode is surrounded by a focusing electrode. A shaping piece on the end along y illustrates a good application of the *TRANSITION* element in **MetaMesh**. With a moderate amount of work we can create a surface that follows the cathode curvature and intersects at an angle of 31° at all points. It is clear in Fig. 1 that although the electrodes follow the general defined shapes, there are many imperfections along the edges. The cathode surface as defined is inadequate for the application. The intention of this appendix is to show how to fix the problems.

As a first step we shall improve the definition of the foundation mesh. While maintaining about the same resolution, we want to ensure that critical cathode surfaces lie on the boundaries of foundation mesh elements. With a good foundation mesh we can remove fitting ambiguities and reduce the amount of work the code must perform. The boundaries of the cathode in the x and y directions are at $x = 5.0$ cm and $y = 4.2$ cm. Accordingly, we modify the *XMESH* and *YMESH* sections to:

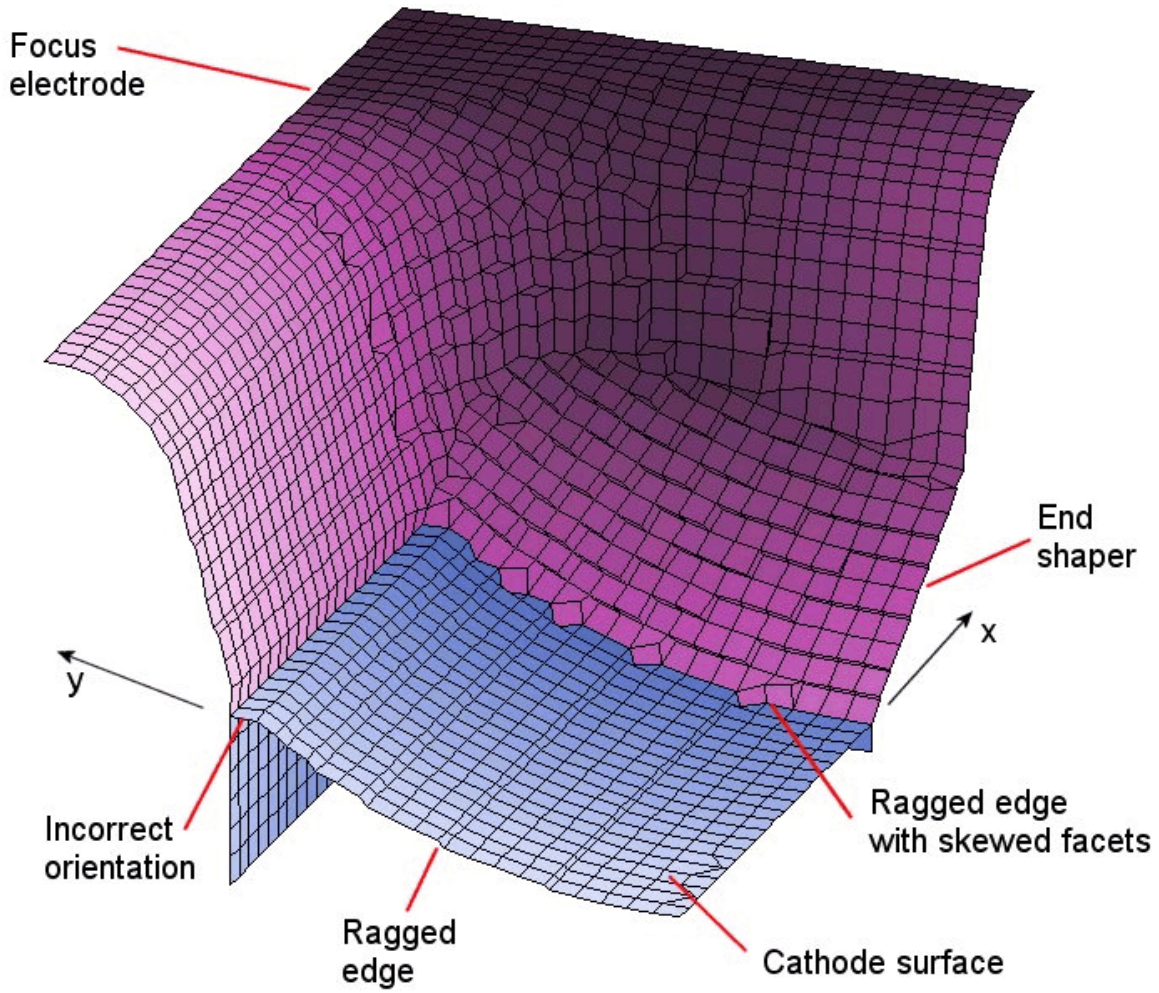


Figure 1. Assembly definitions and the result of the initial fitting attempt.

```

XMesh
  0.00  5.00  0.25
  5.00  9.25  0.25
End
YMesh
  0.00  4.20  0.25
  4.20  6.00  0.25
End

```

The top edge of the cathode lies at $z = 1.40$ cm. In the original mesh, this position was about midway between element boundaries, and the code compromised to fit intersection of the cathode surface and focusing electrode. The result was the row of elements labeled *incorrect orientation* in Fig. 1. Modification of the *ZMESH* section eliminates the problem:

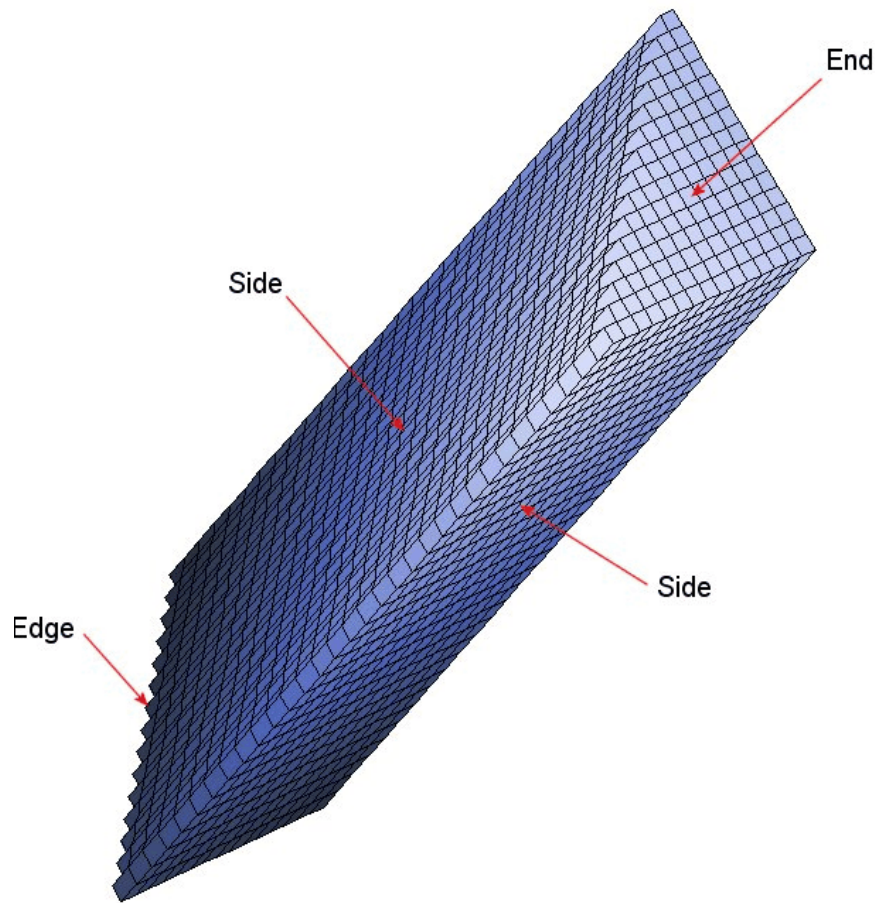


Figure 2. Definition of extrusion terms. A ragged edge may occur if **MetaMesh** has to set fitted nodes to both the sides and the ends.

```
ZMesh
-0.25  1.40  0.25
  1.40 15.00  0.25
End
```

Next, consider the ragged edge at $x = 0.0$ (bottom of Fig. 1). This problem arises because the statements:

```
Fab 5.0
Shift 2.50 0.00 0.00
```

in the definition of the *EXTRUSION* produce a part that extends from $x = 0.0$ to $x = 5.0$. We have defined an end surface (see Fig. 2) of the part at $x = 0.0$ and the code will dutifully try to fit it.

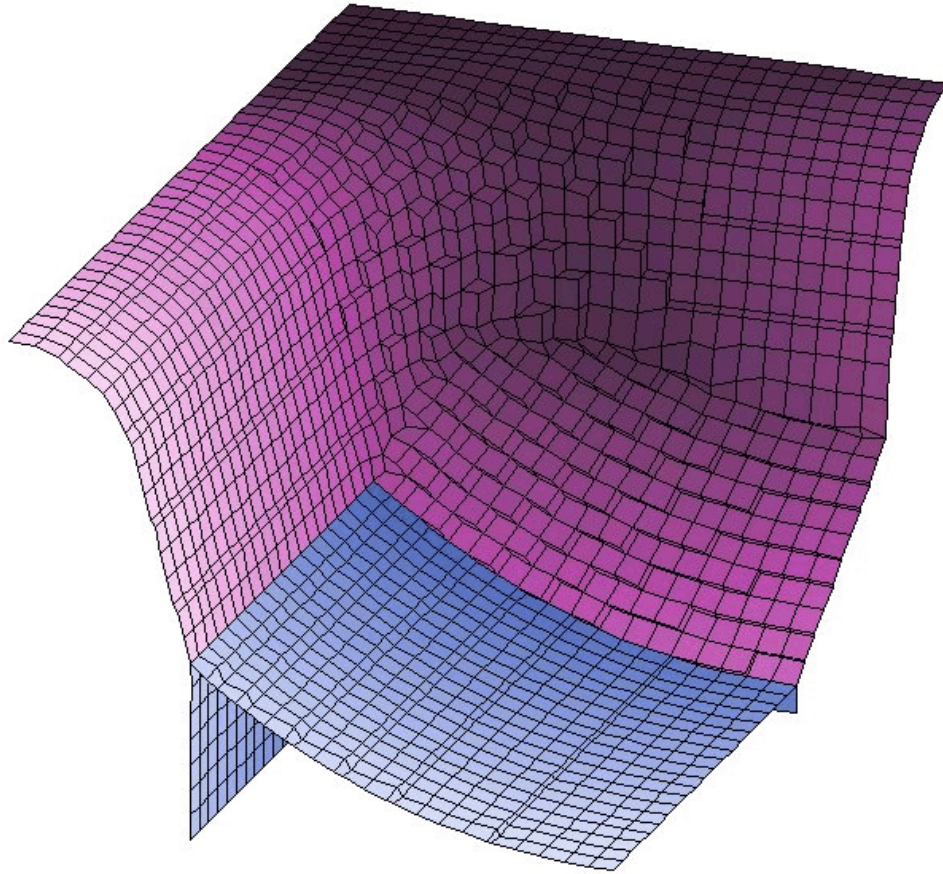


Figure 3. Corrected mesh showing the effect of an improved foundation mesh and application of the *SIDEFIT* option.

Some of the nodes that lie closer to the spurious end than to the sides of the *EXTRUSION* will be shifted resulting in beveled sections along the edge. The solution is to define the full length of the cathode and to let **MetaMesh** perform clipping:

```
Fab 10.0
Shift 0.00 0.00 0.00
```

In this case, an end is not defined at the solution-volume boundary and the code will fit associated nodes only to the *EXTRUSION* sides. Note that we defined the full width of the cathode in the *y*-direction so there was no problem along the boundary at $y = 0.0$. As a general rule, *add extra length to any parts that extend through the solution volume boundaries* and let **MetaMesh** clip them.

Finally, we must complete the boundary correction at $x = 5.0$. The problem again lies with fitting on edges. By default **MetaMesh** assumes that *EXTRUSION*s exist in free space as in Fig.

2. To achieve the best fit the code attempts to move nodes to nearby surfaces on the ends as well as the sides. In many applications we construct objects by butting *EXTRUSIONS* (and possibly *TRANSITIONS*) together so that the ends are not exposed. For example, the focusing electrode in Fig. 1 is a continuous structure. In this case we can assume that all identified surface facets are on extrusion sides. Therefore, we should move nodes only to the sides and disregard the ends. You can signal this intention to **MetaMesh** by adding the keyword *SIDEFIT* to the definitions of *EXTRUSIONS* and *TRANSITIONS*. (Note that severe mesh distortions may occur if the *SIDEFIT* option is specified when end facets are exposed.) Adding the following changes to the script produces the corrected mesh of Fig. 3:

```
Region Cathode
Type Extrusion SideFit

Name EndShaper
Type Transition SideFit
```


Table 1: Complete control script without improvements

```

GLOBAL
  RegName(1) Vacuum
  RegName(2) Cathode
  RegName(3) EmitSurface
  RegName(4) Focus

  XMesh
    0.00  9.25  0.25
  End
  YMesh
    0.00  6.00  0.25
  End
  ZMesh
    -0.25 15.00  0.25
  End
  AutoCorrect ON 9
END
* -----
* SOLUTION VOLUME
* -----
PART
  Region Vacuum
  Type Box
  Fab 18.50 12.00 30.00
END
* -----
* CATHODE AND EMISSION SURFACE
* -----
PART
  Region Cathode
  Type Extrusion
    L  -0.25  4.20  1.40  4.20
    A  1.40  4.20  0.00  0.00  7.00  0.00 S
    A  0.00  0.00  1.40 -4.20  7.00  0.00 S
    L  1.40 -4.20 -0.25 -4.20
    L -0.25 -4.20 -0.25  4.20
  End
  Fab 5.0
  Rotate 0.00 -90.00 0.00
  Shift 2.50 0.00 0.00
  Surface Region Vacuum Edge
  Coat Vacuum EmitSurface
END

```

```

* -----
* FOCUS ELECTRODE
* -----
PART
Region Focus
Name FocusXSide
Type Extrusion
  L  -0.25  0.00  1.40  0.00
  L  1.40  0.00  3.30  0.56  S
  A  3.30  0.56  4.20  1.80  2.90  1.80  S
  L  4.20  1.80  4.20  3.80  S
  L  4.20  3.80  -0.25  3.80
  L  -0.25  3.80  -0.25  0.00
End
Fab 10.0
Rotate 0.00 -90.00 0.00
Shift 0.00 4.20 0.00
Surface Region Vacuum
END
* -----
PART
Region Focus
Name FocusCorner
Type Turning
  L  -0.25  2.30  1.40  2.30
  L  1.40  2.30  3.30  2.86  S
  A  3.30  2.86  4.20  4.10  2.90  4.10  S
  L  4.20  4.10  4.20  6.10  S
  L  4.20  6.10  -0.25  6.10
  L  -0.25  6.10  -0.25  2.30
End
Fab 0.00 90.00
Shift 5.00 1.90 0.00
Surface Region Vacuum
END
* -----
PART
Region Focus
Name FocusYSide
Type Extrusion
  L  -0.25  0.00  1.40  0.00
  L  1.40  0.00  3.30  0.56  S
  A  3.30  0.56  4.20  1.80  2.90  1.80  S
  L  4.20  1.80  4.20  3.80  S
  L  4.20  3.80  -0.25  3.80
  L  -0.25  3.80  -0.25  0.00
End
Fab 3.80
Rotate -90.00 270.00 0.00
Shift 7.35 0.00 0.00
Surface Region Vacuum
END

```

```

* -----
PART
* Approximate cathode edge as four segments
* Add 1.8 cm to all parts to give 31 degrees
Region Focus
Name EndShaper
Type Transition
-0.25 -4.20 3.20 -4.20 -0.25 -4.20 1.40 -4.20
3.20 -4.20 2.71 -3.45 1.40 -4.20 0.91 -3.45
2.71 -3.45 2.32 -2.64 0.91 -3.45 0.52 -2.64
2.32 -2.64 2.03 -1.78 0.52 -2.64 0.23 -1.78
2.03 -1.78 1.86 -0.90 0.23 -1.78 0.06 -0.90
1.86 -0.90 1.80 0.00 0.06 -0.90 0.00 0.00 S
1.80 0.00 1.86 0.90 0.00 0.00 0.06 0.90 S
1.86 0.90 2.03 1.78 0.06 0.90 0.23 1.78 S
2.03 1.78 2.32 2.64 0.23 1.78 0.52 2.64 S
2.32 2.64 2.71 3.45 0.52 2.64 0.91 3.45 S
2.71 3.45 3.20 4.20 0.91 3.45 1.40 4.20 S
3.20 4.20 -0.25 4.20 1.40 4.20 -0.25 4.20
-0.25 4.20 -0.25 -4.20 -0.25 4.20 -0.25 -4.20
End
Fab 3.0
Rotate 0.00 -90.00 0.00
Shift 6.50 0.00 0.00
Surface Region Vacuum
END
* -----
ENDFILE

```