
# OmniTrak 4.0

## Three-dimensional Charged-particle Optics and Gun Design

PO Box 13595, Albuquerque, NM 87192 U.S.A.
Telephone: +1-505-220-3975
Fax: +1-617-752-9077
E mail: techinfo@fieldp.com
Internet: http://www.fieldp.com

# Contents


**1 Introduction** **5**
- 1.1 Program function . . . 5
- 1.2 Operation overview . . . 6

**2 Walkthrough example** **9**
- 2.1 Geometry and mesh definition . . . 9
- 2.2 Electrostatic field solution . . . 12
- 2.3 Particle orbit solution . . . 12
- 2.4 Listing file information . . . 15
- 2.5 Distribution analysis . . . 15
- 2.6 Generating plots . . . 16

**3 Run organization and interactive setup** **19**
- 3.1 Components on an OmniTrak calculation . . . 19
- 3.2 Structure of the **OmniTrak** script . . . 20
- 3.3 Track mode . . . 21
- 3.4 Field line mode . . . 23
- 3.5 Space-charge mode . . . 24
- 3.6 Relativistic beam mode . . . 26
- 3.7 Field-emission mode . . . 26

**4 Running OmniTrak** **28**
- 4.1 Interactive mode commands . . . 28
- 4.2 Charged-particle beam tools . . . 29
- 4.3 Command line operation and batch files . . . 31

**5 OmniView** **32**
- 5.1 File operations . . . 32
- 5.2 2D plots . . . 35
- 5.3 3D plots . . . 38
- 5.4 Saving and loading views . . . 41

**6 Electric field commands – Fields section** **43**
- 6.1 Loading and modifying **HiPhi** solutions . . . 43
- 6.2 Controlling electric field recalculation . . . 44
- 6.3 Program control commands . . . 45
- 6.4 Loading electric field information from a table . . . 46

**7 Applied magnetic fields – Fields section** **48**
- 7.1 Uniform field components . . . 48
- 7.2 Loading and modifying two-dimensional solutions . . . 48
- 7.3 Loading and modifying **Magnum** solutions . . . 49
- 7.4 Loading magnetic field information from a table . . . 50

**8 Temporal modulations – Fields section** **52**
8.1 Tabular modulation functions . . . 52
8.2 Algebraic modulation functions . . . 54
8.3 Electric-field modulation with two voltage levels . . . 56

**9 Single-particle tracking in the Track mode – Particles section** **59**
9.1 Starting orbits from a list . . . 59
9.2 Starting orbits from a source surface . . . 62
9.3 Orbit integration controls . . . 64
9.4 Stopping orbits . . . 66
9.5 Setting material properties . . . 71
9.6 Particle starting times . . . 72
9.7 Application example . . . 72

**10 Beam forces in the SCharge mode – Particles section** **78**
10.1 Beam-generated forces . . . 78
10.2 Control of orbit and field updates . . . 79
10.3 Application example . . . 80

**11 Self-consistent Child law emission – Particles section** **84**
11.1 Mechanics of the space-charge emission model . . . 84
11.2 Space-charge emission control commands . . . 86
11.3 Application examples . . . 88
11.4 Restarting runs with space charge and Child-law emission . . . 92

**12 Beam-generated magnetic fields - Particles section** **94**
12.1 Modeling relativistic beams . . . 94
12.2 Relativistic approximation . . . 96
12.3 Physical basis of the *RelBeam* tracking mode . . . 98
12.4 Commands for the *RelBeam* mode . . . 103
12.5 Application example – planar beam gun . . . 106

**13 Electron field emission – Particles section** **110**

**14 Secondary emission – Particles section** **113**
14.1 Control commands . . . 114
14.2 Benchmark test . . . 117

**15 Electric and magnetic field lines – Particles section** **121**
15.1 Commands for field line tracing . . . 121
15.2 Conventions for field line tracing . . . 123

**16 Diagnostics section** **125**
16.1 Field diagnostics . . . 125
16.2 Particle diagnostics . . . 128
16.3 Particle termination conditions . . . 129
16.4 Field line diagnostics . . . 129

**17 Using field tables** **130**
17.1 Field table format . . . . . 130
17.2 Applications of field tables . . . . . 132
17.3 Using field tables in **OmniTrak** and **OmniView** . . . . . 133

**18 Mapper** **135**
18.1 Loading and viewing field tables . . . . . 135
18.2 Modifying field tables . . . . . 137
18.3 Scan plots . . . . . 139
18.4 Creating tables . . . . . 140
18.5 Using **Mapper** to design a dipole magnet . . . . . 144

**19 OmniTrak output files** **148**

**20 Optimizing cathode surfaces in MetaMesh** **151**

**21 Features for ion-mobility spectrometry** **158**

# 1 Introduction

## 1.1 Program function

**OmniTrak** is a comprehensive 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 package includes the **OmniView** postprocessor for 2D and 3D orbit plotting and **GenDist**, a program for the creation and analysis of beam distributions. **OmniTrak** features include:

- 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.
- 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 magnetic field solutions from the two-dimensional **PerMag** program.
- 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 in **MetaMesh**.
- Flexible methods to add temporal modulations to fields in the single particle tracking mode.

Features of **OmniView** include:

- 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 from the **AMaze** three-dimensional **Magnum** program as well as the two-dimensional **PerMag** 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 Operation 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 this manual, ready-to-run examples and the following three programs:

- `OMNITRAK.EXE`. Trace charged particles in a three-dimensional field environment defined by solutions from **HiPhi** and/or **Magnum**.
- `OMNIVIEW.EXE`. Generate plots of field solutions and orbit traces from **OmniTrak**.
- `GENDIST.EXE`. Create and analyze particle input distributions for and from **OmniTrak**.

**GenDist** is described in a separate manual.

An **OmniTrak** simulation consists of three steps:

1. Generation of conformal meshes for the field solutions.
2. Calculation and analysis of the field solutions.
3. Orbit tracking and analysis.

A run may involve the programs **MetaMesh**, **HiPhi**, **PhiView**, **Magnum**, **MagView**, **GenDist**, **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`. Set control and material parameters for the **HiPhi** solution.
- `BFieldName.MIN`. Input to **Metamesh**, mesh for the **Magnum** solution.
- `BFieldName.GIN`. Control and material parameters for the **Magnum** solution.
- `CoilName.CDF` Applied currents for the **Magnum** solution.

- `ParticleName.PRT`. Input particle parameters for the **OmniTrak** solution.
- `TrackName.OIN`. Control parameters for the **OmniTrak** solution.

The prefixes in the list represent names of up to 50 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 tracing 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 five possible run modes:

- **TRACK**. In this mode the program performs single-particle tracking with no contribution from beam-generated fields. Particles can be initiated from a list of input parameters and/or automatically generated from one or more marked emission surfaces. Lists may be prepared with an editor, spreadsheet or **GenDist**. You may also use lists generated by previous **Trak** or **OmniTrak** runs. The mode is useful for low-current devices.
- **SCHARGE**. Orbit tracking with self-consistent effects of beam space-charge. Particles may be initiated from a list of input parameters and/or automatically generated from one or more 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 ion guns and low-voltage electron guns with thermionic emitters.
- **RELBEAM**. Orbit tracking with self-consistent effects of both beam charge and current. Particles may 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-energy 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.
- **FLINE**. Trace electric or magnetic field lines in three-dimensional solutions. **OmniTrak** generates a file of coordinates for 2D or 3D field-line plots in **OmniView**. The program can also list values of field components along the lines.

Chapter 2 gives a detailed description of an **OmniTrak** calculation. The example illustrates many of the code features. Chapter 3 describes the preparation of scripts to control runs, either through interactive dialogs or directly with an editor. Chapter 4 covers operation of the **OmniTrak** program, while Chap. 5 introduces features of **OmniView**. Chapters 6 through 16 gives detailed descriptions of advanced features in the **OmniTrak** script. Chapter 19 documents the format of **OmniTrak** output files. Chapter 20 is a tutorial on the creation of emission surfaces in **MetaMesh** and Chap. 21 describes how to use **OmniTrak** to model ion-mobility mass spectrometers.

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 available in PDF format on the Field Precision Internet site::

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

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

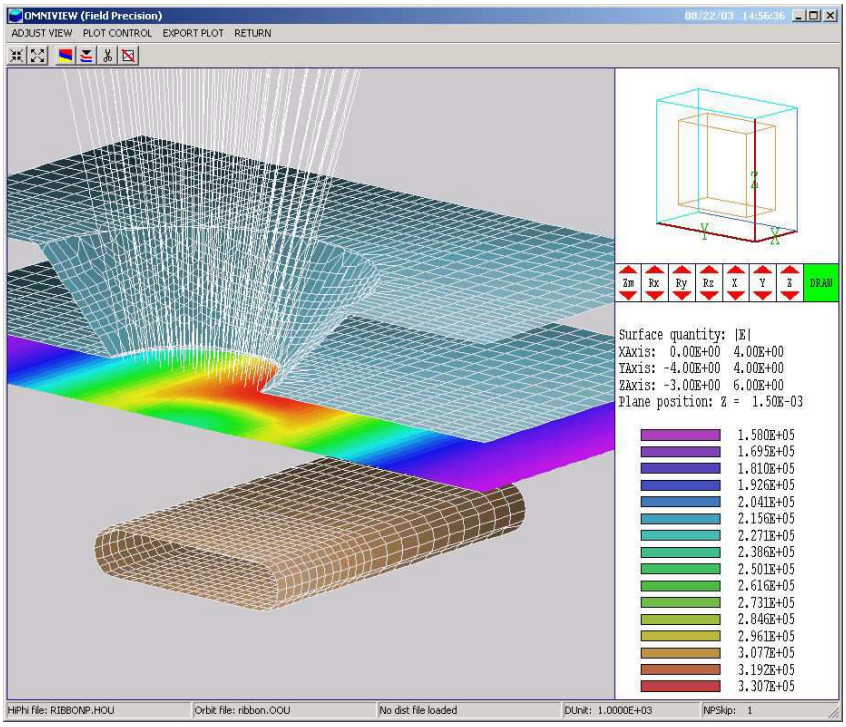


Figure 1: Working environment of the **OmniView** postprocessor.

# 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 1 and 2 show 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 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 file for the example (`RIBBON.MIN`) to a working directory. The file controls the **MetaMesh** program. Run **MetaMesh** and load `RIBBON.MIN`. Table 1 lists the file contents. Spatial quantities are specified in units of mm. The 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 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
```

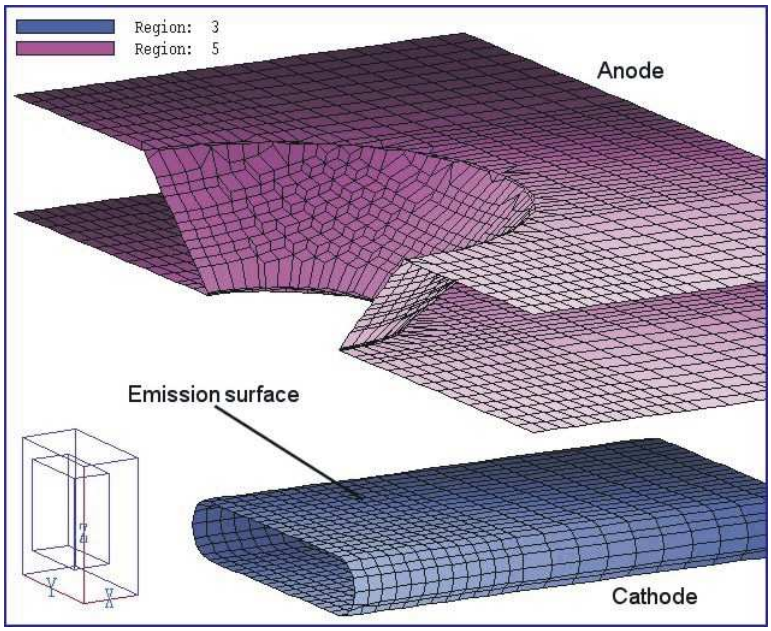


Figure 2: Anode and cathode shapes `RIBBON` example.

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

- To implement automatic generation of electrons in **OmniTrak** we 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 region number equal to that of the emission surface are treated as sites for electron initiation. Section 11.1 describes the process in detail. In the present example, we define the upper surface of the ribbon cathode as an emission region. Figure 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.

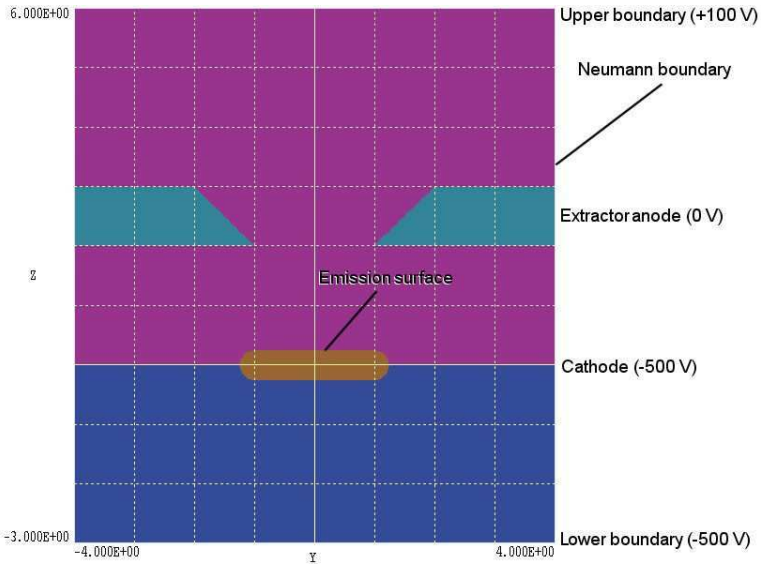


Figure 3: Cross-section of the geometry of example `RIBBON.MIN` in a plane normal to $x$ at $x = 0.0$. Lower vacuum volume: region 1 (blue). Upper vacuum volume: region 2 (violet). Cathode: region 3 (orange). Anode: region 5 (green).

## 2.2 Electrostatic field solution

The next step is to generate a baseline electrostatic solution. Run **HiPhi** and choose the command *Setup*. After you pick the file `RIBBON.MDF`, the program displays the dialog of Fig. 4. In the control section, set $DUnit = 1000.0$ (dimensions in mm), $MaxCycle = 1000$ and $ResTarget = 5.0 \times 10^{-8}$ The *Region properties* grid box has one row for each region of the mesh. The values in the illustration assign the vacuum condition to Regions 1 and 2 and set the potentials of electrodes to those shown in Fig. 3.

Set values in the dialog to those shown in Fig. 4 and then click *OK*. Save the script as `RIBBON.HIN`. Click on the *Run* command and choose the input file. The three-dimensional electrostatic solution takes less than one minute. **HiPhi** creates the output file `RIBBON.HOU` that lists values of node coordinates and electrostatic potential.

## 2.3 Particle orbit solution

We can now proceed to the **OmniTrak** calculation. Start the program and click the *Setup* command. In the initial dialog, choose the option *SCharge* to display the dialog of Fig. 5. Here, we shall set parameters to control the particle orbit calculation. The entries in the *Fields* group define properties of electric and/or magnetic fields. The *DUnit* command specifies that spatial quantities in other fields should be converted from millimeters. The value in the *EFieldPrefix* box specifies that the three-dimensional electrostatic field will be determined from `RIBBON.HOU`. There is no applied magnetic field. Entries in the *Particles* group control tracking of particle orbits and space-charge updates of the field solution. Section 11.1 gives a detailed discussion of the methods used to simulate self-consistent Child law emission. The *Dt* entry 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* entry specifies twelve cycles

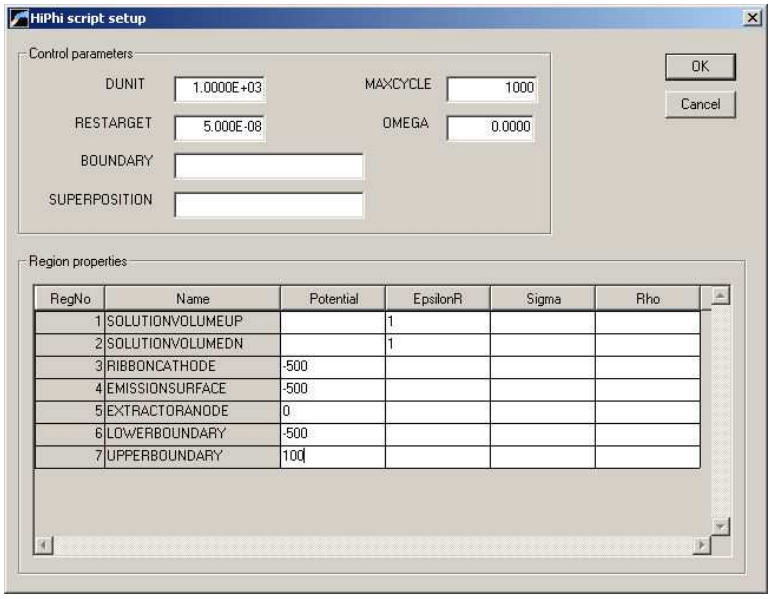


Figure 4: **HiPhi** dialog to create `RIBBON.HIN`.

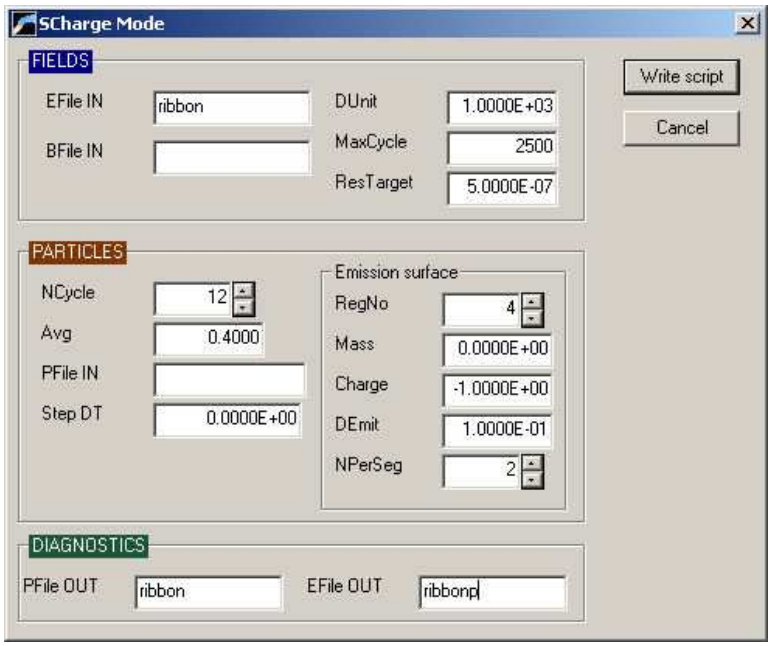


Figure 5: **OmniTrak** dialog to create `RIBBON.OIN`.

Table 2: **OmniTrak** input file `RIBBON.OIN`

```
FIELDS
  EFIELD3D: ribbon.HOU
  DUNIT:   1.0000E+03
  MAXCYCLE(E):   2500
  RESTARGET(E):   5.0000E-07
END

PARTICLES SCHARGE
  NCYCLE:   12
  AVG(E):   4.0000E-01
  EMIT(4):   0.0000E+00 -1.0000E+00  1.0000E-01     2
END

DIAGNOSTICS
  PARTFILE: ribbon
  EDUMP: ribbonp.HOU
  PARTLIST
END

ENDFILE
```

of orbit tracking and field updates while the command *Avg* controls space-charge averaging between cycles. The parameters were chosen to achieve good solution convergence for the specific problem. Values in the *Emission* surface group specify 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 $2 \times 2$. The *Diagnostics* sections entries control the creation of output files. Enter the values shown in Fig. 5 and then click on *Write script*. Save the file as `RIBBON.OIN`. Table 2 shows the contents of the resulting script.

To start the calculation, click on *Start run* from *Run* menu and choose `RIBBON.OIN`. During the run, **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. 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.OOU`. 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 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 may be used as input for subsequent **OmniTrak** runs. It may be loaded in **GenDist** for distribution analyses or it may be ported to your own analysis programs.

## 2.4 Listing file information

In the remaining sections of this chapter, we shall review features of the analysis tools in the **OmniTrak** package. First, we shall check the listing file to determine the convergence history of the run and the final value of total current emitted from the ribbon. Click on File/Edit listing files and pick RIBBON.OLS. You can find the following listing at the end of the file:

```
Current convergence history
    NCycle     Total emitted
                current (A)
    ========================
        1         1.10886E-01
        2         7.96334E-02
        3         1.08935E-01
        4         1.11790E-01
        5         1.11790E-01
        6         1.11790E-01
        7         1.11708E-01
        8         1.11455E-01
        9         1.11291E-01
       10         1.11182E-01
       11         1.11092E-01
       12         1.11092E-01
```

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

## 2.5 Distribution analysis

Run **GenDist** to check the output particle distribution. Click on the command *Load PRT file* and choose `RIBBON.PRT`. By default, the program shows a plot of final particle positions in the $x$-$y$ plane (integrated over all values of $z$). The display is confusing because the positions of extracted particles (that pass 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 $z = 2.0$ mm. To improve the display, click on the *Apply 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 of Fig. 6 showing a slightly elliptical beam. To get quantitative information about the filtered particles, click on the *Beam analysis* command in the *Analysis* menu. In the dialog, specify the file prefix `RIBBONDIST`. **GenDist** records calculated beam quantities relative

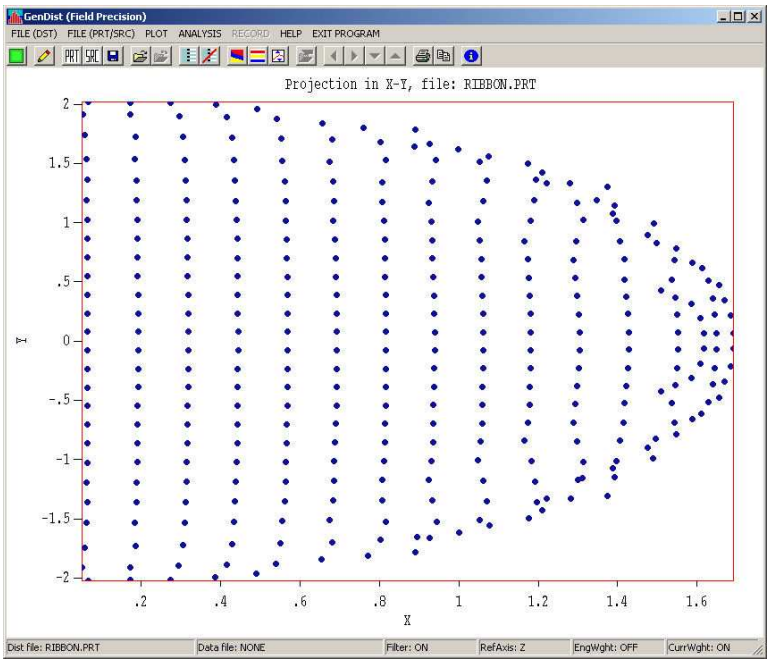


Figure 6: Distribution of extracted particles in the $x$-$y$ plane for the `RIBBON` example.

to a reference axis (in this case, the program uses the default $z$ axis as the reference). Table 3 shows extracts from the file `RIBBONDIST.DAT`. Note that the total extracted current is 0.0115 A, less than 10 per cent of the current emitted from the ribbon. The extracted current for a full circular aperture is 0.0230 A.

## 2.6 Generating plots

**OmniView** can create two- and three-dimensional plots of orbits superimposed on the three-dimensional electric field solution. Run the program 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* slider 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. Pick the file `RIBBON.OOU` and then return to the 2D plots menu. 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 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. 7. The plot

Table 3: **GenDist** data file `RIBBONDIST.DAT`, distribution analysis for the `RIBBON` example.

```
Current magnitude:  1.1528E-02 (A)

Averaged quantities in absolute space
  Total number of entries in file:      338
  Average current per particle:  3.4107E-05 (A)
  Average position
     X:  8.2630E-01
     Y: -5.1553E-05
     Z:  6.0000E+00
  Average direction vector
     Ux:  0.0957824
     Uy: -0.0000001
     Uz:  0.9834995
  Average energy:  5.9936E+02 (eV)

Averaged quantities relative to the reference axis
  Average electron displacement
     X-XRef:  8.2630E-01
     Y-YRef: -5.1553E-05
  Average electron angle (degrees)
     ThetaX:    5.496
     ThetaY:    0.000
  Electron RMS angular spread (degrees)
     dThetaX:    6.429
     dThetaY:    8.253
  Electron RMS position spread
     dX:  9.6139E-01
     dY:  1.0870E+00
     dR:  1.4512E+00
  Electron RMS energy spread:  4.9577E+00 (eV)
```

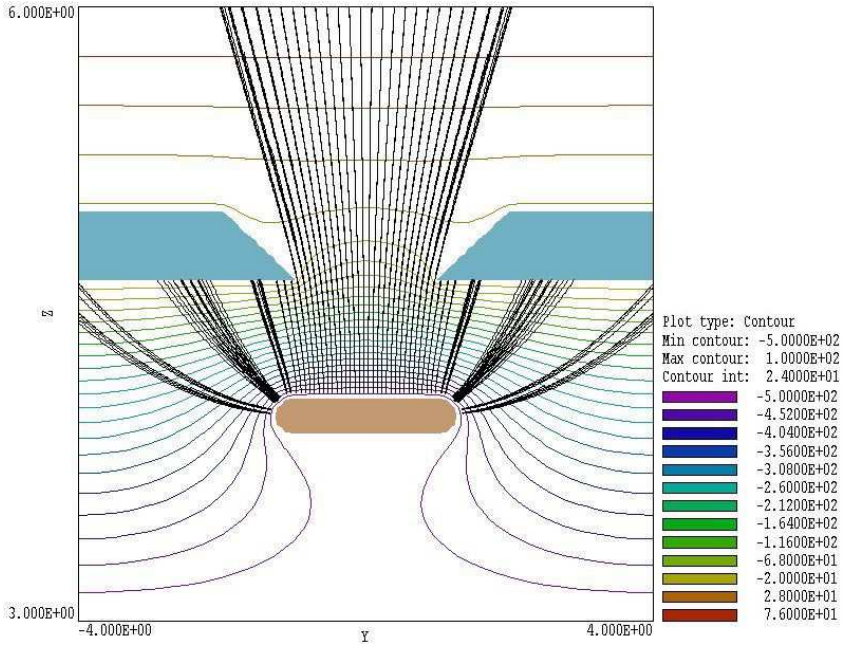


Figure 7: Two-dimensional plot of electrostatic equipotential contours and projected orbits, `RIBBON` example.

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. 1. Chapter 5.3 discusses procedures to generate such plots.

# 3 Run organization and interactive setup

## 3.1 Components on an OmniTrak calculation

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

- A **HiPhi** and/or **Magnum** output file that describes the conformal hexahedron mesh and contains values to find the electric and/or magnetic fields. The files have the name forms `ENAME.HOU` and `BNAME.GOU`. They may be in either binary or text format.
- An **OmniTrak** script (`ONAME.OIN`) that sets control parameters and initial properties of particle orbits.

Other input files that may appear in specific simulations are listed in Table 4. An **OmniTrak** simulation may include the following steps:

- Prepare a **MetaMesh** script (`ENAME.MIN`) that defines the spatial distribution of regions for the electrostatic solution. For this task, you can use the interactive environment of **Geometer** or write the file directly with an editor.
- Prepare a **MetaMesh** script (`BNAME.MIN`) for the magnetostatic solution.
- Run **MetaMesh** to create the binary files `ENAME.MDF` and `BNAME.MDF` with information on the conformal meshes.
- Prepare a **HiPhi** script (`ENAME.HIN`) that sets control parameters and defines the material properties of regions for the electrostatic solution. For this task, you can use the interactive dialog in **HiPhi** or work directly with a text editor.
- Run **HiPhi** to create a solution file `ENAME.HOU`. This file contains information on node spatial coordinates and the applied electrostatic potential.
- Prepare a **Magnum** script (`BNAME.MIN`) that sets control parameters and defines the material properties of regions for the applied magnetostatic solution.
- Run **Magnum** to create a solution file `BNAME.HOU`. This file contains information on node spatial coordinates, applied magnetic field $\mathbf{H}_s$, and the magnetostatic scalar potentials $\phi$ and $\psi$.
- (Optional) Depending on the type of run, prepare a list of starting particle parameters using a text editor, spreadsheet or the `GenDist` utility (`PNAME.PRT`).
- Prepare a script (`ONAME.OIN`) to control the **OmniTrak** run. The script controls reading of field file information and modification of values, initiation of particle orbits, orbit tracking, field updates and diagnostics. Basic scripts may be prepared using the interactive dialogs described in Sects. 3.3 through 3.6. You can modify scripts or add advanced controls with an editor.

Table 4: **OmniTrak** files

| Name | Function |
|---|---|
| ENAME.MIN | Geometry of electric materials, input to **MetaMesh** |
| ENAME.MDF | Output from **MetaMesh**, input to **HiPhi** |
| ENAME.HIN | Control and description of electrical materials, input to **HiPhi** |
| ENAME.HOU | Output from **HiPhi**, input to **OmniTrak** |
| BNAME.MIN | Geometry of magnetic materials, input to **MetaMesh** |
| BNAME.MDF | Output from **MetaMesh**, input to **Magnum** |
| CNAME.CDF | Geometry of drive coils, input to **MagWinder** |
| CNAME.WND | Output from **MagWinder**, input to **Magnum** |
| BNAME.GIN | Control and description of magnetic materials, input to **Magnum** |
| BNAME.GOU | Output from **Magnum**, input to OmniTrak |
| ONAME.OIN | Program control script, input to **OmniTrak** |
| PNAME.PRT | Listing of initial particle parameters, input to **OmniTrak** |
| ONAME.OLS | Run parameters and diagnostics, output from **OmniTrak** |
| ONAME.OOU | Orbit vectors, output from **OmniTrak** |
| PNAMEP.PRT | Final particle parameters, output from **OmniTrak** |

- Run **OmniTrak** to create the files ONAME.OLS (listing of run parameters and particle data), ONAME.OOU (vector file for orbit plotting) and/or PNAMEP.PRT (optional listing of final particle parameters).
- (Optional) Check particle distributions in PNAMEP.PRT with **Gendist** or use **OmniView** to create orbit/field plots from ONAME.OOU.

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 may be used for different **OmniTrak** runs.

## 3.2 Structure of the OmniTrak script

A control script FPREFIX.OIN is required for each **OmniTrak** run. This chapter discusses how to create scripts using the interactive dialogs that appear in response to the *SetUp* command in the main **OmniTrak** menu. Following chapters give detailed information on building scripts and adding advanced features with a text editor.

Whichever method is used, it is useful to understand the script format. A script has three main sections: *Fields*, *Particles* and *Diagnostics*. The file has the following structure:

```
Fields
   (Field commands)
End
Particles RunMode
   (Particle commands)
End
Diagnostics
   (Diagnostic commands)
End
EndFile
```

The sections must appear in the order shown and must terminate with an *End* command. Each section has a set of allowed commands. Within a section, valid commands may appear in any order. **OmniTrak** begins processing a section when all commands have been read. The *EndFile* command closes all files and stops the program.

The commands in the *Fields* section control input of field solution files from **HiPhi**, **Magnum** and/or **PerMag**. You can enter advanced commands with a text editor to adjust field values, introduce time variations, add constant magnetic field components and adjust the potential of individual electrodes.

The *Particles* section controls orbit computations. A string in the section heading specifies the tracking mode: *FLine*, *Track*, *SCharge*, *RelBeam* or *FEmit*. The valid commands of the *Particles* section depend on the tracking mode - the program stops with an error message if it finds an invalid command. The commands of the *Particles* section serve three functions:

- Control orbit calculations (global boundaries, mesh search options, time step, listing options, ...)
- Initiate particle orbits (read particle parameters, set marked emission surfaces, ...)
- Define stopping criteria

**OmniTrak** reads all commands in the *Particles* section, processes the information, and then computes the orbits. Depending on the tracking mode, the program may also update the potential and recalculate orbits over a number of cycles.

The commands of the *Diagnostics* section initiate formatted listings of information on fields and calculated orbits. **OmniTrak** can record scans of applied magnetic fields, self-consistent electric fields and beam-generated magnetic fields. The program makes formatted listings and output files of initial and final particle parameters. The program can generate statistics on final beam distributions. Alternatively, you can also use **GenDist** for distribution calculations if the **OmniTrak** run generates an output particle file.

## 3.3 Track mode

**OmniTrak** features interactive dialogs to help you create basic run control scripts. To start, click the *SetUp* menu command or tool. In the initial dialog, pick the appropriate tracking mode. If you choose the default *Track* mode, the program calls up the dialog of Fig. 8. Note that dialog entries are divided into the same three categories as the **OmniTrak** script: *Fields*, *Particles* and *Diagnostics*. Entries in the different dialog groups create commands in the corresponding

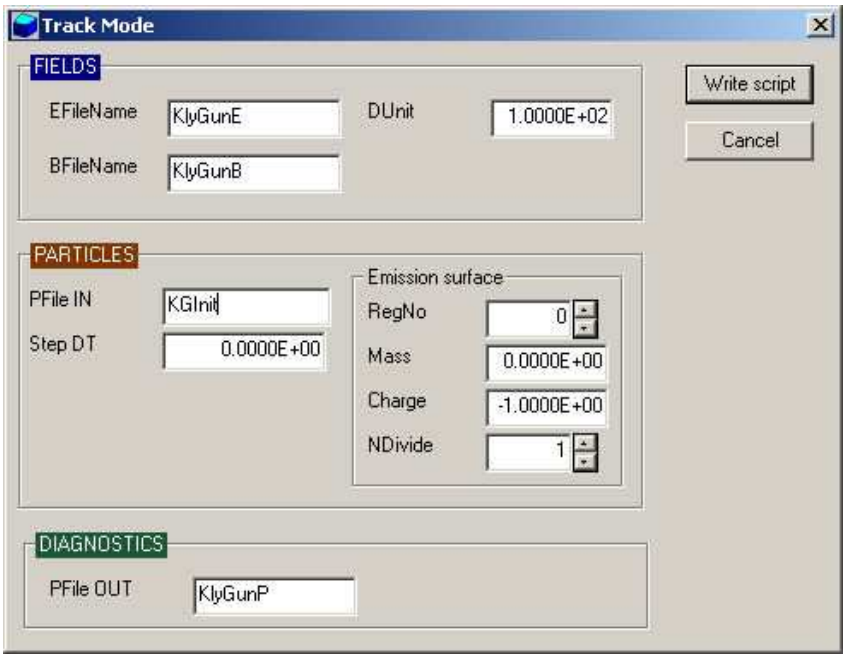


Figure 8: Run setup dialog – *Track* mode.

script section. This chapter gives a brief description of the actions of the dialog entries. Detailed descriptions and commands to control advanced functions are described in Chaps. 6 through 16.

### EPREFIX IN

The prefix of a **HiPhi** solution to provide electric field information for particle tracking. Leave this field blank if the environment for particle tracking has only a magnetic field.

### BPREFIX IN

The prefix of a **Magnum** solution to provide magnetic field informations for particle tracking. Leave this field blank if the environment for particle tracking has only an electric field.

### DUNIT

A conversion factor for lengths that appear in **OmniTrak** script commands, equal to the number of distance units per meter. For example, to supply dimensions in microns, set $DUnit = 1.0 \times 10^6$.

### PPREFIX IN

Supply the prefix of a file `PPREFIX.PRT` that contains a list of start parameters for particles. The format of the file is described in Sect. 9.1.

### DT

Supply a time step (in units of seconds) for orbit integrations. If the box is blank or contains

0.0, **OmniTrak** will try to pick an appropriate value based on the properties of the mesh, the electric-field solution and/or the particle kinetic energy.

The next four commands appear in the *Emission surface* group. An emission surface is an alternate way to initiate particle orbits. **OmniTrak** identifies element facets of a fixed-potential region (electrode) and starts one or more particles per facet close to the surface in an adjacent dielectric element.

#### REGNO
Supply the number of an emission line region on the surface of a fixed-potential region. Section 20 covers techniques to create emission surfaces in **MetaMesh**.

#### MASS
Supply the mass of particles created on the emission surface in AMU (atomic mass units). **OmniTrak** inserts the mass of the electron if 0.0 appears in the box.

#### CHARGE
Supply the charge of particles created on the emission surface in fundamental charge units. Here, protons have $Charge = +1.0$ and electrons have $Charge = -1.0$.

#### NDIVIDE
Controls the number of orbits to start per emission surface facet. The program generates $NDiv \times NDiv$ evenly-spaced particles on each facet.

The *Diagnostics* group contains a single field. You can add additional commands for advanced diagnostics with a text editor.

#### PPREFIX OUT
Supply a prefix if you want **OmniTrak** to write a file `PPREFIX.PRT` of final orbit parameters. The file may be used as input in a subsequent **OmniTrak** run or ported to the Field Precision programs **GenDist**, **Trak** and **GamBet**.

### 3.4 Field line mode

If you choose the *FLine* mode, **OmniTrak** displays the dialog of Fig. 9. You can track three-dimensional electric or magnetic field lines by loading solutions from **HiPhi** or **Magnum**.

#### EPREFIX IN
The prefix of a **HiPhi** solution for electric field line tracking. The program gives an error message if there are entries in both the *EFileName* and *BFileName* boxes.

#### BPREFIX IN
The prefix of a **Magnum** solution for magnetic field line tracking. The program gives an error message if there are entries in both the *EFileName* and *BFileName* boxes.

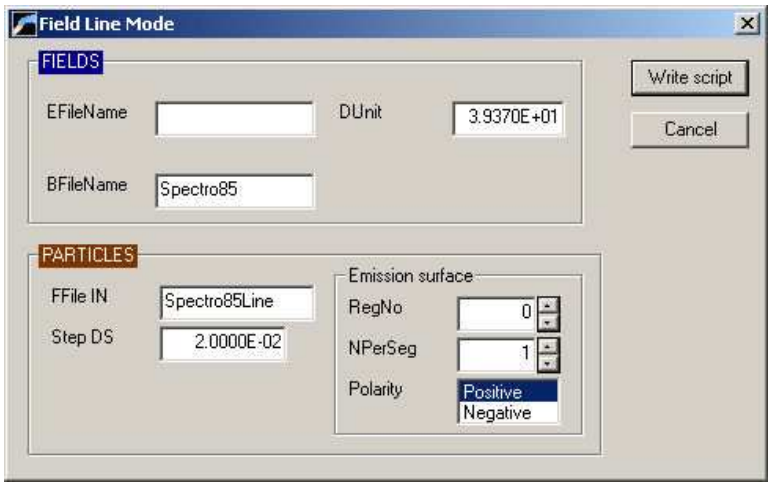


Figure 9: Run setup dialog – *FLine* mode.

#### FFILE IN

Supply the prefix of a file `FPREFIX.FLD` that contains a list of start parameters for field line integrals. The format of the file is described in Sect. 15.1.

#### DS

Field line integrals proceed in small spatial steps rather than time steps. Enter a value in units specified by *DUnit*. **OmniTrak** will pick a default if the field is blank or equals 0.0.

Entries in the *Emission surface* group are similar to those of the *Track* mode dialog. The main difference is the absence of particle parameters and the presence of the following entry.

#### POLARITY

For *Positive* polarity, the spatial integral proceeds along the direction of positive electric or magnetic field.

### 3.5 Space-charge mode

The dialog of Fig. 10 is displayed when you pick the *SCharge* mode. In comparison to the dialog of the *Track* mode, there are two additional entries in the *Fields* group.

#### MAXCYCLE

Particle orbits are computed and the electric field is recalculated over $NCycle$ field/particle cycles to include the effect of beam space charge. The parameter $MaxCycle$ is the maximum number of iterations in the matrix solution for the electrostatic potential. Higher values give more accuracy at the expense of longer run times. The value of $MaxCycle$ should be high enough to ensure convergence at the end of the run. A solution is convergent if the following conditions are satisfied:

- The current from emission surfaces changes little between field/particle cycles.

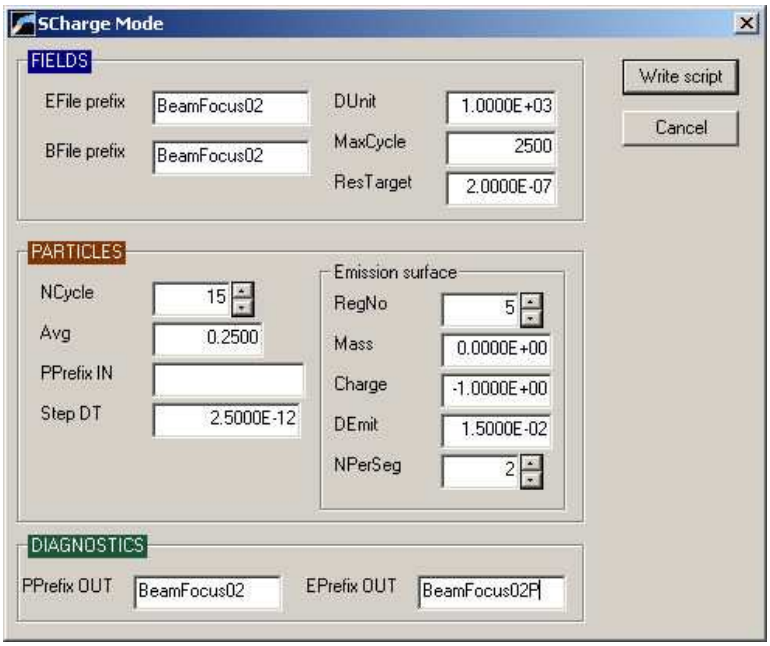


Figure 10: Run setup dialog – *SCharge* mode.

- The initial relative residual (average error in the electrostatic potential) has a low value ($\ll 1.0 \times 10^{-6}$) entering the final particle/field cycle.

### RESTARGET

Set a value for the target *relative residual* in the matrix solution for the electrostatic potential. The quantity is a measure of the accuracy of the solution and should be small compared to 1.0. **OmniTrak** exits the field recalculation routine if the number of interations exceeds $MaxCycle$ or if the residual is less than the target value. Choices of $MaxCycle$ and $ResTarget$ affect the solution accuracy versus the run time.

The *Emission* surface group in the *Particles* section has the following additional parameter.

### DEMIT

The quantity $DEmit$ is the distance between the physical source surface (*e.g.*, cathode surface) and a vitual emission surface required to model Child-law emission. Enter $DEmit$ in units set by the current value of $DUnit$. The virtual surface should be close to the source source and follow its general contours. On the other hand, $DEmit$ must be $\geq 1.5$ times the local element width to ensure accurate calculations of the electric field.

### EPREFIX OUT

Supply a prefix for a file to record the final electrostatic field solution. The file has a name of the form `PREFIX.HOU`. The file can be inspected with **PhiView** or loaded into **OmniTrak** for a subsequent solution.

## 3.6 Relativistic beam mode

The dialog for the *RelBeam* mode is similar to that of the *SCharge* mode (Fig. 10). There are two additional control groups, *BBMesh* with six fields and *BElem*, with two fields. A detailed knowledge of the field calculation technique (described in Chap. 12) is required to choose values of the parameters.

### BBMESH

Define the parameters of a mesh for the computation of the beam-generated field. Supply the minimum and maximum values of $x$, $y$ and $z$ in units specified by DUnit. The parameters $N_x$, $N_y$ and $N_z$ are the number of mesh nodes along the axis. Ideally, the mesh should cover a volume that just encloses all particle orbits.

### LELEM

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 10-20 vectors would be sufficient.

### RELEM

The quantity *RElem* is the effective radius of the current element vector (Sect. 12.3). 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.

### BPREFIX OUT

**OmniTrak** creates information on the beam-generated magnetic field during orbit integrations. If a prefix is suppled in this field, the program records the data in standard **Magnum** format. The file has a name of the form `PREFIX.GOU`. The file can be inspected with **MagView** or loaded into **OmniTrak** for a subsequent solution.

## 3.7 Field-emission mode

Figure 11 show the dialog to generate a script for the *FEmit* mode. The entries are identical to those of the *SCharge*/*RelBeam* dialog with the exception of those in the *Emission* surface group. Because the mode handles only electron field emission, the options *Mass* and *Charge* are not included.

### WORKFUNC

Enter the emission surface work function in units of eV.

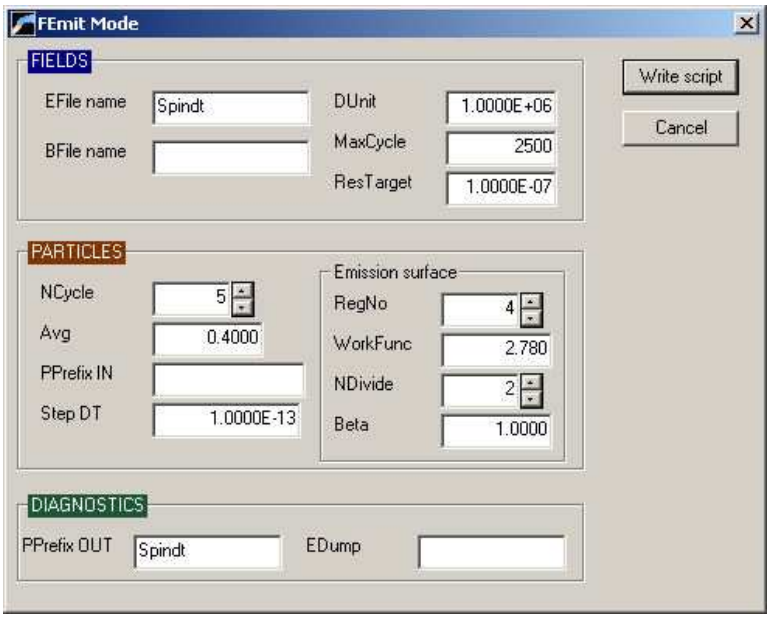


Figure 11: Run setup dialog – *FEmit* mode.

## BETA

The quantity $\beta$ is a parameter to represent field enhancement by submicroscopic structure of the emission surface (e.g., a patch covered with carbon nanotubes). **OmniTrak** calculates the local field amplitude $|\mathbf{E}|$ and uses the quantity $\beta|\mathbf{E}|$ in the Fowler-Nordheim equation to find the local electron current density.

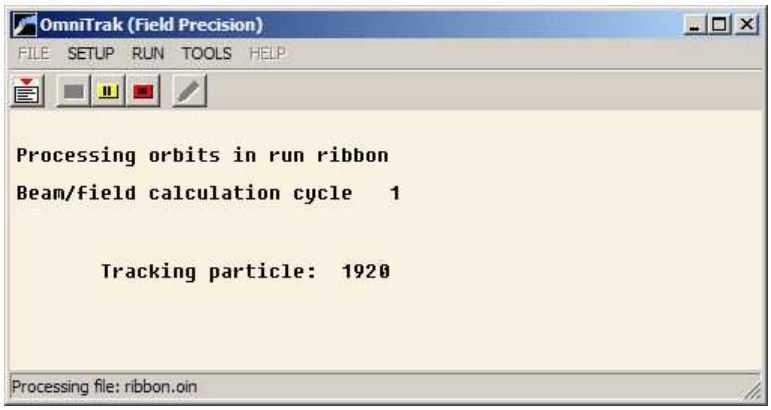


Figure 12: **OmniTrak** running in a window interactive mode.

# 4 Running OmniTrak

## 4.1 Interactive mode commands

**OmniTrak** runs interactively in a window if you launch `omnitrak.exe` from **AMaze** or run the program without a command-line parameter. In this mode you can perform several solutions in a session and temporarily leave the program to work on other tasks. Figure 12 shows the program window.

The menu has four main commands: *Setup*, *File*, *Run* and *Help*.

### SETUP

Launch interactive dialogs to create a control script for an **OmniTrak** run. The procedure was described in detail in Chap. 3.

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

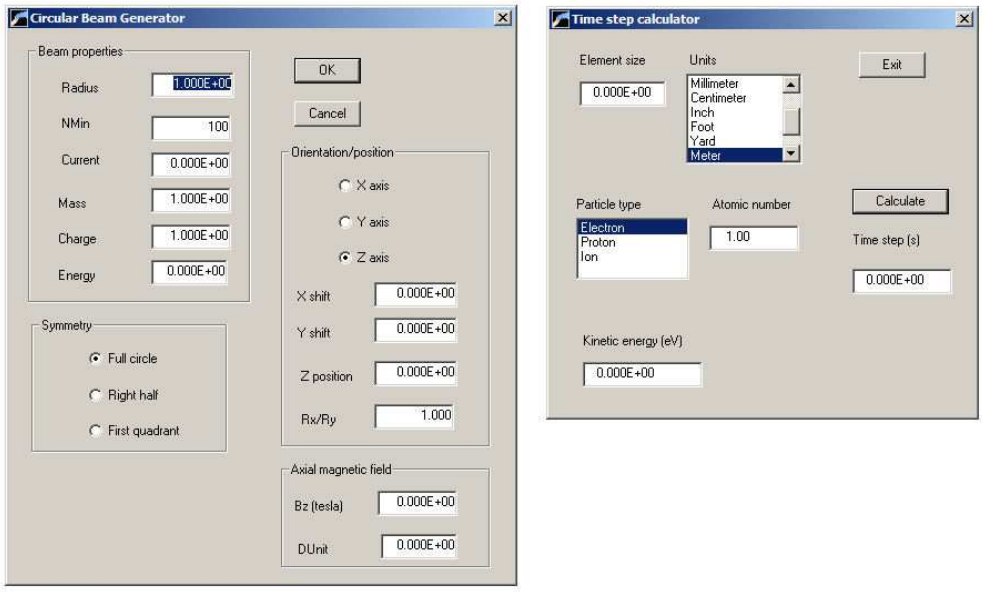


Figure 13: Utilities in the *Tools* menu. Left: circular-beam generator. Right: time-step calculator.

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 OmniOmniTrak 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. This option is usually not necessary on a dual-processor machine. **OmniTrak** will ocuppy the resources of one processor, leaving the second available for other activities. You can utilize the full resources of such a machine by launching two instances of **OmniTrak**.

### STOP RUN

This command terminates the program. **OmniTrak** will attempt to save all available information.

The *Help* menu has a single command.

### INSTRUCTIONS

Displays this manual in your default PDF viewer. The file `omnitrak.pdf` must be in the same directory as `omnitrak.exe`.

## 4.2 Charged-particle beam tools

This menu contains utilities that are helpful for run preparation.

## CIRCULAR BEAM GENERATOR

This command brings up the dialog on the left-hand side of Fig. 13 to create a `PRT` file that describes a laminar, circular (or elliptical) beam with approximately uniform current density. The tool is a quick alternative to **GenDist**. Particle are created in the transverse plane in a hexagonal pattern. In the *Beam properties* group, specify the following quantities:

- The envelope radius $R$. Use dimensions consistent with the value of $DUnit$ specified in the **OmniTrak** script.
- The minimum number of particles, $NMin$. The exact number depends on how the program fills out the hexagonal pattern to approximate a circle.
- The total beam current in amperes.
- The particle type. As in the **PList** command, *Charge* is given in fundamental charge units and the *Mass* in AMU. **OmniTrak** sets the mass of the electron if $Mass = 0.0$.
- The energy in eV.

The *Symmetry* group determines whether the beam fills the entire transverse plane or a subsection. The *Orientation/position* group sets the direction and position of the beam. The options for direction are the $x$, $y$ or $z$ axes. You can include a displacement in the transverse plane and set the position along the main axis. The parameter $Rx/Ry$ determines the ellipticity of the beam distribution. The default value of $Rx/Ry = 1.0$ gives a circular beam.

The *Axial magnetic field* group is used primarily for electron-beam transport in a solenoid field. If the beam has been created in a field-free region (non-immersed cathode) and then injected into the field, this command adjusts the momentum fractions so that all electrons have zero canonical angular momentum. The results are computed from the radial position of particle in meters, so you must supply the value of the conversion factor $DUnit$ that appears in the **OmniTrak** script.

## RECTANGULAR BEAM GENERATOR

This command brings up a dialog to define a beam with a rectangular cross section. Particles are uniformly distributed in the transverse directions and carry equal current.

## TIME-STEP CALCULATOR

In response to this command, **OmniTrak** displays the dialog on the right-hand side of Fig. 13. The function is to calculate values for the $Dt$ command. Enter a value for the minimum element size and choose the appropriate length unit. Enter the particle parameters including maximum kinetic energy. The *Atomic number* value is used only for the particle type *Ions*. When you click the *Calculate* button, the program displays an appropriate value for $Dt$.

## 4.3 Command line operation and batch files

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 `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.OLS` and the plot file `SWITCH.OOU`. During lengthy runs you may perform other tasks in Windows.

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 `SWRUN.BAT` in the data directory using a text editor.

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

The option `/WAIT` ensure that a process ends before the next one starts. Type

```
SWRUN <Enter>
```

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

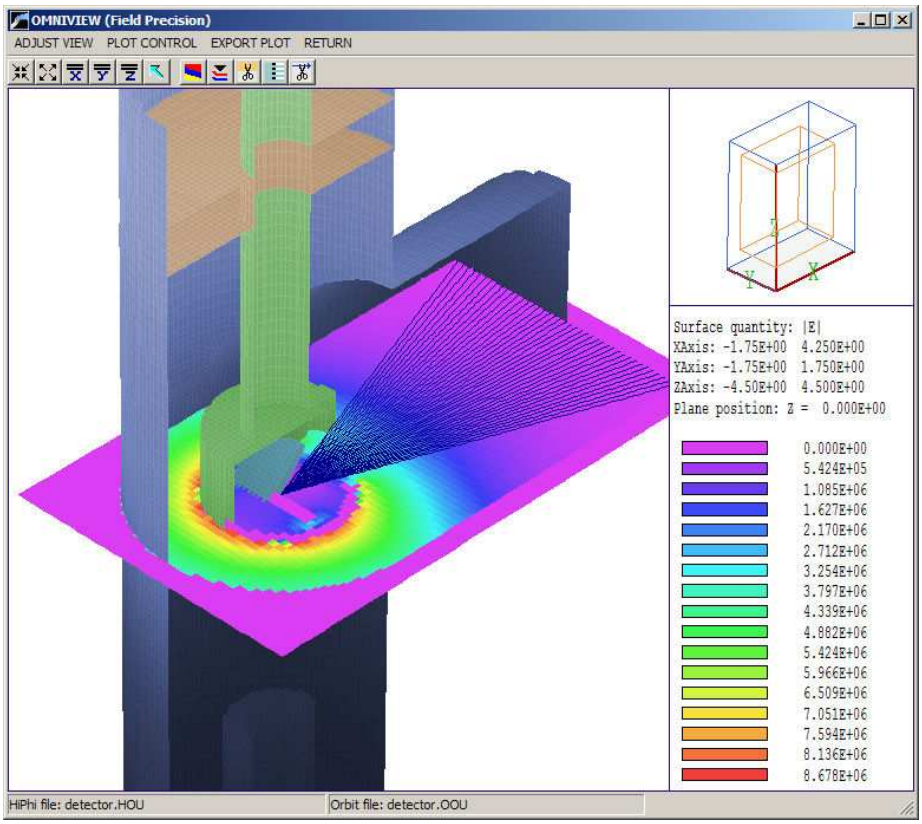


Figure 14: Working environment of the **OmniView** postprocessor - 3D orbit plot with region surfaces in the electrostatic mesh.

# 5 OmniView

## 5.1 File operations

**OmniView** uses orbit and field data files to generate screen and hardcopy plots. The program has the following popup menus: *File operations*, *2D plots*, *3D plots* and *Help*. When the program starts, only the *File operations* and *Help* menus are active. To create plots, you must load one or more field data files. This section reviews options in the *File operations* menu.

### LOAD HIPHI SOLUTION FILE

With this command you can load region-boundary and field information from any **HiPhi** electrostatic field solution. For meaningful plots, you must ensure that the field solution is the one used to calculate superimposed orbits. The dialog shows a list of solution files with names of the form `EPREFIX.HOU`. Changing the directory in the dialog changes the program working directory. Pick an available file and click *OK*. If data retrieval is successful, the *2D plot* and *3D plot* menus become active. At any time, the program can accommodate information from one source: an electrostatic or a magnetostatic solution or a field table.

### LOAD MAGNUM SOLUTION FILE

This command loads region-boundary and field information from solution files generated by **Magnum** (version 2.0 – 2.8). The dialog shows a list of solution files with names of the form `BPREFIX.GOU`.

### LOAD FIELD TABLE

This command loads field information from a table of field values (described in Sect.17). The dialog shows a list of solution files with names of the form `FPREFIX.MTX`. Because the files contain no information on region geometries, the *Region plot* type in 2D plots and region display in 3D plots are inactive.

### 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. The dialog shows a list of **OmniTrak** plot files with names of the form `PPREFIX.OOU`. Changing the directory in the dialog changes the program working directory. Pick an available file and click *OK*. The program loads the solution and updates the menu and status bar.

**OmniView** sets default spatial limits in slice plots from the boundaries of the last file loaded. If you want to plot orbits that extend outside the field solution volume (*i.e.*, the ballistic mode discussed in Sect. 6.3), then load the trajectory file *after* the field file.

### CLOSE PARTICLE FILE

Close the current particle file to load new orbit data.

### CLOSE FIELD PARTICLE FILE

Close the current field file to load a new field solution or table.

### CLOSE FIELD/PARTICLE FILES

Close both the field and particle files to load new data.

### ORBIT FILE INFORMATION
### FIELD FILE INFORMATION
### FIELD MAP INFORMATION

Show information on the data files (*e.g.*, boundaries, number of orbits,...).

### EDIT FILE

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

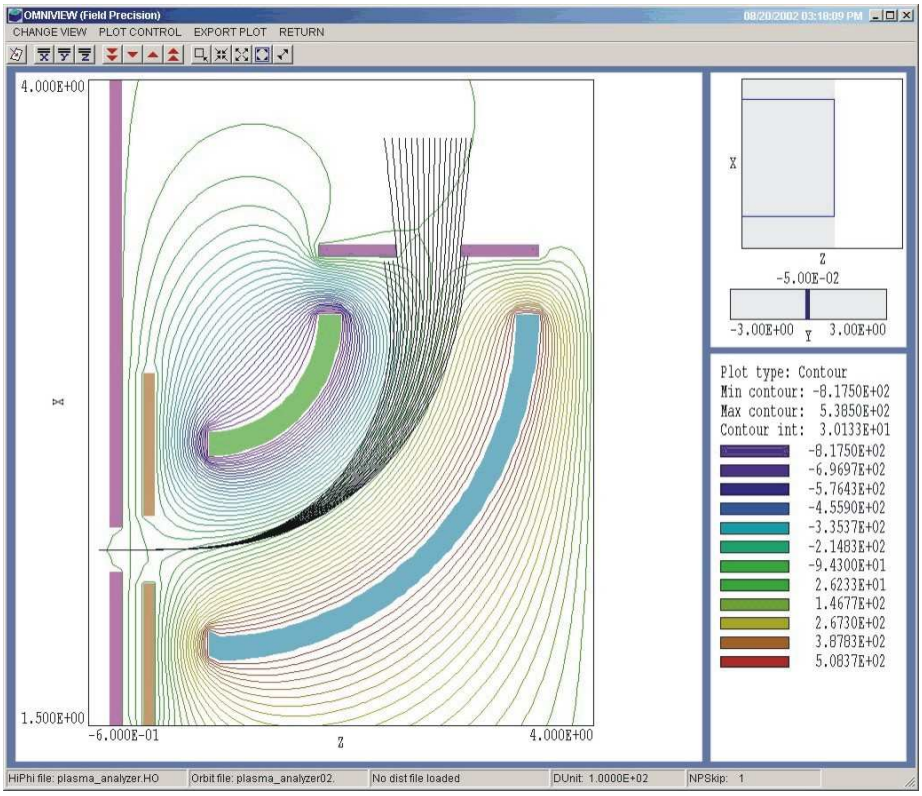


Figure 15: Working environment of the **OmniView** postprocessor - 2D projected orbit plot with electrostatic equipotential lines.

## 5.2 2D plots

The *2D plot* menu is active if a data file has been loaded. Two-dimensional plots (Fig. 15) show the variation of field quantities over a plane normal to one of the Cartesian axes. When a field solution is loaded, 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. When a field table is loaded, plots are constructed in node planes. It is important to note that orbit traces in plots are projections to the slice plane, independent of their location along the normal axis. You can use the *Orbit cut planes* command to restrict orbit vectors to the vicinity of the plot plane.

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

### SET SLICE PLANE PROPERTIES

This command calls up a dialog where 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 normal 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. 15) shows the present location.

### ZOOM WINDOW

As an alternative to the entries in the *Set slice plane properties* 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 *F1* 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. 15) 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.

### 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* 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. There are four available plot types. The *Region* plot style is a cross-section view of the mesh elements color-coded by region. This plot type is available only when **HiPhi** or **Magnum** solutions have been loaded. The *Filled contour* plot makes an attractive display for slices with extended field regions. It shows discrete bands of color coding according to values of the current plot quantity. The *Contour* style shows lines of constant values of the plot quantity. Finally, an *Element* plot has color coding by the average value of the plot quantity in the element volume. The element plot is best in a slice with extensive region boundaries. 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 different dielectric regions when **HiPhi** and **Magnum** solutions are loaded. When *Element outline* is checked, **OmniView** shows element boundaries in *Region* and *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 QUANTITY
Choose the quantity for plot color coding and contour lines. The choices for **HiPhi** solutions are $\phi$, $|\mathbf{E}|$, $E_x$, $E_y$ and $E_z$. The quantities for **Magnum** solutions are $|\mathbf{B}|$, $B_x$, $B_y$ and $B_z$. When a field table is loaded, the choices $|\mathbf{F}|$, $F_x$, $F_y$ and $F_z$ may refer to either electric or magnetic fields, depending on the type of table.

### PLOT LIMITS
In the autoscale mode, **OmniView** chooses defaults for the minimum and maximum values of the plotted quantity. Deactivate *Autoscale* to set the values manually.

### NUMBER OF CONTOUR LINES
Change the number of lines for filled contour and contour line plots.

### ORBIT CUT PLANES

This command brings up a dialog where you can restrict the spatial limits for orbit plotting. You can use the feature to improve the appearance of two-dimensional plots. One application is to restrict plotted orbits to a region along the normal axis near the plot plane.

### ORBIT FILTERS

This command brings up a dialog where you can set orbit plotting according to particle properties. There are three filters on the left hand side where you can restrict orbits plotted by current, mass or charge. (The first option functions only if the present plot file was created in the *SCharge*, *RelBeam* or *FEmit* mode.) As an example, suppose the simulation contains counterflowing electrons and ions, and that we want to plot ions in blue and electrons in red. For Filter 1, set $Mass_{min} = 0.9$. Activate Filter 2 and set $Mass_{max} = 0.9$. The $NPSkip$ field on the right-hand side is useful when a run includes a large number of particles that would make the plot confusing. The value of the parameter *NPSkip* controls the fraction of plotted orbits. All orbits are included when $NPSkip = 1$; every tenth orbit is included when $NPSkip = 10$. The default value depends on the number of particle in the plot file. The present value of $NPSkip$ is displayed in the status bar. Finally, you can restrict the range of plotted particles with the $NPMin$ and $NPMax$ fields. You can use information in the listing file (`OLS`) to determine the indices of different classes of particles.

### TOGGLE SNAP MODE

Mouse coordinates for commands such as *Zoom window*, *Pan*, and *Scan in slice* may 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 present value is displayed in the status bar.

### TOGGLE GRID

A set of dashed grid lines may be superimposed on slice plots. **OmniView** 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). The grid intervals are listed in the information window. 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.

The command brings up a 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.

#### RESET PLOTS ON LOAD

In the default mode, **OmniTrak** automatically resets plot parameters (current view, plot type, magnification mode,..) when you load new orbit or field files. Sometimes, you may want to compare a series of similar runs maintaining a specific view or magnification limits. In this case, use this command to deactivate autoscaling. The current setting (*Autoset plot parameters* or *Fixed plot parameters*) is shown on the right-hand side of the status bar.

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.

#### SAVE PLOT FILE

Use this command to create a graphics file of the current plot in either Windows Bitmap (BMP) or Portable Network Graphics (PNG) formats. In the dialog, specify the format, the size in pixels and the file prefix. The graphics file is created in the current directory.

#### COPY TO CLIPBOARD

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

### 5.3 3D plots

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

The method to control the three-dimensional display with the mouse is identical to that used in **MetaMesh**. Figure 16 shows the active areas of the screen. The central zone ($A$) is used for zooming in (left button) and out (right button). Hold down the left mouse button in zones $B, C, D$ and $E$ to walk around the object. Hold down the right mouse button in zones $B, C, D$ and $E$ to move the viewpoint to the right, upward, to the left and downward. Note that changes are reflected in the orientation box in the upper-right portion of the screen. The plot is updated when you release the mouse button. To set specific view parameters, you

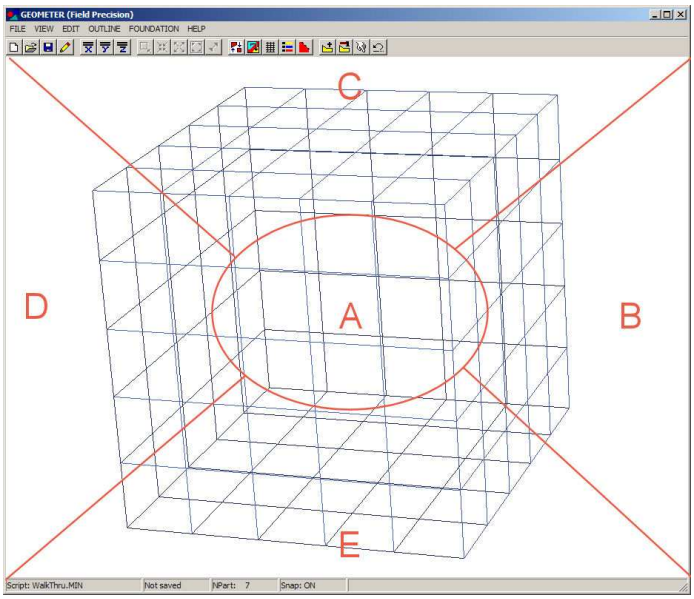


Figure 16: Active areas for mouse control of the 3D view.

can use the *Set surface view* command in the *Adjust view* menu. The command calls up the dialog where you can set the view parameters. The quantity $DView$ controls the distance from the average position of displayed elements to the viewpoint. Decrease $DView$ to increase the 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* menu control the appearance of the plot.

## PLOT STYLE

This command brings up the dialog of Fig. 17 that controls the appearance of the plot. 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. Check the *Add plane plot* box to include a slice plane. In this case, color coding is determined from the current plot quantity. The *Resolution* field determines the number of calculated points along the axes in the slice plane. Higher numbers give better resolution with slower regeneration tinme. The default is $150 \times 150$ points. The commands of the *Surface plot style* and *Surface plot display* groups control the appearance of the three-dimensional region boundaries.

## PLOT QUANTITY

Pick a quantity to control color coding in the slice plane. The choices for **HiPhi** solutions are $\phi$, $|\mathbf{E}|$, $E_x$, $E_y$ and $E_z$. The quantities for **Magnum** solutions are $|\mathbf{B}|$, $B_x$, $B_y$ and $B_z$. When a field table is loaded, the choices $|\mathbf{F}|$, $F_x$, $F_y$ and $F_z$ may refer to either electric or magnetic fields, depending on the type of table.

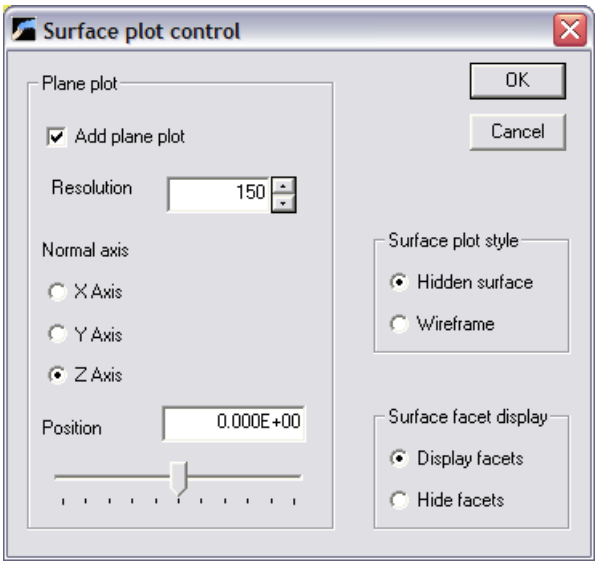


Figure 17: Three-dimensional plot style dialog.

## PLOT LIMITS

In the autoscale mode, **OmniView** chooses defaults for the minimum and maximum values of the plotted quantity. Deactivate *Autoscale* to set the values manually.

## DISPLAYED REGIONS

The command brings up a dialog where you can pick the region boundaries that will be included in the plot. It functions when **HiPhi** or **Magnum** solutions are loaded. Region numbers correspond to those defined in **MetaMesh**.

## REGION CUT PLANES

In a hidden surface plot, internal details may be obscured by surrounding parts or regions. This command brings up a dialog where you can 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 (*i.e.*, all facets included).

## ORBIT CUT PLANES

## ORBIT FILTERS

The functions of these commands were described in the previous section.

## 5.4 Saving and loading views

The creation of plots for presentations may involve some effort. With the following two commands, you can save all the current view parameters and immediately restore the plot.

### SAVE NAMED VIEW

Save the view parameters for the current plot. Quantities such as the slice axis, slice position and zoom limits are saved for two dimensional plots. Parameters for three-dimensional plots include the viewpoint position, displayed regions and cut planes. The information is stored in a text file in the current directory with a name of the form `FPREFIX.FPV`.

### LOAD NAMED VIEW

Load a view file and refresh the plot. Note that you must be in the appropriate plot menu to retrieve a view. Views of plane plots must be loaded to the Plane Plot Menu.

The file contains the complete set of plot parameters. This excerpt illustrates the format:

```
Program: OmniView
2D/3D: 2D
DisplayBy: Regions
Outline: On
NSlice:     40
SliceAxis: YAxis
PlotType: LogElemUp
XPMin: -1.500000E+00
XPMax:  4.250000E+00
...
```

If a specific solution file is loaded, the plot will be restored exactly. The saved view feature in **OmniView** has two useful features if a different meshes are loaded:

- Dynamic adaptation to different solutions.
- Option for user control of the view parameter set.

Regarding the first feature, there are situations where you want to create consistent views of a set of solutions with different geometries, maintaining a similar appearance. Some plot properties (like the viewpoint rotation matrix) are applicable to any solution, but others (like region cut planes or slice plot limits) depend on the geometry. **OmniView** checks each plot parameter for validity. If a parameter is outside the allowed range for the currently-loaded solution, the program computes an alternative. The goal is to preserve as many features of the view as possible.

You can modify view files with an editor. The order of entries is not rigid. On input, **OmniView** uses a free-form parser. If a parameter is missing, the program simply makes no change from the value current in the program. The implication is that you can modify a saved view to include only elements essential to your application. For example, you could compare a series of assemblies with different sizes, maintaining an orthographic 3D view from the same point in Cartesian space. In this case, the view file would contain only the entries:

```
DView:  1.000000E+37
R11:  8.660253E-01
R12: -5.000002E-01
R13:  0.000000E+00
R21:  1.669031E-01
R22:  2.890846E-01
R23:  9.426408E-01
R31: -4.713208E-01
R32: -8.163510E-01
R33:  3.338061E-01
```

# 6 Electric field commands – Fields section

## 6.1 Loading and modifying HiPhi solutions

This chapter and following ones give in-depth descriptions of available commands in the **Omni-Trak** script. The first step in a calculation 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 [FMult]**
**EFIELD3D = KlyGun.HOU**
This command loads an electric field solution from **HiPhi**. The quantity *FileName* is the full name of the file (*e.g.*, `EFTEST.HOU`). The **HiPhi** file must be available in the working directory. The optional real-number parameter *FMult* is a field scaling factor. Values of electrostatic potential $\phi$ are multiplied by *FMult* when the file is loaded. The default value is $FMult = 1.0$.

**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

$$x_{new} = x_{old} + X_{shift},$$
$$y_{new} = y_{old} + Y_{shift},$$
$$z_{new} = z_{old} + Z_{shift}.$$

Enter the real-number parameters $X_{shift}$, $Y_{shift}$ and $Z_{shift}$ in the units set by the *DUnit* command. The *Shift* 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 $X_{rot}$, $Y_{rot}$ and $Z_{rot}$ 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. The string parameter 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 $Y_{rot}$ is ignored. Rotations are performed

before shifts. Electric field rotations are valid 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.

**CHANGEPOT(RegNo) = PotNew**
**CHANGEPOT(6) = 25000.0**
This command may 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 for different regions. After reading the full *Fields* section, **OmniTrak** performs an iterative field solution using the new values of potential. Commands in the next section control this relaxation process.

## 6.2 Controlling electric field recalculation

**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* and *FEmit* options. The field solution routines are the same as those used in **HiPhi**. Therefore, most of 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.

**RESTARGET E ResTargE**
**RESTARGET(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.

**EBOUND [111111]**
**EBOUND 111100**
This command is used in the if the input electric field is a microscopic solution calculated inside a macroscopic solution using the **HiPhi** *Boundary* command. In such a solution, the nodes on one or more surfaces of the solution volume are treated as fixed-potential points. The *EBound* command signals that **OmniTrak** should not change potential values on specified boundaries from those of the input electric field file. The user should ensure that the boundaries are far enough from the beam volume to give a valid solution. The optional command parameter is a string of six characters, either 0 (zero) or 1 (one). A value of 1 (one) indicates that potential values on the boundary should not be changed. The order of characters in the string is *XDn, XUp, YDn, YUp, ZDn* and *ZUp*. The default value is 111111 (all boundaries fixed).

## 6.3 Program control commands

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

**DUNIT Unit**
**DUNIT = MM**
**DUNIT = 39.37**
This command controls conversion of spatial quantities that appear 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 *Unit* argument may be one of the following strings for common units: `angstrom`, `nanometer`, `micrometer`, `mil`, `mm`, `cm`, `inch`, `foot`, `yard`, `meter`, `kilometer` or `mile`. For custom units, enter a real number equal to the number of mesh units per meter. For example, to enter values in cm for the *Shift* command, set $Unit = 100.0$. In contrast to other *Fields* commands, the order of appearance of *DUnit* is significant. The present value of *Unit* is applied in commands that follow. Multiple *DUnit* commands may appear in the *Fields*, *Particle* 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 (`ONAME.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 $\mathbf{E} = 0.0$ and $\mathbf{B} = 0.0$ 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.

**PARALLEL [NProc]**
**PARALLEL = 4**
This command invokes multithread support for the 64-bit program running on a multi-core machine. When the command appears with no value of $NProc$, **OmniTrak** uses the full resources of the computer during electric field recalculations in the SCharge mode and beam-generated magnetic field calculations in the RelBeam mode. Set a value of $NProc$ to assign a specific number of threads. Because **OmniTrak** operations (such as particle tracking) must be performed serially, the time reduction from parallel operation is typically 30%. If the command does not appear, the operating system assigns the program to a single thread, leaving the others free for other tasks. Do not employ parallel processing if you are running multiple instances of **OmniTrak**. The command has no effect on the 32-bit program version.

## 6.4 Loading electric field information from a table

As an alternative to electric field solutions from **HiPhi**, **OmniTrak** can read electric field information from a text file produced by **PhiView** or by another finite-element program. Section 17.1 reviews the file format.

**ETABLE3D = TabName [FMult, EMax]**
**ETABLE3D = ElecLens.DAT 100.0**
Load values from an electric field table. The quantity $TabName$ is the full name of a data file available in the current directory. The optional real-number parameter $FMult$ is a field scaling factor. Values of the electric field are multiplied by $FMult$ when the file is loaded. The quantity may also be used to covert units if the electric field values are not in the standard units of V/m. The default value is $FMult = 1.0$.

The following rules apply to tabular electric fields 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 table contains no conformal mesh information. Consequently, **OmniTrak** does not detect mesh regions along the particle orbit. Stopping conditions based on region properties do not function.
- Without mesh information, the electric field cannot be recalculated. Therefore, tabular electric fields may be used only in the *Track* and *FLine* modes.
- Tabular fields may be translated and oriented with the *Shift E* and *Rotate E* commands.

- 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 $\mathbf{E} = 0$.

- If a modulation function is defined, the temporal variation applies to the components of electric field calculated from a table.

- 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 not accurate near conducting or dielectric boundaries. Section 17 gives a detailed discussion of applications of electric field tables.

# 7 Applied magnetic fields – Fields section

## 7.1 Uniform field components

Magnetic fields within the computational 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 12 describes commands for magnetic field recalculation.

In most applications only applied magnetic fields are important, 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 may be combined with either two-dimensional or three-dimensional magnetic solutions. The default values are $B_{x0} = 0.0$, $B_{y0} = 0.0$ and $B_{z0} = 0.0$.

## 7.2 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 **PerMag**, the **TriComp** magnetostatic program. Note that you can load either a two-dimensional or three-dimensional magnetic field solution, but not both.

**BFIELD2D = FileName [FMult]**
**BFIELD2D = HelmHoltz.POU 0.75**
This command loads a magnetic field solution from **PerMag**. The quantity *FileName* is the full name of the file. The solution file must be available in the working directory. The optional real-number parameter $FMult$ is a field scaling factor. Values of $A_z$ or $rA_\theta$ are multiplied by $FMult$ when the file is loaded. The default value is $FMult = 1.0$.

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 may 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 *Rotate* 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.

## 7.3 Loading and modifying Magnum solutions

This section introduces commands to load and to modify three-dimensional magnetic field solutions from **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 file must be available in the working directory. The optional real-number parameter *FMult* is a field scaling factor. Values of applied field ($\mathbf{H}_s$), reduced potential $\phi$ and dual potential $\psi$ are multiplied by $FMult$ when the file is loaded. The default value is $FMult = 1.0$.

**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

$$x_{new} = x_{old} + X_{shift}.$$
$$y_{new} = y_{old} + Y_{shift}.$$
$$z_{new} = z_{old} + Z_{shift}.$$

Enter the real-number parameters $X_{shift}$, $Y_{shift}$ and $Z_{shift}$ 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 mzgnetic field mesh or table. The parameters $X_{rot}$, $Y_{rot}$ and $Z_{rot}$ 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 $Y_{rot}$ is ignored. Rotations are performed before shifts.

## 7.4 Loading magnetic field information from a table

In addition to magnetic field information from **Magnum**, **OmniTrak** can load field information from a text file with values of **B** tabulated on a regular mesh. With this feature, you can track orbits in magnetic fields calculated from other programs. Section 17.1 reviews the file format.

**BTABLE3D = TableName [FMultBT3D]**
**BTABLE3D = HelmHoltz.DAT**
Load a magnetic field table.The quantity *TableName* is the full name of a data file available in the current directory. The optional real-number parameter $FMult$ is a field scaling factor. Values of the magnetic field are multiplied by $FMult$ when the file is loaded. The quantity may also be used to covert units if the magnetic field values are not in the standard units of tesla. For example, use $FMult = 1.0 \times 10^{-4}$ if the field values are listed 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, but not both.
- The field table 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 may be translated and oriented with the *Shift B* and *Rotate B* commands.
- 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 $\mathbf{B} = 0$.

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. Section 17.2 gives a detailed discussion of applications of magnetic field tables.

# 8 Temporal modulations – Fields section

**OmniTrak** can add arbitrary time variations to electric and/or magnetic fields (both 2D and 3D). The modulations apply to field components, whether calculated directly from a solution or a table. Temporal effects are generally 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

$$\Delta t \ll \frac{L\sqrt{\epsilon_r}}{c}. \tag{1}$$

In Eq. 1, $\Delta t$ is the time scale for field variations, $L$ is the approximate dimension of the solution volume, and $\epsilon_r$ 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 may be applied only in the *Track* mode (single-particle tracking). Modulation functions may be defined either numerically (Sect. 8.1) or algebraically (Sect 8.2).

## 8.1 Tabular modulation functions

Tables of values may be used to define arbitrary functions of time. The resulting *modulation functions* may be applied to the electric and/or magnetic fields. A temporal table is a text file that contains up to 256 data lines. Each line has two real numbers in any valid format:

```
Time   ModFunc
```

The following rules apply to temporal tables:

- Time values are in units of seconds. The modulation function is dimensionless.
- Entries should be in ascending order of time. The time interval between values need not be uniform.
- 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 $\tau$ exceeds $t_{max}$, **OmniTrak** restarts the table and returns the value $MFunc_E(t')$, where $t' = \text{MODULUS}(\tau, t_{max})$. This feature facilitates the representation of periodic functions.
- **OmniTrak** uses a cubic spline interpolation, so the table values should approximate a smooth function of time with continuous first derivative.
- The file must end with the *EndFile* command. You may place any amount of documenting text after the command.

The flexible table format makes it easy to import digitized data or values generated by a spreadsheet.

To illustrate, consider a pulse with the following variation in the range $0.0 \leq t \leq 2\Delta t$, where the width at half-height is $\Delta t = 0.01$ s:

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

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 of 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
```

**MODFUNC E TABLE TableName**
**MODFUNC(E,TABLE) 50NSPULSE.DAT**
Load the table contained in file `TABLENAME` and use the value to compute a modulation function $M_e(t)$. The file must be available in the working directory. Electric field values are multiplied by $M_e(t)$ over the transit time of each particle. The command functions only in the *Track* mode.

**MODFUNC B TABLE TableName**
**MODFUNC(B,TABLE) = 50NSPULSE.DAT**
Load the table contained in file `TABLENAME` and use the value to compute a modulation function $M_b(t)$. Magnetic field values are multiplied by $M_b(t)$ over the transit time of each particle. Note that the modulation function is applied to the total magnetic field, including uniform field contributions.

## 8.2 Algebraic modulation functions

**OmniTrak** features a flexible and robust algebraic interpreter to define temporal functions. An algebraic function is a string (up to 230 characters) that may include the following entities:

- The time as a variable: `$t`. The dollar sign is required so that the parser can differentiate the time from the letter $t$ in functions like `tan`.

- Real and/or integer numbers in any valid format (*e.g.*, `3.1415`, `476`, `1.367E23`, `6.25E-02`, `8.92E+04`,... ). Integers are converted to real numbers for evaluation.
- Binary operations: `+` (addition), `-` (subtraction), `*` (multiplication), `/` (division) and $\wedge$ (exponentiation).
- Functions: `abs` (absolute value), `sin` (sine), `cos` (cosine), `tan` (tangent), `ln` (normal logarithm), `log` (base 10 logorithm), `exp` (normal exponent) and `sqt` (square root).
- Up to 20 sets of parentheses to any depth.
- Any number of spaces as delimiters.

The parser conforms to the standard algebraic rules and features comprehensive error checking. Errors may include unbalanced parentheses, unrecognized characters and sequential binary operations. As an example, the expression

```
1 - exp(-1.0*($t^2)/24))
```

corresponds to

$$1 - \exp\left[-\left(\frac{t^2}{24}\right)\right]. \tag{3}$$

**MODFUNC E > Me(t)**
**MODFUNC(E) > 5.0*cos(($t-1.5E-6/5.0E-6)**
Enter a temporal modulation function as an algebraic expression. The portion of the line string following the > sign defines the modulation function $M_e(t)$. Electric field values are multiplied by $M_e(t)$ over the transit time of each particle.

**MODFUNC B > Mb(t)**
**MODFUNC(B) > 0.5*(1.0 - cos($t/6.0E-9))**
Enter an temporal modulation function $M_b(t)$ for the total magnetic field.

Some applications may require a modulation function that can best be expressed as a product of an algebraic function and a tabular function. For example, the electric field in the acceleration cavity of a circular accelerator may have a sinusoidal variation with a frequency sweep (algebraic function). The sine may have a long-term change of amplitude (tabular function). You can implement such a variation by including two `MODFUNC E` or `MODFUNC B` commands in the script. One command should specify an algebraic function and the other should specify a table. In this case, the total modulation function will be a product of the two.

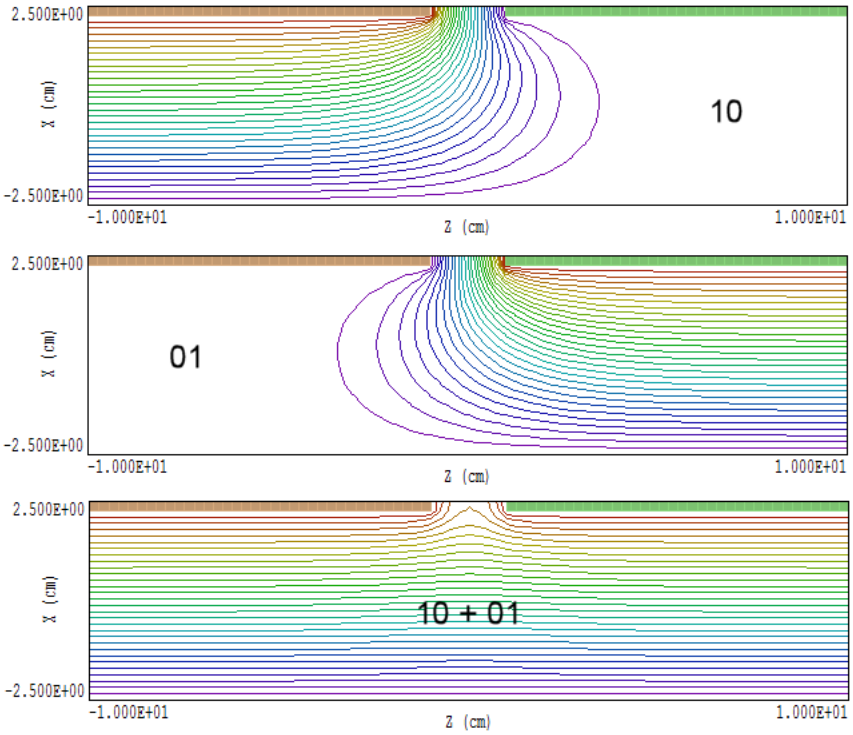


Figure 18: Equipotential contours for the base functions of a simple three-electrode system, ground reference at the bottom. Electrode 1 at top left, electrode 2 at top right. The lower illustration shows the sum of the base solutions.

## 8.3 Electric-field modulation with two voltage levels

This section describes an advanced capability of **OmniTrak** developed to study RF ion extractors that may include an electrode with a DC bias. This situation is fundamentally different from the ones discussed in the previous sections. In applying a simple modulation, the assumption was that the spatial field distribution was self-similar. The only change with time was the normalization. This type of field solution involves a reference potential (ground) and one independent value of the time-varying voltage.

Biased electrodes introduce the possibility of two (or more) independent time-varying voltages. In this case, the spatial distribution of the field changes at each instant. Clearly, it would be impractical to solve the three-dimensional electric-field at each particle integration step. Fortunately, there is a simple way to find the time-varying field with two independent voltages. We can understand the method with reference to the field solution illustrated in Fig. 18. The simple three-electrode system has a ground plane at the bottom. Independent voltages may be applied to the electrode at the top left ($V_1$) and right ($V_2$). We can construct two normalized base solutions: solution $S_{10}$ has $V_1 = 1.0$ V and $V_2 = 0.0$ V while solution $S_{01}$ has $V_1 = 0.0$ V and $V_2 = 1.0$ V. In the absence of nonlinear materials, the set of all possible electrostatic solutions with arbitrary voltages $V_1$ and $V_2$ may be generated with a linear combination of the base functions:

$$S = V_1\ S_{10} + V_2\ S_{01}. \tag{4}$$

To illustrate, the bottom solution in Fig. 18 is the sum of the base functions. The same solution would follow from a direct **HiPhi** solution with $V_1 = V_2 = 1.0$ V.

The following procedure is used for modulated dual-voltage solutions in **OmniTrak**:

1. Prepare two base electrostatic solutions with **HiPhi** using the same mesh. One or more reference electrodes in both solutions are grounded ($\phi = 0.0$ V). In the first solution, one electrode (or set of electrodes) has $V_1 = 1.0$ and the other has $V_2 = 0.0$ V. The voltages levels are reversed in the second solution.

2. Load the first solution with the `EFIELD3D` command. The program reads the mesh characteristics as well as the potential values at all nodes, $\phi_1(i, j, k)$. A normalization factor may be included in the command to set the amplitude $V_1$ in solution $S_{10}$.

3. Load the second solution with the `EFIELD3D2` command. In this case, **OmniTrak** checks that the mesh is identical to the previous one and stores the potential values $\phi_2(i, j, k)$. Again, a normalization factor may be used to set the amplitude $V_2$ for $S_{01}$.

4. The command `MODFUNC E` is used to define a modulation function $M_1$ for $S_{10}$, while the command `MODFUNC E2` sets the modulation function $M_2$ for $S_{01}$. Following the discussion of the previous sections, there may be two instances of the `MODFUNC` command for each solution to define both an algebraic variation and an envelope function given by a table.

During orbit tracking, the program uses electrostatic potential values of the form

$$\phi = M_1(t)S_{10} + M_2(t)S_{01}, \tag{5}$$

as input to the electric field interpolation routine. Because much of the computational work involves identification of the element occupied by the particle, there is little change in the orbit tracking time penalty for a dual-voltage solution.

There are two new commands to implement dual-voltage simulations in the *Track* mode:

**EFIELD3D2 FileName [FMult]**
**EFIELD3D2 = Extractor02.HOU 5000.0**
Load potential values from the second base solution. The string *FileName* is the full name of a **HiPhi** output file. If the optional real-number parameter *FMult* appears, potential values are multiplied by the factor when they are loaded (default, $FMult = 1.0$). This command must occur after the `EFIELD3D` command in the script. If the commands `SHIFT` and/or `ROTATE` appear, the operations are applied to both base solutions.

**MODFUNC2 E TABLE TableName**
**MODFUNC2 E > Me(t)**
Set a modulation function for the second base solution from a table, an algebraic function or both.

The example of Fig. 18 illustrates the procedure. The mesh is described by the file `THREEELECT.MIN`. The **HiPhi** input files `THREEELECT01.HIN` and `THREEELECT02.HIN` generate the two normalized base solutions. The *Field* section of the **OmniTrak** script `THREEELECT.OIN` includes the following commands:

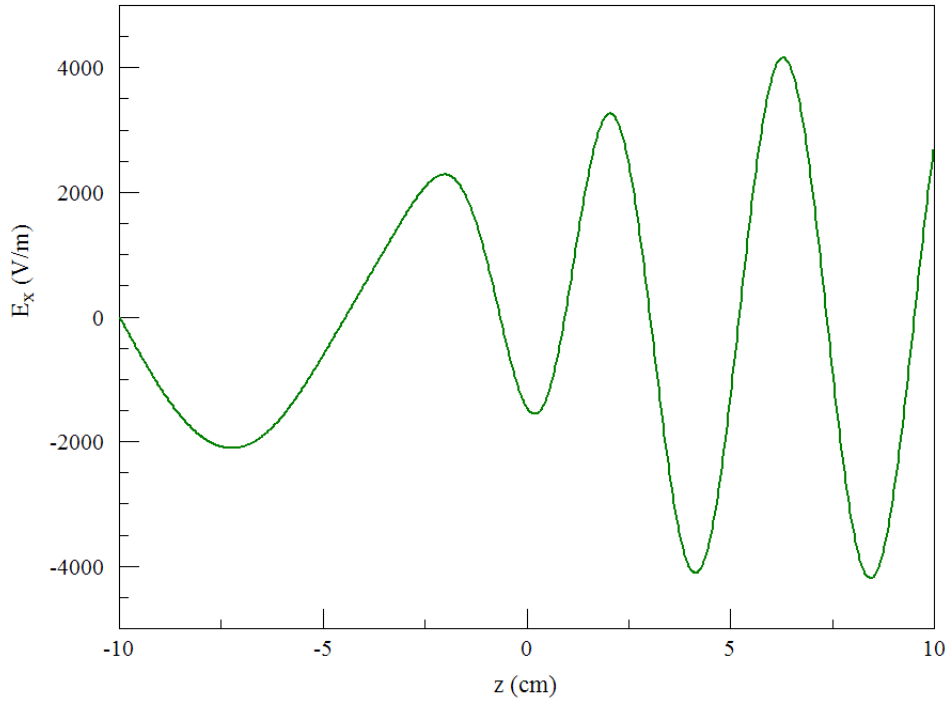


Figure 19: Electric field $E_x$ as a function of position $z$ experienced by a proton moving from the left to the right in the test solution with velocity $v_z \cong 2.2 \times 10^5$ m/s.

```
EFIELD3D: ThreeElect01.HOU 100.0
MODFUNC(E) > sin(1.2566E7*$t)
EFIELD3D2: ThreeElect02.HOU 200.0
MODFUNC(E2) > sin(3.1416E7*$t)
```

The left-hand electrode has a sinusoidal variation at 2.0 MHz with amplitude 100 V. The peak electric field far from the gap should be $E_x = -2000$ V/m. The voltage on the right-hand side varies at 5.0 MHz with amplitude 200 V ($E_x = -4000.0$ V/m). A 250 eV proton moving in $z$ is injected near the left-hand boundary. The transit time over the 20.0 cm distance is about 1.0 $\mu$s. The commands

```
LISTON 1
LISTTYPE EField
```

specify an electric-field listing along the particle trajectory. Figure 19 shows the resulting electric field experienced by particle as a function of axial position.

# 9 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 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 program computes contributions of beam current to the three-dimensional magnetic field.
- **FEmit**. This mode is similar to the *SCharge* mode except that the particle current is determined by the Fowler-Nordheim equation.
- **FLine**. In this mode OmniTrak carries out precise calculations of electric or magnetic field lines.

Table 5 summarizes allowed commands in the different tracking modes.

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 9.1 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 9.2 discusses how to define a source surface and how particle orbits are generated from the surface. The material in Sections 9.3, 9.4 and 9.5 applies to all tracking modes. Section 9.3 describes commands to set the integration time step and other control parameters. Stopping and diagnostic planes are described in Section 9.4. The option for accurate stopping conditions is useful for lens characterization or the design of isochronous systems. Section 9.5 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).

## 9.1 Starting orbits from a list

**OmniTrak** follows up to 10,000 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

Table 5: Allowed commands for tracking modes

| **ALL MODES** | **SCHARGE/RELBEAM/FEMIT** |
|---|---|
| STOPPLANE XUp XStopUp | AVGE AvgE |
| DIAGPLANE XUp XStopUp | NCYCLE NCycle |
| DUNIT DUnit | SUPPRESS SFract(1) SFract(2) ... |
| SMAX SMax | SCHARGE MODE |
| LISTON ListSkip | EMIT NReg Mass Chrg DEmit NDiv |
| ORBINFO | JLIMIT JLimit |
| PLOTOFF | DTHETA DTheta |
| MATERIAL E NReg | EDIRECT [Surface, Field] |
| VACUUM E NReg | RELAPPROX |
| NSTEPMAX NStepMax | |
| **ALL PARTICLE MODES** | **RELBEAM** |
| DT Dt | EMIT NReg Mass Chrg DEmit NDiv |
| TMAX TMax | JLIMIT JLimit |
| SECONDARY E NReg Delta0 Eng0 | DTHETA DTheta |
| SECONDPARAM KECutOff MinFact | MU NReg Mu |
| PLIST | BBELEM LElem RElem |
| PFILE PFileName | RELAPPROX |
| LISTTYPE Particle | BBMESH xd xu nx yd yu ny zd zu nz |
| RECORDPLANE X Up Xs1 Xs2 ... | BBSYMPLANE Axis |
| **TRACK** | **FEMIT** |
| EMIT NReg Mass Charge NDiv | EMIT NReg WorkFunc NDiv [Beta] |
| **FLINE** | |
| FLIST | |
| FFIELD | |
| DS | |
| EMIT NReg NDiv [Polarity] | |

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 may 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 **GenDist** utility program. Also, an **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 may 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   ux   uy   uz
mass          energy
```

The quantities in the line have the following meanings.

1. **Rest mass**. The particle rest mass in units of (atomic mass units), where 1.0 AMU = $1.660538782 \times 10^{-27}$ kg. The rest mass of a proton is about 1.0073 AMU. 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, $1.60210 \times 10^{-19}$ 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 = $1.6021 \times 10^{-19}$ joules.

4. **x, y, z**. The initial particle position in meters or units set by *DUnit*.

5. **ux, uy, uz**. 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 $u_x = 1.0$, $u_y = 0.0$, and $u_z = 0.0$. A particle orbit in the $x$-$y$ plane initially inclined at an angle $\alpha$ with respect to the $x$ axis has $u_x = \cos\alpha$ and $u_y = \sin\alpha$. For a particle with zero kinetic energy use $u_x = u_y = u_z = 1.0$.

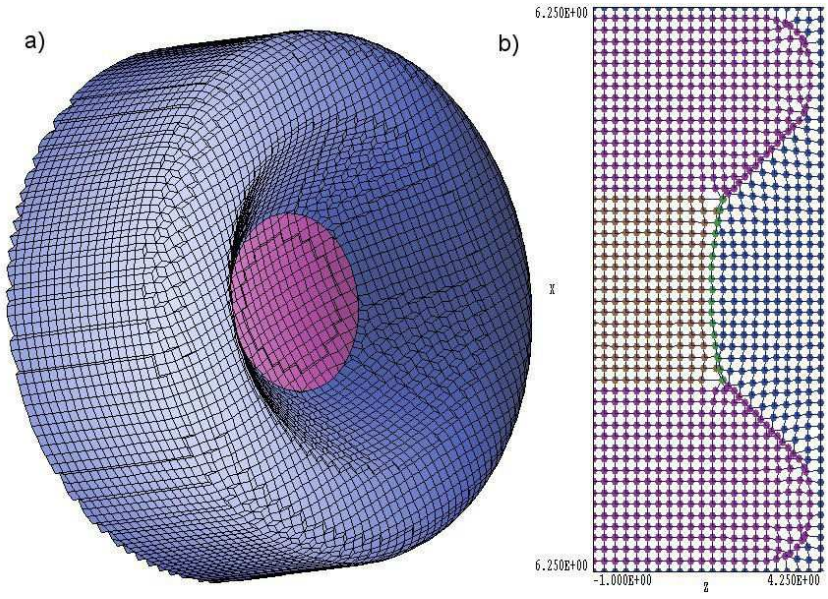


Figure 20: Illustration of the *Coat* operation. a*)* Three dimensional view of facets on region boundaries: region 2 (blue), region 3 (violet). *b*) Cross section in midplane showing node identities: region 1 (blue), region 2(violet), region 3(brown), region 4(green).

Note that line entries may be separated by any of the delimiters listed in Sect. 3.2.

**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 may appear anywhere in the *Particles* section.

**PFILE PPrefix**
**PFILE = TCathode**

This command specifies that **OmniTrak** should read particle parameter lines from an external file with a name of the form `PPREFIX.PRT`. The file must be available in the working directory. A particle file may consist of blank lines, comments and from 1 to 10,000 data lines. The file must terminate with the *End* command.

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

## 9.2 Starting orbits from a source surface

Particles with zero kinetic energy may be generated automatically near special areas on the surfaces of electrodes. Use of the areas, *source surfaces*, requires that a properly-prepared

Table 6: **MetaMesh** script definitions to create the cathode assembly in Fig. 20

```
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
```

electric field file has been loaded with the *EFile3D* 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 include up to 20 source surfaces. Surface marking is performed in **MetaMesh** using the *Coat* command.

As an example, consider the convergent cathode with focusing electrode shown in Fig. 20. 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 **MetaMesh** script commands shown in Table 6. 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. 20*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 (*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 point out of the fixed potential element and into the vacuum element. The facet is divided into $NDiv \times NDiv$ 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**: 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 represents 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 Chrg NDiv**
**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 *Chrg* is the particle charge in units of elementary charge, $1.60210 \times 10^{-19}$ coulombs. The charge of a proton is 1.00 while the electron has a charge -1.00. The quantity $NDiv$ determines the number of particles per facet. Each facet is divide into $NDiv$ sections along the two normal coordinate directions, so that the program creates $NDiv^2$ 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.

## 9.3 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:

1. Search the electric and/or magnetic meshes to find the minimum average element dimension along the $x$, $y$ and $z$ axes.

2. If an electric mesh has been loaded, find the maximum potential difference in the solution space.

3. Find the maximum initial kinetic energy (in eV) for particles.

4. Using the maximum of the potential difference and initial kinetic energy, estimate the maximum velocity that particles can attain in the solution space.

5. 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. This option may 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 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 list 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 contains the keyword *Mass*. Apply the command when runs include particles with different mass (*e.g.*, ions and electrons). Here the value $DtRef$ is a reference time step for particles with a mass equal to 1.0 AMU (atomic mass unit). If a particle has a rest mass $m_p$ (expressed in AMU), then **OmniTrak** picks a time step.

$$Dt = \sqrt{m_p}\ DtRef. \tag{6}$$

For reference, 1 AMU equals $1.6598 \times 10^{-27}$ kg. The electron rest mass is $9.1091 \times 10^{-31}$ 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 `OPREFIX.OLS`. 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 (`OPREFIX.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 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 points 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 `OPREFIX.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 may 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.

**PLOTSKIP NPlotSkip [NPartSkip]**
Trajectories with a large number of steps or distributions with a large number of particles may lead to very large plot files. Use this command to control the size of the `OOU` file. The parameter $NPlotSkip$ is the number of integration steps per plot vector. The default is $NPartPlot = 5$. Use a larger number if the time step is very small (for instance, it a strong magnetic field). Use $NPartPlot = 1$ to make a detailed record of trajectories in the `OOU` file. The optional parameter *NPartSkip* limits the number of trajectories included in the `OOU` file. For example, if $NPartSkip = 5$, only the trajectories for particles 5, 10, 15, ... are included. The default is $NPartSkip = 1$ (all trajectories plotted).

**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 *OrbInfo* to check the error condition.

**DUNIT DUnit**
**DUNIT = 39.37**
This command was described in Sect. 6.3.

## 9.4 Stopping orbits

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 (Sect. 6.3). 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 program enters the *ballistic mode*. Here, the field interpolation routines return $\mathbf{E} = 0$. 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 $t_{max}$. 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 $t_{max}$ parameter may also be used to avoid infinite orbits (such as a electrons trapped in a magnetic field). The default value is $t_{max} = \infty$.

### SMAX SMax
### SMAX 4.567

Orbits stop when their total track length exceeds the quantity $s_{max}$. Enter the real-number value in units set by $DUnit$. **OmniTrak** performs an interpolation over the final integration step to determine particle parameters at the precise termination distance..The $s_{max}$ parameter may also be used to avoid infinite orbits. The default value is $s_{max} = \infty$.

### NSTEPMAX NSMax
### NSTEPMAX = 20000

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

### KELIMIT TMin TMax
### KELIMIT = -1.0 250.0E6

Orbits stop if their kinetic energy is less than $T_{min}$ or greater than $T_{max}$. Enter the values in eV (electron volts). Use a negative value for $T_{min}$ if you need only a maximum limit. Conversely, set $T_{max}$ to a very large value to define only a lower limit. The default condition is no energy limit.

**STOPPLANE POrient PPos**
**STOPPLANE ZUp 99.0**

The *StopPlane* command sets a plane normal 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.

Note that the particle may start on either side of the plane. You may 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 stopping plane, a particle orbit does not terminate when it crosses a diagnostic plane. Instead, **OmniTrak** records interpolated particle parameters (position, momentum, energy, transit time and track length) at the plane in the output particle data arrays. 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 calculated at 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 may include multiple *ReflectPlane* commands in the script.

### RECORDPLANE [UP,DOWN] [X,Y,Z] RP1 RP2 ... RPN
### RECORDPLANE (Up, Z) = 19.0 20.0 21.0

With this command you can record precise particle parameters at a number of positions normal to an axis. There are three differences from the *DiagPlane* command: 1) you can define up to 10 record planes, 2) the information is transferred to individual `PRT` files during the orbit integrations and is not available for operations with commands of the *Diagnostics* section and 3) only a single *RecordPlane* command may appear in the script. The second and third string parameters specify the direction of particle motion and the axis for normal planes. Up to 10 real-number parameters may follow giving the positions along the axis for diagnostics. If there are three entries, **OmniTrak** opens three files with names of the form `RUNNAME01.PRT`, `RUNNAME02.PRT` and `RUNNAME03.PRT` and records values of energy, position and momentum as each particle crosses the corresponding planes. In simulations of the type *SCharge*, *RelBeam* and *FEmit*, data are recorded only on the final cycle. The following rules apply to the *RecordPlane* command:

- The file may contain only one *RecordPlane* command. Therefore, all planes are normal to a single axis.
- The positions $RP1$, $RP2$, ... $RPN$ must appear in order of increasing value for the *Up* option or decreasing value for the *Dn* option.
- The command must contain from 1 to 10 position values.
- For the *Up* option, only particles with starting positions less than $RP1$ will be recorded in any of the files. For the *Dn* option, a particle must start at a position greater than $RP1$ to be recorded.
- The recording process may not function correctly if the spacing between planes is less than the particle integration step size.

### PERIODPLANE POrient PPos [FPrefix] [NPSkip]
### PERIODPLANE YDn 0.0 CYCLO01

This command is useful for beams with periodic orbits in cyclic accelerators (cyclotrons, synchrotrons,...). Like the *DiagPlane*, a *PeriodPlane* does not stop a particle orbit. Instead, it records interpolated orbit parameters each time the particle crosses the plane in the specified direction. Results are recorded as data lines in a particle file in the standard format (Sect. 9.1).

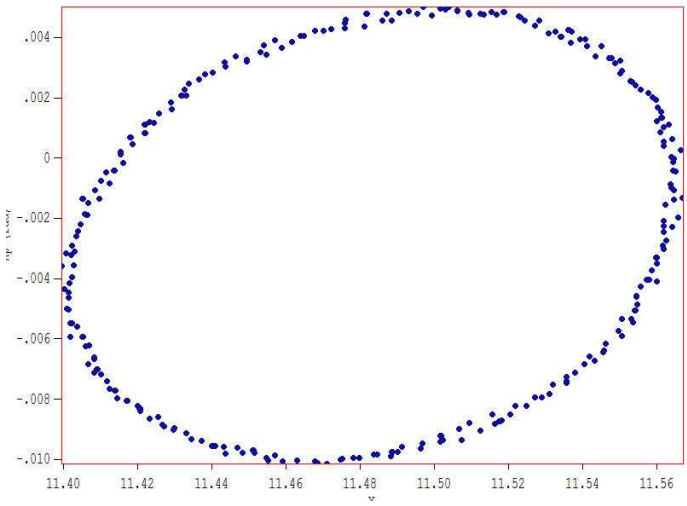


Figure 21: Radial phase-space plot of crossing points for an orbit in the plane of a cyclotron with an azimuthal field asymmetry. The points (created using the *PeriodPlane* command) trace the stationary distribution at the location.

The parameter *POrient* (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 *PPos* is the position of the plane along the normal axis. Enter the value in units set by *DUnit*. The optional string parameter *FPrefix* is the prefix of the file where particle parameters will be recorded. The default file name is `PERIOD.PRT`. To reduce the length of the `PRT` file, you can optionally include the integer parameter *NPSkip*. In this case, records are written at intervals of *NPSkip* turns. The default is $NPSkip = 1$. You can use **GenDist** to analyze particle properties in the plane. As an example, Fig. 21 illustrates generation of a stationary radial phase-space distribution of particles in a cyclotron with an azimuthal field perturbation. The particle completes 382 rotations in the machine. If you include multiple *PeriodPlane* commands, entries for all crossings are recorded in the particle file. In this case, you can limit the analysis to a crossing class by using the filter features of **GenDist**. In the *SCharge* and *RelBeam* modes, crossings are recorded only on the last cycle.

### TABSTOP [E,B] [EMax, BMax]
### TABSTOP(B) = 1.0E6

This command functions only if an electric or magnetic field table has been loaded. The usual spatial stopping conditions do not apply for field interpolations on a table because there is no region information. This command gives an option to set a stopping condition based on the interpolated field values. When you construct the table, you can set large values of $|\mathbf{E}|$ and/or $|\mathbf{B}|$ at nodes to designate regions inside an electrode or pole piece. The orbit terminates if $|\mathbf{E}| \geq E_{max}$ and/or $|\mathbf{B}| \geq B_{max}$.

## 9.5 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 14 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 $RegNo$ 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 **PerMag** is loaded, **OmniTrak** sets 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.

## 9.6 Particle starting times

In *Track* mode runs that include modulations, the default condition is that all particles start at $t = 0.0$ s. You can specify individual particle start times relative to the modulation function by adding a column to particle lists or files. In this case, the data lines have the form:

```
PLIST
*    Mass Chrg   Eng       x    y       z    px    py    pz      ts
*   ============================================================
     0.0  -1.0 0.7399E6  0.1  0.0   -8.0  0.00  0.00  1.00  0.0E-6
     0.0  -1.0 0.7399E6  0.2  0.0   -9.0  0.00  0.00  1.00  0.1E-6
     0.0  -1.0 0.7399E6  0.3  0.0   -9.0  0.00  0.00  1.00  0.2E-6
     0.0  -1.0 0.7399E6  0.4  0.0   -9.0  0.00  0.00  1.00  0.3E-6
     0.0  -1.0 0.7399E6  0.5  0.0   -9.0  0.00  0.00  1.00  0.4E-6
     ...
END
```

Specify the start time in units of seconds. If an entry is missing, **OmniTrak** takes the default $t_s = 0.0$ s. When a start time is given, the modulation function is evaluated during an orbit trace as:

$$f_e(t + t_s), \quad f_b(t + t_s), \tag{7}$$

where $t$ is the elapsed time from the initiation of the particle trace and $t_s$ is the start time. The adjusted time is recorded in particle lists in the `OLS` file and in the `OOU` plot file, Note that any particles that start from an emission surface have $t_s = 0.0$ s.

## 9.7 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 22 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:

$$E_r \cong -\frac{V_0}{r \ \ln(r_o/r_i)}. \tag{8}$$

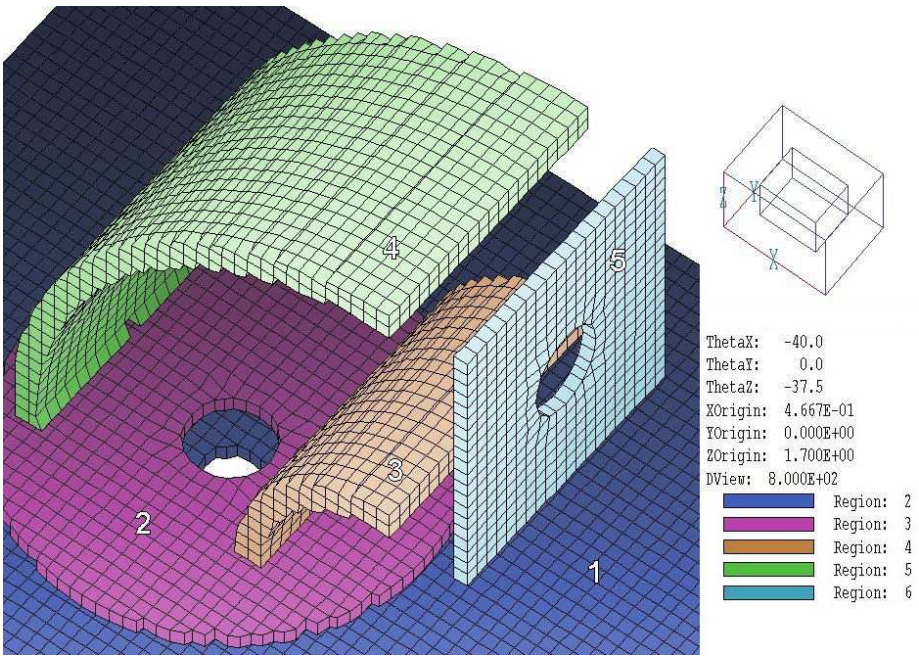


Figure 22: Geometry of the plasma ion energy analyzer. 1) Plasma chamber wall with extraction aperture. 2) Electron suppressor. 3) Inner deflection plate. 4) Outer deflection plate. 5) Detector shield. The assembly is located inside a grounded box (not shown).

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

$$\frac{m_0 v_0^2}{r} = qE_r, \tag{9}$$

or

$$T_o = \frac{qV_0}{2\ \ln(r_o/r_i)}. \tag{10}$$

In the second equation $T_0$ 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 $T_0 = 0.59V_0$. The mesh file `PANALYZER.MIN` defines the geometry shown in Fig. 22. We have employed a fairly coarse mesh so the example runs quickly. The voltage levels in the file `PANALYZER.HIN` are optimized for ions with $T_0 = 1000.0$ 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 23 shows the electrostatic solution. The equipotential surface at $\phi = 25.0$ 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.

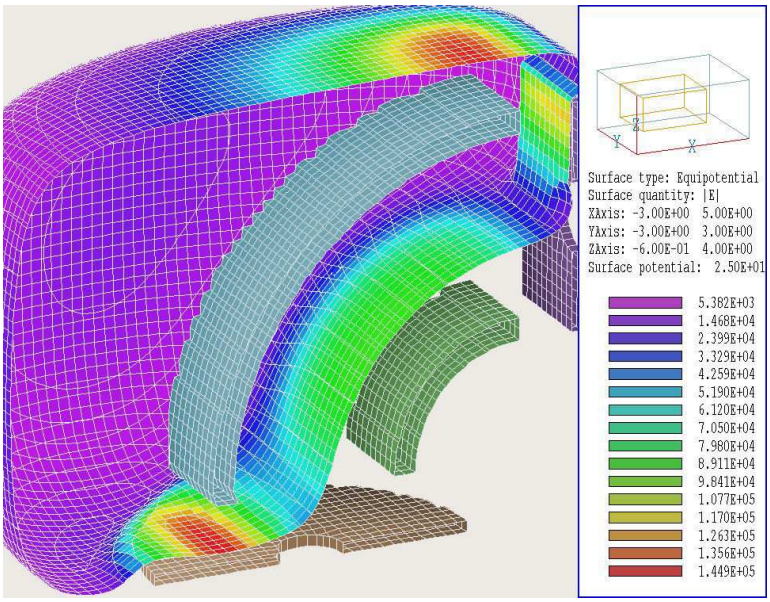


Figure 23: Equipotential surface at $\phi = 25.0$ V, color-coding on the surface according to $|\mathbf{E}|$.

We shall first use **OmniTrak** to calibrate the detector. Using a $C^+$ ion with $T_0 = 1.0$ keV, we seek to adjust the electrode potentials so that the ion passes through the middle of the exit aperture. Table 7 shows the **OmniTrak** input file is used for the run. Consider first the commands of the *Fields* section. The line `DUnit = 100.0` specifies that are coordinates will be entered in units of centimeters. The *EField3D* command loads the electrostatic solution of Fig. 23. 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 $1.3 \times 10^5$ m/s. The time to cross an element of width $10^{-3}$ m is 7.9 ns. The choice $Dt = 5.0$ ns means that particles move less than one element per time step. The next command, `TMax = 1.0E-6`, limits the maximum orbit lifetime. Setting $T_{max}$ 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 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 1.0 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 24*a* shows equipotential lines and the resulting particle orbit projected in the $y = 0.0$ plane. Because the fields are lower than the values for a ideal cylindrical system, the particle is not deflected to the detector axis. Additional runs with *FMult ¿ 1.0* 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

Table 7: File PANALYZER.OIN

```
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
```

to include ions with different energies over the range 800 to 1200 eV. The values in Table 8 were prepared with a spreadsheet and pasted into the script, giving the orbits show in Fig. 24*b*.

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 **GenDist** utility that automatically generates PRT files. We want to create a distribution of particles initially 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.

Table 9 shows the **GenDist** input file. Briefly, the first set of commands defines a beam

Table 8: Expanded *PList* structure for the PANALYZER example

```
* 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  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
```

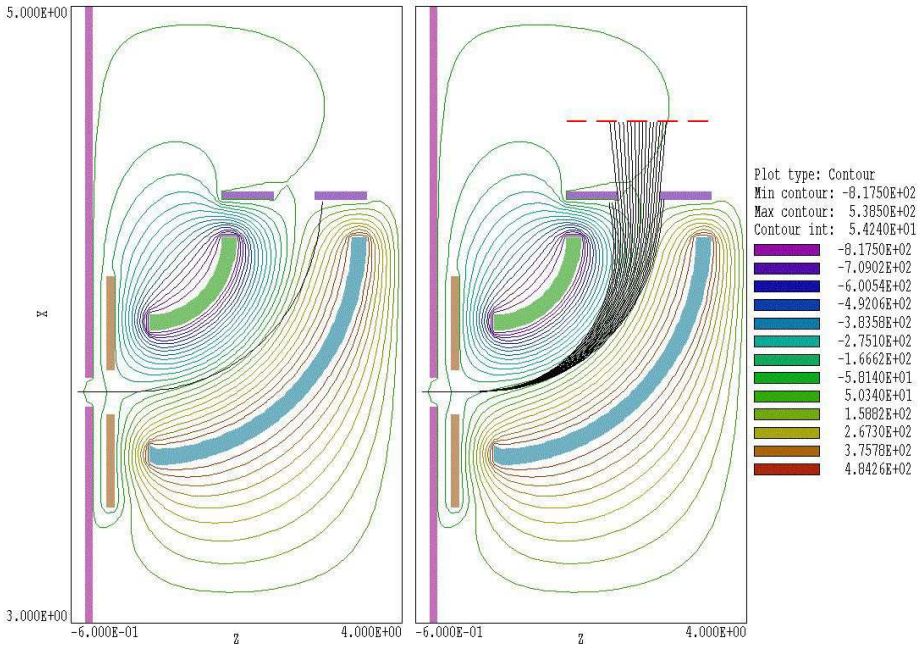


Figure 24: Equipotential lines and projected particle orbits in the plane $y = 0.0$. a) Single $C^+$ ion with $T_0 = 1000.0$ eV. *b*) Multiple orbits over the kinetic energy range 800 – 1200 eV. Dashed red line shows the stopping plane.

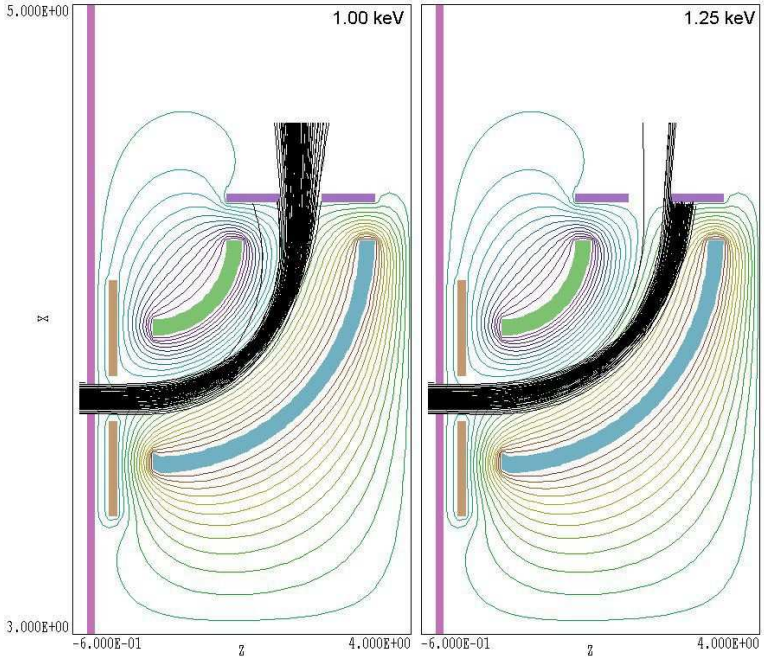


Figure 25: Equipotential lines and orbits projected in the plane $y = 0.0$ for a spatially-extended group of particles with a $\pm 100$ eV energy spread for two values of the central energy.

Table 9: File PANALYZER.DST, input to **GenDist**.

```
* File PANALYZER.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
    ...
   300.0  0.00012
End
EndFile
```

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 **GenDist** structure define an energy distribution with Gaussian weighting. We run the program to create the file PANALYZER.PRT. Figure 25 shows orbits projected in a plane at $y = 0.0$ for two central energies.

# 10 Beam forces in the SCharge mode – Particles section

## 10.1 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 (*e.g.*, focal limits for low-emittance probe beams). Forces resulting from beam charge and current are discussed in the two texts supplied with **OmniTrak** (Sect. 1.1).

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

$$\frac{F_m}{F_e} \cong \beta^2. \tag{11}$$

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 a 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 additional 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 11 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 includes ten entries in the following order:

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

Here, $I$ is the current (in amperes) 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 can 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 perform 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 $\Delta t$. 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.

Statistics determine the granularity of the space-charge density. Large numbers of particles give a more continuous function and smoother variations of electric field.

## 10.2 Control of orbit and field updates

When model particles carry current, 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. 6.2. The following list summarizes the command functions:

**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 (with deposition of space-charge) followed by relaxation iterations to correct the electric field solution. The number of cycles may be small ($NCycle \sim 5$) for low current beams with laminar orbits or large ($NCycle \geq 20$) 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 make large contributions to the electric field. 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 $\Delta Q_i^n$ is an increment of charge determined for the present orbit integration, then the element charge changes according to

$$Q_i^n = (1 - AvgE)\ Q_i^{n-1} + (AvgE)\Delta Q_i^n. \tag{12}$$

A value of $AvgE = 1.0$ means that only the present orbits contribute to the space-charge. A value $AvgE \ll 1.0$ 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 $NCycle \geq 1.0/AvgE$. The default value is $AvgE = 0.30$.

## 10.3 Application example

The benchmark example `SCHARGETEST` illustrates techniques for 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 26 shows the geometry. The solution volume covers an area 1.0 cm × 1.0 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 $\phi = 0.0$ V. The input electric field has $\mathbf{E} = 0$ 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. Table 10 shows a portion of

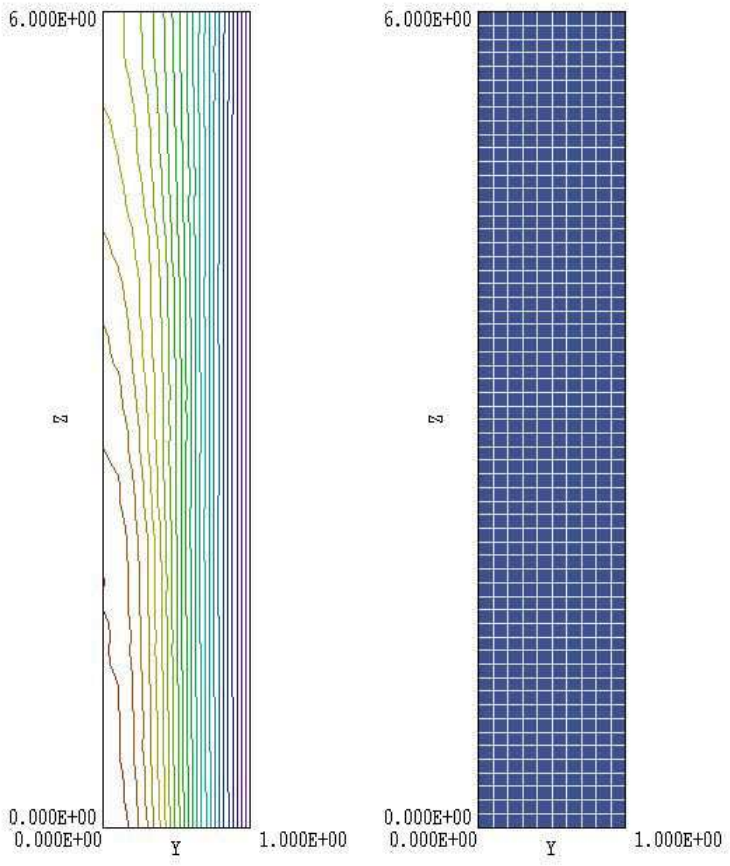


Figure 26: Expansion of a planar beam in vacuum. Left: equipotential contours from 0.0 to 3000 V. Beam waist at the bottom, beam midplane on the left-hand side. Right: mesh projected in $y$-$z$ plane.

the **OmniTrak** script `SCHARGETEST.OIN`. 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 $31.0 \times 10^{-4}$ 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 $J = 31.0$ 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 SCHARGETESTP.HOU` creates an output file of the beam-generated electric field in **HiPhi** format.

Formulas in Sect. 9.1 of **Charged Particle Beams** give an on-axis potential at the beam waist of 3000 V with a potential drop of 1000 V across the beam. The planar generalized perveance is $K_y = 2.0$. The envelope of an ideal sheet beam drifting a distance $L$ is predicted to increase by an amount

$$\Delta y = \frac{K_y L^2}{2}. \tag{13}$$

Despite the coarse mesh, the on-axis potential at z = 0.0 in the simulation is within 1% of

Table 10: File SCHARGETEST.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
```

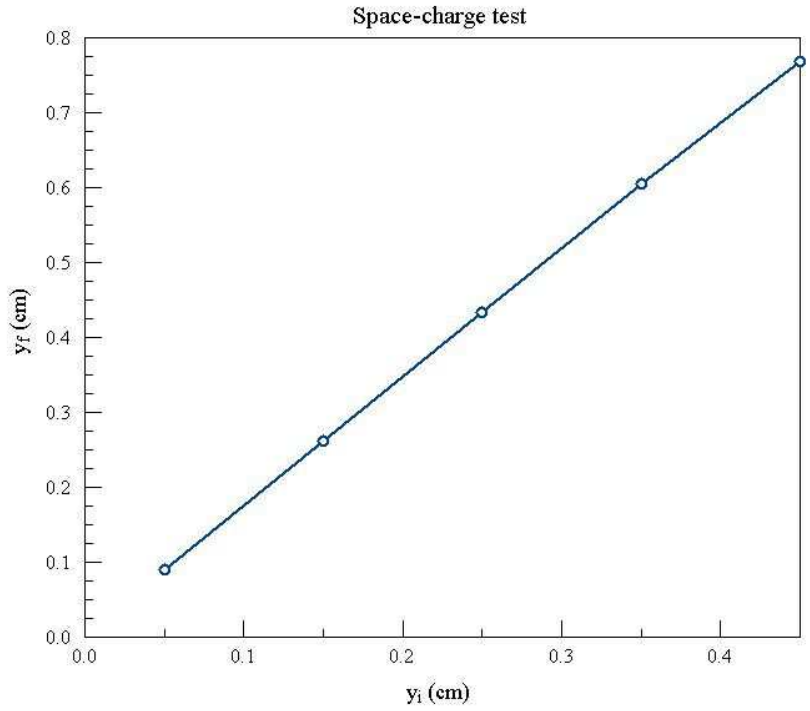


Figure 27: Expansion of a sheet beam, final displacement versus initial displacement.

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:

$$y_f = 0.0045 + (0.45/0.50)(0.06)^2/2 = 0.774 \text{ cm}. \tag{14}$$

Figure 27 shows a plot of the final orbit positions in $y$ versus their initial positions. Note the linearity of forces acting in the expansion. The outer orbits have a final position $y = 0.768$ cm, within 0.8% of the theoretical value.

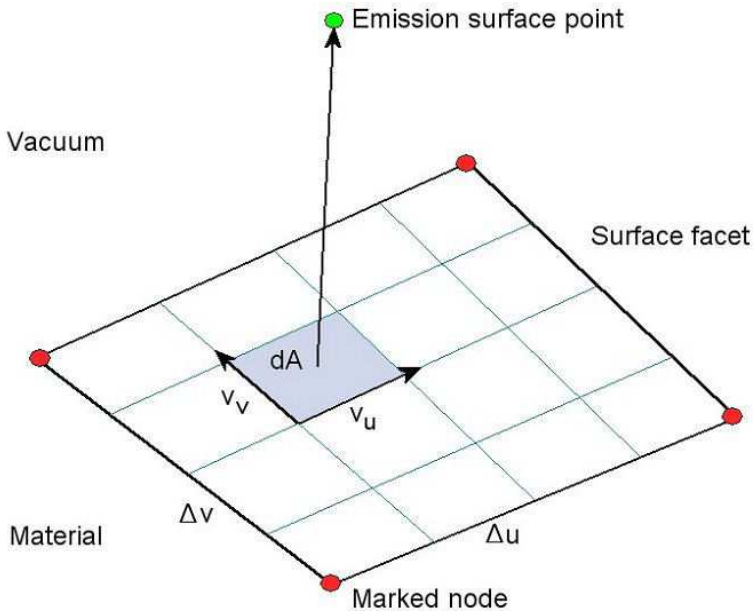


Figure 28: Projection of segments of a surface facet to an emission surface.

# 11 Self-consistent Child law emission – Particles section

## 11.1 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.9.2. 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 $D_e = DEmit$ from the physical surface. The reason is that 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 formulas to assign kinetic energy and current to the particles.

To process a source surface, **OmniTrak** 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. 26). As in the *Track* mode, the facet is divided into $NDiv \times NDiv$ sections using the normal coordinates ($u$, $v$) of the surface. For example, when $NDiv = 2$ the first model particle on a facet extends over the range $-1.0 \leq u \leq 0.0$ and $-1.0 \leq v \leq 0.0$. 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 $(x_0, y_0, z_0)$ is the position of the first node of the facet. The shape functions are $N_0 = (1-u)(1-v)/4$, $N_1 = u(1-v)/4$, $N_2 = uv/4$ and $N_3 = (1-u)v/4$. The surface area of the facet segment corresponding to the range of normal coordinates is calculated with a $4 \times 4$ Gaussian-quadrature integral. The following equations give the surface normal vector $\mathbf{n}$ at the emission location:

$$\mathbf{x}_u = \left[\frac{\partial x_s}{\partial u_s}\ \mathbf{x} + \frac{\partial y_s}{\partial u_s}\ \mathbf{y} + \frac{\partial z_s}{\partial u_s}\ \mathbf{z}\right],$$
$$\mathbf{x}_v = \left[\frac{\partial x_s}{\partial v_s}\ \mathbf{x} + \frac{\partial y_s}{\partial v_s}\ \mathbf{y} + \frac{\partial z_s}{\partial v_s}\ \mathbf{z}\right],$$
$$\mathbf{L} = \mathbf{x}_u \times \mathbf{x}_v,$$
$$\mathbf{n} = \mathbf{L}/|\mathbf{L}|.$$

The particle emission position is given by:

$$x_p = x_s + D_e\ n_x,$$
$$y_p = y_s + D_e\ n_y,$$
$$z_p = z_s + D_e\ n_z.$$

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:

$$j = \frac{4\epsilon_0}{9}\ \sqrt{\frac{2q}{m}}\ \frac{\Delta\phi^{3/2}}{D_e^2}, \tag{15}$$

$$T_0 = \sqrt{\frac{2q\Delta\phi}{m}}. \tag{16}$$

**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 $4T_0/9$ 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 To and momentum components directed away from the source and tracks particle orbits through the solution volume.

## 11.2 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 Chrg DEmit NDiv
### EMIT 4 1.0 1.0 0.05 1

The *Emit* command in the *SCharge* mode is similar to that of the *Track* mode. The command specifies region number $NReg$ 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 $Chrg$ is the particle charge relative to the magnitude of the electron/proton charge. To specify electrons, set $Mass = 0.0$ and $Chrg = -1.0$. 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 $NDiv$ (integer) governs how many particles are emitted per surface facet. It equals the number of divisions along the normal axes of the facet. Figure 28 shows a case with $NDiv = 4$. The number of particles per facet equals $NDiv^2$.

### 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 is the source-limited current density (in A/m$^2$). 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 $JLimit$.

**TSOURCE kTs**
**TSOURCE = 0.08**

Use this command to assign a small angular spread to particles leaving the previously-defined emission surface based on a source temperature. Note that the command should immediately follow the associated *Emit* command. Enter the value $kT_s$ in electron volts.

When you set a value for the source temperature, **OmniTrak** assigns an angular spread to particles leaving the emission surface based on a Maxwell distribution in transverse velocity. The direction of particle emission is determined from a unit vector normal to the emission surface and an angular displacement is calculated from the transverse kinetic energy and the longitudinal kinetic energy at the surface. The default value for all emission regions is $kTs = 0.0$ eV. Use the following equation to convert a value of source temperature in °C to electron volts:

$$kT_s(\text{eV}) = \frac{T_s(^{\circ}\text{C}) + 273}{11,594}. \tag{17}$$

The probability distribution of transverse velocity $v_\perp$ of particles emitted from a source at temperature $T_s$ approximates the Maxwell distribution:

$$f(v_\perp)dv_\perp = v_\perp \left(\frac{m}{kT_s}\right) \exp\left(-\frac{mv_\perp^2}{2kT_s}\right). \tag{18}$$

We can rewrite Eq. 18 in terms of the normalized variable

$$\chi = \sqrt{\frac{m}{2kT_s}}\, v_\perp, \tag{19}$$

as

$$f(\chi)d\chi = 2\chi \exp\left(-\chi^2\right)\, d\chi. \tag{20}$$

The integral of Eq. 20 gives the cumulative probability, a variable in the range from 0.0 to 1.0:

$$\int_0^\chi f(\chi)d\chi = \zeta = 1 - \exp(-\chi^2). \tag{21}$$

The procedure in **OmniTrak** is to generate a random value of $\zeta$ and then to find a weighted value of $\chi$ from the inverse of Eq. 21:

$$\chi = -\frac{\ln(1-\zeta)}{2}. \tag{22}$$

The angle of the particle relative to the surface normal is given by

$$\Delta\theta \cong \frac{v_\perp}{v_0} = \chi \sqrt{\frac{kT_s}{T_p}}, \tag{23}$$

where $T_p$ is the kinetic energy (in eV) of the particle at the emission surface. Particles are emitted with a random distribution of azimuthal angle about the surface normal vector. Note that the derivation is valid in the limit that $v_\perp \ll v_0$. **OmniTrak** generates an error message if $kT_s$ is too high relative to the average potential of the emission surface.

**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 | Supression 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 ($DTheta$) does not function in the *Field* mode.

## 11.3 Application examples

This section reviews three examples that illustrate **OmniTrak** techniques for space-charge-limited emission. The first (`SCEMIT1D`) 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 $z = 1.0cm$ 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/m$^2$, giving a total current of $1.844 \times 10^{-2}$ A over the solution cross section. The equation (**Charged Particle Beams**, Sect. 5.2):

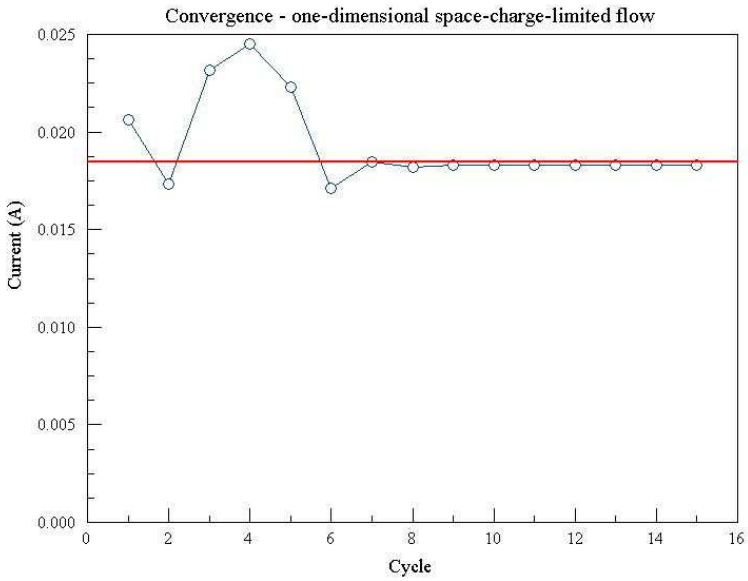


Figure 29: Convergence history for one-dimensional space-charge-limited flow - total current versus cycle number. Red line shows the theoretical current value. Example `SCEMIT1D`.

$$\Delta\phi = V_0 \left(\frac{z}{d}\right)^{4/3}, \tag{24}$$

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 a half element per step. Figure 29 shows the convergence history. The final value of current was $1.830 \times 10^{-2}$ A, within 0.76% of the prediction. Accumulated noise from the field interpolations was small - the orbits at the exit were parallel to within $\pm 0.03$ milliradian. The final average kinetic energy for the 400 particles was $998.69 \pm 0.06$ eV, implying energy conservation to within 0.13%.

In the next example (`CLCYLIN`) 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 30 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 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. 30.

The emission surface area for an ideal cylinder of radius 1.75 cm and length 2.0 cm is 5.498 cm$^2$. The facets created by **MetaMesh** have a calculated surface area of 5.497 cm$^2$. One model particle is assigned to each facet for a total number of 680 particles. The simulation occupies

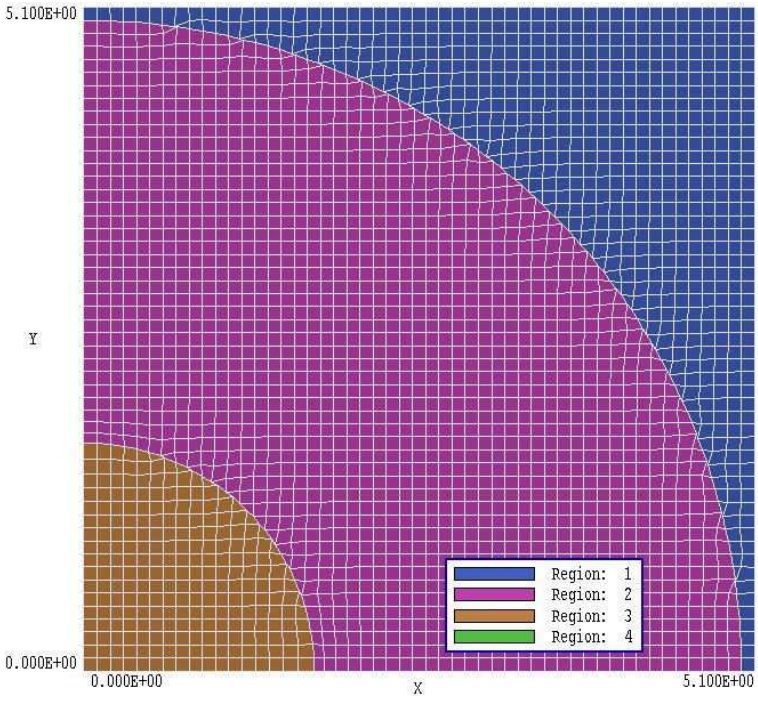


Figure 30: Computational mesh - simulation of space-charge-limited flow between concentric cylinders. Dimensions in cm. Example `CLCYLIN`.

fifteen iteration cycles with an averaging parameter of $AvgE = 0.35$. 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 (`RODEMIT`) 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 32 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 rounded 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

The mesh for the **OmniTrak** simulation contained 245760 nodes and 234171 elements. The simulation treats one quadrant of the space with symmetry boundaries at $x = 0.0$ and $y = 0.0$. The variable-resolution mesh had a minimum element length of 0.10 cm in the emission region. Region 1 represented the grounded vacuum chamber and Region 2 represented the vacuum volume. 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

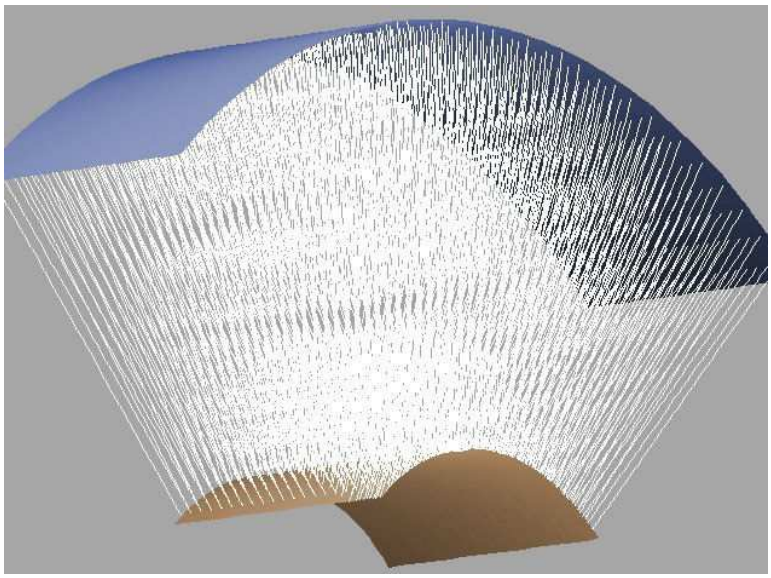

Figure 31: Three-dimensional plot of orbits - space-charge-limited flow between coaxial cylinders. Example `CLCYLIN`.

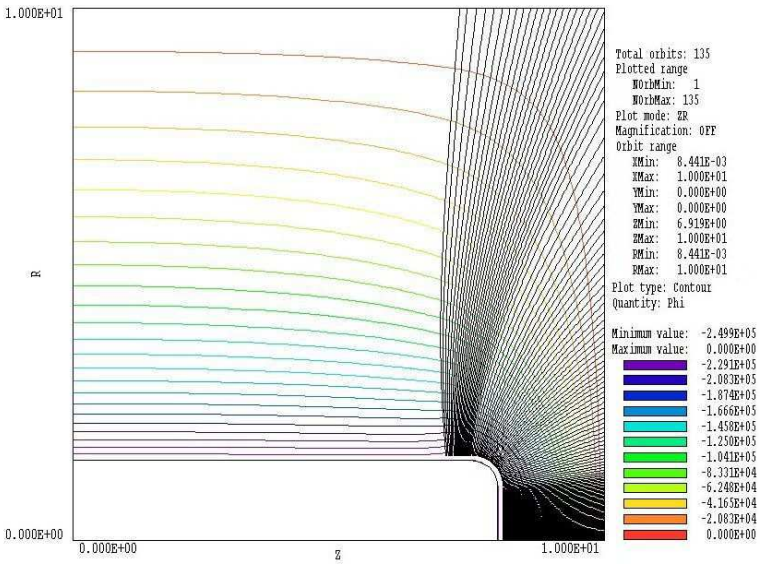


Figure 32: **Trak** simulation electron emission from a shaped rod.

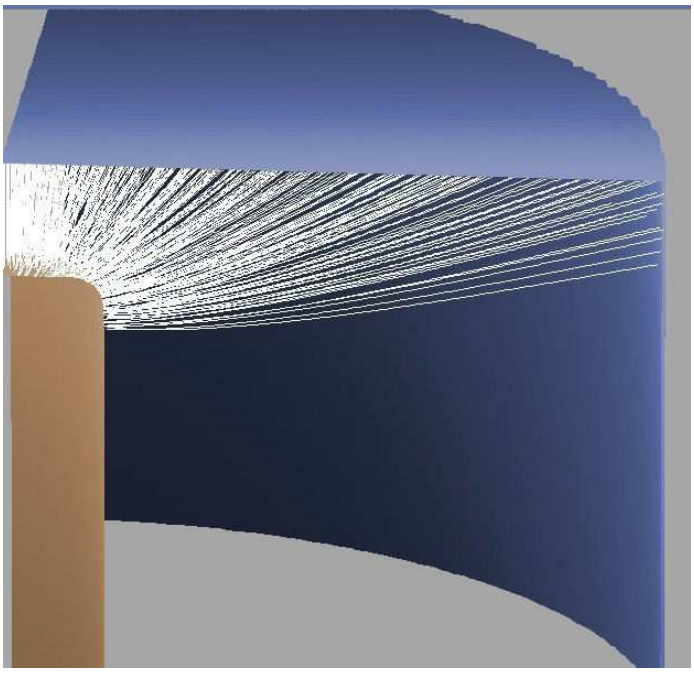

Figure 33: Calculated orbits (1 out of 5 shown) for the **OmniTrak** simulation of the complex emission surface. Example `RODEMIT`

.

emission nodes (Region 5) covered the electrode tip. The simulation used 1 electron per facet for a total of 469 particles.

Figure 33 shows a 3D plot of selected orbits ($NPSkip = 5$). The irradiated area on the anode is in excellent agreement with that of Figure 32. **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 $249.4 \pm 1.0$ keV, implying energy conservation to within $\pm 0.4\%$.

## 11.4 Restarting runs with space charge and Child-law emission

You can restart **OmniTrak** runs that include beam-generated electric and/or 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 efficient 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 smaller 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 *EField3D* 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.

# 12 Beam-generated magnetic fields - Particles section

## 12.1 Modeling relativistic beams

Three-dimensional simulation of a high-current relativistic electron injector is a difficult challenge for a gun-design code. The capability is essential for work on non-circular guns (such as sheet beam injectors) and for studies of perturbations in circular guns. **OmniTrak** is the only ray-tracing code that can handle the complete problem with moderate setup and run times.

This section reviews some features of relativistic beam transport and the methods used in **OmniTrak** to calculate beam-generated magnetic fields. This material shows 1) why beam-generated magnetic fields play an important role in the generation and transport of relativistic beams and 2) when to use the two available modeling methods (*relativistic approximation* and *relativistic beam mode*). The text **Charged Particle Beams** (`http://www.fieldp.com/cpa.html`) gives detailed information.

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 ($v_z \cong c$) and 2) orbits make small angles with respect to the axis:

$$z' = \frac{dr}{dz} \ll 1.0. \tag{25}$$

Equation 25 implies that changes in the beam radius $r_o$ take place over axial distances much greater than $r_o$. 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:

$$E_z \cong 0.0, \tag{26}$$

$$E_\perp = E_r(r) = -\frac{e}{2\pi\epsilon_0 r} \int_0^r 2\pi r' dr' n(r'), \tag{27}$$

$$B_z \cong 0.0, \tag{28}$$

$$B_\perp = B_\theta(r) = -\frac{e\beta c\mu_0}{2\pi r} \int_0^r 2\pi r' dr' n(r'). \tag{29}$$

Equations 27 and 29 imply that the transverse electric and magnetic forces acting on individual electrons are related by

$$F_\perp(\text{magnetic}) = -\beta^2 \; F_\perp(\text{electric}). \tag{30}$$

The quantity $\beta$ is small compared to unity for non-relativistic beams; therefore, the magnetic force is usually small. In contrast, for highly relativistic beams ($\beta \cong 1.0$) the repulsive electric force and attractive magnetic force are almost balanced.

The transverse forces of a planar beam also satisfy Eq. 30. In fact, the relationship holds for paraxial beams of any shape and with any transverse distribution of space-charge. 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. The derivation also shows that the criteria underlying paraxial beam motion and the definition of *gradual* changes is less stringent than that implied by Eq. 25. 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. 30 holds if

$$\frac{D}{\gamma L} \ll 1.0, \tag{31}$$

where

$$\gamma = \frac{1}{\sqrt{1-\beta^2}}. \tag{32}$$

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

$$\gamma = 1 + \frac{T}{m_0 c^2}. \tag{33}$$

Equations 25) through 33 suggest a strategy (that we shall refer to as the *relativistic approximation*) to avoid calculations of three-dimensional magnetic fields for relativistic beams. The total transverse force is related to the electric force by:

$$F_\perp(\text{total}) = F_\perp(\text{electric}) - F_\perp(\text{magnetic}) = (1-\beta^2)\ F_\perp(\text{electric}), \tag{34}$$

or

$$F_\perp(\text{total}) = \frac{F_\perp(\text{electric})}{\gamma^2}\ . \tag{35}$$

We simply calculate the electric-field force and then divide by the square of the average particle $\gamma$ to include the effect of the magnetic field. The relativistic approximation holds under the following conditions:

- Transverse electric fields arise entirely from the beam space charge.
- Beam particles move predominantly in one direction and have about the same axial velocity, $v_z \cong c$.
- The axial scale length for changes in the transverse dimensions of the beam and surrounding boundaries satisfies Eq. 31.

Section 12.2 discusses applications of the relativistic approximation and its implementation in **OmniTrak**. It is important to recognize that the method does not apply in the acceleration gap of an electron gun. In this case there are strong transverse applied electric fields and beam properties may change significantly over a short axial distance. Gun simulations require the complete three-dimensional calculation of the magnetic field carried out in the **OmniTrak** *RelBeam* mode (Sects. 12.3 through 12.5). To summarize, there are two ways to include beam-generated magnetic fields in **OmniTrak**:

- For beam transport without acceleration, apply the relativistic approximation in the *SCharge* mode
- For electron guns and acceleration gaps, use the *RelBeam* mode.

## 12.2 Relativistic approximation

Run times are shorter if you can use the relativistic approximation. To review, the system must satisfy the following criteria:

1. The electric field results solely from the beam space-charge. In general, the relativistic approximation applies to transport rather than acceleration solutions.
2. Particles in the beam have about the same kinetic energy.
3. Orbits of individual particles 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.
4. The axial distance $L$ for changes in the beam direction, changes in the envelope width, or changes in the transport tube geometry satisfies the inequality $L \gg \gamma D$. Here, $D$ is a characteristic transverse dimension of the beam and/or transport tube.
5. Applied magnetic fields may be included and the main axis of the beam need not be a straight line. The relativistic approximation could be employed, for example, to simulate the dynamics of a high-intensity beam in a bending magnet.

The relativistic approximation may be applied to **OmniTrak** calculations in the *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. The list should contain only a single particle species. In most practical cases the particles are electrons. To invoke the relativistic approximation, simply include the command *RelApprox* in the *Particles* section. The section then has the general form:

```
PARTICLES Scharge
   ...
  PLIST
      ...
  END
   ...
  RELAPPROX
   ...
END
```

### RELAPPROX

When this command appears, **OmniTrak** applies the general relativistic approximation. The

program divides all Cartesian components of the electric force by $\gamma^2$ for the calculation of particle dynamics. The quantity $\gamma$ is the local relativistic energy factor of the present particle.

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

$$F_x = \frac{qE_x}{\gamma^2}, \tag{36}$$

$$F_y = \frac{qE_y}{\gamma^2}, \tag{37}$$

$$F_z = \frac{qE_z}{\gamma^2}. \tag{38}$$

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 $\phi$ at positions inside the beam is much smaller than the average kinetic energy expressed in eV:

$$\phi \ll T_e/q. \tag{39}$$

Therefore, values of $\beta$ and $\gamma$ for the particle do not change significantly over the spatial extent of the simulation.

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

### RELAPPROX Axis
### RELAPPROX Z

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

For example, in response to the command `RELAPPROX Z`, **OmniTrak** calculates forces according to:

$$F_x = \frac{qE_x}{\gamma^2}, \tag{40}$$

$$F_y = \frac{qE_y}{\gamma^2}, \tag{41}$$

$$F_z = qE_z. \tag{42}$$

The set of equations would 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 Eqs. 40, 41 and 42 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 application of the relativistic approximation. The goal is to transform a circular high-perveance electron beam into a stretched elliptical beam. The

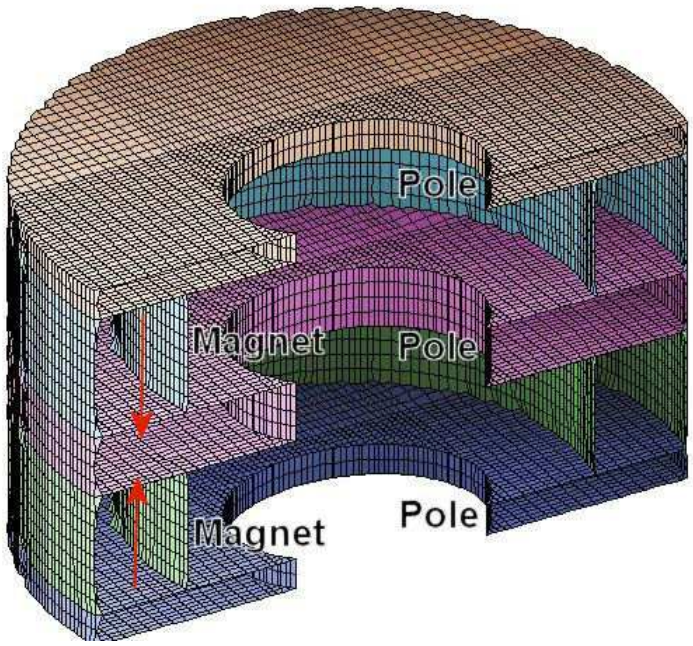


Figure 34: Mesh representation of the elliptical solenoid lens.

approach is to use solenoid lens with non-circular apertures to apply unequal forces along two normal axes. Figure 34 shows the geometry (`ELLIPLENSB.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 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 35 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.

## 12.3 Physical basis of the *RelBeam* tracking mode

In the RelBeam mode, **OmniTrak** uses a special mesh to calculate full three-dimensional beam-generated magnetic fields and forces. The code determines the component of magnetic field

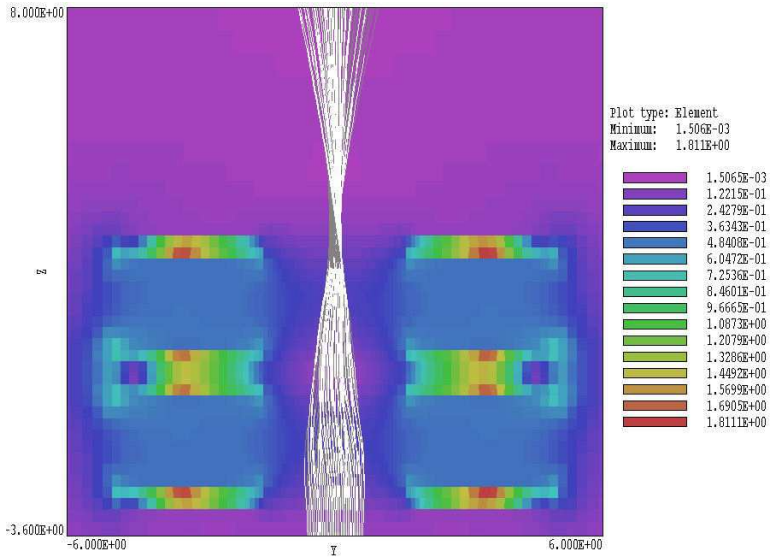


Figure 35: Electron orbits projected to the plane $x = 0.0$ in example `ELLIPLENS`. Light orbit vectors are in front of the projection plane and dark vectors are behind the plane. Color coding shows $|\mathbf{B}|$.

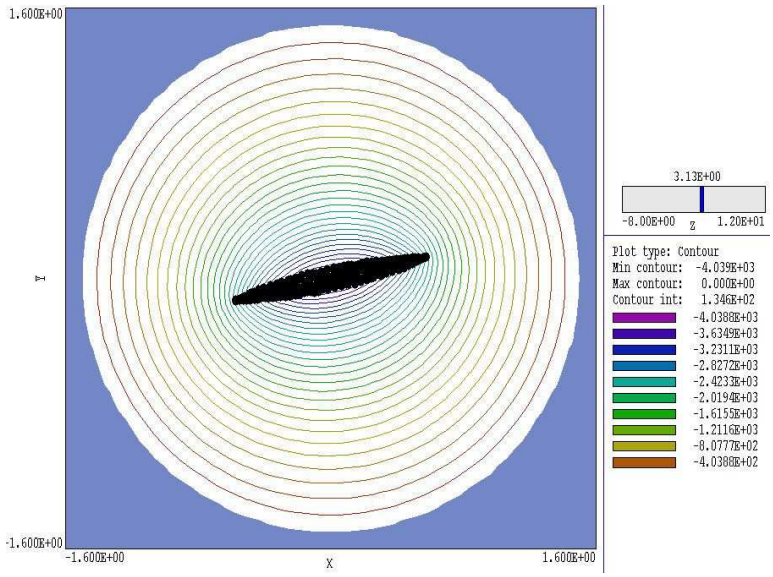


Figure 36: Orbit projections in the plane $z = 3.13$ cm for the example `ELLIPLENS`. Plot includes electrostatic equipotential lines. Blue area is the grounded transport tube.

Table 11: 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
  RelApprox
  Vacuum(E) = 2
  Material(E) = 1
END
DIAGNOSTICS
  PartList
  EDump ELLIPLENSEP.HOU
  PartFile ELLIPLENS
END
ENDFILE
```

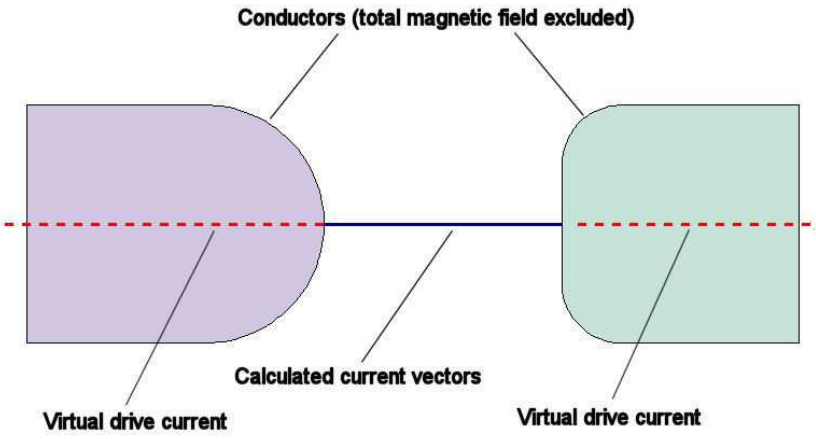


Figure 37: Virtual drive currents are required to calculate the correct beam-generated applied field in the vacuum region between conductors.

$\mathbf{B}_s$ created by the beam by taking a Biot-Savart integral over the current elements represented by the model particles. The contribution from each element is evaluated at every node of the mesh. The procedure can be quite work-intensive. In a simulation there may be hundreds of particle orbits, each composed of hundreds of integration steps. There is 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 $+\infty$ (Fig. 37). If we do not include the contribution of the drive currents, the calculated applied field values near the start and end points will about 50% too low.

The problems arise because a complete Biot-Savart calculation represents considerably more work than is warranted by the simple magnetic field distributions created by beams. 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 run time:

- 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 entire length of each orbit (including associated drive currents). 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 procedure illustrated in Fig. 38 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

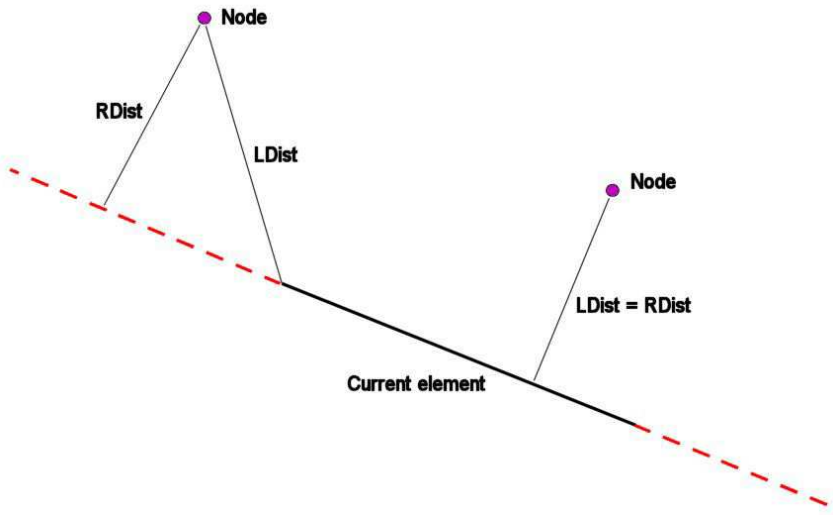


Figure 38: Treatment of a current element as a section of an infinite-length cylinder.

when the trajectory 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 beam-field mesh. For each node, the program identifies the minimum distance to each particle vector ($LDist$ in Figure 38). 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,

$$\mathbf{dH}_s = (\mathbf{u}_e \times \mathbf{u}_r)\,\frac{I}{2\pi R_{dist}}, \quad (R_{dist} > R_{elem}) \tag{43}$$

$$\mathbf{dH}_s = (\mathbf{u}_e \times \mathbf{u}_r)\,\frac{I R_{dist}}{2\pi R_{elem}^2}. \quad (R_{dist} \leq R_{elem}) \tag{44}$$

In the equation, the vector is treated as a cylinder of radius $R_{elem}$ with uniform current density. The procedure smooths the current distribution of the beam to avoid divergent field values.

An advantage of the algorithm is that the applied fields are calculated as though the current extended an infinite distance in the upstream direction from the start point and in the downstream direction from the end point (Fig. 38). In other words, drive currents are automatically included in the applied field calculation. It is important to realize the limitations of the method. First, although the approach saves work, the magnetic field calculation could add a significant burden to the program. It is important to limit the magnetic field mesh only to the region occupied by beam particles. A second consideration is that the method may not be accurate in some cases. Figure 39*a* 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

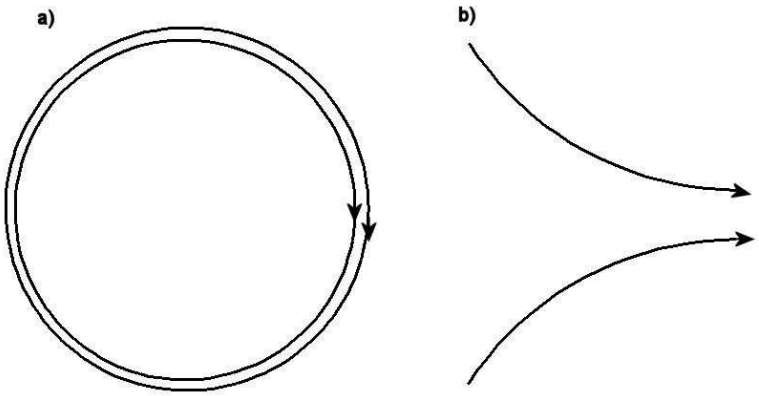


Figure 39: Accuracy of the infinite cylinder approximation is poor when all particles of a beam have the same inflection ($a$) and good when beam particles have opposite inflection ($b$).

the beam. The method does give good results for the more practical case of Fig. 39$b$. Here field gradients cancel because orbits on opposite sides of the beam have opposing inflections. Finally, the method does not apply if there are magnetically-active objects *inside* the beam volume (e.g., field excluding metal, iron, permanent magnets,...).

As a final topic, consider the use of symmetry to reduce the run time. For example, in a simulation of a planar-beam injector, it is often valid to model only the first quadrant in the $x$-$y$ plane with both particle and field symmetry about $x = 0.0$ and $y = 0.0$. Previous chapters have discussed how to use special boundary conditions to create symmetric solutions for the electric field and applied magnetic field. Reflection conditions may be applied to represent symmetry of particle orbits.

**OmniTrak** ensures symmetry of the beam-generated magnetic field by creating virtual current elements at mirror positions and summing their contributions. Figure 40 illustrates the approach for a symmetry boundary at $x = 0.0$. Although there is some computational overhead to integrate over the extra elements, a symmetry boundary usually gives a substantial reduction in run time. **OmniTrak** can represent symmetry of the beam-generated magnetic field under three special conditions:

- Symmetry about $x = 0.0$. One set of virtual particles is created at $[-x, y]$.
- Symmetry about $y = 0.0$. One set of virtual particles is created at $[x, -y]$.
- Symmetry about both $x = 0.0$ and $y = 0.0$. Three sets of virtual particles are created at $[-x, y]$, $[-x, -y]$ and $[x, -y]$.

## 12.4 Commands for the *RelBeam* mode

The command `Particles RelBeam` signals that **OmniTrak** should prepare for calculations of three-dimensional beam-generated magnetic fields. 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. The first two commands control the calculation of applied fields from orbit vectors in both modes.

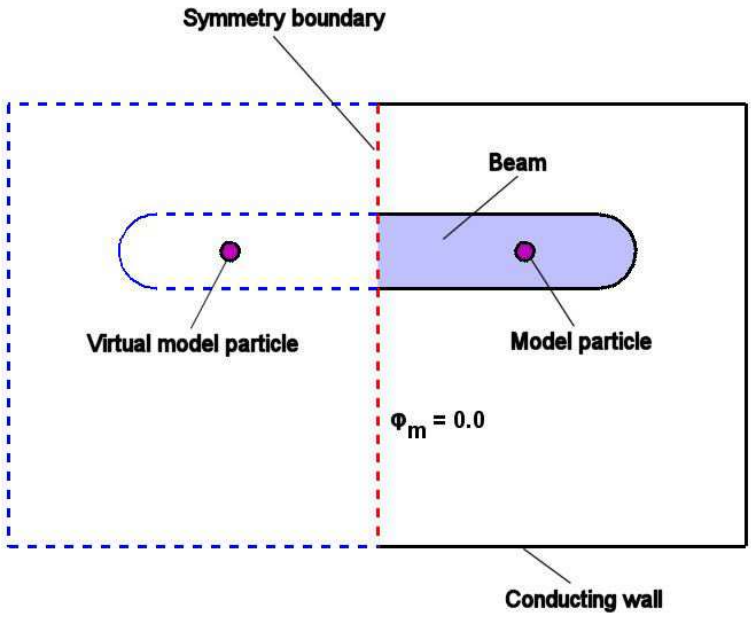


Figure 40: Bilateral symmetry boundary for the calculation of beam-generated magnetic fields.

**BBMESH XMin XMax Nx YMin YMax Ny ZMin ZMax Nz**
**BBMESH (-0.25 0.25 20) (-0.25 0.25 20) (0.0 2.50 50)**

Set parameters for a regular mesh of node positions to record values of the beam-generated magnetic field ($B_{sx}, B_{sy}, B_{sz}$). The real-number quantities $x_{min}, ..., z_{max}$ define the mesh volume. Enter dimensions in units set by *DUnit*. The integer quantities $N_x, N_y$ and $N_z$ set the number of nodes along each axis. Because **OmniTrak** uses second-order interpolations between nodes,it is not necessary to use large number of nodes along the axes. If the script does not contain a *Boundary* command, the mesh volume must enclose the particle trajectories. The code stops with an error if a particle moves outside the mesh. When the *Boundary* command is present (ballistic mode), **OmniTrak** continues particle orbits outside the mesh using zero values for $\mathbf{B}_s$.

**BBELEM LElem RElem**
**BBELEM 1.0 0.05**

The parameters in this command control the applied field calculation described in Sect. 12.3. The real-number quantity *LElem* is the minimum length of current-element vectors stored during orbit integrations. 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 10-20 vectors would be sufficient. The second parameter *RElem* is the effective radius of the current element vector (Eq. 43). Again, enter the value in units set by *DUnit*. Non-zero values of *RElem* prevent divergence of the 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.

**BBSYMPLANE Axis**
**BBSYMPLANE 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 [$X$, $Y$ or $Z$] 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 *BBSymplane X* defines a symmetry plane located at $x = x_{min}$. To avoid confusion, it is usually best to set up the calculation so that $x_{min} = 0.0$. **OmniTrak** 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 $x' = -x$, $y' = y$ and $z' = z$. The contribution of the mirror vectors ensures that the beam-generated magnetic field is normal to the boundary.

**BBSYMPLANE Axis1**
**BBSYMPLANE Axis2**
**BBSYMPLANE X**
**BBSYMPLANE Y**

To define a solution with quadrilateral symmetry, include two instances of the *BBSymPlane* command. The sample commands specify that the beam distribution has symmetry about both the planes $x = x_{min}$ and $y = y_{min}$. In response to the two commands, **OmniTrak** creates three sets of mirror vectors for each particle orbit with start and end coordinates related by $x'_1 = -x$, $y'_1 = y$, $z'_1 = z$, $x'_2 = x$, $y'_2 = -y$, $z'_2 = z$, $x'_3 = -x$, $y'_3 = -y$ and $z'_3 = z$.

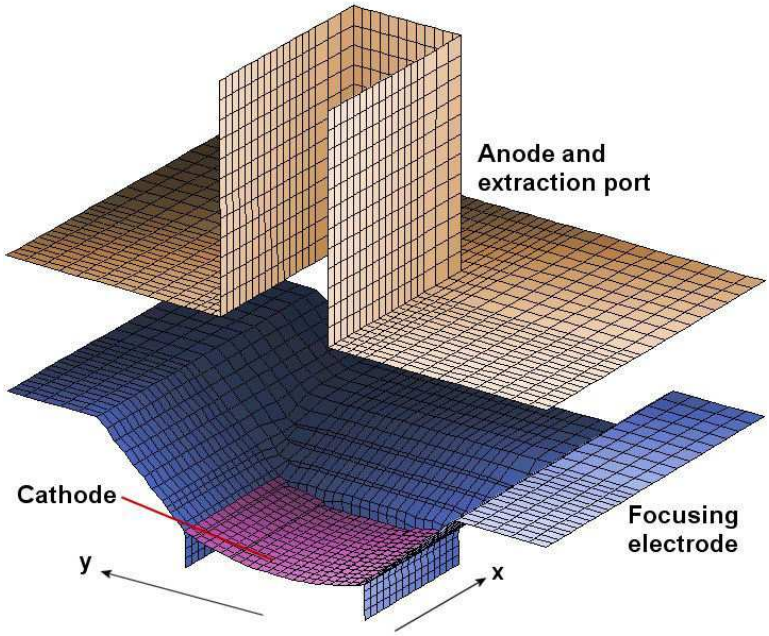


Figure 41: Electric field mesh for the example `PLANAR`.

## 12.5 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. 41 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.

Two **OmniTrak** input files are supplied: `PLANAR01.OIN` has no beam-generated magnetic field while `PLANAR02.OIN` models all effects. Table 12 lists the contents of `PLANAR02.OIN` with added line numbers. Several 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 *BBMesh* command (Line 22) defines a box mesh the brackets all electron orbits in the calculation. The *BBElem* command (Line 23) sets $LElem = 0.4$" and $RElem = 0.05$". The choice of $LElem$ gives about 6 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 20) guarantees particle orbit symmetry about the plane $x = 0.0$. The *DiagPlane* command (Line 21) instructs the program to record orbit parameters near the waist point at $z = 1.60$" while continuing orbits to the axial boundary at $z = 2.50$". In response to the *BBSymPlane* command in Line 24, **OmniTrak** will create virtual current elements for evaluation of the beam-generated magnetic field at mirror positions with respect to $x = 0.0$, The final statement

Table 12: **OmniTrak** input file for the PLANAR example

```
01: FIELDS
02:   DUnit = 39.37
03:   EField3D = PLANAR.HOU 2.0
04:   NCheck(E) = 10
05:   MaxCycle(E) = 250
06: END
07:
08: PARTICLES RelBeam
09:   NCycle = 10
10:   Emit(5) = (0.0, -1.0, 0.110, 1)
11:   EDirect = Field
12:   Vacuum(E) = 1
13:   Material(E) = 2
14:   Material(E) = 3
15:   Material(E) = 4
17:   Material(E) = 5
18:   Avg(E) = 0.40
19:   Dt = 5.0E-12
20:   ReflectPlane(XDn) = 0.0
21:   DiagPlane(ZUp) = 1.60
22:   BBMesh  0.00  1.00  25   -0.50   0.50  25  -0.10  2.50  25
23:   BBElem = (0.40, 0.05)
24:   BBSymPlane X
25: END
26:
27: DIAGNOSTICS
28:   EDump PLANAR02E.HOU
29:   BBDump PLANAR02BB.GOU
30:   PartList
31:   PartFile PLANAR02
32: END
33:
34: ENDFILE
```

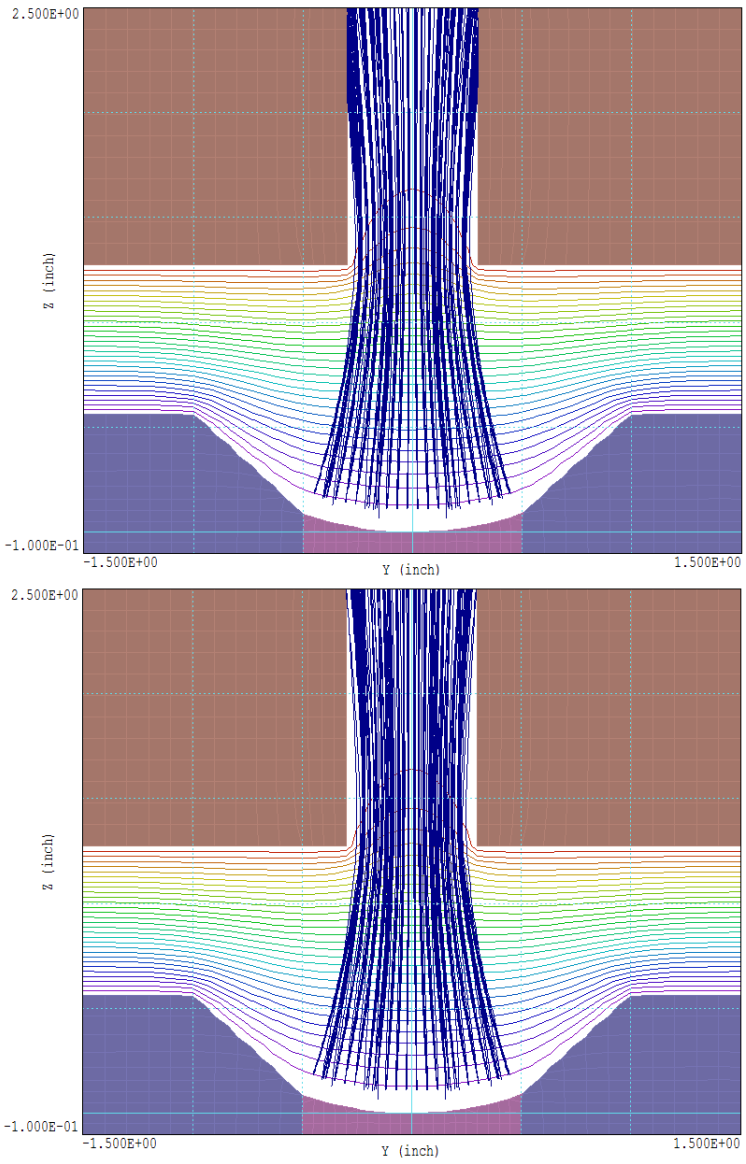


Figure 42: Projected particle orbits and equipotential lines in the plane $x = 0.0$, `PLANAR` example. Top: No beam-generated magnetic field. Bottom: With self-consistent beam-generated field.

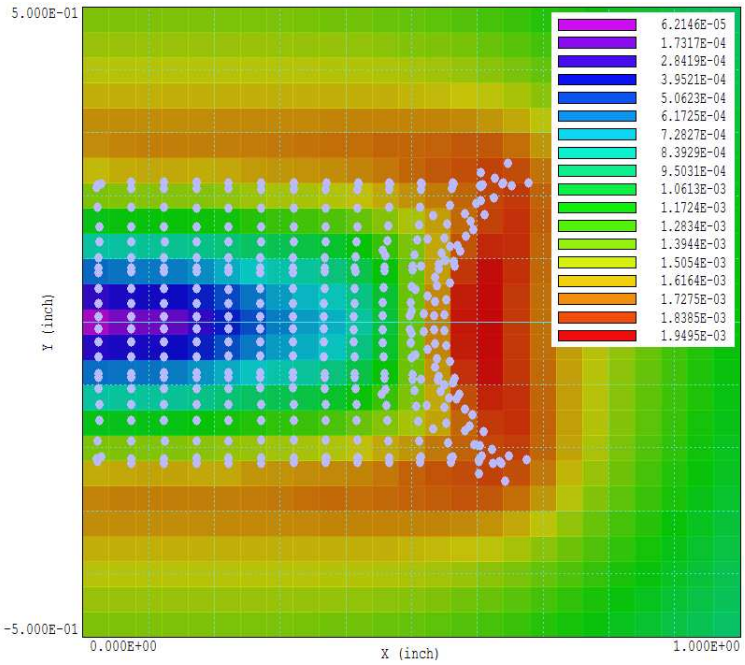


Figure 43: Orbit intersections with and $|\mathbf{B}|$ in the plane $z = 1.60$" for the `PLANAR` example.

of interest is `EDIRECT = EFIELD` in Line 11. 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.Figure 42 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 43 shows particle orbit intersections and elements color-coded by —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.

# 13 Electron 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 are created on source surfaces.
- Because the electric field must have 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 $\phi$, then the current density in A/m$^2$ is given by

$$j_{FE} = 1.537 \times 10^{-6} \ \frac{e^{\Gamma} E^2}{\phi t(\chi)^2}, \tag{45}$$

where

$$\Gamma = -6.83 \times 10^{9} \ \frac{\phi^{3/2} v(\chi)}{E}, \tag{46}$$

and

$$\chi = 3.79 \times 10^{-5} \ \frac{\sqrt{E}}{\phi}. \tag{47}$$

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

The set of allowed commands is similar to that for the *SCharge* mode. The major difference is that alternate parameters should appear in the *Emit* command.

### EMIT WFunc NDiv [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

Table 13: Fowler-Nordheim functions

| $\chi$ | $v(\chi)$ | $t(\chi)$ |
|---|---|---|
| 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 |

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 $WFunc$ is the work function $\phi$ in eV (electron volts). The integer parameter $NDiv$ 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 $NDiv^2$. The optional parameter $\beta$ is a field enhancement factor that may be useful 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. 45 through 47.

**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 Chap. 11 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.

# 14 Secondary emission – Particles section

**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 may be included 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 may represent a multi-generational set (incident electron plus one or more secondaries from 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 $\delta$ from the energy $E$ of the incident electron and the angle $\alpha$ 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 $\delta$. 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 the following references:


1. J.L.H. Jonker, Phillips Research Reports **6**, 372 (1951).
2. J.L.H. Jonker, Phillips Research Reports **7**, 1 (1952).
3. J.L.H. Jonker, Phillips Research Reports **12**, 249 (1957).
4. R.M. Vaughn, IEEE Trans. Electron Devices **ED-36**, 1963 (1989).
5. R.M. Vaughn, IEEE. Trans. Electron Devices **ED-40**, 830 (1993).


The model involves the angle $\alpha$ between the direction of the incident electron and the surface-normal vector. The defining function for the angular dependence of $\delta$ is

$$F(\alpha) = \frac{1}{\sqrt{\cos(\alpha)}}. \tag{48}$$

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

$$\delta_m = \delta_{m0}\ F(\alpha), \tag{49}$$

$$E_m = E_{m0}\ F(\alpha). \tag{50}$$

where $\delta_{m0}$ and $E_{m0}$ are the values at normal incidence. Table (*) lists these quantities for a variety of materials. The secondary emission coefficient as a function of the angle $\alpha$ and kinetic energy $E$ of the incident electron is given approximately as

Table 14: Secondary emission parameters for elements (adapted from D.R. Lide, ed., **Handbook of Chemistry and Physics**, 74th Edition (CRC Press, Boca Raton, 1993), 12-107)

| Element | $\delta_{m0}$ | $E_{m0}$ | Element | $\delta_{m0}$ | $E_{m0}$ |
|---|---|---|---|---|---|
| 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 |

$$\delta(\alpha, E) \cong \delta_m(\alpha) \ \left[ f e^{(1-f)} \right]^a , \tag{51}$$

where $f = E/E_m(\alpha)$. The parameter $a$ has the value 0.62 for $f < 1.0$ and 0.25 for $f \geq 1.0$. To prevent infinite values of $\delta$, the code takes $\alpha = 80^o$ for orbits with incident angles that exceed $80^o$. The reference S. Humphries, N. Dione and J. Petillo, *Secondary-electron emission modeling on a conformal mesh*, **Proc. WorkShop on RF Superconductivity**, Santa Fe, 1999. (included with the package) gives a detailed description of numerical methods used to treat secondary emission in the **OmniTrak** code.

## 14.1 Control commands

The following commands may appear in the *Track*, *SCharge*, *RelBeam* and *FEmit* sections.

**SECONDARY [E,B] NReg DMax0 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 DMax0 (real) is the maximum value of the secondary emission coefficient for normal incidence, $\delta_{m0}$. The quantity $EngMax0$ (real) is the kinetic energy of the incident electron (in eV) at which the maximum occurs.

Table 15: Secondary emission parameters for compounds (adapted from D.R. Lide, ed., **Handbook of Chemistry and Physics**, 74th Edition (CRC Press, Boca Raton, 1993), 12-107)

| Compound | $\delta_{m0}$ | $E_{m0}$ | Compound | $\delta_{m0}$ | $E_{m0}$ |
|---|---|---|---|---|---|
| **Alkali halides** | | | **Oxides** | | |
| CsCl | 6.5 | | Ag2O | 1.0 | |
| KBr (crystal) | 14 | 1800 | A12O3 (layer) | 2.0-9.0 | 400 |
| KCl (crystal) | 12 | 1600 | Ba0 (layer) | 2.3-4.8 | 2000 |
| KCl (layer) | 7.5 | 1200 | Be0 | 3.4 | 500 |
| KI (crystal) | 10 | 1600 | Ca0 | 2.2 | 400 |
| KI (layer) | 5.6 | | Cu20 | 1.2 | 1500 |
| LiF (crystal) | 8.5 | | Mg0 (crystal) | 20-25 | 400-1500 |
| LiF (layer) | 5.6 | 700 | Mg0 (layer) | 3-15 | |
| NaBr (crystal) | 24 | 1800 | MoO2 | 1.2 | 400 |
| NaBr (layer) | 6.3 | | SiO2 (quartz) | 2.1-4 | 640 |
| NaCl (crystal) | 14 | 1200 | SnO2 | 3.2 | |
| NaCl (layer) | 6.8 | 600 | **Others** | | |
| NaF (crystal) | 14 | 1200 | BaF2 (layer) | 4.5 | |
| NaF (layer) | 5.7 | | CaF2 (layer) | 3.2 | 1000 |
| NaI (crystal) | 19 | 1300 | BiCs3 | 6 | 1000 |
| NaI (layer) | 5.5 | | BiCs | 1.9 | 700 |
| RbCl (layer) | 5.8 | | GeCs | 7 | 450 |
| **Sulfides** | | | Rb3Sb | 7.1 | 700 |
| MoS2 | 1.1 | | SbCs3 | 6 | 350 |
| PbS | 1.2 | 500 | Mica | 2.4 | 300-450 |
| WS2 | 1.0 | | Glasses | 2-3 | |
| ZnS | 1.8 | 350 | | | |

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

This command sets global parameters that control the termination of multi-generation orbits to prevent infinite loops. The quantity $ECut$ 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 $ECut = 2.5$ eV. During a multi-generation 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, $MultFact(N) = \delta_1 \times \delta_2 \times ... \times \delta_N$. An orbit is terminated if the multiplication factor drops below the value $MinFact$. The default is $MinFact = 10^{-4}$.

**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.

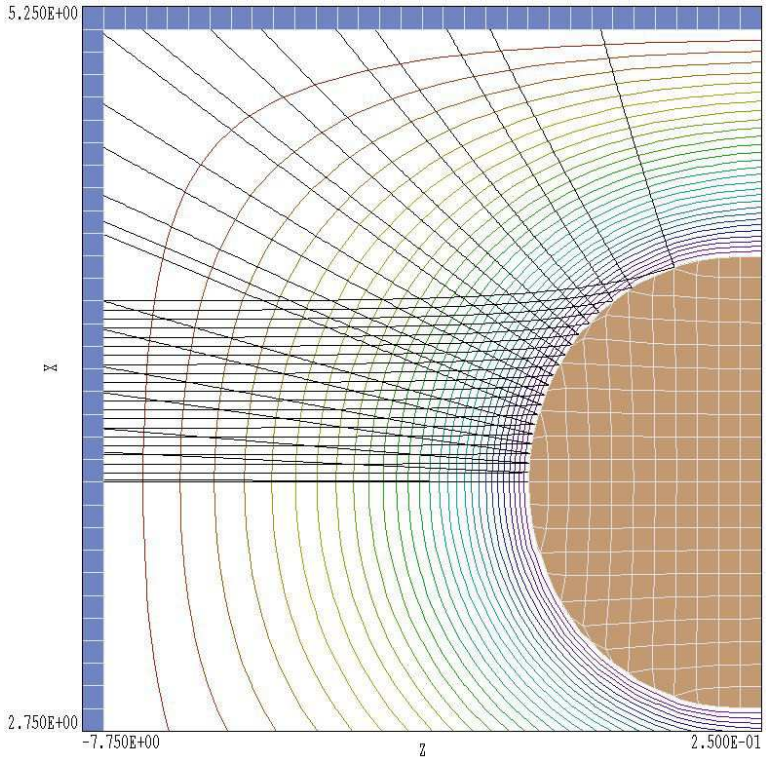


Figure 44: Orbits and electrostatic equipotential lines for the example SECONDTEST in the plane $y = 0.0$.

## 14.2 Benchmark test

The example SECONDTEST illustrates the mechanism for treating secondary emission. The file SECONDTEST.OIN is listed in Table 16. Figure 44 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 $\delta_{m0} \cong 3.5$ and $E_{m0} = 400$ eV. For each orbit, **OmniTrak** performs the following operations: 1) track the incident 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 $\delta$ according to Eqs. 48 through 51, 4) restart an electron near the surface penetration point with low kinetic energy and a weight adjusted by $\delta$, 5) track the secondary-electron orbit until it 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 $0.5 \times 10^{-10}$ 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 where successful.

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

Table 16: 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   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
```

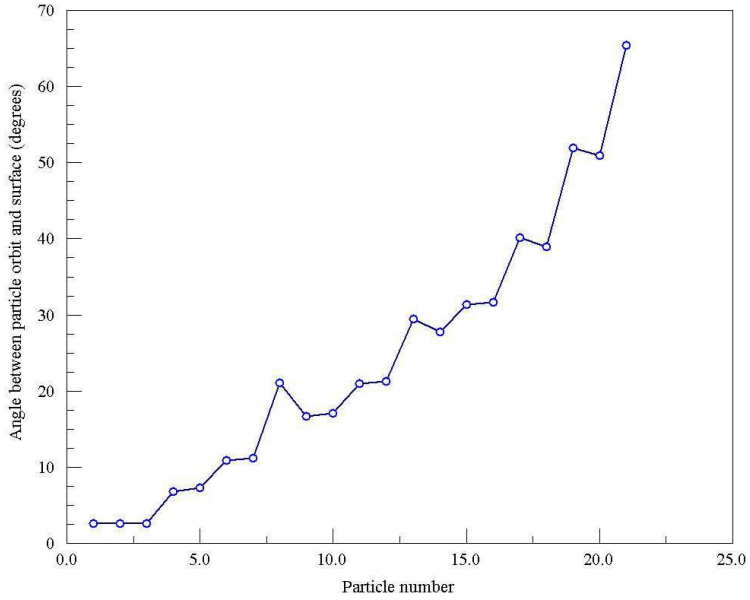


Figure 45: Calculated angle between the particle orbit and a vector normal to the spherical surface as a function of particle number in example `SECONDTEST`.

Inspection of Figure44 shows that orbit 1 strikes the sphere at approximately normal incidence while orbit 21 strikes at grazing incidence. Figure 45 shows a plot of the calculated surface angle a 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 $\overline{\delta} = 3.455$. Finally, in response to the *MultFact* command, the code records the data shown in Table 17. For the single-event simulation, the quantity *MultFact* equals the secondary emission coefficient on the surface of Region 3. The value of $\delta$ increases with decreasing angle of incidence.

Table 17: File SECONDTEST.OIN

| 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 |

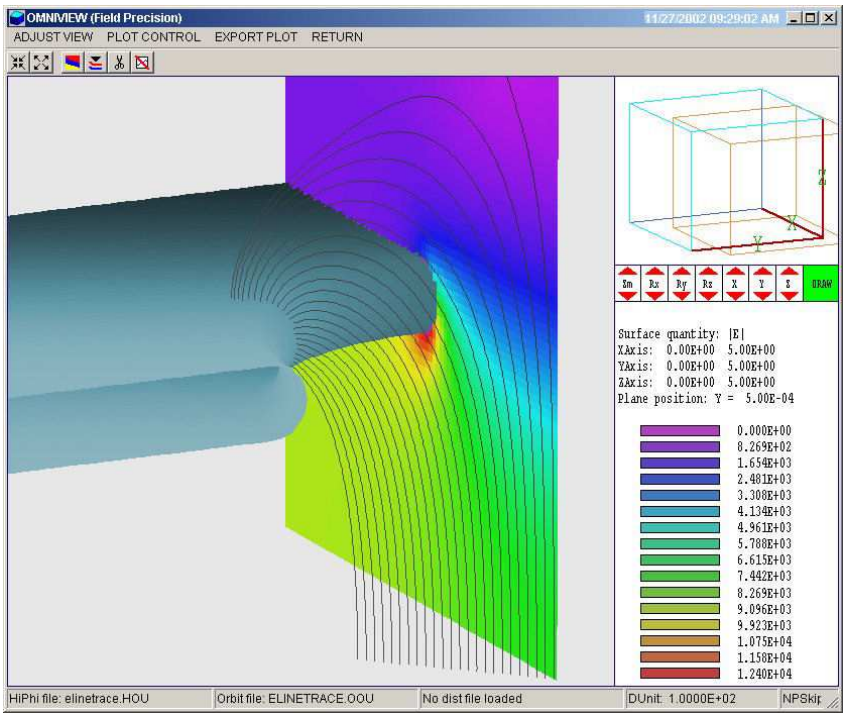


Figure 46: Three-dimensional plot of electric field lines near a circular aperture.

# 15 Electric and magnetic field lines – Particles section

**OmniTrak** includes a mode for precision three-dimensional tracing of lines of electric field (**E**) and magnetic flux density (**B**). In this case the section begins with the command `PARTICLES FLINE`. Calculated coordinates of points along field lines are recorded in the plot file (`OOU`) 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.

## 15.1 Commands for field line tracing

**OmniTrak** can trace up to 10000 field lines in a run. The initiation of lines can be accomplished with a list or from source surfaces. Regarding lists, field-line information lines may be inserted directly into the input command file with the *FList* command or it may be read from an independent file with the *FFile* command. For both 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 may 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 contains the following components:

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

Enter the coordinates in units set by *DUnit*. The optional polarity string may have the values *Neg* or *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.

### 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 may 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 32 characters. The file must be available in the working directory. A field line file may consist of blank lines, comments and from 1 to 10000 data lines. The file must terminate with the *End* command. The file has the following organization:

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

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

### EMIT NReg NDiv [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 $NDiv$ governs the number of field lines created per facet ($NDiv^2$). The string parameter *Polarity* (*Neg* or *Pos*) governs the tracing direction (default: *Pos*).

Two additional commands are available to control the tracing process.

**DS Ds**
**DS = 0.010**
Set the length step for spatial integrations. Enter the value of $Ds$ in units set by $DUnit$. If this command does not appear, **OmniTrak** sets a default approximately equal to the smallest element length along $x$, $y$ or $z$.

**SMAX SMax**
**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.

## 15.2 Conventions for field line tracing

This section lists some considerations for using the FLine mode:

- 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, $\epsilon_r \neq 1.0$ and $\mu_r \neq 1.0$ as *Material*. For field line tracking into dielectric and ferromagnetic materials, you should actively set the region properties using *Vacuum* commands.

- Precision stopping conditions may be used in field line tracking. Stopping and diagnostic planes function the same as 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 the 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 *ListType* command is inactive.

- In two-dimensional magnetic calculations, contours of vector potential $A_z$ or stream function $rA_\theta$ 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 (`OOU`): $t$, $x$, $y$ and $z$. Because the elapsed time $t$ has no meaning in a field line calculation, **OmniTrak** records the following quantities in the plot file: $d$, $x$, $y$ and $z$. The quantity $d$ 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 $\int \mathbf{E} \cdot \mathbf{dl}$ 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.

# 16 Diagnostics section

**OmniTrak** records information on the progress of the run in the listing file `OPREFIX.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.

## 16.1 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 $EMult$ or $BMult$ do not equal unity in the *EField3D*, *BField2D* and *BField3D* commands.
- Modifications of individual region potentials in the electric field solution (*ChangePot*).
- Coordinate shifts (*Shift*) and rotations of two-dimensional magnetic field solutions (*RotB2*).
- 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 field file into **OmniView**, the mesh structures and field values will match to 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 $\phi$ 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)**
Record 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)**
Write 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)**
Record 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)**
Write 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 command works if either a two-dimensional or three-dimensional magnetic field solution has been loaded. Uniform field components are included. A 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
================================================
   0.0000E+00  0.0000E+00 -7.0000E+00  1.2559E-01
   0.0000E+00  0.0000E+00 -6.7200E+00  1.2683E-01
   0.0000E+00  0.0000E+00 -6.4400E+00  1.2576E-01
   0.0000E+00  0.0000E+00 -6.1600E+00  1.2122E-01
```

```
...

      By          Bz          |B|      NReg
 ========================================
   0.0000E+00  6.2331E-01  6.3584E-01    1
   0.0000E+00  6.9604E-01  7.0750E-01    1
   0.0000E+00  7.6926E-01  7.7947E-01    1
   0.0000E+00  8.4164E-01  8.5032E-01    1
   ...
```

**BBPOINT Xp Yp Zp**
**BBPOINT = (0.567, 12.27, 18.5)**
Record 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 functions 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)**
Write 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 command functions only in the *RelBeam* mode.

**NSCAN NScan**
**NSCAN 150**
Set the number of intervals for *EScan*, *BScan* and *BBScan* listings. The default value is $NScan = 25$.

**EDUMP FileName [Text]**
**EDUMP = KlyMod.HOU**
Record 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**
Record 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 `GOU`. 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 (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 `GOU`. 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**.

## 16.2 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**
Write ordered tables of the initial and final particle positions and momenta to the listing file. The information may 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 PPrefix**
**PARTFILE = RFGOutDist**
Creates a file with the name `PPREFIX.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. 9.1. This file may be used as the input target file for the *PFile* command in a succeeding **OmniTrak** run.

**PARTDIST**
Analyze particle parameters (either at the diagnostic or stopping points) and calculate 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 **GenDist**.

**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.*, $ZOrig = \overline{Z}$ and $XOrig = \overline{X}$). 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 the distribution calculation.

## 16.3 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 $\delta > 1.0$. In the Track mode the absorbed current is zero.

**MULTFACT**
Write a list of final multiplication factors for electrons. The information is useful in runs with *Secondary* regions and multi-generation electron orbits. The listing also includes a global analysis of electron multiplication.

## 16.4 Field line diagnostics

**FLINELIST**
Write ordered tables of the initial and final particle positions of field lines to the listing file. The command functions only for *FLine* simulations. The information may be used directly in spreadsheets, plotting programs, and mathematical analysis programs. The final field line information is calculated at the termination position unless diagnostic planes have been defined. For electric fields, the list also contains the total voltage drop along the line as an indication of accuracy.

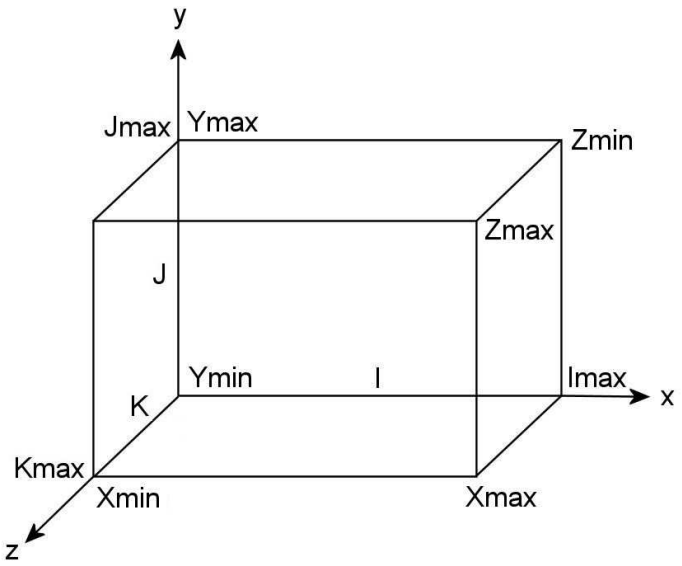


Figure 47: Regular mesh for field tables.

# 17 Using field tables

**OmniTrak** has several options for entering data on electric and magnetic fields. Most calculations involve the direct input of three-dimensional field solutions from **HiPhi** and/or **Magnum**. These files are organized according to the conformal meshes, so they contain a complete description of the system geometry. As mentioned in Sects. 6.4 and7.4, it is also possible to load information from tables of field values recorded at the nodes of a regular mesh. Although these files lack detailed geometric information, there are circumstances where the method is preferable or necessary. In particular, field tables provide a path to import field information from other finite-element programs into **OmniTrak**. Field tables may be created from **HiPhi** and **Magnum** solutions using the *Matrix file* command of **PhiView** and **MagView**. Tables may be created or modified with the **Mapper** utility. This chapter reviews the nature of field tables and their uses in **OmniTrak**. The following chapter describes **Mapper**.

## 17.1 Field table format

The organization of field table files is similar to that of **HiPhi** and **Magnum** output files in text format. The body consists of data lines with electric field values $(E_x, E_y, E_z)$ or magnetic flux density values $(B_x, B_y, B_z)$ calculated at points of the rectangular mesh illustrated in Fig. 47. 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 $0 \le i \le I_{max}, 0 \le j \le J_{max}, 0 \le k \le K_{max}$.

he file consists of a header followed by $N$ data lines, where

$$N = (I_{max} + 1) \times (J_{max} + 1) \times (K_{max} + 1). \tag{52}$$

Table 18: Initial section of a field map file

```
TEXT
NREG: 0
IMAX:      142
JMAX:       70
KMAX:       10
DUNIT:   3.937000E+01

RegNo RegUp        X             Y             Z
=================================================
     1     1 -2.413005E-01 -2.413005E-01 -6.350013E-03
     1     1 -2.349505E-01 -2.413005E-01 -6.350013E-03
     1     1 -2.286005E-01 -2.413005E-01 -6.350013E-03
     1     1 -2.222504E-01 -2.413005E-01 -6.350013E-03
   ...

                  Bx             By             Bz
              ========================================
                0.000000E+00  0.000000E+00  0.000000E+00
                0.000000E+00  0.000000E+00  0.000000E+00
                0.000000E+00  0.000000E+00  0.000000E+00
                0.000000E+00  0.000000E+00  0.000000E+00
                  ...
```

Table 18 illustrates the file format. The first line contains the string `TEXT`. The second line, `NREG: 0`, indicates that a field map contains no region structure and there are no named regions. The next three lines give the integer mesh dimensions $I_{max}$, $J_{max}$ and $K_{max}$. The quantity $DUnit$ is a factor for converting the mesh dimensions to meters. For example, use $DUnit = 39.37$ if the spatial quantities are in inches. Note that coordinates recorded in a field map are in dimensions of meters.

Each data line contains the following information:

- Two dummy region numbers (integers) for compatibility with HiPhi and Magnum. OmniTrak and OmniView ignore these numbers.
- Three real number (in any valid format) giving the node coordinate position in meters.
- Three real numbers given the electric field components $(E_x, E_y, E_z)$ in V/m or the components of magnetic flux density $(B_x, B_y, B_z)$ in tesla.

The order of data line storage is:

```
DO K=0,KMax
  DO J=0,JMax
    DO I=0,IMax
      ...
```

```
    END DO
  END DO
END DO
```

## 17.2 Applications of field tables

Conformal mesh field solutions from **HiPhi** and **Magnum** are required under the following conditions:

- The electric field must be recalculated.
- There are stopping conditions that depend on the system geometry, such as a beam target.

The important implication of the first item is that an electric field table may be used only in the *Track* mode of **OmniTrak**. On the other hand, you may combine a conformal mesh for the electric field and a table for the magnetic field in any tracking mode.

There are several circumstances where field tables are useful:

- Position searches and field interpolations for orbit calculations are faster on the regular meshes of tables. You may save time in complex orbit calculations by creating a matrix file in **PhiView** or **MagView** and using it to define three-dimensional field variations in the **OmniTrak** run.
- The volume for orbit calculations may be small compared to the solution volume in a **HiPhi** or **Magnum** calculation (*e.g.*, the gap region of a dipole magnet). In this case, you save time and improve accuracy in the **OmniTrak** calculation by creating a table that covers only the critical region.
- Tables provide a path to incorporate three dimensional field solutions from other finite-element software into **OmniTrak**.
- With tables it is relatively easy to set up fields that follow specified mathematical forms. You can write your own programs to define any physically-valid field variation.
- You can use the **Mapper** utility (described in Chap. 18) for quick and efficient design studies of dipole field devices.
- Symmetry planes reduce the time for field calculations, but they may introduce small errors in particle orbits. You can use **Mapper** to project tables through a symmetry plane to eliminate reflection planes in long orbit calculations.
- **Mapper** includes sophisticated routines to smooth noisy field data for high-accuracy orbit calculations.

## 17.3 Using field tables in OmniTrak and OmniView

The following two **OmniTrak** commands (defined in Sects. 6.4 and 7.4) are used to load field tables.

**ETABLE3D = TabName [FMult]**
This command loads an electric field table. The quantity $TabName$ is the full name of a data file available in the current directory. The optional real-number parameter $FMult$ is a field scaling factor. Values of the electric field are multiplied by $FMult$ when the file is loaded.

**BTABLE3D = TabName [FMult]**
This command loads values from a magnetic field table. The quantity $TabName$ is the full name of a data file available in the current directory. The optional real-number parameter $FMult$ is a field scaling factor. Values of the magnetic field are multiplied by $FMult$ when the file is loaded.

The position and orientation of tables may be modified with the following commands

**SHIFT E XShift YShift ZShift**
**SHIFT B XShift YShift ZShift**
The commands shift node positions in the electric or magnetic field table. Enter the real-number parameters $X_{shift}$, $Y_{shift}$ and $Z_{shift}$ in the units set by the $DUnit$ command.

**ROTATE E XRot YRot ZRot [RotOrder]**
**ROTATE B XRot YRot ZRot [RotOrder]**
The commands rotate the coordinates and field components of the electric or magnetic field table. The parameters $X_{rot}$, $Y_{rot}$ and $Z_{rot}$ 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. The string parameter contains from one to three of the characters $X$, $Y$ or $Z$. The default is $RotOrder = XYZ$.

Scaling, shift and rotation operations on tables present opportunities for errors. Therefore, it is important to check the validity of field interpolations before proceeding with extensive orbit calculations. The following commands in the *Diagnostics* section of the **OmniTrak** script may be used to return values from modified field tables:

**EPOINT Xp Yp Zp**
**BPOINT Xp Yp Zp**
Record values of the electric or magnetic field 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**
**BSCAN Xs Ys Zs Xe Ye Ze**

Write a series of data lines containing the electric or magnetic field values along a line from $(x_s, y_s, z_s)$ to $(x_e, y_e, z_e)$. Enter coordinates in units set by *DUnit*.

**NSCAN NScan**
**NSCAN 150**
Set the number of intervals for the *EScan* and *BScan* commands.The default value is $NScan = 25$.

Finally, **OmniView** can load tables and superimpose plots of field quantities on orbit plots (Chap. 5).

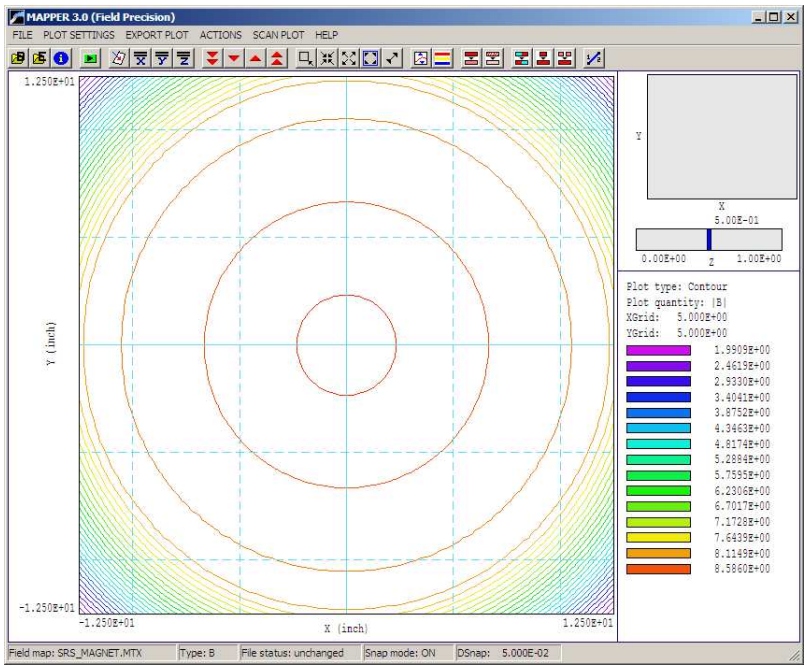


Figure 48: Mapper main window

# 18 Mapper

The interactive utility program `mapper.exe` is a versatile tool to inspect and to modify field tables, You can also create tables for dipole-type fields with specified properties. Figure 17.1 shows the main program screen.

## 18.1 Loading and viewing field tables

The first step to use the plotting and transformation capabilities of **Mapper** is to load an existing table file. The following commands appear in the *File* popup menu.

### LOAD B TABLE
### LOAD E TABLE

Load a file of electric or magnetic field values in the format described in Sect. 17.1. The file must have a name of the form `FPREFIX.MTX`. The choice of $E$ or $B$ determines labels used in plots. **Mapper** issues a save prompt if the current file in memory has been modified.

### TABLE INFORMATION

List the properties of the currently-loaded table.

### SAVE CURRENT TABLE

Save the current table. Specify a different name if you do not want to over-write the existing file.

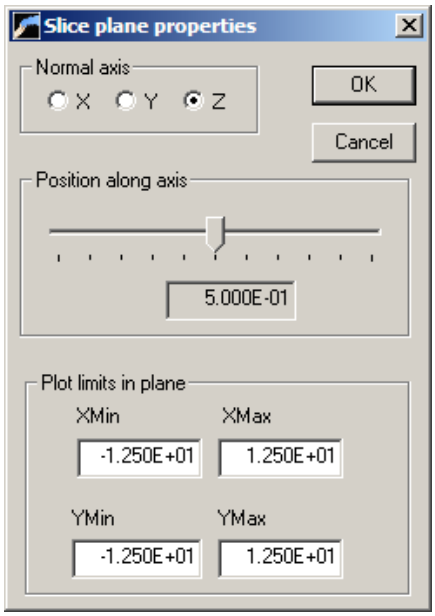


Figure 49: Set slice plane dialog.

**Mapper** creates a default slice plot when the table is loaded. Slice plots are two-dimensional plots that show the variation of table quantities over a plane normal to one of the Cartesian axes. Slice positions are limited to node planes of the table. The program remembers the plot settings if a new table has the same header parameters as the previous one. The following commands in the *Plot settings* popup menu control the slice plane and the plot dimensions and style.

## SLICE PLANE

This command calls the dialog of Fig. 49 to set the slice plane. Set the normal axis with the radio buttons at the top. For example, for a choice of $z$ the plot will be created in the $x$-$y$ plane. You can use the slider bar to set the position along the normal axis or type a value in the box. The range of the slider bar is automatically set to the limits of the solution volume along the normal axis. The boxes at the bottom determine the plot range in the normal plane. The default settings are the limits of the solution volume.

## TOGGLE CONTOUR/ELEMENT

The *Contour* style shows lines of constant value of the plotted quantity. An *Element* plot shows cells of the table color-coded according to the value of the plotted quantity.

## PLOT LIMITS

By default, **Mapper** chooses limits of contour or color values that span the range of values of the plot quantity in the current slice plane. Use this command to set fixed values to compare different planes. Uncheck the *Autoscale* box and supply real-number minimum and maximum values for the plot quantity.

**PLOT QUANTITY**
Choose the quantity to be plotted. The choices are $E_x$, $E_y$, $E_z$ and $|\mathbf{E}|$ for $E$ tables and $B_x$, $B_y$, $B_z$ and $|\mathbf{B}|$ for $B$ tables.

The following commands have functions identical to those of the slice plot menu in **OmniView**, described in Sect. 5.2:

**X NORMAL**
**JUMP DOWN**
**ZOOM WINDOW**
**ZOOM IN**
**EXPAND VIEW**
**GLOBAL VIEW**
**PAN VIEW**
**NUMBER OF CONTOURS**
**TOGGLE GRID DISPLAY**
**TOGGLE SNAP MODE**
**SET SNAP DISTANCE**

## 18.2 Modifying field tables

**Mapper** offers several options to transform field tables. The following commands appear in the *Actions* popup menu:

**SCALE TABLE**
Multiply all field values of the currently loaded table by a scaling factor. This command is useful to change the amplitude of fields or to convert units from other software into (V/m) or (tesla) for use in **OmniTrak**. Supply the factor as a real number in the initial dialog. Use the S*ave current table* command to record the modified values in a table.

**SMOOTH TABLE**
Noisy data may affect the accuracy of orbit integrations in **OmniTrak**. Use this command to smooth the data. **Mapper** uses a sophisticated relaxation routine that preserves the physical nature of the fields as solutions of the Poisson equation. You can apply the command several times to achieve the desired degree of smoothing. Use the *Scan plot* command to check the current state of the data. Use the *Save current table* command to record the modified tables in a table.

**SMOOTHING PARAMETERS**
Set controls for the smoothing process. The number $NCycleS$ is the number of relaxation cycles. Set a high value for more agressive smoothing. The default is $NCycleS = 25$. The quantity $FmaxS$ is a maximum value of $|\mathbf{E}|$ or $|\mathbf{B}|$ for which nodes will be included in the average. For example, in a table you could set a high value for $B_z$ to mark points inside a

material region. In this case, **Mapper** does not change the value at matked nodes or at points adjacent to them. The default value is $FmaxS = \infty$ (*i.e.*, all points included in the average).

## STRETCH TABLE

Change the spatial limits of the table. This command has several uses: 1) shift the table along any three dimensional vector, 2) shrink or expand the table, 3) stretch or compress the table along one or more directions or 4) create a mirror image of the table. When you click on the command, **Mapper** displays a dialog with the dimensional limits of the currently-loaded table. You can implement any of the transformations by entering appropriate new values in the boxes. To make a mirror reflection in $x$, enter limits such that $x_{min} > x_{max}$. You can check the validity of the transformation by loading and viewing the resulting map.

## REFLECT TABLE

Double the extent of the table by reflecting values about a symmetry boundary. In the opening dialog specify the reflection axis (*X, Y* or *Z*) and the reflection type (*Mirror*, *Cusp*). To illustrate the operation, consider a reflection along the $z$ axis. Suppose the original table has dimensions $x_{min}$ to $x_{max}$, $y_{min}$ to $y_{max}$ and $z_{min}$ to $z_{max}$ with indices the range $0 \le i \le I_{max}$, $0 \le j \le J_{max}$ and $0 \le k \le K_{max}$. The final table has the same dimensions and indices in $x$ and $y$ with the following properties in $z$:

$$z'_{min} = z_{min} - (z_{max} - z_{min}), \tag{53}$$
$$z'_{max} = z_{max}, \tag{54}$$
$$K'_{max} = 2K_{max}. \tag{55}$$

The field values in the new section depend on the type of reflection. In a *Mirror* reflection, the quantity parallel to the reflection direction has the same sign and the normal field components are reversed. For example, the field of a magnetic mirror symmetric about the plane $z = 0.0$ obeys the relationships:

$$B_x(-z, x, y) = -B_x(z, x, y), \tag{56}$$
$$B_y(-z, x, y) = -B_y(z, x, y), \tag{57}$$
$$B_z(-z, x, y) = B_z(z, x, y). \tag{58}$$

A *Cusp* type reflection satisfies the conditions

$$B_x(-z, x, y) = B_x(z, x, y), \tag{59}$$
$$B_y(-z, x, y) = B_y(z, x, y), \tag{60}$$
$$B_z(-z, x, y) = -B_z(z, x, y). \tag{61}$$

## ADD TABLES

Superimpose the values in two tables. In the dialog, specify the prefixes of two input tables and a prefix for the output table. Both input tables must be available in the working directory. Optionally, you can specify scaling factors for values in the tables. Note that the input tables must have the same spatial and mesh sizes (*i.e.*, all values in the headers must be equal.)

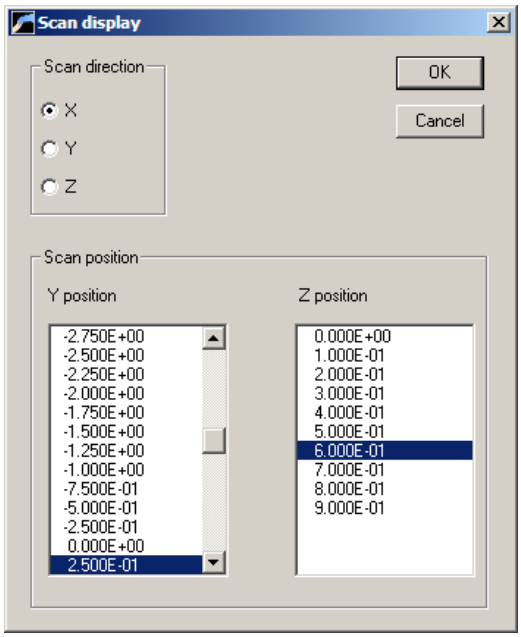


Figure 50: Scan plot dialog.

## 18.3 Scan plots

Use the *Scan plot* command in the main menu to plot a scan of node values in the table along a chosen axis and position in the normal plane. Scan plots are useful for finding individual values in the table and checking the result of operations like smoothing. In response to the command, **Mapper** displays the dialog of Fig. 50. Use the radio buttons at the top to specify the direction of the scan. The boxes at the bottom list the node positions along the axes normal to the scan. **Mapper** switches to the scan view and menu when you press *OK*. The following commands are available in the *Scan* menu:

### SCAN DISPLAY

Return to the scan dialog to change the displayed values.

### PLOT QUANTITY

The choices of plotted quantities are the same as those of slice plots.

### TOGGLE SYMBOL DISPLAY

Display or suppress symbols to mark the node positions along the scan axis.

### OSCILLOSCOPE MODE

In oscilloscope mode, the scan plot assumes characteristics of a digital oscilloscope. **Mapper** superimposes a cross-hair pattern on the graph. Plot values at the intersection are displayed in the information box at the bottom. Move the marker along the plot by moving the mouse. Press *Esc* or the right mouse button to exit oscilloscope mode.

## 18.4 Creating tables

The function of the *Create table* command of **Mapper** is to generate tables for dipole fields. The most familiar example is the field distribution in the gap between the poles of a magnet. Dipole magnets have extensive applications to the transport of charged-particle beams:

- Bending magnets in synchrotrons.
- Weak-focusing cyclotrons and betatrons.
- Sector-field cyclotrons.
- Magnetic spectrometers

Although it is possible to generate numerically-exact fields for these devices with **Magnum**, the process may be time-consuming. The table-generation capability of **Mapper** is useful for quick preliminary designs that can then be examined in detail with full three-dimensional field calculations. With **Mapper**, it is easy to experiment with different pole shapes and mathematically-specified field variations.

Dipole fields generated with the *Create table* have a symmetry plane normal to $z$. The fields follow the relationships of Eq. 58 and satisfy the condition:

$$B_z \gg B_x, B_y. \tag{62}$$

We shall refer to the $z$ axis as the *vertical* direction and to the $x$-$y$ plane as the *horizontal* direction. The field components $B_x$ and $B_y$ equal 0.00 in the midplane.

**Mapper** follows the procedure described in the Ref. 1. The computation involves the following steps:

1. Define the geometry ($x_{min}, x_{max}, ...$) and mesh size ($I_{max}$, $J_{max}$, $K_{max}$) for the table.
2. Use a list of pole boundary vectors to divide the midplane ($z = 0.0$) into regions of free space and one or more magnet gaps.
3. Set midplane variations of vertical field, $B_{z0}(x, y)$, according to mathematical functions or numerical tables.
4. Smooth the fields at the boundaries of gaps for physically-reasonable edge variations.
5. Determine the off-midplane field values using an expansion based on the Maxwell equations. The approximation is valid if the table dimension in z (which is less than or equal to the gap height) is small compared to the transverse gap dimensions.
6. Write the table in the standard format of Sect. 17.1.

The smoothing operation for edge fields gives only a rough approximation for the actual fields. Reference 2 shows that orbits are not sensitive to the form of the edge fields as long as the focal length for edge focusing is long compared to the gap width. The main requirement is to ensure smooth variations for field extrapolations. For reference, the field expansion of step 5

follows from the equations $\nabla \cdot \mathbf{B} = 0$ and $\nabla \times \mathbf{B} = 0$. If the field variation in the midplane is $B_{z0}(x, y)$, then the following expressions give the field components off the midplane for small $z$:

$$B_x(x, y, z) \cong \frac{\partial B_{z0}}{\partial x}\ z, \tag{63}$$

$$B_y(x, y, z) \cong \frac{\partial B_{z0}}{\partial y}\ z, \tag{64}$$

$$B_z(x, y, z) \cong B_{z0} - \left(\frac{\partial^2 B_{z0}}{\partial x^2} + \frac{\partial^2 B_{z0}}{\partial y^2}\right)\ \frac{z^2}{2}. \tag{65}$$

The partial derivatives are calculated from difference operations on the midplane field.

Table generation is controlled by a file with a name of the form `FPREFIX.FTS`, where the suffix denotes *Field Table Script*. The script follows the standard Field Precision conventions for delimiters, comment lines and number format. The syntax to define two-dimensional geometries is quite similar to that of the **Mesh** program. The file has the following structure:

```
GLOBAL
  (Global commands)
END
REGION RegName
  (Region 2 commands)
END
REGION RegName
  (Region 3 commands)
END
REGION RegName
  (Region 4 commands)
END
  ...
ENDFILE
```

The commands of the *Global* section set the size of the field table and control program operation. Each *Region* section represents a magnet gap. You can define up to 39 regions. The following commands appear in the *Global* section:

**XMESH XMin XMax Nx**
**YMESH YMin YMax Ny**
**ZMESH ZMin ZMax Nz**

The three commands are required to set the properties of the table. The keyword is followed by two real-number parameters and an integer. **Mapper** uses dynamic memory allocation, so the size of the table is limited only by the installed memory. You can use any dimensions for the spatial quantities. The code checks that $K_{max}$ is an even number to ensure that a node plane lies in the midplane at $(z_{min} + z_{max})/2$. You can use *Shift* and *Rotation* commands in **OmniTrak** to move or to orient the table.

**TYPE [E,B]**
Specify whether the table should be recorded in the format for an electric or magnetic field file. If the command does not appear, the default is a magnetic field file.

**DUNIT DUnit**
Set dimension units to be recorded in the table file. The quantity *DUnit* is the number of dimension units per meter. For example, if the table limits are given in millimeters, set $DUnit = 1000.0$.

**DGAP DGap**
The quantity *DGap* is the effective gap width for midplane-field smoothing on boundaries between magnets or at the boundary of a field free region. **Mapper** sets a smooth variation of field values for all points within a distance *DGap* of the boundary. Enter the quantity in units set by *DUnit*.

From one to 39 Regions sections may follow the *Global* section. **Mapper** automatically assigns Region 1 as free space. The region initially fills the midplane ($x_{min}$ to $x_{max}$ and $y_{min}$ to $y_{max}$). The field value $B_{z0} = 0.0$ is assigned to associated nodes. You can define additional regions to represent magnet gaps. A region overwrites previously-defined regions. The division of the midplane therefore depends on the order of region sections in the script. You must exercise some caution in choosing the shapes of regions. Because of the method used to identify nodes near boundaries, a region does not erase boundary markers set by previously defined regions. This can lead to field anomalies during the edge-smoothing process. Figure 51 illustrates the incorrect and correct ways to define multiple regions. Here, Region 2 is a cylindrically-symmetric base field. Region 3 is a field perturbation that covers a 60°angle. Finally, Region 4 is a uniform-field area that covers the inner portion of the magnet. If Region 3 is defined as a wedge, the process leaves phantom boundaries in the area covered by Region 4. To generate a correct table, Region 3 should have the shape shown on the right-hand side of Fig. reffig:tabboundary. In this case, boundaries are created only at the intersections of regions.

The *Region* command begins a section. **Mapper** assigns a default name of the form `REGION03`, or you can specify a descriptive name in the command line. The section terminates with the *End* command. The following commands may appear in the section:

**L Xs Ys Xe Ye**
**L = (0.00, 1.00) (5.00, 1.00)**
Define a line vector on the boundary of the region. The four real-number parameters give the coordinates in the midplane of the start and end points in units set by *DUnit*. The sum of line and arc vectors must form a closed, connected outline. **Mapper** orders the vectors, checks that adjacent start and end points are connect and ensure that the end point of the last vector equals the start point of the first.

**A Xs Ys Xe Ye Xc Yc**
**A = (5.00, 0.00) (1.00, 4.00) (1.00 0.00)**
Define an arc vector on the boundary of the region. The six real-number parameters give the

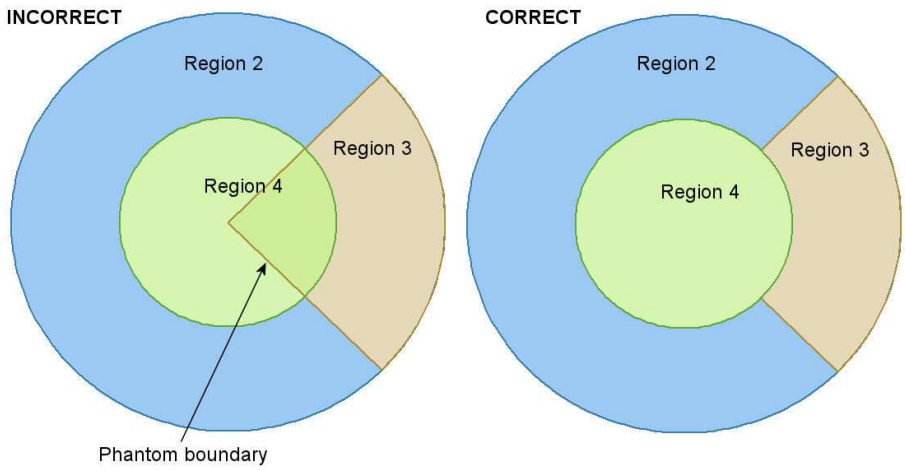


Figure 51: Region boundaries in a field table. Region 3 overwrites Region 2, and Region 4 overwrites both. The wedge shape of Region 2 on the left-hand side leaves a phantom boundary beneath Region 3. The right-hand side illustrates the correct shape for Region 2.

coordinates in the midplane of 1) the start point, 2) the end point and 3) the center point. **Mapper** checks that the arc coordinates are consistent (*i.e.*, the start and end points are equidistant from the center).

**FFUNC > f($x,$y)**
**FFUNC > 0.15*sin($x/3.67)*cos($y/4.55)**
Define the spatial variation of magnetic field (in tesla) over the region from a mathematical function. The function is a string of up to 230 characters that follows the > symbol. The symbols $x and $y denote the position in the midplane in units set by *DUnit*.

The function parser recognizes the following entities:

- Spatial variables `$x` and `$y`.
- Real and/or integer numbers in any valid format (*e.g.*, `3.1415`, `476`, `1.367E23`, `6.25E-02`, `8.92E+04`,... ). Integers are converted to real numbers for evaluation.
- Binary operations: `+` (addition), `-` (subtraction), `*` (multiplication), `/` (division) and $\wedge$ (exponentiation).
- Functions: `abs` (absolute value), `sin` (sine), `cos` (cosine), `tan` (tangent), `ln` (normal logarithm), `log` (base 10 logorithm), `exp` (normal exponent) and `sqt` (square root).
- Up to 20 sets of parentheses to any depth.
- Any number of space delimiters.

The parser conforms to the standard algebraic rules and features comprehensive error checking. Errors may include unbalanced parentheses, unrecognized characters and sequential binary operations. To illustrate a valid example, the expression

```
1 - exp(-1.0*(($z^2 + $r^2)/24))
```

corresponds to

$$1 - \exp\left[-\left(\frac{z^2 + r^2}{24}\right)\right]. \tag{66}$$

**FFUNC > f($r)**
**FFUNC > 0.25*exp(-0.567*$r)**
This alternate form of the *FFunc* command is convenient for fields that have cylindrical symmetry in the midplane. The variable `$r` replaces the expression `sqt($x*$x + $y*$y)`.

**FTABLE [X,Y,R] TabName**
**FTABLE(R) = FGradient.DAT**
The spatial variation of magnetic field in the midplane may also be defined through interpolation on A table of values. The tables have the same format as the field modulation tables discussed in Sect. 8.1. Tables are inherently one-dimensional, so the field MAY vary only along one direction. Each data line of the table contains two values: spatial position and magnetic field (in tesla). The string parameter $[X,Y,R]$ sets the interpretation of the independent variable: $x$, $y$ or $r = \sqrt{x^2 + y^2}$.

## 18.5 Using Mapper to design a dipole magnet

The example described in this section illustrates techniques for dipole magnet design using field tables constructed with **Mapper**. The calculation addresses a permanent-magnet electron spectrometer designed and fabricated for Los Alamos National Laboratory. The design was confirmed with extensive experiments described in Ref. 3. Figure 52 shows the midplane geometry. The field has the uniform value $B_z = -0.24$ tesla inside a shaped pole. The pole is rectangular with a small extension to ensure uniform edge fields near the entrance point. The goal is to provide confinement in the vertical direction while ensuring a horizontal point focus as a detector plane over a wide range of electron energy (10-24 MeV). The target plane is defined by a line passing through the injection point at an angle -15°.

The input script `WHEBY.FTS` to generate the field table `WHEBY.MTX` is shown in Table 19. The vectors of the single region section define the outline shown in Fig. 52. The command `FFUNC > -0.24` sets a uniform field value over the gap region. The output file with 111,683 nodes is created in a fraction of a second.

In the **OmniTrak** run, we want to stop electrons at a target plane that is not normal to one of the Cartesian axes. One possible approach is to rotate the field solution and initial orbit vectors by 15° so that the detector lies in the plane $y = 0.0$. We use an alternative technique for this solution that takes advantage of the **AMaze** conformal mesh. We set up a zero-field electrostatic solution that covers the region near the detector. The target, a fixed-potential region defined by an extrusion, occupies the volume below the 15° line. **OmniTrak** sets this region as *Material* by default so that electrons stop when they enter any element of the target. Table 20 shows the file `WHEBY01.OIN` that controls an **OmniTrak** run to produce the orbits of Fig. 52. The script loads both the conformal-mesh electric field solution and the magnetic field

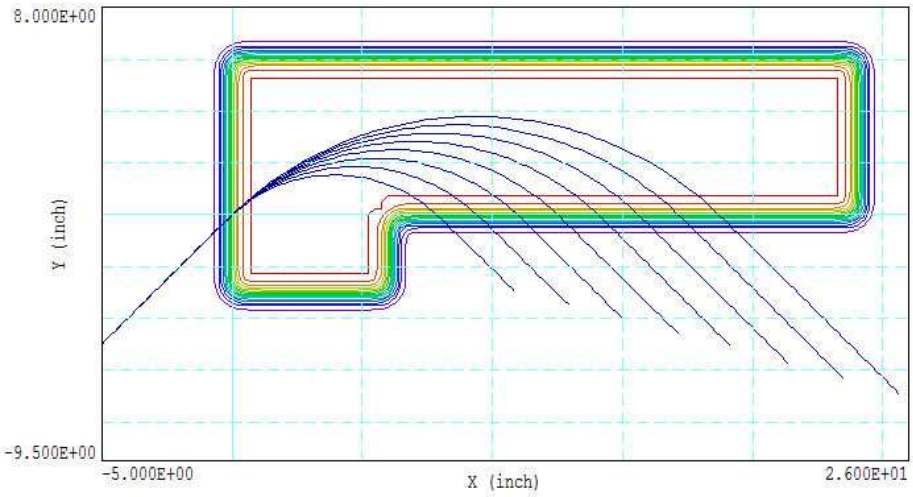


Figure 52: Midplane field variation and central electron orbits with kinetic energy from 10.0 MeV to 24.0 MeV (Dimensions in inches). The red cntour corresponds to $|\mathbf{B}| = 0.24$ tesla. Edge smoothing parameter: $DGap = 0.7$".

table. The electrons start outside the volume covered by the electric field solution. Therefore, the command

```
Boundary  -9.50  -9.50  -0.25  26.00  8.00   0.25
```

is necessary to activate the ballistic mode (Sect. 6.3).

Figure 53 shows results of a second run (`WHEBY02.OIN`) to check focal properties in the horizontal and vertical directions. The true-scale projection of Figure 53*a* shows a horizontal point focus at the detector plane for good energy resolution. Figure 53*b* shows an expanded plot of vertical orbits. Focusing and defocusing at the inclined entrance and exit edges of the magnet are apparent. The systems has good vertical acceptance over the full energy range.

**References**

1. S. Humphries, R. Baltrusaitis, C. Ekdahl, C. Young and C. Warn, *BMAP – dipole magnet field analysis and orbit tracking*, in **Computational Accelerator Physics** edited by R. Ryne (American Institute of Physics, New York, 1994), 590.

2. S. Humphries, **Principles of Charged Particle Acceleration** (Wiley, New York, 1986), Sect. 6.8

3. S. Humphries, R. Chrein, C. Ekdahl, C. Young, C, R. Baltrusaitis and C. Warn, *Compton spectrograph for Bremsstrahlung spectra measurements* (Proc. IEEE Conf. Plasma Sci., ISBN: 0-7803-0147-1, 1991).

Table 19: Contents of the file WHEBY.FTS

```
GLOBAL
  XMesh -9.50  26.00  142
  YMesh -9.50   8.00   70
  ZMesh -0.25   0.25   10
  Type B
  DUnit 39.37
  DGap 0.700
END
REGION WHEBY
  L   0.00  -3.00   6.00  -3.00
  L   6.00  -3.00   6.00   0.00
  L   6.00   0.00  24.00   0.00
  L  24.00   0.00  24.00   6.00
  L  24.00   6.00   0.00   6.00
  L   0.00   6.00   0.00  -3.00
  FFUNC > -0.24
END
ENDFILE
```

Table 20: Contents of the file WHEBY01.OIN

```
FIELDS
  DUnit = 39.37
  EField3D WHEBY.HOU
  BTable3D WHEBY.MTX
  Boundary  -9.50  -9.50  -0.25  26.00  8.00   0.25
END
PARTICLES TRACK
  PList
   0.0 -1.0  10.0E6 -9.4999  -9.4999  0.000  0.707  0.707  0.000
   0.0 -1.0  12.0E6 -9.4999  -9.4999  0.000  0.707  0.707  0.000
    ...
   0.0 -1.0  22.0E6 -9.4999  -9.4999  0.000  0.707  0.707  0.000
   0.0 -1.0  24.0E6 -9.4999  -9.4999  0.000  0.707  0.707  0.000
  End
  Dt = 2.0E-11
END
DIAGNOSTICS
  PartList
END
ENDFILE
```

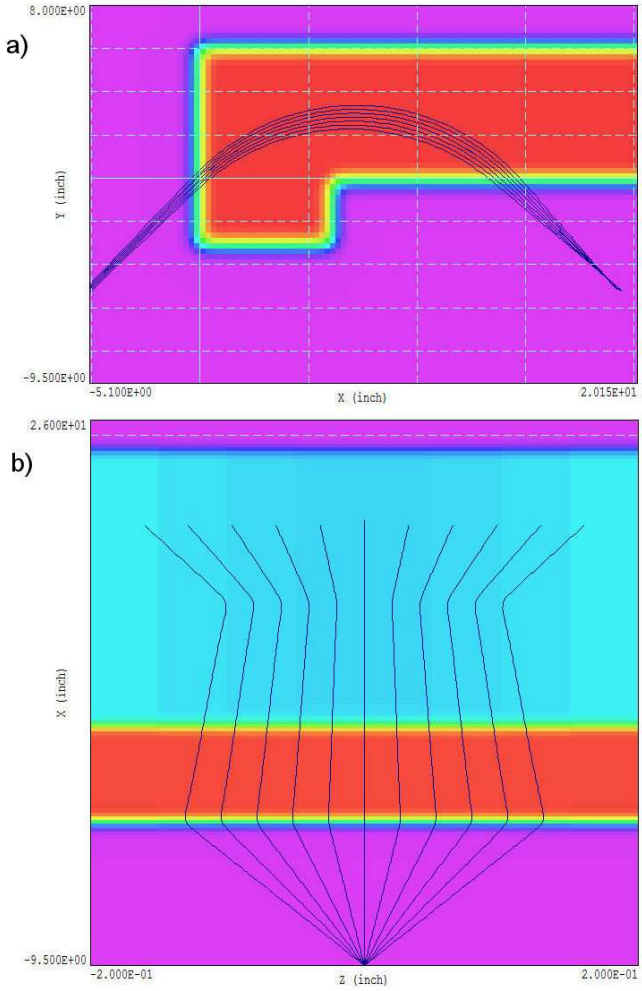


Figure 53: Trajectories to illustrate focusing of 18 MeV electrons. *a*) True-scale projection in the horizontal plane. *b*) Orbits projected to the $z$-$x$ plane, plot expanded in $z$.

# 19 OmniTrak output files

**OmniTrak** creates a variety of output files. The listing file (`OPREFIX.OLS`) contains valuable information in text format. The contents of the file depend on commands in the *Diagnostics* section. **OmniTrak** produces two types of binary files:

- Electric field files are written in response to the *EDump* command. They have a name of the form `HPREFIX.HOU` and are in the standard **HiPhi** format (either text or binary).
- Files of beam-generated magnetic field may be created in *RelBeam* mode simulations in response to the *BBDump* command. They have a name of the form `MPREFIX.GOU` and are in the standard **Magnum** formats.

You can load files created using the *EDump* or *BDump* commands into **OmniView** for plotting or use them as input for subsequent **OmniTrak** runs. The files may also be inspected with **PhiView** or **MagView**. The BBDump command creates a standard Magnum with the following characteristics:

- The solution has a single region with the name *BBVolume*. The region is marked as *I* (included).
- The node values of $\phi$ and $\psi$ equal zero. In other words, materials make no contributions to the beam-generated fields.
- The components of applied magnetic field, $H_x, H_y$ and $H_z$, represent contributions from the beam.
- The relative magnetic permeability has the value $\mu_r = 1.0$ in all elements.

You can use MagView to investigate the beam-generated field and also display values in OmniView.

**OmniTrak** normally generates orbit vector files (`OPREFIX.OOU`) in text format for all runs. The files 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 shows an extract from a 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. The line contains information on the particle mass and charge as well as current for calculations in the *SCharge*, *RelBeam* and *FEmit* modes. There is no particle information for calculations in the *FLine* mode. Subsequent lines 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: OmniTrak plot file extract

```
OmniTrak Plot File (Field Precision)
NPart:    4
XMin: -5.000000E-03
YMin: -5.000000E-03
ZMin: -1.000000E-02
XMax:  5.000000E-03
YMax:  5.000000E-03
ZMax:  1.000000E-02
DUnit:  1.000000E+03
-----------------------------------------------------
Particle    1  Mass:   5.4881E-04  Charge:  -1.0000E+00
  0.000000E+00  3.000000E-03  0.000000E+00  0.000000E+00
  1.171335E-11  3.000000E-03  2.489503E-11  1.323749E-06
  2.342670E-11  3.000000E-03  9.941516E-11  5.294994E-06
  3.514005E-11  3.000000E-03  2.225707E-10  1.191373E-05
  ...
 -1.000000E+00 -1.000000E+00 -1.000000E+00 -1.000000E+00
-----------------------------------------------------
Particle    2  Mass:   5.4881E-04  Charge:   1.0000E+00
  0.000000E+00  2.000000E-03  0.000000E+00  0.000000E+00
  1.171335E-11  2.000000E-03 -2.708960E-11 -1.327755E-06
  ...
  9.136414E-10  1.873576E-03 -8.270741E-08 -8.158983E-03
 -1.000000E+00 -1.000000E+00 -1.000000E+00 -1.000000E+00
-----------------------------------------------------
Particle    3  Mass:   1.0000E+00  Charge:   1.0000E+00
  0.000000E+00  1.000000E-03  0.000000E+00  0.000000E+00
  5.000000E-10  1.000000E-03 -2.935036E-11 -1.331028E-06
  ...
  3.768097E-08  9.329809E-04 -1.559170E-08 -7.689098E-03
 -1.000000E+00 -1.000000E+00 -1.000000E+00 -1.000000E+00
-----------------------------------------------------
Particle    4  Mass:   2.0000E+02  Charge:  -1.0000E+00
  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
  7.071068E-09  2.043055E-11  2.988346E-11  1.332288E-06
  ...
  5.256115E-07  5.649159E-08  1.020528E-07  7.499931E-03
 -1.000000E+00 -1.000000E+00 -1.000000E+00 -1.000000E+00
```

# 20 Optimizing cathode surfaces in MetaMesh

The procedures used in **MetaMesh** produce the best mesh possible within the limits of the user-specified foundation mesh and objectgeometries. 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 may 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 54 illustrates the geometry. The full original script is listed at the end of this chapter. 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. 54that although the electrodes follow the general defined shapes, there are many imperfections along the edges. As defined, the cathode surface is inadequate for the application. The material in this chapter shows 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 set the *XMesh* and *YMesh* sections to:

```
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. 54. Modification of the *ZMesh* section eliminates the problem:

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

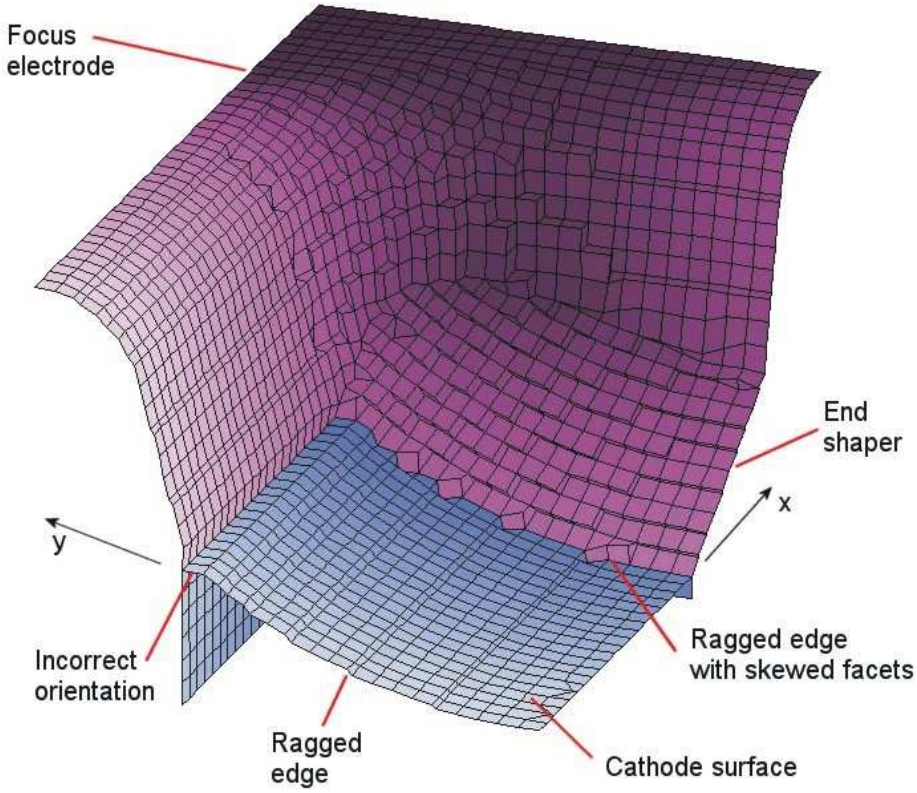


Figure 54: Assembly definitions and the result of the initial fitting attempt.

Next, consider the ragged edge at $x = 0.0$ (bottom of Fig. 54). The 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. 55) of the part at $x = 0.0$ and **MetaMesh** will dutifully try to fit it.

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 *Extrusions* exist in free space as in

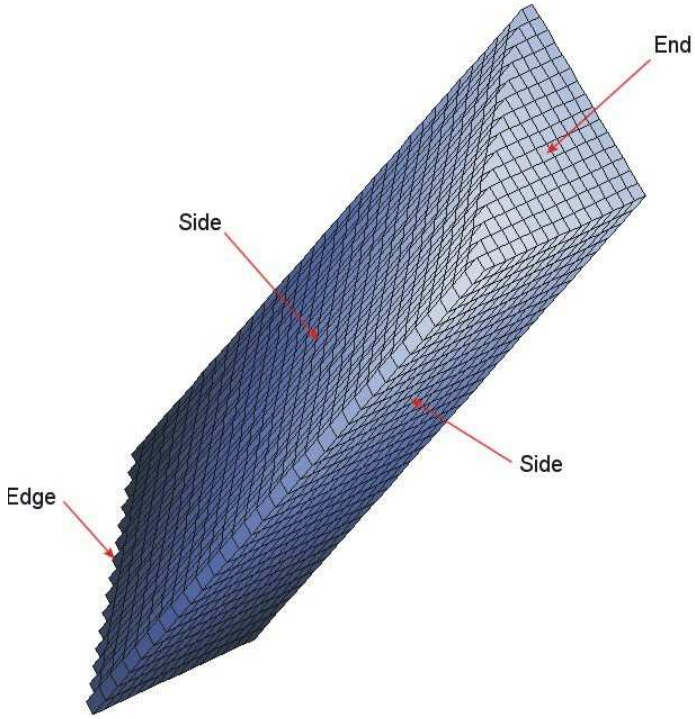


Figure 55: Terminology for the *Extrusion* model. A ragged edge may occur if MetaMesh must fit facets on both the sides and ends.

Fig. 55. 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. 54 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**: 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. 56:

```
Region Cathode
Type Extrusion SideFit

Name EndShaper
Type Transition SideFit
```

### Control script before improvements

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

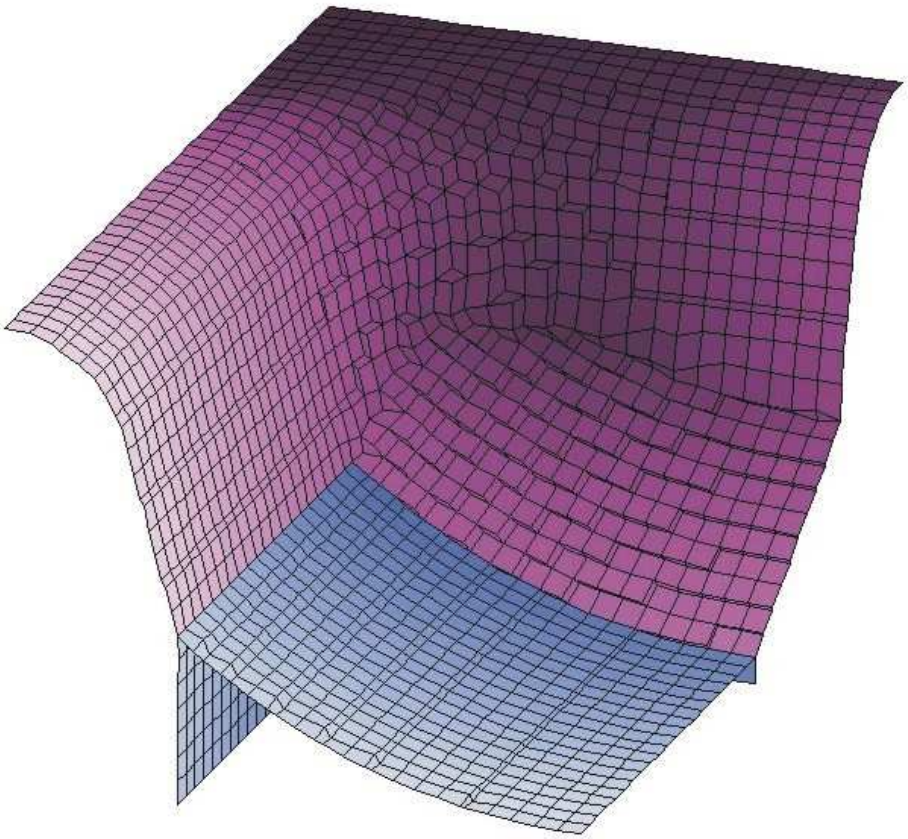

Figure 56: Corrected mesh showing the effects of a corrected foundation mesh and application of the *SideFit* option.

```
  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
```

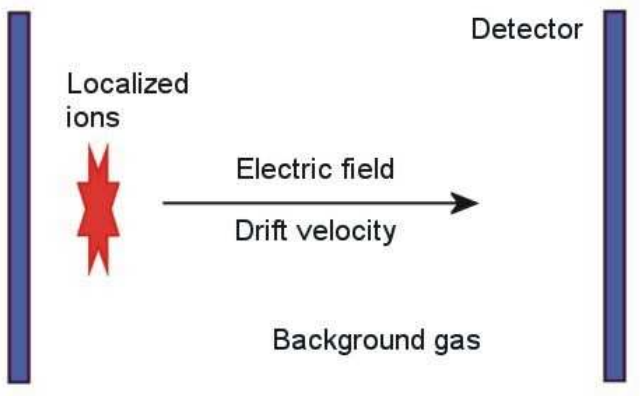


Figure 57: Principle of the ion-mobility mass spectrometer.

# 21 Features for ion-mobility spectrometry

**OmniTrak** includes useful features for designing time-of-flight ion-mobility spectrometers. Figure 57 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:

$$\mathbf{v} = \mu \mathbf{E}, \tag{67}$$

where $\mathbf{E}$ is the electric field vector (in V/m) and $\mu$ is the mobility (in $m^2$/V-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 $\mu$.

Because the drifting ions follow lines of $\mathbf{E}$, the electric-field tracking capabilities of **OmniTrak** (Chapter 15) are ideal for the 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 $\Delta t$ for an ion to move a distance $\Delta s$ is

$$\Delta t = \frac{\Delta s}{|\mathbf{v}|} = \frac{\Delta s}{\mu |\mathbf{E}|}. \tag{68}$$

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

$$t_{AB} = \int_A^B \frac{ds}{\mu |\mathbf{E}|}. \tag{69}$$

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

$$tau_{AB} = \mu t_{AB} = \int_A^B \frac{ds}{|\mathbf{E}|}. \tag{70}$$

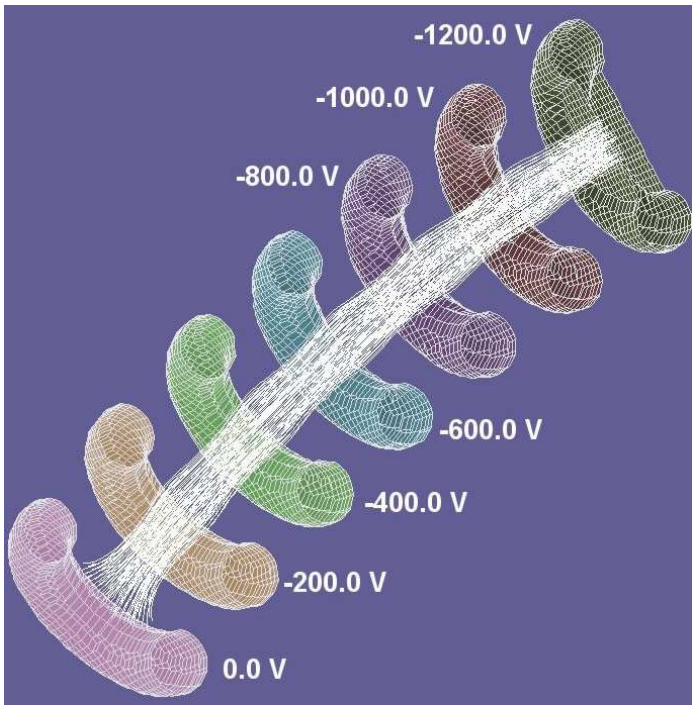


Figure 58: Benchmark calculation of a curved ion-mobility spectrometer showing a cutaway view of electrodes to create an accelerating electric field and selected ion orbits for an intial plasma radius of 0.4 cm.

The quantity $\tau$ (in $m^2/V$) depends only on the nature of the electric field solution. The properties of the background gas and ions are contained in $\mu$. 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 Fig. 58. 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 $\tau$ 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 values of $x$ (displacement in the direction of curvature). Table 22 shows the **OmniTrak** script to control the run. Figure 59 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 versus the start position in x. The ions that start at the bottom of Fig. 59 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. 59 require additional time to pass through the low-field region near the entrance.

In a second run (`MOBILITYDEMO02`) 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. 58. In response to the *FLineList*

Table 22: File MOBILITYDEMO01.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
```

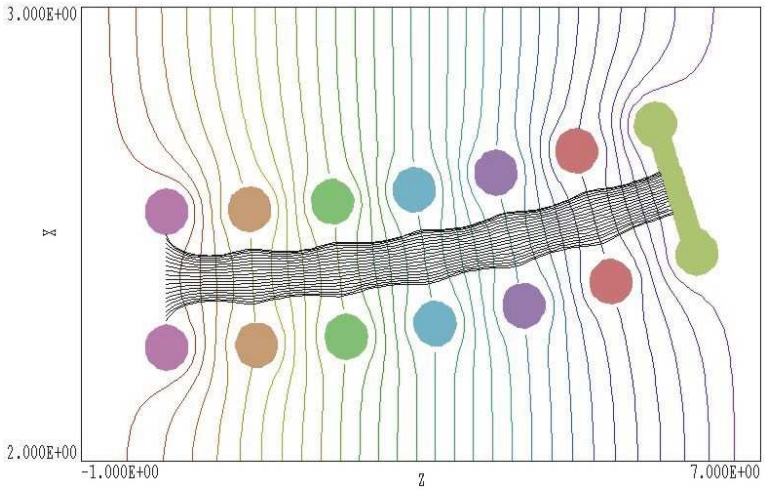


Figure 59: View of equipotential lines and ion orbits in the plane $y = 0.0$ for run MOBILITYDEMO01.

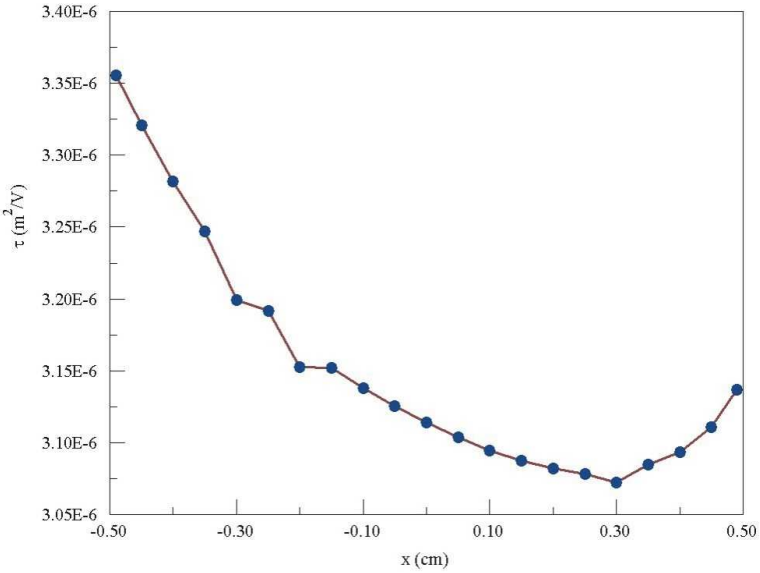


Figure 60: Plot or normalized transit time as a function of initial position in $x$ for ions initiated on the line $y = 0.0$, $z = 0.0$.

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 $\overline{\tau} = 3.14 \times 10^{-6}$ m$^2$/V. If the ion mobility is $\mu = 1.36 \times 10^{-4}$ m$^2$/V-s, the predicted average transit time is 23.1 ms. The dispersion in normalized transit time is $\Delta\tau = 4.69 \times 10^{-8}$ m$^2$/V. Therefore, we expect that the field-magnitude and trajectory variations limit the detector resolution to about $R \geq \Delta\tau/\overline{\tau} = 1.5\%$.

For reference, the absolute value of mobility $\mu$ is related to the reduced mobility $\mu_0$ (defined at standard temperature and pressure) by:

$$\mu = \mu_0 \left(\frac{760}{P}\right) \left(\frac{T + 273.15}{273.15}\right). \tag{71}$$

In the equation $P$ is the pressure in mm of mercury and $T$ is the temperature in °C.

Table 23: Statistical analysis listing, MOBILITYDEMO02

```
Number of values:  239
Normalized transit time (Tau = 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)
```

# Index


algebraic expressions, 54
angular divergence, 87
atomic mass unit, 61

ballistic mode, 46, 67
batch file control, 31

charged-particle beams
    beam-generated forces, 78, 94
    paraxial approximation, 94
    references, 8
Child law, 25, 85
circular accelerators, 70
convergence conditions, 24
conversion, centigrade to eV, 87
current, particle, 79

distribution analysis, 16

electric field solution, 25, 43
electron charge, 61
emission surface, 11, 23, 62, 84

field
    dipole, 140
    importing information, 132
    modifications in OmniTrak, 125
    recalculation, 44
    static approximation, 52
    symmetry, 132
field line
    polarity, 24
    tracking, 23, 121
field modulation, 72
field table, 46, 50, 130
    applications, 132
    format, 130
    rules of use, 46, 50
Fowler-Nordheim functions, 110

GenDist, 6, 15, 75, 128

HiPhi
    Boundary command, 45
    run setup, 12

magnetic field
    applied, 48, 98
magnetic field solution, 48
Mapper, 135
    application example, 144
    create dipole tables, 140
    reflection symmetry, 138
    scan plots, 139
    slice plots, 135
    transform tables, 137
Mapper commands
    Add tables, 138
    Arc, 142
    Contour/element, 136
    Create table, 140
    DGap, 142
    DUnit, 142
    FFunc, 143
    FTable, 144
    Line, 142
    Load B table, 135
    Load E table, 135
    Oscilloscope mode, 139
    Plot limits, 136
    Plot quantities, 137
    Reflect table, 138
    Scale table, 137
    Scan display, 139
    Slice plane, 136
    Smooth table, 137
    Smoothing parameters, 137
    Stretch table, 138
    Table information, 135
    Type, 142
    XMesh, 141
Mapper script, 141
    Global commands, 141
    Region commands, 142
material default properties, 71
Maxwell distribution, 87
MetaMesh
    source surface definition, 63, 84
    turning, 11

variable resolution, 9
modulation function, 52
dual, 55
two voltage, 56

Neumann boundary, 9, 69

OmniTrak, 6
batch file run, 31
Diagnostics section, 21, 125
emission surface, 11, 84
features, 5, 6
Fields section, 21, 48
file formats, 148
input/output files, 6, 14, 19
instruction manual, 29
interactive run, 28
listing file contents, 15
magnetic field procedure, 101
package components, 6
Particles section, 21, 59, 78, 84, 94, 110, 113, 121
relativistic approximation, 94, 96
RelBeam mode symmetry, 103
restart procedure, 92
run setup, 7, 12, 19
script format, 14, 20
secondary emission model, 113
space-charge solution procedure, 78, 84
tracking modes, 22
OmniTrak commands
Avg, 80
BBDump, 26, 128, 148
BBElem, 26, 104
BBMesh, 26, 104
BBPoint, 127
BBScan, 127
BBSymPlane, 105
BDump, 128, 148
BField3D, 22, 23
Boundary, 46, 67
BoundZUp, 11
BPoint, 126, 133
BScan, 126, 134
BTable3D, 133
ChangePot, 44
DiagPlane, 68
Ds, 24, 123
Dt, 23, 30, 65
Dt DtRef, 65
DUnit, 22, 45
EBound, 45
EDirect, 88
Edit file, 28
EDump, 25, 127, 148
EField3D, 22, 23, 43
EField3D2, 57
Emit, 23–26, 64, 86, 112, 123
EPoint, 125, 133
EScan, 126, 134
ETable3D, 46, 133
FFile, 24, 122
FLineList, 129
FList, 122
JLimit, 86
KELimit, 67
ListOn, 65
ListType, 66
Material, 71
MaxCycle, 25, 44, 79
ModFunc, 54, 55
ModFunc2, 57
MultFact, 129
NCycle, 80
NScan, 127, 134
NStepMax, 67
Omega, 45, 79
OrbInfo, 66
Parallel, 46
PartDist, 128
PartFile, 23, 128
PartList, 128
Pause run, 29
PeriodPlane, 70
PFile, 22, 62, 78
PList, 62, 78
PlotOff, 66
PlotSkip, 66
RecordPlane, 69
Reference, 129
ReflectPlane, 69
RegList, 129

RelApprox, 97
ResTarget, 25, 44, 79
Rotate, 44, 133
Secondary, 114
SecondList, 116
SecondParam, 116
Setup, 28
Shift, 43, 133
Slice plane properties, 35
SMax, 67, 123
Start run, 29
Stop run, 29
StopPlane, 68
summary, 59
Suppress, 88, 112
TMax, 67
TSource, 87
Vacuum, 71
OmniTrak modes, 7, 21
FEmit, 26, 59, 110
FLine, 23, 59, 121
RelBeam, 26, 59, 94
SCharge, 24, 59, 78, 84
Track, 21, 59
OmniTrak script
Diagnostics section, 21
Fields section, 21, 43
Particles section, 21
structure, 20
OmniView, 6
changing view, 35, 39
features, 5, 32
making plot files, 38
plot types, 16, 35
surface plot controls, 38
OmniView commands
BField2D, 48
BField3D, 50
BTable3D, 50, 133
BUni, 48
Close data files, 33
Copy to clipboard, 38
Default printer, 38
Displayed regions, 40
Edit file, 33
Expand view, 36
Field plot style, 36
File information, 33
Global view, 36
Load HiPhi solution, 32
Load Magnum solution, 33
Load named view, 41
Load Plot file, 33
Magnification mode, 38
Number of contour lines, 36
Orbit cut planes, 37
Orbit filters, 37
Pan view, 36
Plot limits, 36, 40
Plot quantity, 39
Plot style, 39
Region cut planes, 40
Reset plots on load, 38
Rotate, 49, 50
Save named view, 41
Save plot file, 38
Shift, 49, 50
Snap distance, 37
Toggle grid, 37
Toggle snap mode, 37
Zoom in, 35
Zoom window, 35

particle starting time, 72
planar beam gun, 106
Plot views, saving, 41
PRT file format, 61

reflection conditions, 69
relative residual, 25
relativistic approximation, 94, 96

secondary emission, 113
coefficient values, 114
source limit, 86
source surface, 63
source temperature, 87
stopping conditions, 66, 68, 71, 72
superposition
electric field, 56

tabular functions, 52
temporal modulation, 52

text editor, 28, 33
time step
    automatic calculation method, 64
    calculator tool, 30
tools
    circular beam generator, 30, 97
    rectangular beam generator, 30
    time-step calculator, 30

virtual emission surface, 25

work function, 26, 110