

Email: [techinfo@fieldp.com](mailto:techinfo@fieldp.com)

URL: <http://www.fieldp.com>

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Program function . . . . .	4
1.2	Walkthrough example . . . . .	5
1.3	Theoretical background . . . . .	12
<b>2</b>	<b>Organizing HiPhi calculations</b>	<b>17</b>
2.1	Procedures and files . . . . .	17
2.2	Script conventions . . . . .	18
<b>3</b>	<b>Script commands</b>	<b>20</b>
3.1	Program control . . . . .	20
3.2	Basic material properties . . . . .	22
3.3	Modeling small details in a large-scale solution . . . . .	24
<b>4</b>	<b>Variations of potential and material quantities from tables and mathematical functions</b>	<b>28</b>
4.1	Program capabilities . . . . .	28
4.2	Function syntax and table format . . . . .	30
4.3	Benchmark example . . . . .	31
<b>5</b>	<b>Running the HiPhi program</b>	<b>33</b>
5.1	Interactive operation . . . . .	33
5.2	Automatic runs under batch file control . . . . .	35
5.3	Calculations with large meshes . . . . .	36
<b>6</b>	<b>PhiView – file operations and plane plots</b>	<b>37</b>
6.1	File operations . . . . .	37
6.2	Plane plots . . . . .	41
6.3	Saving and loading views . . . . .	44
<b>7</b>	<b>PhiView – numerical analysis</b>	<b>46</b>
7.1	Automatic analyses operations . . . . .	46
7.2	PhiView analysis scripts . . . . .	50
7.3	Script commands . . . . .	51
<b>8</b>	<b>PhiView – slice plots</b>	<b>54</b>
8.1	Setting the slice view . . . . .	54
8.2	Setting slice plot properties . . . . .	55
8.3	Analyses in a slice . . . . .	58
8.4	Scan menu . . . . .	58
8.5	Slice vector tools . . . . .	62
<b>9</b>	<b>PhiView – surface plots</b>	<b>64</b>

<b>10 Customizing PhiView</b>	<b>70</b>
10.1 Defining calculated quantities . . . . .	70
10.2 Introduction to RPN . . . . .	72
10.3 Variables, operations and parameters . . . . .	73
10.4 Setting up the calculated-quantity sections . . . . .	75
<b>11 Force calculations in PhiView</b>	<b>78</b>
11.1 Surface integrals for force and torque . . . . .	78
11.2 Expressions for electrostatic forces . . . . .	79
11.3 Commands for force calculations . . . . .	80
11.4 Example . . . . .	81
<b>12 Calculating mutual capacitance</b>	<b>84</b>
<b>13 HiPhi file formats</b>	<b>87</b>

---

# 1 Introduction

## 1.1 Program function

**HiPhi** calculates electrostatic fields in three-dimensional systems. The program generates either *dielectric* or *conductive* solutions. In dielectric solutions, regions in the solution-space may represent electrodes, dielectrics and space-charge density. In conductive solutions, the regions correspond to electrodes or resistive media. You may include up to 250 material types in a solution. **HiPhi** has several advanced capabilities:

- High accuracy calculations using finite-element techniques on conformal hexahedron meshes.
- Unmatched computation speed through structured meshes optimized for multicore Windows PCs.
- Dynamic memory allocation for large mesh capacity (20-50 million elements on a standard PC).
- Techniques to assign continuous spatial variations of potential, dielectric constant, conductivity and/or space-charge density to regions from mathematical expressions.

The **HiPhi** package includes four programs:

- `geometer.exe` – interactive environment for modeling three-dimensional assemblies.
- `metamesh.exe` – generation of finite-element meshes from geometric information.
- `hiphi.exe` – generates electrostatic solutions interactively in a window or automatically under batch file control
- `phiview.exe` – postprocessor generates plots and calculates field quantities from **HiPhi** solution files.

**HiPhi** has the power to address the full range of your applications and keep pace with your growing expertise. Learning the program can be relatively easy if you proceed in manageable steps. We suggest the following sequence of activities:

- Follow the walkthrough example in the next section. This will give you quick introduction to the programs.
- Section 1.3 gives a brief review of electrostatic theory and introduces important terms.
- Read Chap. 2 to understand the procedure for **HiPhi** calculations and the organization of data files.
- Try some of the ready-to-run solutions supplied in the example library. The examples illustrate many important techniques and may serve as templates for your applications.

- Chapters 6 through 9 gives detailed descriptions of the capabilities of the **PhiView** post-processor.
- Tackle advanced topics when you have some experience with the programs. Chapter 3 describes how to modify **HiPhi** scripts to invoke advanced functions. Other topics include creation of microscopic solutions to model small details (Sect. 3.3), definition of continuous spatial variations of potential and material properties (Chap. 4) and automatic operation of programs in the background under batch file control (Sect. 5.2).
- Chapter 10 describes how to customize the operation of **PhiView** for your application. You can change the order of plot and analysis quantities and create your own definitions.
- Chapter 11 explains how to use **PhiView** to calculate electrostatic forces. Chapter 12 reviews techniques for calculating mutual capacitances in a multi-electrode system.
- Finally, Chapter 13 describes **HiPhi** output file formats. The information is useful if you want to port data to your own analysis programs.

## 1.2 Walkthrough example

A quick way to understand the solution procedure is to step through an example. The **WALKTHRU** calculation describes fields created by a spherical high-voltage electrode on a dielectric support at an offset position inside a grounded box. Move the file **WALKTHRU.MIN** (supplied in the example library) to a working directory. Table 1 shows a portion of the input file to define the system geometry. The **MetaMesh** manual describes the format and logic of the script. The listed parts are: 1) a box representing vacuum that fills the entire solution volume, 2) the cylindrical dielectric support and 3) the spherical electrode. To ensure the proper assignment of node region numbers, the dielectric regions appear *before* the fixed-potential electrode region. Not listed are five parts that set a fixed-potential condition on the boundaries of the solution volume at  $y_{max}$ ,  $x_{min}$ ,  $x_{max}$ ,  $z_{min}$  and  $z_{max}$ . Because the system is symmetric about  $y = 0.0$ , we leave the boundary  $y_{min}$  unassigned. Therefore, the boundary automatically assumes the special Neumann condition (*i.e.*, electric fields parallel to the surface).

To begin, we shall use **Geometer** to check the geometry of the assembly. Run the program, choose the command *File/Load script* and pick **WALKTHRU.MIN**. Figure 1 shows views normal to the  $y$  and  $x$  axis. The display was created by modifying the default view with the following operations:

- Click on *View/Toggle parts/regions* to display objects by their region number.
- Use *View/Displayed parts/regions* to bring up the view settings dialog. To see the internal parts, uncheck the *Solid* box for Region 4 so that the boundary is displayed as a wireframe. (Note that Region 1 is not displayed by default.)
- Click the *View/Grid control* command and uncheck the *Display grid* box.
- Choose *View/X normal axis* or *View/Y normal axis*.

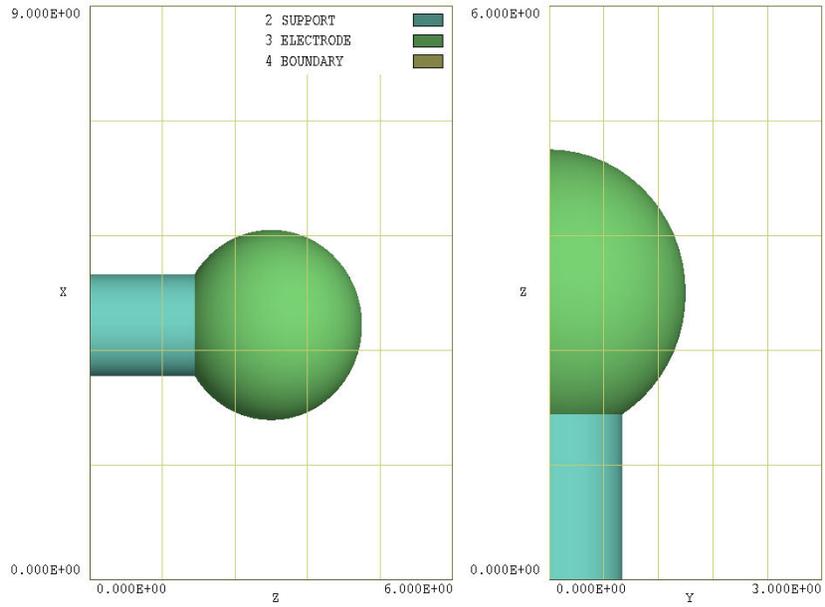


Figure 1: Orthographic views of the WALKTHRU geometry. Left: view normal to  $y$  axis. Right: view normal to  $x$  axis.

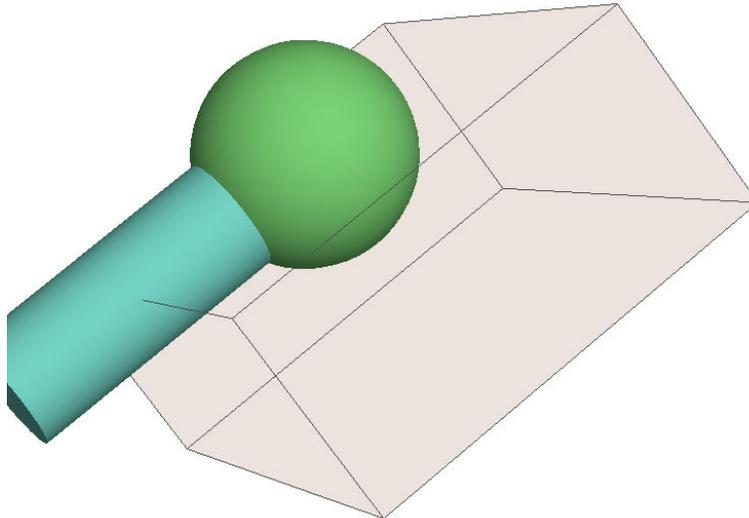


Figure 2: Perspective view of the WALKTHRU geometry. The boundaries of the solution volume have been shaded.

Table 1: Excerpts from file WALKTHRU.MIN, input to **MetaMesh**

```
GLOBAL
  XMesh
    0.00  9.00  N 90
  End
  YMesh
    0.00  3.00  N 30
  End
  ZMesh
    0.00  6.00  N 60
  End
  RegName 1 Vacuum
  RegName 2 Support
  RegName 3 Electrode
  RegName 4 Boundary
END
PART 1
  Type Box
  Region Vacuum
  Fab 18.0  6.00  12.00
END
PART 2
  Type Cylinder
  Region Support
  Fab 0.80  4.00
  Shift 4.00  0.00  0.00
  Surface Region Vacuum
END
PART 3
  Type Sphere
  Region Electrode
  Fab 1.5
  Shift 4.00  0.00  3.00
  Surface Region Vacuum
  Surface Region Support
END
...
ENDFILE
```

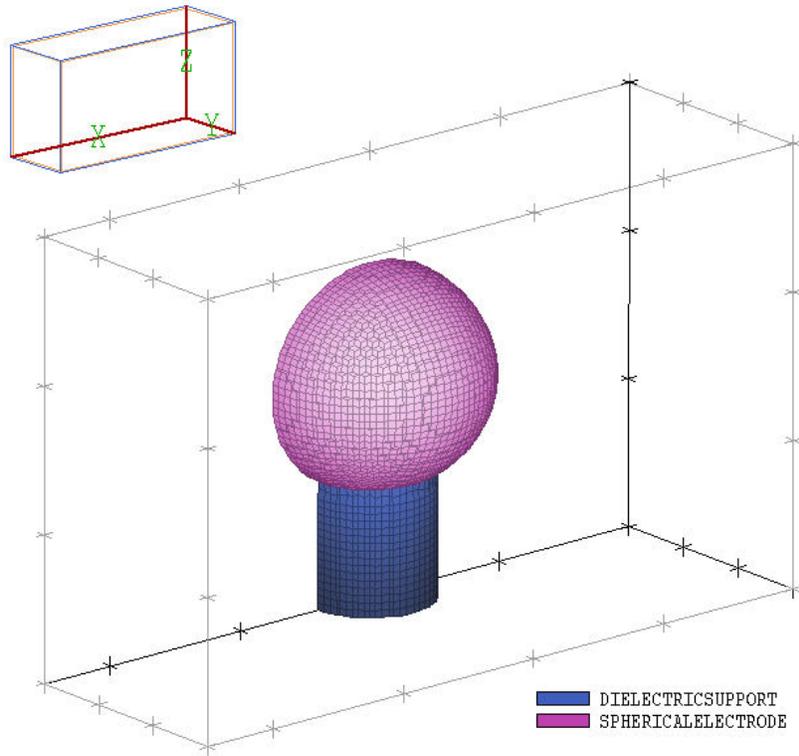


Figure 3: Three-dimensional view of the completed mesh for the WALKTHRU example,

You can set up 3D views like that of Fig. 2 by clicking the *View/Orthogonal/perspective* command. The figure illustrates the use of clipping in **MetaMesh**. Although we define a full sphere, only the portion of the electrode inside the solution volume is represented. Similarly, **MetaMesh** clips the cylindrical dielectric support at the solution-volume boundary. Exit or minimize **Geometer** and run **MetaMesh**. Choose the command *File/Load MIN file* and pick WALKTHRU.MIN. Then choose *Process mesh* to generate the mesh of Fig. 3. Use the command *File/Save mesh* to create the file WALKTHRU.MDF.

We are now ready to use **HiPhi**. Run the program from **AMaze**, choose the *Setup* command and pick the file WALKTHRU.MDF. The program displays the dialog of Fig. 4. Note that the grid contains a row for each mesh region. There are four columns in the grid where you can enter values: *Potential* (fixed voltage), *EpsilonR* (relative dielectric constant), *Sigma* (electrical conductivity in S/m) and *Rho* (space-charge density in units of coulomb/m<sup>3</sup>).

The entries in the *Control parameter* section of the dialog have the following functions:

- DUNIT. Sets a factor to convert coordinate units used in **MetaMesh** to meters. The menu sets the value of *DUnit*, equal to the number of mesh units per meter: 39.37 for inches, 100.0 for cm.
- RESTARGET. Accuracy tolerance for the iterative matrix solution of the finite-element equations.
- MAXCYCLE. Maximum number of cycles in the iterative solution.
- BOUNDARY and SUPERPOSITION. Advanced program capabilities described in Sect. 3.3.

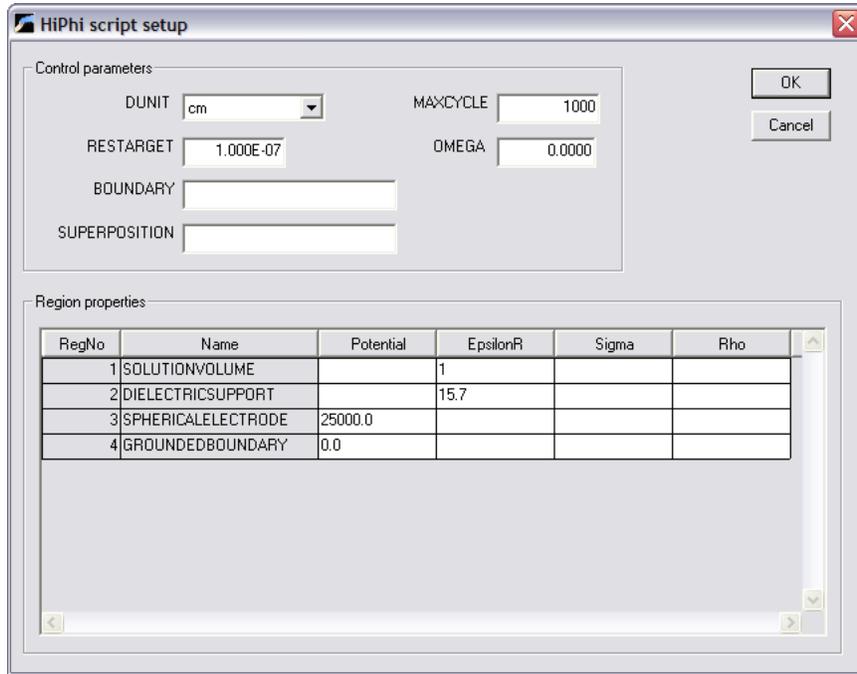


Figure 4: Dialog to set **HiPhi** control parameters and region properties.

- **OMEGA**. A parameter in the range 0.0 to 2.0 to control the iterative matrix solution.

The column options in the region grid box determine the material properties of the regions and the solution type (dielectric or conductive). An entry in the *Potential* column specifies that the region is a fixed-potential electrode in both solution types. The appearance of entries in the *Sigma* column indicates a conductive solution, while entries in the *EpsilonR* and/or *Rho* column signal a dielectric solution. **HiPhi** issues an error messages if regions have mixed conductive and dielectric properties. If only *EpsilonR* appears, the program sets *Rho* = 0.0 by default. Similarly, **HiPhi** assumes that *EpsilonR* = 1.0 if only *Rho* is specified. The values in Fig. 4 define a dielectric solution with the following region characteristics:

- Region 1. Dielectric (vacuum):  $\epsilon_r = 1.0$ .
- Region 2. Dielectric (support):  $\epsilon_r = 15.7$ .
- Region 3. Fixed potential (electrode):  $\phi = 2.5 \times 10^4$  V.
- Region 4. Fixed potential (boundary):  $\phi = 0.0$  V.

Set up the dialog with the values shown and click *OK*. **HiPhi** uses the information in the dialog to create the script `WALKTHRU.HIN` shown in Table 2. Chapter 3 reviews the script format and advanced program capabilities.

Choose *Run* in the main **HiPhi** menu and pick `WALKTHRU.HIN`. The program uses information in `WALKTHRU.MDF` and `WALKTHRU.HIN` to create finite-element equations appropriate for the conformal mesh and material properties and then solves the equations. The complete procedure takes about 25 seconds. **HiPhi** creates the binary file `WALKTHRU.HOU` which contains information on the mesh geometry and the electrostatic potential at nodes.

Table 2: **HiPhi** script WALKTHRU.HIN

```

Mesh = walkthru
DUnit = 1.0000E+02
ResTarget = 1.0000E-07
MaxCycle = 1000
* Region 1: VACUUM
Epsi(1) = 1.0000E+00
* Region 2: SUPPORT
Epsi(2) = 1.5700E+01
* Region 3: ELECTRODE
Potential(3) = 2.5000E+04
* Region 4: BOUNDARY
Potential(4) = 0.0000E+00
EndFile

```

To analyze the solution, start **PhiView** from **AMaze**. Pick *File/Load solution file* or click on the tool. Pick the file WALKTHRU.HOU in the dialog. **PhiView** can generate a wide variety of 2D and 3D plots. Chapters 6, 8 and 9 give detailed information. Here, we shall look at two examples. To create the 2D equipotential plot of Fig. 5, go to the *Slice plot* menu. On the *Change view* popup menu, pick the command *Set slice plane properties*. Click the *X Axis* button, move the slider to  $x = 4.05$  and then click *OK*. Note the slope discontinuity of equipotential lines at the edge of the dielectric.

To create a 3D plot, return to the main menu and click on *Surface plots*. On the *Plot control* popup menu choose *Surface plot style*. The default plot shows the variation of electric field in a plane normal to  $z$ . Click the button for the  $y$  axis, move the slider to  $y = 0.0$  and click *OK*. Next, we shall add the spherical electrode to the plot for reference. On the *Plot control* menu click on *Displayed regions*. Activate Region 3 and click *OK*. To get a better view, we need to rotate the plot. Move the mouse into the view area and note the change in cursors as you move around the screen. When you hold down the left or right mouse buttons, the view orientation box on the right changes. The main plot is regenerated when you release the button. With some experimentation you should be able to create a view like that of Fig. 6. It shows the variation of  $|\mathbf{E}|$  over the symmetry plane. (The dielectric support is at the top).

To illustrate the numerical-analysis capabilities of **PhiView**, return to the main menu, click *File/Run script* and choose WALKTHRU.SCR. **PhiView** carries out an extended analysis based on instructions in the file. To view the script, click on *Edit script* and choose the file. The internal program editor loads and shows the following content:

```

OUTPUT WallThru.DAT
INPUT WalkThru.HOU
VOLUMEINT
SURFACEINT 3 -2 -1
ENDFILE

```

The first statement ensures that the proper solution file is loaded while the second statement opens a data file to write the results. The command *VolumeInt* instructs the program to take

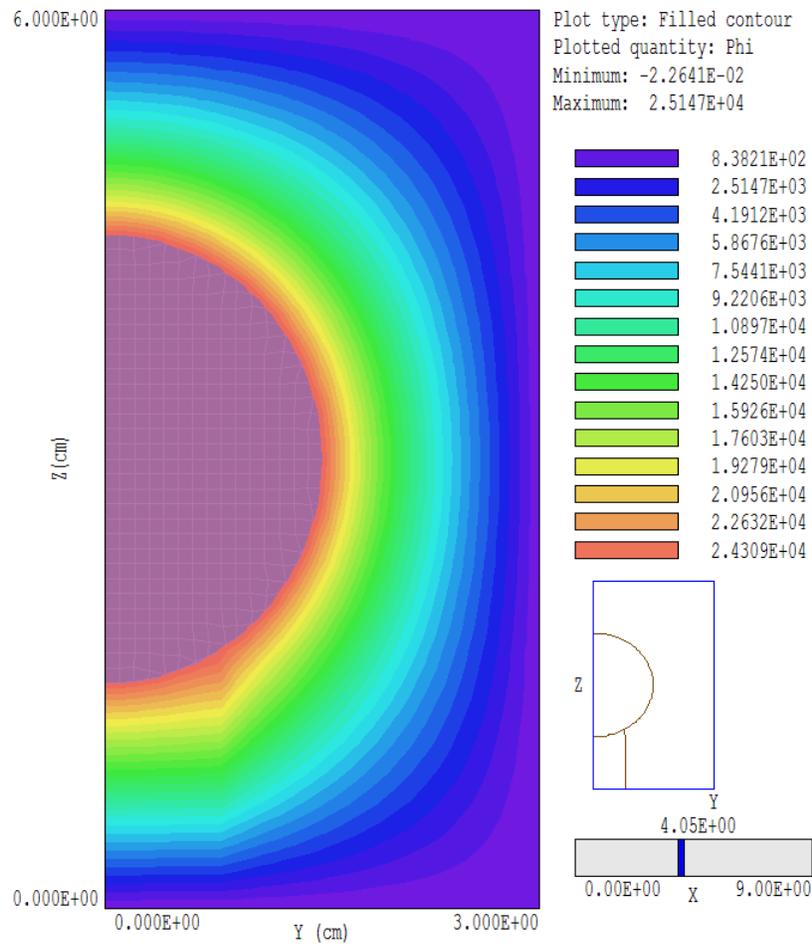


Figure 5: Slice plot normal to the  $x$  axis at position  $x = 4.05$  cm.

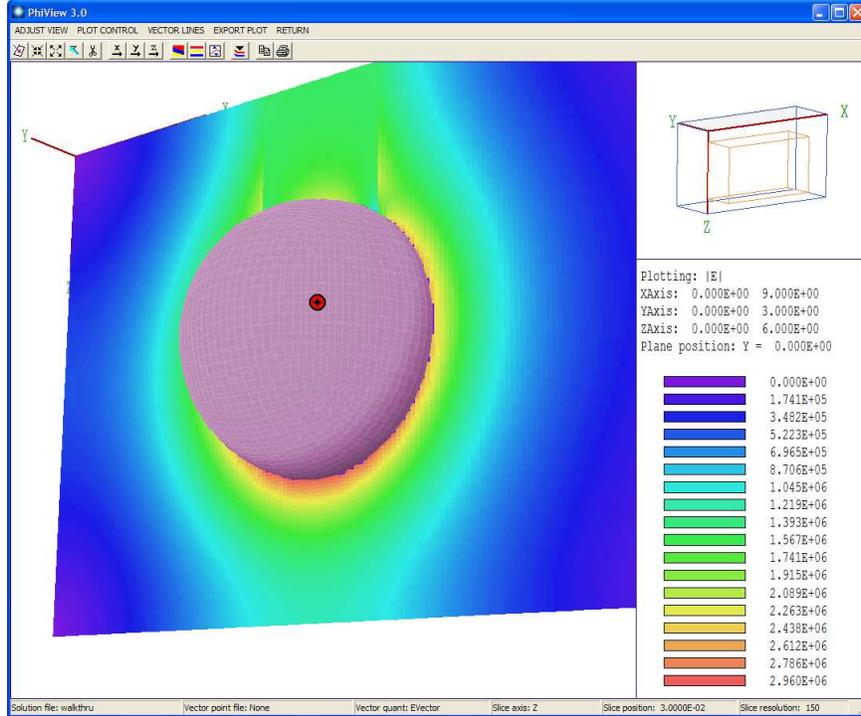


Figure 6: Working environment of **PhiView** showing a surface plot of  $|E|$  in a normal plane.

volume integrals over all regions of the solution volume. The *SurfaceInt* command instructs the program to take a surface integral over the electrode surface (Region 3) adjacent to vacuum (Region 1) and the dielectric support (Region 2). To inspect the output file `WALKTHRU.DAT`, use the command *File/Edit data file*. Table 3 summarizes results listed in the file. **PhiView** has computed the volume of regions by taking Gaussian integrals over individual hexahedron elements. The computed volume of the spherical electrode (Region 3) is  $7.066 \times 10^{-6} \text{ m}^3$ , close to the theoretical value of the half-sphere of radius 1.5 cm:  $7.069 \times 10^{-6} \text{ m}^3$ . The computed total electrostatic field energy is  $7.0204 \times 10^{-04} \text{ J}$  at an applied potential of  $2.5 \times 10^4 \text{ V}$ . Using the expression  $U = CV^2/2$ , we find that the capacitance of the half-assembly is 2.247 pF. Next consider the surface integral values determined in response to the command Region 3. The calculated surface area is  $1.427 \times 10^{-3} \text{ m}^2$  while the theoretical prediction is  $1.414 \times 10^{-3} \text{ m}^2$ . We can also double-check the capacitance calculation using the surface charge value determined by integrating the normal electric field over the surface of the electrode. The calculated result is  $5.658 \times 10^{-8} \text{ coulombs}$ . The formula  $C = Q/V$  gives the capacitance as 2.263 pF, within 0.7% of the volume integral result.

### 1.3 Theoretical background

This section discusses dielectric and conductive electrostatic solutions and reviews the nature of boundary conditions. To begin, consider a volume with electrodes and ideal dielectrics (*i.e.*, zero conductivity). If we apply a pulsed voltage, the distribution of charges on electrodes (and hence the fields in the volume) is determined by capacitive coupling. The fields for the dielectric-type solution are given by the Poisson equation

Table 3: Contents of WALKTHRU.DAT produced by WALKTHRU.SCR.

----- Volume Integrals -----

Global volume: 1.620000E-04

RegNo Region volume

=====

1 1.533153E-04

2 1.618928E-06

3 7.065768E-06

Quantity: Energy

Global value: 7.060260E-04

RegNo Region volume

=====

1 4.304516E-04

2 2.755744E-04

LineNo: 8 SURFACEINT 3

----- Surface Integrals -----

Included region set

RegNo Name

=====

3 SPHERICALELECTRODE

Surface area of region set (m2): 1.427121E-03

Charge: 5.660779E-08

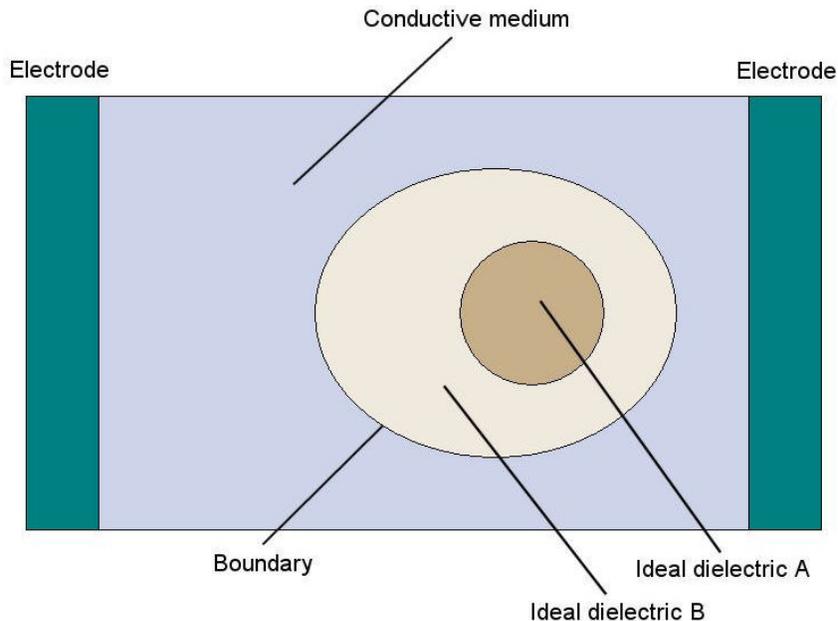


Figure 7: Example of a mixed dielectric-conductive electrostatic solution.

$$\nabla \cdot (\epsilon_r \nabla \phi) = -\frac{\rho}{\epsilon_0}, \quad (1)$$

where  $\phi$  is the electrostatic potential,  $\epsilon_r$  is the relative dielectric constant,  $\rho$  is the space-charge density and  $\epsilon_0 = 8.854 \times 10^{-12}$  F/m. Next consider a system with imperfect dielectrics with non-zero conductivity. If we apply a rapidly-pulsed voltage, the charge distribution on electrodes early in time is determined by capacitive coupling. Later the initial charge relaxes by flow through the conductors and the distribution is determined entirely by the media conductivities. Therefore, we seek a dielectric-type solution when 1) media in the solution volume have zero conductivity or 2) the voltage pulse is short. The resistive relaxation time for a material with relative dielectric constant  $\epsilon_r$  and conductivity  $\sigma$  is given by

$$\tau_{RC} = \frac{\epsilon_r \epsilon_0}{\sigma}. \quad (2)$$

As an example, consider water with  $\epsilon_r = 81.0$  and a volume resistivity of 1 M $\Omega$ -cm ( $\sigma = 10^{-4}$  S/m). Equation 2 implies a relaxation time of 7  $\mu$ s. Therefore the dielectric model would be justified if we apply a 200 ns voltage pulse to a small volume of water.

Conductive effects always dominate over long time scales. We need to qualify this statement for the special case of a conductive system that includes regions of ideal dielectrics as in Fig.7. We can determine the static field solution in the conductive regions by treating the dielectrics as perfect insulators ( $\sigma = 0.0$ ). On the other hand, the field solution inside the dielectric region depends on the spatial variation of dielectric constant. Here the conductive solution determines a fixed-potential condition along the dielectric boundary. We solve for the fields inside by applying Eq. 1 with a given potential distribution on the boundary (Dirichlet condition).

For a dielectric-type solution with no space charge, **HiPhi** solves the Laplace equation:

$$\nabla \cdot (\epsilon_r \nabla \phi) = 0 \quad (3)$$

Table 4: Material types in dielectric and conductive solutions

Object	Dielectric	Conductive
Electrode	Fixed potential	Fixed potential
Insulator		$\sigma \ll 1$
Floating electrode	$\epsilon_r \gg 1$	$\sigma \gg 1$

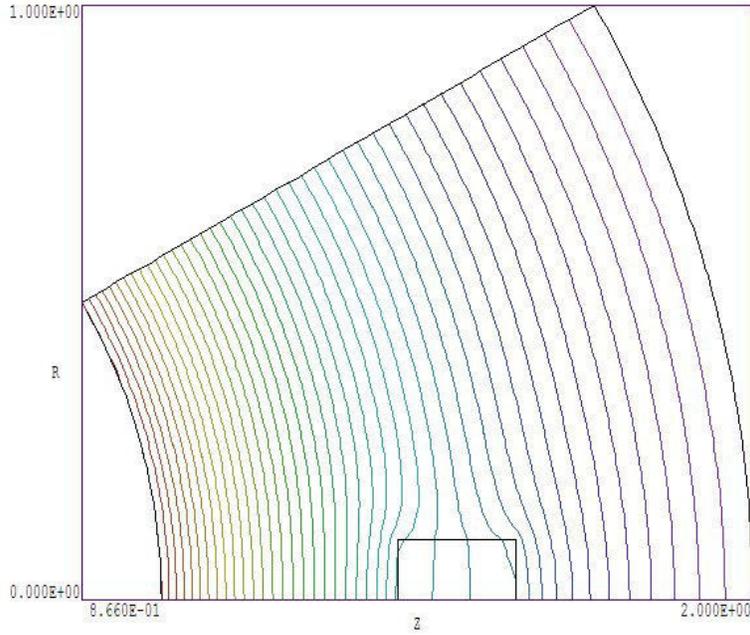


Figure 8: Special Neumann boundaries.

In a conductive medium, the current density is related to the gradient of potential by:

$$\mathbf{j} = \sigma \nabla \phi. \quad (4)$$

Conservation of charge in the medium implies that:

$$\nabla \cdot \mathbf{j} = 0, \quad (5)$$

or

$$\nabla \cdot [\sigma \nabla \phi] = 0. \quad (6)$$

Equation 6 has the same form as Eq. 3 with the substitution  $\epsilon_r \rightarrow \sigma$ . Table 4 reviews how to represent materials in the two solution types.. Note that a *floating electrode* has zero internal electric field and an initially-unknown potential determined by the dielectric or conductive properties of surrounding materials.

The *boundary* is the outer edge of the solution volume. For electrostatic problems, node potentials on the boundary can be set to one of two conditions.

- **Dirichlet.** Boundary points have a fixed potential that is not affected by the potential at neighboring nodes. A region of uniform-potential nodes represents an electrode (equipotential volume). Electric field lines are normal to such a surface.
- **Neumann.** The normal derivative of the potential is specified. The boundaries in **HiPhi** are limited to the special case where  $\partial\phi/\partial n = 0$ . The special Neumann condition implies that the electric field is parallel to the boundary. One of the advantages of the finite-element method is that all boundaries that are not fixed automatically satisfy the special Neumann condition, even if they are slanted or curved. Neumann boundaries are often used to reduce computation time for symmetric systems. For example, Fig. 8 shows an equipotential plot for a solution of the fields between two spherical electrodes with a dielectric inclusion. By utilizing automatic Neumann boundaries at the top and bottom, we can limit the section to a fraction of the interelectrode volume.

---

## 2 Organizing HiPhi calculations

### 2.1 Procedures and files

The **HiPhi** package consists of the program `hiphi.exe` that computes the physical solution and `phiview.exe`, a post-processor for analyses of the results. The solution program can run in two modes: interactively in a window or autonomously in the background under batch file control. Similarly, **PhiView** can run as an interactive application under user control or autonomously under script file control. The autonomous modes allow automatic processing of large or repetitive data sets.

Two input files are required for a **HiPhi** solution:

- A **MetaMesh** output file (`MPrefix.MDF`) that describes the conformal hexahedron mesh. The file contains node coordinates and the region numbers of elements and nodes.
- A **HiPhi** script (`FPrefix.HIN`) that sets control parameters for the solution program and describes the physical properties associated with regions.

The **HiPhi** file is usually prepared with the dialog described in Sect.1.2. You can use an editor to compose a script or to add advanced features (Chap. 3).

A **HiPhi** calculation generally consists of the following steps:

- Use **Geometer** or an editor to prepare a **MetaMesh** script with a name of the form `MPREFIX.MIN` that defines the solution space and describes the division into spatial regions.
- Run **MetaMesh** to create a file `MPREFIX.MDF` of standard mesh information. This file could be used as input for multiple **HiPhi** simulations or for other **AMaze** programs.
- Use the the *Setup* dialog of **HiPhi** to prepare a script (`FPREFIX.HIN`) that sets control parameters and the material properties of regions. You can also compose the script directly with a text editor.
- Use the **HiPhi Process** command to create a solution file `FPREFIX.HOU`. This file contains information on node spatial coordinates and electrostatic potential.
- (Optional) Prepare a standard analysis script (`SPREFIX.SCR`) to control a **PhiView** analysis session.
- Use **PhiView** to create plots or to generate numerical data using the information in `FPREFIX.HOU`. The analysis may be performed interactively or automatically under the control of a script.

Table 5 summarizes the input and output file used in **HiPhi**.

Table 5: **HiPhi** file types

Name form	Function	Status
MPREFIX.MIN	Simulation geometry, input to <b>MetaMesh</b>	Required
MPREFIX.MDF	Output from <b>MetaMesh</b> , input to <b>HiPhi</b>	Required
FPREFIX.HIN	Material properties, input to <b>HiPhi</b>	Required
FPREFIX.HOU	Output from <b>HiPhi</b> , input to <b>PhiView</b>	Required
SPREFIX.SCR	Analysis control, input to <b>PhiView</b>	Optional
PREFIX.DAT	Analysis data output, from <b>PhiView</b>	Optional

## 2.2 Script conventions

The **HiPhi** input script is a text file with data lines containing commands and parameters. The script must end with the *EndFile* command. The programs make no distinction between upper and lower case. Entries on a line may be separated by the following delimiters:

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

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

```
Epsi 2 5.56
Epsi(2) = 5.56
```

are equivalent.

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

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

The final number is interpreted as 5.0.

**HiPhi** accepts commands in any order. The program reads and analyzes all commands before starting the solution. Generally, it is good practice to put control commands at the beginning and to group commands that set material properties by region. The following example illustrates a complete script for **HiPhi**:

```
* File XFORM01.HIN
DUnit = 39.37
ResTarget = 5.0E-8
MaxCycle = 1000
NCheck = 1
Omega = 1.80
* Region assignments
* 1: Solution volume, oil
* 2: Casing, ground
* 3: Dielectric coil support
* 4: Oil penetration
* 5: High voltage electrode
Epsi(1) = 2.1
Potential(2) 0.0
Epsi(3) = 5.3
Epsi(4) = 2.1
Potential(5) = 380.0E3
EndFile
```

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

---

## 3 Script commands

### 3.1 Program control

This chapter reviews the full set of commands that may be included in **HiPhi** scripts. The first group, discussed in this section, controls program operation. Each command is shown symbolically and in a form that it might assume in a program.

#### **MESH MPrefix**

##### **MESH = TriggerConfig**

The prefix of the **MetaMesh** output file (`MPrefix.MDF`) to use for the definition of the solution geometry. If the command does not appear in a script `FPrefix.HIN`, **HiPhi** searches for the default file `FPrefix.MDF`.

#### **DUNIT Unit**

##### **DUNIT = MM**

##### **DUNIT = 100.0**

Use this command to set the units used in **MetaMesh** to create the input mesh. 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, the parameter should equal to the number of mesh units per meter. For example, if you used dimensions of centimeters in the **MetaMesh** script, set *Unit* = 100.0. The quantity *Unit* is recorded in the output file and is used in **PhiView** for the input and output of positions. Default: *Unit* = 1.0.

#### **RESTART ResTarget**

##### **RESTART = 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 mesh. **HiPhi** 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  $10^{-6}$ . If the value of *ResTarget* is too low, the program may not converge because of roundoff errors. In this case, you can terminate a solution manually if you running **HiPhi** in the interactive mode by using the *Stop* command. Default value: *ResTarget* =  $5.0 \times 10^{-7}$ .

**MAXCYCLE MaxCycle****MAXCYCLE = 2500**

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

**OMEGA Omega****Omega = 1.92**

The real-number parameter *Omega* 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 this command does not appear, **HiPhi** picks values that vary with the iteration cycle following the Chebyshev prescription. With this command, you can set a value manually. The value must be between 0.0 and 2.0 for numerical stability. Higher values generally give faster convergence.

**PARALLEL [NProc]****PARALLEL = 4**

This command invokes multithread support when the program runs on a multi-core machine. When the command appears with no value of *NProc*, **HiPhi** uses the full resources of the computer during the matrix inversion, giving a substantial reduction of the run time. Set a value of *NProc* to assign a specific number of threads. 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 **HiPhi**.

The remaining control commands serve specialized functions and are used infrequently.

**NCHECK NCheck****NCHECK = 20**

The operations to calculate *ResTarget* take time; therefore, **HiPhi** usually does not make a check on every iteration cycle. The integer parameter is the number of relaxation cycles between convergence checks. Default value: *NCheck* = 50.

**FORMAT [Text, Binary]****Format = Text**

By default, **HiPhi** creates output files `FPREFIX.HOU` in binary format. This format preserves full numerical accuracy and minimizes the file size. Use text format if you want to port results to your own analysis programs. **PhiView** can read either format. The string parameter may be either *Binary* or *Text*.

**INITVAL FPrefix****INITVAL = DistortionSwitch**

In some circumstances, you can reduce the solution time for large meshes by using precomputed initial values of potential. In normal operation, **HiPhi** initializes the electrostatic potential to  $\phi = 0.0$  at all nodes that do not have a fixed potential. When this command appears,

the program sets initial values at variable nodes equal to those in the file `FPrefix.HOU`. One application for the command is to run additional relaxation cycles to improve the accuracy of an existing solution. You can also use the command if you make small changes to applied potentials or the geometry of electrodes or dielectrics. The solution file `FPrefix.HOU` should be available in the working directory and must use the same foundation mesh as the input file `MPrefix.MDF`. A necessary condition is that the values of  $I_{max}$ ,  $J_{max}$ ,  $K_{max}$ ,  $x_{min}$ ,  $x_{max}$ ,  $y_{min}$ ,  $y_{max}$ ,  $z_{min}$  and  $z_{max}$  must match. Within this limitation, there may be small geometric differences between objects in the two files. If there are large differences, the procedure may not significantly reduce the solution time.

### **INTERP [Linear, Spline]**

#### **INTERP = Spline**

Set the interpolation method used when spatial variations of  $\phi$ ,  $\epsilon_r$ ,  $\sigma$  and/or  $\rho$  are defined from data tables. Use the *Linear* interpolation method for noisy or discontinuous data. The default is *Spline*.

## **3.2 Basic material properties**

Material properties in electrostatic solutions are relatively simple so only a few commands are required. **HiPhi** handles either dielectric solutions or conductive solutions (Sect. 1.3). In a dielectric solution you can set both the relative dielectric constant and space-charge density of a filled region. In a conductive solution you can specify the conductivity of a filled region. **HiPhi** issues an error message if you mix quantities in a script. The basic commands of this section set properties that are uniform throughout a region. Chapter 4 describes how to define material properties that vary according to a mathematical expression over the region volume.

### **POTENTIAL RegNo Phi**

#### **POTENTIAL(5) = 2500.0**

This command sets all elements or nodes associated with a region number to a fixed potential condition. The integer parameter is a region number defined in **MetaMesh**. The real number is the value of electrostatic potential in volts (V). The fixed-potential condition may be applied to both filled and open regions.

### **EPSI RegNo EpsiRel**

#### **EPSI(4) = 12.6**

This command sets a relative dielectric constant for all elements associated with a filled region. The integer parameter is the number of a region defined in **MetaMesh**. The second parameter is the value of relative dielectric constant  $\epsilon_r = \epsilon/\epsilon_0$  (dimensionless). To model a floating electrode (conducting object whose potential is not previously know), assign a relatively large value of dielectric constant (*i.e.*,  $\epsilon_r \cong 10^4$ ) to the region. Note that excessive values ( $\epsilon_r \cong 10^8$ ) may slow the solution convergence. The default for all unspecified regions is  $\epsilon_r = 1.0$ .

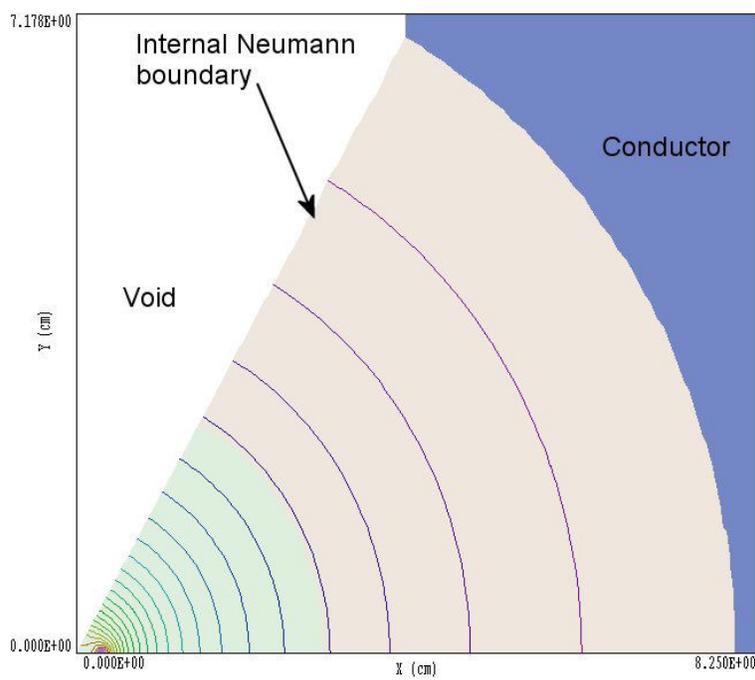


Figure 9: Use of a void to define a slanted Neumann boundary.

### **RHO RegNo ChgDens**

#### **RHO 3 4.5E-10**

This command assigns a space-charge density to elements within the specified region. The real-number parameter is the value of  $\rho$  in coulombs/m<sup>3</sup>. A script may contain both *Rho* and *Epsi* commands for the same region. The default for all unspecified regions is  $\rho = 0.0$ .

### **SIGMA RegNo Cond**

#### **SIGMA(5) = 0.682**

This command assigns a conductivity to elements within the specified filled region. The real-number parameter is the value of  $\sigma$  in siemens/m. To model a floating electrode, assign a relatively high value of conductivity. Do not assign the value  $\sigma = 0.0$  for a perfect insulator. Instead, use the *Void* model. The default for all unspecified regions in a conductive solution is  $\sigma = 1.0$  S/m.

### **VOID RegNo**

#### **VOID (4)**

A void material has the property  $\epsilon_r = 0.0$  or  $\sigma = 0.0$ . Electric field lines in the medium adjacent to a void are parallel to the boundary and equipotential lines are normal. Voids can be used to define symmetry boundaries that are not parallel to one of the axes. For example, Figure 9 shows a simulation of a 3-wire probe array in one sextant of the  $x$ - $y$  plane. Internal voids can also be used to represent ideal insulators of any shape in conductive solutions. As shown in the figure, **PhiView** does not calculate or plot electrical quantities inside a void.

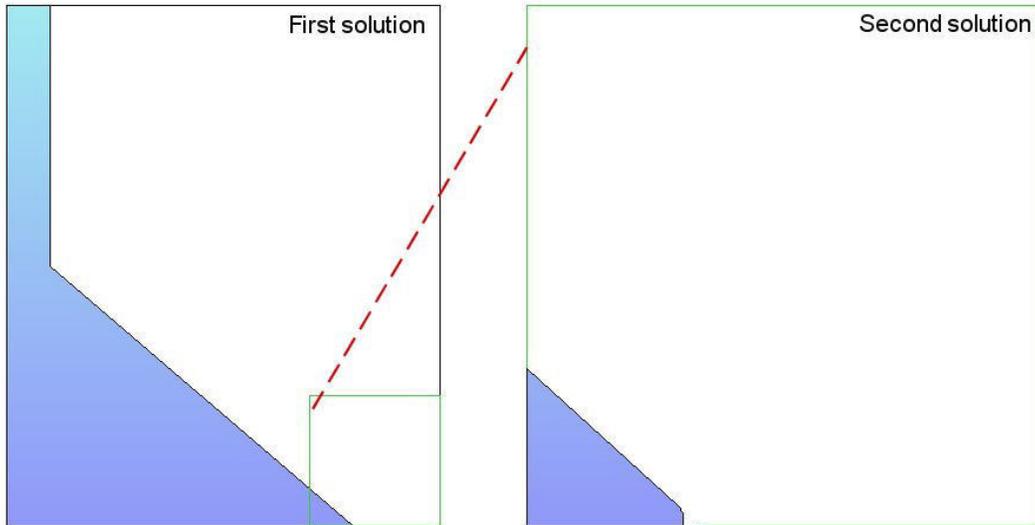


Figure 10: Creating a microscopic solution using the *Boundary* command.

### 3.3 Modeling small details in a large-scale solution

**HiPhi** incorporates a powerful feature that enables accurate calculations of fields near small features in a large solution space. Suppose we wanted a precision calculation of the fields on a field emission tip. The radius of the tip is much smaller than the scale size of the electrodes that create the field and hold the tip. We need to do a large-scale calculation to find the macroscopic fields (**Solution 1**), but we would need very small elements near the tip to resolve the curvature. One approach is to use variable mesh resolution to create small elements near the tip. A limitation to this technique arises from the structured mesh used in the **AMaze** programs – the region of small elements must extend the full length of the solution volume.

Figure 10 illustrates an alternate approach. We create a second solution that extends over a small subregion of the original solution (green outline). The microscopic solution contains any electrodes that are located within the subregion. The difference is that the electrode shapes may be resolved in much greater detail. The question is how to incorporate the macroscopic fields correctly into the microscopic solution. One resolution is to enclose the second solution in a variable-potential Dirichlet boundary. Values of  $\phi$  on the boundary are calculated by interpolation at the corresponding point in the macroscopic solution (dashed red line). The total solution will be approximately correct as long as new features (such as the rounded tip) are well-removed from the variable-potential boundary.

Definition of the variable Dirichlet boundary is controlled by the following command that appears in the script of **Solution 2**:

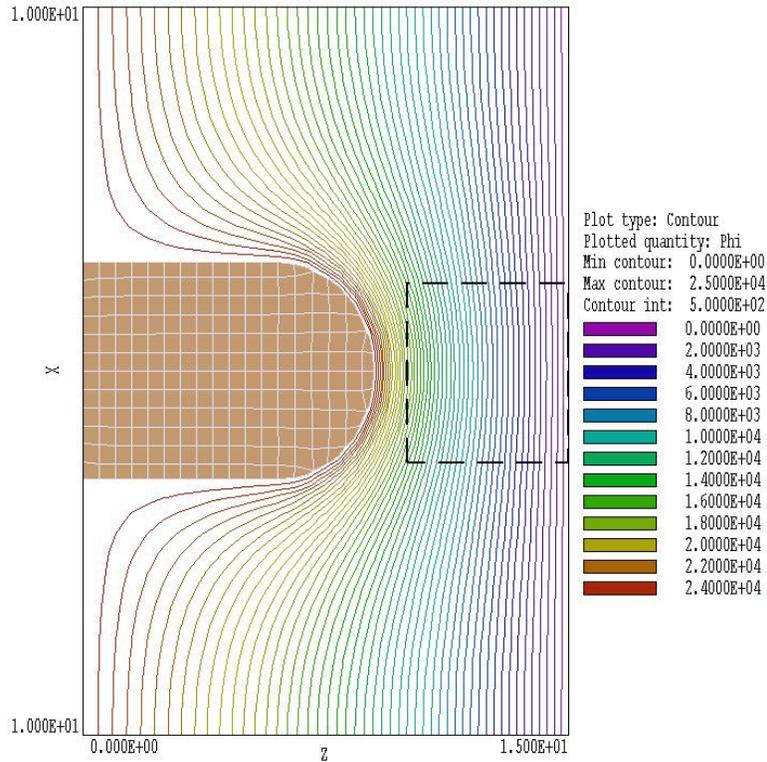


Figure 11: Equipotential lines and element resolution in the large-scale solution BOUNDTEST01. The dashed line shows the boundary of the small-scale solution.

**BOUNDARY FPrefix [BScale] [BStatus]**  
**BOUNDARY FEmitMacro 1.0 101111**

The string quantity *FPrefix* is the prefix of the output file for **Solution 1**. The real-number quantity *BndScale* is an optional scaling factor applied to potential values transferred from **Solution 1** to **Solution 2**. The default is *BndScale* = 1.0. The optional parameter *BStatus* is a string with six characters, either '0' or '1'. It can be used to control whether a specific boundary is assigned the computed Dirichlet condition. The order of boundaries in the string is *XDn*, *XUp*, *YDn*, *YUp*, *ZDn* and *ZUp*. An entry of '1' indicates that the Dirichlet condition should be applied. Use a '0' entry for symmetry boundaries that assume the Neumann condition. The default is *BStatus* = '111111'.

In response to the *Boundary* command, **HiPhi** loads **Solution 1** into memory and sets up the apparatus for making second-order interpolations of potential in the solution. Next, the program loops through all nodes of **Solution 2**. If the node is on the boundary of the solution box, **HiPhi** marks it as a fixed-potential point and sets the potential equal to the interpolated value calculated at the position in **Solution 1**. **HiPhi** issues an error message under the following conditions: 1) the output file for **Solution 1** is not available in the working directory, 2) **Solution 2** does not fit completely inside **Solution 1** or 3) an interpolation failed. The program makes no further validity checks. You must ensure that the geometry of **Solution 2** represents a correct microscopic section of **Solution 1**.

Files in the example folder illustrate the process. The file set BOUNDTEST01.MIN and BOUNDTEST01.HIN describe a large-scale solution. A microscopic solution, a subset of the

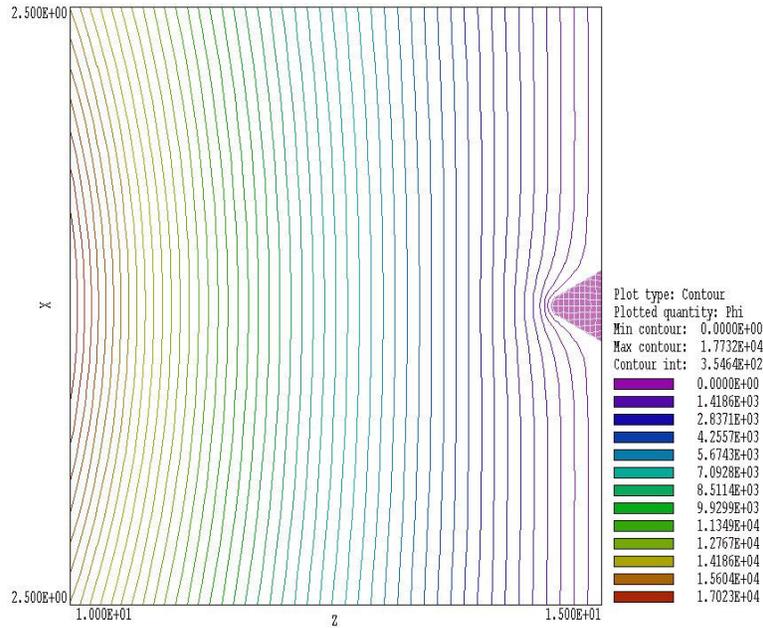


Figure 12: Equipotential lines and element resolution in the small-scale solution BOUNDTEST02.

first, is defined by BOUNDTEST02.MIN and BOUNDTEST02.HIN. Figure 11 shows the large-scale solution which covers a volume extending from  $-10.0''$  to  $10.0''$  in  $x$  and  $y$  and from  $0.0''$  to  $15.0''$  in  $z$ . A shaped electrode creates a field on a flat grounded plate. We want to find precise fields near a small protrusion on the plate. The protrusion is smaller than the scale size of elements in the large-scale mesh ( $0.5''$ ). The microscopic solution covers a portion of the original solution volume extending from  $-2.5''$  to  $2.5''$  in  $x$  and  $y$  and from  $10.0''$  to  $15.0''$  in  $z$ . (dashed line in Fig. 11). The element scale size is  $0.25''$  in the bulk of the solution and  $0.05''$  in the region of the protrusion. The script BOUNDARYTEST02.HIN has the following content:

```
* File BOUNDARYTEST02.HIN
* NReg      RegName
* =====
*      1      VACUUM
*      2      PROTRUSION
RESTARTGET = 5.0E-8
MAXCYCLE = 2500
DUNIT = 39.37
NCHECK = 10
BOUNDARY BOUNDARYTEST01
EPSI(1) = 1.0
POTENTIAL(2) = 0.0
ENDFILE
```

Figure 12 shows equipotential lines and the element division of the protrusion in the microscopic electrostatic solution, a subset of the original. The surface of the protrusion is well-resolved and electric field interpolations in the vicinity will be accurate. The process is valid because the presence of the small protrusion does not significantly influence values of potential at the boundary of **Solution 2**.

In response to the *Superposition* command, **HiPhi** adds values from one solution (**Solution 1**) to another (**Solution 2**). The following statement may appear in the command script for **Solution 2**.

**SUPERPOSITION FPrefix [SScale]  
SUPERPOSITION UniField**

The string *FPrefix* is the prefix of the output file for **Solution 1**. The restriction is that **Solution 1** must fit inside **Solution 2**. The real quantity *SScale* is a scaling factor applied to potential values transferred from **Solution 1** to **Solution 2**. The default is *SuperScale* = 1.0.

In response to the *Superposition* command, **HiPhi** opens the solution file *FPrefix.HOU* after **Solution 2** has been completed. The program performs an interpolation in the space of **Solution 1** to determine the potential  $\phi_1$  at each node location in **Solution 2** and adjusts values according to

$$\phi'_2 = \phi_2 + SScale \times \phi_1, \quad (7)$$

before writing the output file. You must ensure that the superposition is physically correct. In electrostatic solutions the presence of electrodes and dielectrics in **Solution 2** can make significant local changes to the macroscopic field of **Solution 1**, so that a simple superposition would be invalid.

---

## 4 Variations of potential and material quantities from tables and mathematical functions

### 4.1 Program capabilities

Section 3.2 covered commands to define potential values or material properties ( $\epsilon_r$ ,  $\sigma$  and/or  $\rho$ ) that are uniform throughout a region. This section describes how to represent quantities that vary continuously in space following a mathematical prescription. It is straightforward to assign variable region properties with following commands:

#### **POTENTIAL RegNo > Function**

**POTENTIAL(5) > 1.50E04\*cos(\$x/20.5)\*sin(\$y/15.0) + 1.3E03**

**POTENTIAL(2) > 80.245\*(1.0 - 0.0625\*\$z^2)**

Set values of  $\phi$  (in volts) at the nodes of a fixed-potential region according to a specified function of space. The keyword *Potential* followed by the region number and the > symbol designates that a function string occupies the remainder of the line. The function may be up to 230 characters in length and follows the format described in the next section. The function defines a variation in three-dimensional space:  $f(x, y, z)$ . The parser uses the Perl standard for variables where \$x designates the variable  $x$ , \$y stands for  $y$ , and \$z for  $z$ . The potential at a node equals the value of the function evaluated at the node position. **Note 1:** There is no limit on the number of functions – any region may be associated with a function. **Note 2:** Positions are passed to the function in units set by *DUnit* (cm, inches,  $\mu\text{m}$ ,...). **Note 3:** You can model discontinuous functions by dividing a volume into two or more regions. **Note 4:** Variations may be assigned to both filled regions and contiguous or disconnected line or surface regions.

#### **POTENTIAL RegNo TABLE [x,y,z,r] TabName**

**POTENTIAL(3) = TABLE R ZUpBoundary.DAT**

Set values of  $\phi$  (in volts) at the nodes of a fixed-potential region according to a spatial variation specified in a table. A table is a text file available in the current directory with format described in the following section. The keyword *Potential* is followed by the region number and the keyword *Table*. Because tables are one-dimensional, the command must include a symbol that designates the direction of the variation. The options are  $x$ ,  $y$ ,  $z$  and  $r$ , where

$$r = \sqrt{x^2 + y^2}. \quad (8)$$

The string *TabName* is the full file name of the table. The node positions (in units set by *DUnit*) are passed to the table-interpolation routine. The table should return potential values in units of volts. Tables are particularly useful when you want to add small three-dimensional details to a global two-dimensional solution. A scan of potential values in an **EStat** solution may be used to set boundary values in a microscopic **HiPhi** solution.

**EPSI RegNo > Function****EPSI(7) > 1.0 + (\$x - 0.25)/5.50****EPSI(4) > 1.0 - 0.625\*\$z^2**

Assign values of the relative dielectric constant  $\epsilon_r$  to elements of the region according to a specified spatial variation. The spatial function is evaluated at the center of mass of each element. **HiPhi** issues an error message if the value of the function in any element gives  $\epsilon_r \leq 0.0$ .

**EPSI RegNo TABLE [x,y,z,r] TabName****EPSI(5) = TABLE Z GradedDielectric.DAT**

Set values of the relative dielectric constant  $\epsilon_r$  in elements of the region according to a spatial variation defined by a table. The position of the element center-of-mass (in units set by *DUnit*) is passed to the table-interpolation routine. **HiPhi** issues an error message if the returned value in any element gives  $\epsilon_r \leq 0.0$ .

**SIGMA RegNo > Function****SIGMA(7) > 0.5 + 2.0\*(1.0 - cos(3.14156\*\$x/15.0))****SIGMA(4) > 100.0 - 50\*exp((\$z/10)^2)**

Assign values of electrical conductivity to elements of the region according to a specified spatial variation. The spatial function is evaluated at the center of mass of each element and should return values in units of S/m. **HiPhi** issues an error message if the value of the function in any element is  $\sigma < 0.0$ .

**SIGMA RegNo TABLE [x,y,z,r] TabName****SIGMA(5) = TABLE X SwitchBreakdown.DAT**

Set values of the conductivity  $\sigma$  in elements of the region according to a spatial variation defined by a table. The position of the element center-of-mass (in units set by *DUnit*) is passed to the table-interpolation routine. The table should return values of  $\sigma$  in units of S/m. **HiPhi** issues an error message if the returned value in any element gives  $\sigma < 0.0$ .

**RHO RegNo > Function****RHO(5) > 5.235E-6 + 4.33E-6\*(\$x/2.3E-6)**

Assign values for the space-charge density in dielectric solutions to elements of the region according to a specified spatial variation. The spatial function is evaluated at the center of mass of each element and should return values in units of coulomb/m<sup>3</sup>.

**RHO RegNo TABLE [x,y,z,r] TabName****RHO(2) = TABLE Y PulsedBeam.DAT**

Set values of the space-charge density  $\rho$  in elements of the region according to a spatial variation defined by a table. The position of the element center-of-mass (in units set by *DUnit*) is passed to the table-interpolation routine. The table should return values of  $\rho$  in units of coulomb/m<sup>3</sup>.

You can use the plotting and analysis functions of **PhiView** to check the validity of mathematical expressions for spatially-varying  $\phi$ ,  $\epsilon_r$ ,  $\rho$  or  $\sigma$ .

## 4.2 Function syntax and table format

**HiPhi** features a flexible and robust algebraic function interpreter. A function is a string (up to 230 characters) that may include the following entities:

- Spatial variables,  $x$ ,  $y$  or  $z$ .
- 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 ^ (exponentiation).
- Functions: **abs** (absolute value), **sin** (sine), **cos** (cosine), **tan** (tangent), **ln** (normal logarithm), **log** (base 10 logarithm), **exp** (normal exponent) and **sqrt** (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*((x^2 + y^2 + z^2)/24))`

corresponds to

$$1 - \exp \left[ - \left( \frac{x^2 + y^2 + z^2}{24} \right) \right]. \quad (9)$$

A table is a text file that contains a set of data lines of the format

`IndVar DepVar`

followed by the *EndFile* command. The independent variable may be  $x$ ,  $y$ ,  $z$  or  $r$  in units set by *DUnit*. In other words, if the mesh dimensions were in centimeters, use the same units for the independent variable. Note that the intervals between values of the independent variable need not be uniform – values may be clustered near positions where there are large changes in the dependent variable. The dependent variable may represent the potential (in volts), the relative dielectric constant, the conductivity (in S/m) or the space-charge density (in coulombs/m<sup>3</sup>). The entries may be in any valid floating point format and may be separated by any of the standard delimiters listed in Sect. 2.2. You may include comment lines that start with an asterisk and text in any format after the *EndFile* command.

The table interpolation method may be linear or cubic spline, according to the form of the *Interp* command. The default is *Spline*. The cubic spline method requires a good data set with continuous values of the function and its derivative. Use the *Linear* option for noisy or discontinuous data. The interpolation routine returns a value of zero for out-of-range input values of the independent variable.

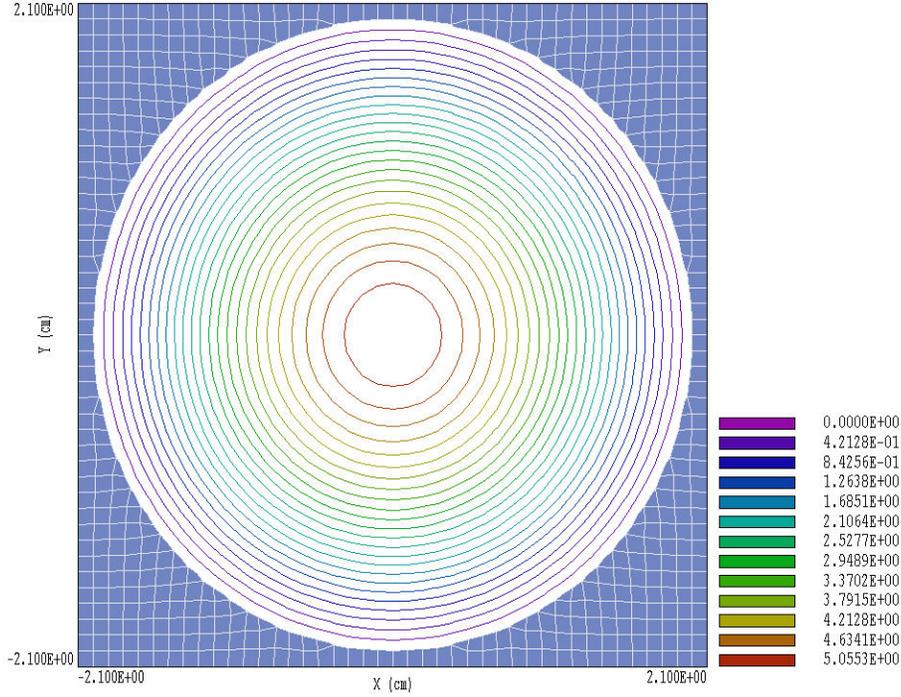


Figure 13: Geometry for the RHOTEST example showing the element size. Dimensions in cm.

### 4.3 Benchmark example

The RHOTEST example (Fig. 13) illustrates the procedure to define a continuous variation of space-charge density. We choose a simple geometry for comparison with analytic results. The calculation determines the electrostatic potential generated by a symmetric charge distribution  $\rho(R)$  inside a grounded metal sphere of radius  $R_0 = 2.0$  cm. In this case, the potential is determined by the Poisson equation:

$$\frac{1}{R^2} \frac{d}{dR} R^2 \frac{d\phi}{dR} = -\frac{\rho(R)}{\epsilon_0}. \quad (10)$$

If the space-charge density has the uniform value  $\rho_0$ , solution of Eq. 10 gives the following expression for the potential at center of the sphere:  $\phi_0 = \rho_0 R_0^2 / 6\epsilon_0$ . For  $\rho_0 = 1.0 \times 10^{-6}$  C/m<sup>3</sup> and  $R_0 = 0.02$  m, the potential is  $\phi_0 = 7.529$  V. If the space-charge density varies with radius as:

$$\rho(R) = \rho_0 \left[ 1 - \left( \frac{R}{R_0} \right)^2 \right], \quad (11)$$

then the potential at the center is  $\phi_0 = (\rho_0 R_0^2 / \epsilon_0)(1/6 - 1/20) = 5.271$  V.

Figure 13 shows the geometry for the numerical solution. The element size is approximately 0.10 cm. Region 1 is a volume of fixed-potential nodes with  $\phi = 0.0$  V that initially fills the solution volume. Region 2 is a dielectric with  $\epsilon_r = 1.0$  that carves out the spherical volume. To create a solution with uniform charge density  $\rho_0 = 1.0 \times 10^{-6}$  C/m<sup>3</sup>, the **HiPhi** script contains the following entry:

```
Rho(1) = 1.0000E-06
```

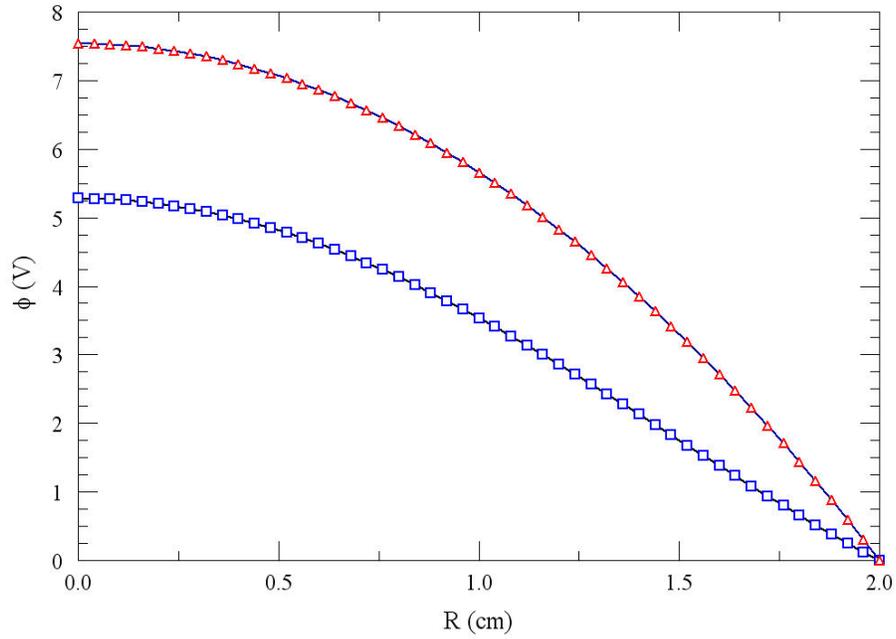


Figure 14: Variation of potential with  $R$  for the RHOTEST example. Red: uniform space-charge density. Blue: non-uniform space-charge density.

The red curve in Fig. 14 shows the calculated radial variation of potential. The potential at the center is  $\phi_0 = 7.533$  V, within 0.11% of the theoretical value. To introduce the variable space-charge density of Eq. 11, we simply change the above line to

$$\text{Rho}(2) > 1.0\text{E-}6*(1.0 - 0.25*(x^2+y^2+z^2))$$

The resulting variation of  $\phi$  is plotted as the blue curve in Fig. 14. The calculated central potential is  $\phi_0 = 5.266$  V, within 0.13% of the theoretical value.

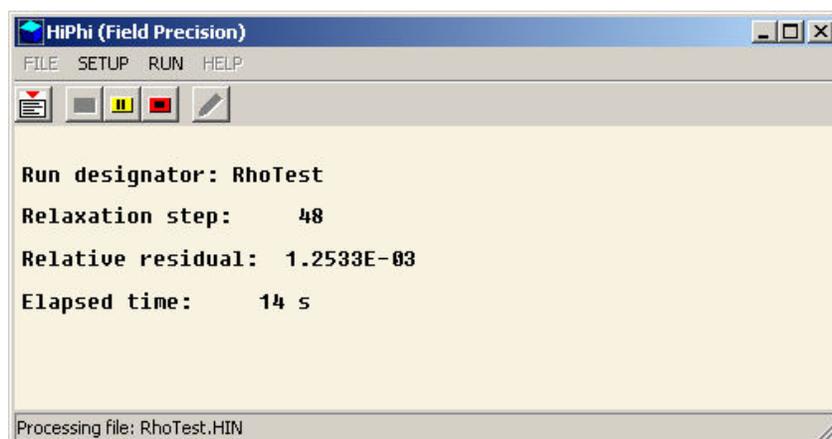


Figure 15: **HiPhi** display during a solution.

## 5 Running the HiPhi program

### 5.1 Interactive operation

The program `hiphi.exe` can run interactively in a window. In this mode you can perform several solutions in a session and temporarily leave the program to work on other tasks. You can launch the program from **AMaze** or create you own shortcuts. Figure 15 shows the program window.

The program menu has four main commands: *File*, *Setup*, *Run* and *Help*. The following commands appear in the *File* popup menu.

#### EDIT SCRIPT

This command calls the internal editor to inspect or to modify **HiPhi** input files (Fig. 16). When the command is chosen, **HiPhi** shows a load dialog with a listing of files with names of the form `FPREFIX.HIN`. Choosing a file from an alternate directory does not change the working directory. In this mode, the editor has supplementary functions to help you remember command options and to edit scripts. Click on an entry in the *Control commands* or *Material properties* menu column to add a selected command line above the cursor. The *Block commands* column has entries to *comment/uncomment* or *indent/unindent* selected lines. You must exit the editor to resume normal operation of the program.

#### EDIT LISTING FILE

#### EDIT FILE

The commands call up the internal editor to inspect or to modify **HiPhi** listing files or general text files. With the *Edit listing file* command you can pick files with names of the form `FPREFIX.HLS`. The *Edit file* command shows all available

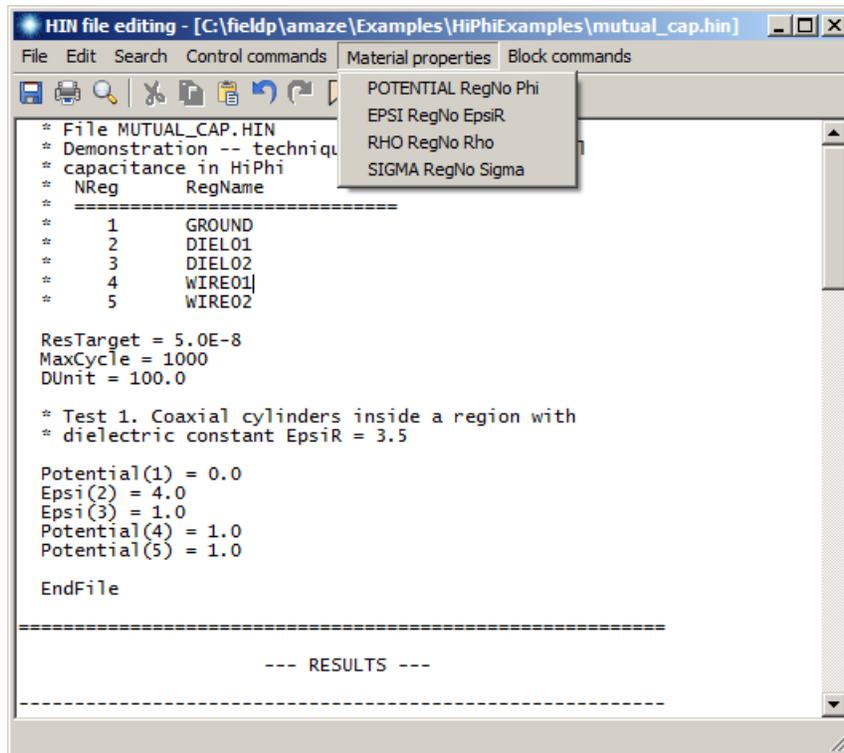


Figure 16: **HiPhi** display during a solution.

files. Choosing a file from an alternate directory does not change the working directory of the program. Note that the main program will be inactive until you exit the editor.

## SETUP

In response to this command, **HiPhi** prompts for the name of a **MetaMesh** file and brings up the dialog of Fig. 4. Fill in values for control parameters and material properties to create a basic script. You can use an editor to modify the script or to add the advanced functions described in Chap. 3.

The *Run* menu has three commands.

## START RUN

Pick an input file with a name of the form **FPREFIX.HIN** to start a solution. The working directory changes if you pick a file from an alternate directory. The run begins if the requested **MetaMesh** file **MPREFIX.MDF** is available in the working directory. **HiPhi** displays information in the window to show the progress of program operations (Fig. 15).

## PAUSE RUN

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

## STOP RUN

This command terminates the program during the relaxation process and saves output data. For example, you may want to stop a run at a moderate value of convergence to check whether the problem has been correctly defined.

The *Help* menu has a single command.

## INSTRUCTION MANUAL

The command displays this manual using your default PDF viewer. The file `hiphi.pdf` must be in the same directory as `hiphi.exe`.

## PHYSICAL CONSTANTS

The command displays the file `physcons.pdf`, a useful list of physical constants included with the **HiPhi** distribution.

## 5.2 Automatic runs under batch file control

Batch file control is a useful option for running large technical programs like **HiPhi**. You can prepare scripts to organize complex operations. The sequenced programs run automatically in the background. This feature is particularly attractive on dual-processor machines.

To run a single **HiPhi** calculation in the background, go to the Command Prompt from Windows and log to the data directory that contains the required MDF and HIN files. For example, suppose the data files `SWITCH.MDF` and `SWITCH.HIN` are stored in `\AMAZE\BUFFER` and that the program `hiphi.exe` is in the directory `\AMAZE`. From `\AMAZE\BUFFER` type

```
..\HIPHI SWITCH <Enter>
```

The program runs silently, writing detailed information in the listing file `SWITCH.HLS`. If the solution is successful, the program creates the output file `SWITCH.HOU` in the data directory.

The main function of the command mode is autonomous operation under batch file control. As an example, suppose you have prepared the input files `SWITCH01.MIN`, ..., `SWITCH08.MIN` and `SWITCH01.HIN`, ..., `SWITCH08.HIN` 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 ..\METAMESH.EXE SWITCH02
START /WAIT ..\HIPHI.EXE SWITCH02
...
START /WAIT ..\METAMESH.EXE SWITCH08
START /WAIT ..\HIPHI.EXE SWITCH08
```

Type

```
SWRUN <Enter>
```

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

### 5.3 Calculations with large meshes

The best strategy for three dimensional calculations is to work with the smallest mesh consistent with the desired accuracy. There are several ways to minimize the mesh size::

- Avoid representing small details that have little effect on the field solution.
- Use variable element sizes with a coarse representation of remote objects.
- Exploit symmetries so that only a portion of the system is included in the calculation.
- Use the *Boundary* building detailed microscopic solutions inside a global solution.

If a large mesh ( $> 1$  million elements) is unavoidable, you must ensure that the computer has sufficient resources. The mesh size may be limited by the installed RAM. **HiPhi** requires about 160 bytes/element. Multiply by the number of elements to estimate the required memory. (Note, when using the *Boundary* command, you must include the number of elements in both the microscopic and macroscopic meshes.)

---

## 6 PhiView – file operations and plane plots

The functions of the **PhiView** post-processor are creation of plots and calculation of numerical quantities from **HiPhi** solution files. The program has the popup menus: *File operations*, *Analysis*, *Plane plots*, *Slice plots*, *Surface plots* and *Help*. Initially, only the *File operations* and *Help* menus are active. You must load a data file in order to create plots or to perform analyses.

### 6.1 File operations

This section reviews options in the *File operations* menu.

#### LOAD SOLUTION FILE

**PhiView** displays a dialog with a list of solution files with names of the form `FPREFIX.HOU`. Changing the directory in the dialog changes the program working directory. Pick an available file and click *OK*. The program loads the solution and updates the status bar. If data retrieval is successful, the analysis and plot menus become active.

#### SOLUTION FILE INFORMATION

The command shows a message box with information on the currently-loaded data file.

#### LOAD CONFIGURATION FILE

Configuration files determine the plot and analysis quantities of **PhiView**. The files have names of the form `FPREFIX.CFG`. Three files are supplied with the program:

`phiview_dielectric.cfg` – analysis of dielectric-type solutions.

`phiview_conductive.cfg` – analysis of conductive-type solutions.

`phiview_force.cfg` – calculations of electrostatic forces and torques in dielectric-type solutions.

**PhiView** loads the configuration `phiview_dielectric.cfg` the first time you run the program. Use this command to change the configuration. The program remembers the last configuration used and reloads it. You can customize **PhiView** by creating your own configuration file (see Chap. 10).

#### SAVE REFLECTED SOLUTION

The use of a symmetry boundary saves considerable time in a large 3D calculation but produces half a solution. There are instances where it is useful to have numerical values for the full solution volume (*e.g.*, creating plots or tracking orbits in **OmniTrak** or **GamBet**). Use this command to create a data file for the currently-loaded solution that includes values of electrostatic potential reflected across a symmetry boundary. To begin, you must set up the

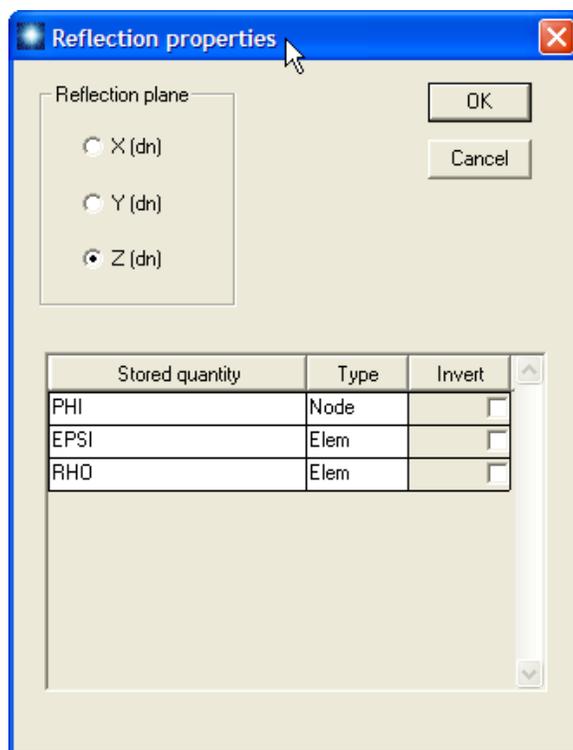


Figure 17: Dialog to create a reflected solution

original solution so that the symmetry plane corresponds to the lower solution-volume boundary in either  $x$ ,  $y$  or  $z$ . Load the solution, click on the command and use the *Save* dialog to specify an output file name. The next dialog (Fig. 17) shows a set of options for the reflection. Designate a reflection plane with the radio buttons. The grid field shows the stored quantities in the solution file. The *Invert* check box in the row for electrostatic potential  $\phi$  determines the reflection symmetry. Leave the box unchecked if the electric field is parallel to the symmetry plane. Check the box if the field is normal to the plane. In electrostatic solutions, the quantity  $\epsilon_r$  is always a positive number and should not be inverted. You must ensure that the solution is appropriate for the reflection parameters that you choose. **PhiView** does not check the physical consistency of the output values. You can load the modified file and check validity with the plot functions of **PhiView**.

## RUN SCRIPT

Sometimes you may want to perform complex or repetitive analyses on a set of similar solutions. Analysis script operation is a powerful feature of **PhiView**. This command displays a dialog with a list of analysis scripts (suffix **SCR**) that you have created. Pick a file and click *OK*. Changing directories in the dialog changes the working directory of the program. The analysis script can load data files, open and close history files, and perform any of the numerical functions described in this manual. Section 7.1 reviews the analysis script language.

## CREATE SCRIPT

Use this command to create scripts using the internal editor. A box requests a file prefix. The resulting script file will be saved as `FPREFIX.SCR`. Next, the program opens the file in the editor and writes the reference list of allowed commands shown in Table 6. Enter commands in the space above *EndFile*. After saving the file, you can run it using the *Run script* command.

## EDIT SCRIPT

Use this command to change an existing script file. The dialog lists files in the current directory with the subscript `SCR`. Changing directories does not change the working directory of the program.

## OPEN DATA FILE

Several of the analysis commands like *Point calculation* and *Line scan* generate quantitative information. You can automatically record the data generated during an analysis session by opening a data file. Supply a file prefix in the dialog or accept the default. The text data file has a name of the form `FPREFIX.DAT` and will be stored in the working directory. You can use an editor to view the file or to extract information for mathematical analysis programs or spreadsheets.

## CLOSE DATA FILE

Use this command if you want to start a new file to record data. The data file is automatically closed when you exit the postprocessor. Otherwise, you must close the file before using the *Edit data file* command or loading the file into another program. Failure to close the file may result in a Windows Resource Sharing Error.

## EDIT DATA FILE

View or modify files with names of the form `FPREFIX.DAT`.

## EDIT FILE

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

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

## INSTRUCTION MANUAL

Displays this document in your default PDF viewer. The file `hiphi.pdf` must be in the same directory as `phiview.exe`.

Table 6: Create script - default file content

```
* Analysis script

* Insert commands here...

ENDFILE

    --- Command summary ----

INPUT InFileName
  [Close current solution file and load FileName]
OUTPUT OutFileName
  [Close current data file and open OutFileName]
CONFIGURATION ConfigFileName
  [Load a new configuration, ConfigFileName]
POINT xp yp zp
  [Point field calculation at the given coordinates]
SCAN xp1 yp1 zp1 xp2 yp2 zp2
  [Scan along a line between the given coordinates]
GENSCAN
  xp1 yp1 zp1
  ...
  xpn ypn zpn
END
  [Write field values at listed coordinates]
VOLUMEINT [RegNo]
  [Volume integrals, optionally for region RegNo]
SURFACEINT Reg1 Reg2 ... RegN
  [Surface integrals around the region set]
MATRIX FileName XMin XMax NX YMin YMax NY ZMin ZMax NZ
  [Write a matrix of field values to the file FileName]
NSCAN 100
  [Set the number of points in a line scan]
INTERPOLATION [LSQ,Linear]
  [Set the interpolation method]
ENDFILE
  [Terminate the analysis]
```

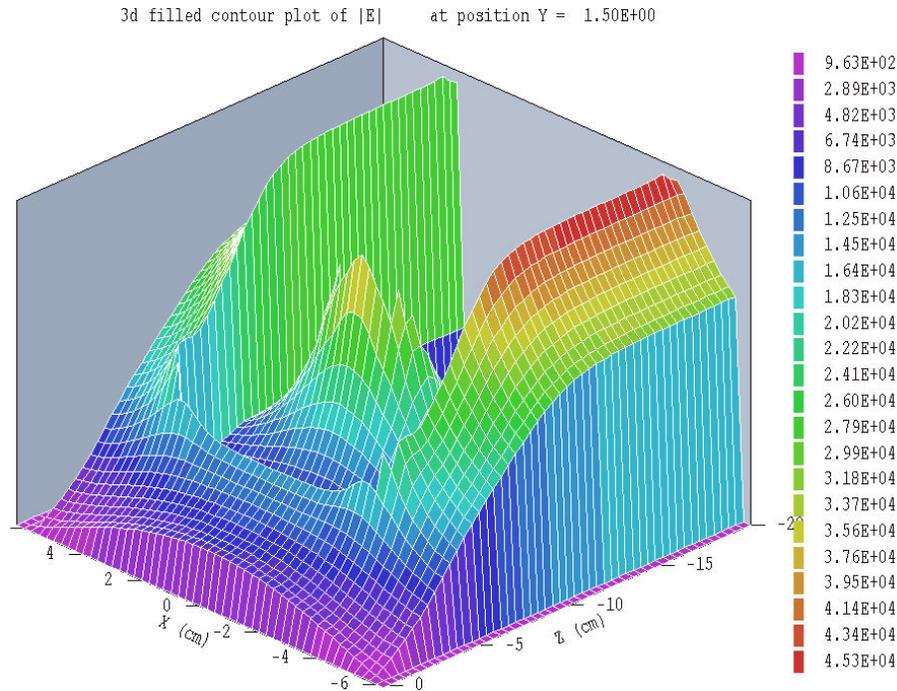


Figure 18: Plane plot of electrostatic potential - type *3D filled contour*.

## 6.2 Plane plots

Plane plots (Fig. 18) are two-dimensional plots that show the variation of quantities over planes that cut the solution volume. There are three differences from the slice plots discussed in the next section:

Plane plots are created from a rectangular mesh of values while slice plots built using the computational mesh. Slice plots show precise region boundaries and equipotential lines in true scale.

Because they are simpler to construct, plane plots support a wider variety of plot styles.

Plane plots are not limited to slices normal to a Cartesian axis.

### SET PLANE

This command brings up the dialog of Fig. 19 to set the plane for the plot. 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 field group at the bottom-left determines the plot range in the normal plane. The default settings are the limits of the solution volume. Note that plane plots are constructed to fill the maximum area. They do not preserve scaling in the normal plane. The final command group at the bottom-right control an optional rotation of the plot plane. The radio buttons set the pivot axis, the quantity *Rotate* is the rotation angle in degrees and *Position* is the pivot position along the direction normal to the pivot axis.

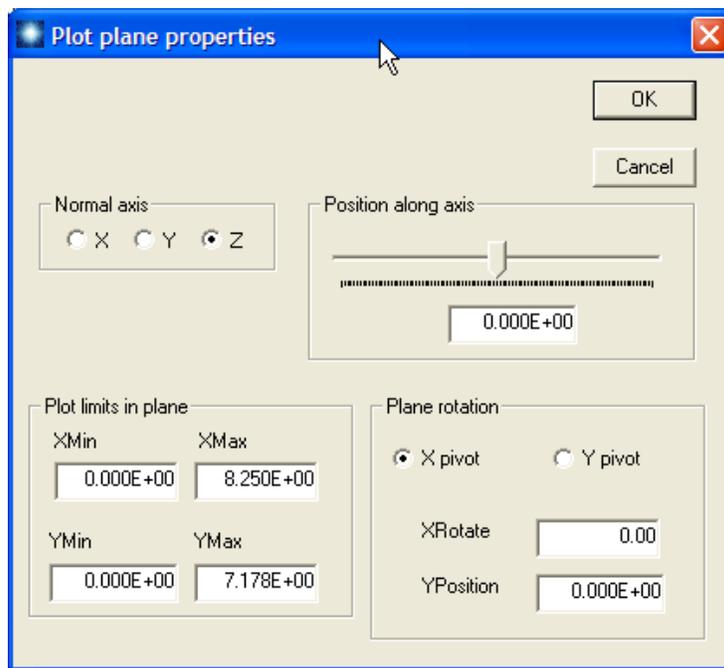


Figure 19: Set plane dialog.

The following commands are in the *Plot control* popup menu:

## PLOT STYLE

This command brings up the dialog of Fig. 20 to set the plot style. As an example, Fig. 18 shows the *Filled contours 3D* plot style. The numbers at the bottom give the resolution of the mesh used to create the plot. Higher values give more detail but require longer regenerate times. The default is a  $100 \times 100$  mesh.

## PLOT QUANTITY

Set the quantity to be plotted. the choice depends on the current configuration file. For `phiview_dielectric.cfg`, the quantities are  $E_x$ ,  $E_y$ ,  $E_z$ ,  $|\mathbf{E}|$ ,  $\phi$ ,  $U_e$ ,  $\epsilon_r$  and  $\rho$ , where  $U_e$  is the electrostatic field energy density and  $\rho$  is the space charge density. The configuration file `phiview_conductive.cfg` does not include  $\epsilon_r$  and  $U_e$  but adds the quantities  $j_x$ ,  $j_y$ ,  $j_z$ ,  $|\mathbf{j}|$ ,  $P$  and  $\sigma$ . Here,  $P$  is the resistive power density and  $\sigma$  is the electrical conductivity. Note that discontinuous quantities like the electric field and current density plots may appear jagged at boundaries because of the limited resolution of the mesh.

## PLOT LIMITS

Set limits for the plotted quantity. When *Autoscale* is active, **PhiView** automatically sets limits based on the range of values in the solution file. To plot an absolute range, uncheck the box and enter numbers for the minimum and maximum values. In either case, the program shows lines in plots of type *Contour lines 2D* at a uniform interval. You can set values for contour lines manually. See Sect. 8.2 for information.

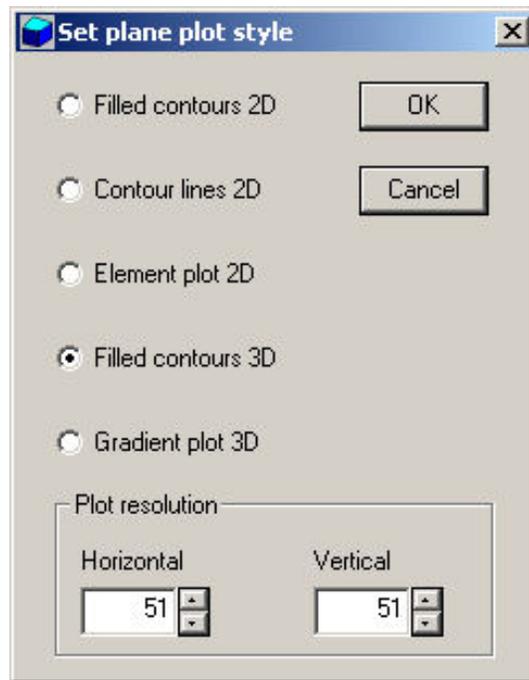


Figure 20: Plot style dialog.

## ROTATE PLOT

This command is active only for the *Filled contours 3D* and *Gradient plot 3D* styles. You can rotate the plot in  $90^\circ$  increments for the best view.

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

## DEFAULT PRINTER

With this command, a **PhiView** plot can be ported to any installed Windows printer (including network printers, postscript drivers, PDF drivers...). Note that the current screen plot is sent to the default Windows printer. If necessary, change the default in Windows before running **PhiView**.

## 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.

## 6.3 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: AMaze
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 **PhiView** 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. **PhiView** 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, **PhiView** 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

---

## 7 PhiView – numerical analysis

The commands of the *Analysis* popup menu are used to determine numerical values for the solution. In the interactive mode, the menu is active only when a data file is open.

### 7.1 Automatic analyses operations

#### POINT CALCULATION

The command brings up a dialog where you enter the  $x$ ,  $y$  and  $z$  coordinates of any point in the solution space. In contrast to the *Point calculation* command of the *Slice plots* menu, the point is not limited to a slice plane. When you click *OK*, **PhiView** displays an information box and also writes the results to the data file in the following format:

```
--- Point Field Analysis ---
Position: [ 3.0000E+00,  1.5000E+00,  7.4205E-01]
Region number:  3
Ex:  5.511018E+00
Ey:  2.770687E+00
Ez:  2.803520E+00
|E|:  6.775526E+00
Jx:  5.511018E+00
Jy:  2.770687E+00
Jz:  2.803520E+00
|J|:  6.775526E+00
Phi:  1.675318E-01
PowerDens:  4.590775E+01
Sigma:  1.000000E+00
```

The set of values depends on the quantities defined in the *INTERPOLATION* section of the currently-loaded configuration file.

#### LINE SCAN

In the *Analysis* menu, scans may be performed along arbitrary straight lines between any two points in the solution volume. When you click the *Line scan* command a dialog appears in which you can specify the start and end points. Enter values in units set by *DUnit*. In other words, if you used dimensions of cm in **MetaMesh** and the **HiPhi** script contains the command  $DUnit = 100.0$ , then enter the dimensions in cm. In the interactive mode, **PhiView** creates a screen plot of a selected quantity determined by the *Set scan quantity* command. Features of the scan plot environment are discussed in Sect. 8.3. The following is an example of a listing.

```

----- Field scan between points -----
  XStart:  5.0000E-01  YStart:  0.0000E+00  ZStart:  7.4205E-01
    XEnd:  3.0000E+00    YEnd:  1.5000E+00  ZEnd:  7.4205E-01

```

```

X          Y          Z          NReg
5.000000E-01  0.000000E+00  7.420529E-01  4
5.166667E-01  1.000000E-02  7.420529E-01  4
5.333333E-01  2.000000E-02  7.420529E-01  4
5.500000E-01  3.000000E-02  7.420529E-01  4
5.666667E-01  4.000000E-02  7.420529E-01  4
5.833333E-01  5.000000E-02  7.420529E-01  4

```

...

```

Ex          Ey          Ez
1.149979E+02  1.439398E+00  1.499865E+01
1.063150E+02  4.025448E+00  1.563426E+01
9.868382E+01  6.680476E+00  1.612934E+01
9.173597E+01  8.884391E+00  1.655349E+01
8.575326E+01  1.005728E+01  1.686338E+01
8.055398E+01  1.091353E+01  1.705007E+01

```

...

## VOLUME INTEGRALS

In response to the command, **PhiView** performs volume integrals of quantities defined in the **VOLUME** section of the configuration file over all elements of the mesh. The calculations are intensive, so there may be a delay for large meshes. **PhiView** records results to the data file in the following form:

```

----- Volume Integrals -----

```

```

Global volume:  9.474960E-04

```

```

RegNo  Region volume

```

```

=====

```

```

 1  1.737214E-04
 2  2.377571E-04
 3  5.161524E-04
 4  1.974895E-05
 5  5.175411E-08
 6  6.440011E-08

```

```

Quantity: Power

```

```

Global value:  3.158969E-02

```

```

RegNo  Region volume

```

```

=====

```

```

 3  2.063170E-02
 4  1.005183E-02
 5  9.061643E-04

```

The first table always shows the global and region volumes. Additional tables list results for quantities defined in the **VOLUME** section of the configuration file. In a dielectric analysis, the integral is taken over the electrostatic field-energy density. You can use the energy values to determine capacitance in a solution with a single electrode at voltage  $V_0$  from the equation:

$$U_0 = \frac{C V^2}{2}. \quad (12)$$

In Eq. 12 the quantity  $U_0$  is the global field energy. In a resistive analysis, the calculated quantity is the resistive power density to yield the regional and global power ( $P_0$ ). You can use the equation:

$$P_0 = \frac{V_0^2}{2R}, \quad (13)$$

for a single-electrode system to find the resistance.

## SURFACE INTEGRALS

In response to this command, **PhiView** takes integrals of defined quantities over the selected region surfaces. The program displays the dialog of Fig. 21 which includes a row for each region along with the name assigned in **MetaMesh** and forwarded by **HiPhi**. Activate the check boxes in the *Internal* column to specify the regions to include in the set. The default is to include the full exterior surface of the region set (*i.e.*, all remaining regions are *External*). Use the check boxes to specify *External* regions. For example, to take a surface integral over the boundary between Regions 2 and 5, set Region 2 as *Internal* and Region 5 as *External*.

The calculated quantities are defined in the **SURFACE** section of the configuration file. For a dielectric analysis, the surface integral gives the total charge on the region set:

$$Q = \int \int_S \epsilon_r \epsilon_0 \mathbf{E} \cdot \mathbf{n} dA. \quad (14)$$

The induced charge can be used to calculate mutual capacitances of a multi-electrode system. As an example of a region set, suppose we want to find the charge on an electrode with sharp corners. A useful technique is to enclose the electrode in a vacuum diagnostic region with a simple shape. We then take the integral over the surface of the combined set that includes the electrode and the diagnostic region. For conductive solutions, the surface integral is the net current out of the region set.

## MATRIX FILE

This command controls a feature that is useful if you want to write your own analysis routines or port results to mathematical software. In response to the command, **PhiView** performs interpolations over a specified box region on a regular grid of values. It is much easier to use results in this form than to deal with the conformal mesh directly. The command calls up the dialog of Fig. 22. Specify the dimensions of the box along each axis (in units set by *DUnit*) and the number of calculation intervals. To illustrate, calculations are performed at positions with  $x$  coordinates given by

$$x = x_{min} + \frac{n (x_{max} - x_{min})}{n_x}, \quad (15)$$

where  $n = 0, 1, 2, \dots, n_x$ . For example, if you set  $x_{min} = 0.5$ ,  $x_{max} = 1.5$  and  $n_x = 10$ , the calculations are performed at points with  $x = 0.5, 0.6, \dots, 1.4, 1.5$ . The recorded quantities depend on definitions in the configuration file and the setting specified in the *Recorded quantities*

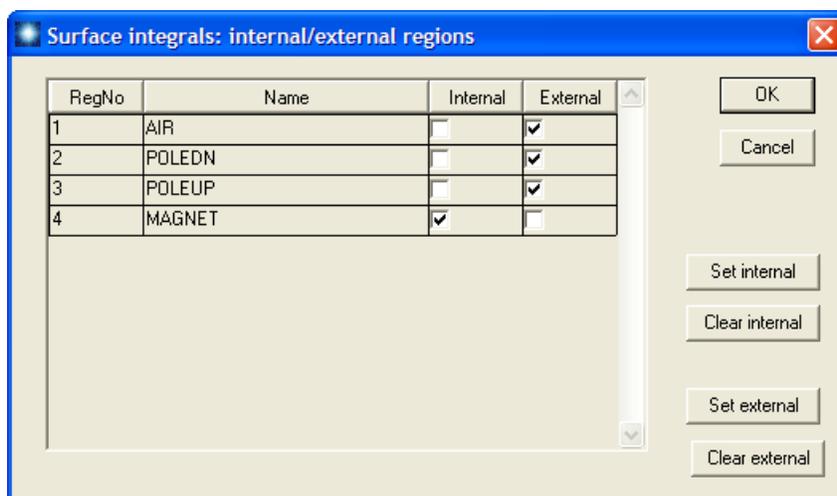


Figure 21: Dialog to pick the region set for surface integrals.

command. The interpolation routine returns zero if a portion of the box is outside the solution volume or inside an electrode. You can also specify an output file name. The format of the text file is described in Chap. 13. If you want a two-dimensional matrix of values in a single plane, set either  $N_x$ ,  $N_y$  or  $N_z$  equal to zero. For example, if  $N_x = 0$ , then a set of calculations is performed over the specified range of  $y$  and  $z$  at the single position  $x = (x_{min} + x_{max})/2.0$ .

## SCAN QUANTITY

With this command you can pick the quantity that will be displayed in screen and exported plots of line scans. Pick the quantity from the list box and click *OK*. The listed quantities depend on the entries in the INTERPOLATION section of the configuration file. This setting has no effect on the data file listing which includes all values specified by the RECORDED QUANTITIES command.

## SET NUMBER OF SCAN POINTS

This command sets the number of line scan points in the screen plot and data file listing. The default value is 150 and the maximum number is 500.

## LSQ/LINEAR INTERPOLATION

Set the interpolation method for analysis and plots. The default is a second-order least-squares-fit procedure that uses data collected from nearby elements. Under the *Linear* option, only node values from the target element are included. This option is useful in tight spaces near material boundaries.

## RECORDED QUANTITIES

This command determines the calculated quantities included in scan listings and matrix files. Under the option *Field quantities*, **PhiView** records the first three quantities defined in the INTERPOLATION section of the configuration file. The quantities are  $E_x$ ,  $E_y$  and  $E_z$  in the standard configurations. Use this option to export field tables to **OmniTrak**. Under the *Full set* option, **PhiView** records all quantities defined in the configuration file.

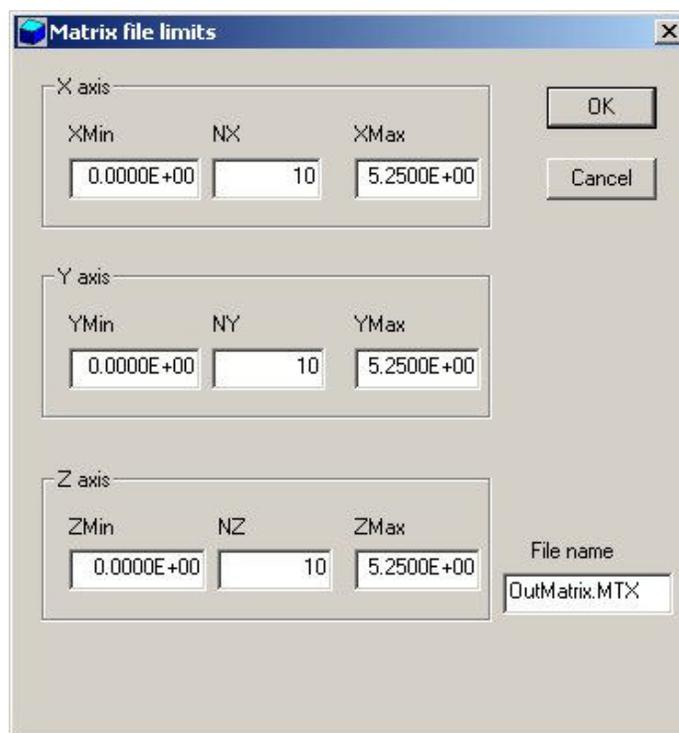


Figure 22: Create matrix file dialog.

## 7.2 PhiView analysis scripts

**PhiView** scripts are useful for automating complex or repetitive analysis procedures. For example, you may want to compare a line scan of values between two points in a set of similar solutions. In the interactive mode, it would be necessary to type in the same coordinates for each scan. An analysis script can load each solution file, perform the scan and record all information in a single data file. In the **PhiView** main menu, the following commands in the *File* popup menu (Sect. 6.1) apply to analysis scripts.

### **RUN SCRIPT**

Pick and run a script file with a name of the form `FPREFIX.SCR`. The script can operate on the currently-loaded solution file or load a different file. The last solution file loaded by the script becomes the current file for plots and interactive analysis.

### **CREATE SCRIPT**

Use this command to create an analysis script using the internal program editor.

### **EDIT SCRIPT**

Use this command to change an existing script with the internal program editor.

**PhiView** may be called as a command line task. The calling syntax is

```
[progbath\]PHIVIEW [scriptpath\]SPREFIX
```

where the script has the file name **SPREFIX.SCR**. You should supply the full path `[scriptpath]` if the script is not in the current directory.

Command line operation opens several possibilities.

**PhiView** runs quickly and silently in the background when launched from the *Command Prompt* in a terminal window. To minimize typing, open the terminal in the working directory. Use the batch command **CHDIR** to change directories.

The program may be called from the command prompt with commands of the form

```
START /w [progbath]\PhiView [scriptpath\]SPREFIX  
START [progbath]\PhiView [scriptpath\]SPREFIX
```

Use batch files to organize calculations that run autonomously. The option `/w` in the first example signals that the batch file waits for completion of the program to continue. This form is useful if data from **PhiView** is required for the next operation. Omit the option if you want to launch several instances of **PhiView** to take advantage of a multiple-core machine. Use the **CHDIR** command to set the working directory.

Run **PhiView** as a subtask from your own compiled or interpreted programs. One application is to integrate finite element solutions and analysis into an optimization loop.

### 7.3 Script commands

Entries on a command line may be separated by the following set of delimiters: space, tab, equal sign [=], colon [:], left parenthesis [(] or right parenthesis [)]. Each command listed below is shown in symbolic form along with an example of how they might appear in a script.

**INPUT [datapath\] FileName**

**INPUT = E:\FEData\AprilRuns\HVCross.HOU**

Close the currently-opened solution file and load a solution file from the current directory. If `[datapath]` is not specified, the solution file must be in the working directory.

**OUTPUT [datapath\] FileName**

**OUTPUT: WGUIDE1.DAT**

Close the current data record file and open a new one. If a `[datapath]` is not specified, the file will be written in the working directory. A data file must be opened to use the following analysis commands.

**CONFIGURATION [datapath\] FileName**

**CONFIGURATION = \FieldP\AMaze30\PHIVIEW\_FORCE.CFG**

Load a new PhiView configuration file to change the analysis characteristics. If a `[datapath]` is not specified, the configuration file must be available in the working directory. If this command does not appear, the program uses the current configuration or the one in force in the last run.

**POINT xp yp zp****POINT = (0.00, 0.05, 4.67)**

Perform a point calculation and write values of the quantities defined in the **INTERPOLATION** section of the configuration file to the data record file. Enter coordinates in the units set by *DUnit*.

**SCAN xp1 yp1 zp1 xp2 yp2 zp2**

**SCAN = (0.00, 0.00, 15.00) (12.00, 0.00, 15.00)** Perform ( $NScan + 1$ ) calculations along a line in space. At each point, write the first  $NRecord$  quantities defined in the **INTERPOLATION** section of the configuration file to the data record file. Enter coordinates in units set by *DUnit*.

**GENSCAN**

Perform any number of calculations along an arbitrary path in space. At each point, write the first  $NRecord$  quantities defined in the **INTERPOLATION** section of the configuration file to the data record file. The command must be part of a structure with the following form:

```
GENSCAN
  xp1 yp1 zp1
  xp2 yp2 zp2
  ...
  xpn ypn zpn
END
```

Each data line contains three real numbers separated by spaces to define a point in the solution volume. Enter coordinates in units set by *DUnit*.

**VOLUMEINT [NReg]****VOLUMEINT****VOLUMEINT 8**

Perform a volume integral of quantities defined in the **VOLUME** section of the configuration file and write the results to the data record file. If a region number does not appear, the integral is taken over the full solution volume. Otherwise, the integral extends over elements with region number  $NReg$ . Volume integrals may take a long time, so use the region-number option if there are several regions that are not of interest.

**SURFACEINT Reg1 Reg2 ... RegN****SURFACEINT 5 7 9 12**

Perform a surface integral of quantities defined in the **SURFACE** section of the configuration file and write the results to the data record file. Positive integer values correspond to *Internal* regions and negative to *External* regions. With no specification, all regions that are not *Internal* are taken as *External*. The integral is taken over the surface facets between any *Internal* and *External* elements. The command may include any number of regions. The regions in the *Internal* set may or may not be contiguous. Note that **PhiView** does not include facets on the boundary of the solution volume in surface integrals.

**MATRIX FileName XMin XMax NX YMin YMax NY ZMin ZMax NZ**  
**MATRIX WGUIDE.MAT 1.00 1.00 10 10.00 2.00 10 2.00 12.00 20**

Write values calculated at an array of location to a data file in text format. The file is named *FileName* and is created in the current directory. The real-number values  $x_{min}$  and  $x_{max}$  are the limits of the listing volume along the  $x$  direction. The parameter  $N_x$  is an integer. Listings are made at  $(N_x + 1)$  evenly spaced positions along  $x$ .

**NSCAN NScan**  
**NSCAN = 100**

Set the number of intervals for line scans. The default is  $NScan = 150$  and the maximum value is  $NScan = 500$ .

**RECORD [Field, Full]**  
**RECORD = Full**

Set the number of quantities to include in data files in response to the *Scan* and *Matrix* commands. The *Field* set (default) consists of  $E_x$ ,  $E_y$  and  $E_z$ . Use this option to create data tables for **OmniTrak**. The *Full* set includes all quantities defined in the INTERPOLATION section of the configuration file.

**INTERPOLATION [LSQ,Linear]**  
**INTERPOLATION = Linear**

Set the interpolation type to a multi-element least-squares fit or a single-element linear fit.

**PLOT FSaveView FOutput Nx Ny**  
**PLOT (XAxis3D VIEW001 800 600)**

Write a plot file based on data in the currently loaded solution. The string *FSaveView* is the prefix of the file **FSaveView.FPV**, a set of view parameters created with the *Save view* command (Sect. 6.3). The view file must be available in the working directory. The plot is saved in the working directory. It is in Windows Bitmap format and has the name **FOutput.BMP**. The integers  $N_x$  and  $N_y$  give the width and height of the image in pixels. To optimize compatibility with graphics format converters, pick values that are multiples of 16 (e.g.,  $1024 \times 768$ ).

**ENDFILE**

Terminate the analysis

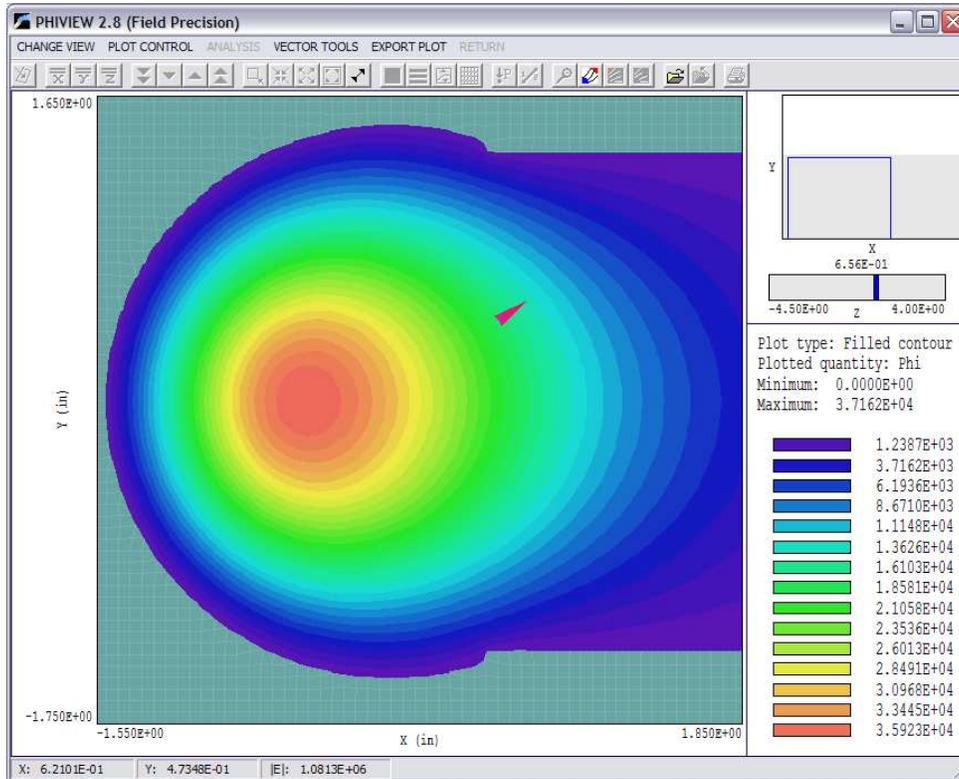


Figure 23: Slice plot – filled-contour plot style.

## 8 PhiView – slice plots

### 8.1 Setting the slice view

Slice plots are two-dimensional plots that show the variation of quantities over a plane normal to one of the Cartesian axes. In contrast to plane plots, slice plots are based on the structure of the mesh projected to a slice plane. This structure may be complex for a conformal mesh. To facilitate the process, slices are constructed at discrete locations along the normal axis corresponding roughly to the planes of the foundation mesh. The precise rendering of mesh information enables point-and-click analysis operations (point calculation, line scan, ...) in the slice.

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

#### SET SLICE PLANE PROPERTIES

This command calls a dialog similar to that of the *Set plane* command in the plane plot menu (Fig. 19). You can change the normal axis, change the position along the normal axis, and set plot limits in the normal plane.

**SLICE NORMAL TO X**  
**SLICE NORMAL TO Y**  
**SLICE NORMAL TO Z**

Quick commands to change the normal axis.

**JUMP FORWARD**  
**STEP FORWARD**  
**STEP BACKWARD**  
**JUMP BACKWARD**

Move along the slice axis by small or large steps. The small step is approximately one layer of the foundation mesh and the large step is 5 layers. The term *forward* implies motion toward higher indices of the normal axis. The slider bar in the orientation area to the right of the plot (Fig. 23) shows the current location.

**ZOOM WINDOW**

As an alternative to the entries in the *Set slice plane* dialog, you can interactively change plot limits in the normal plane using the mouse. Choose the command and move the mouse pointer into the plot area. The status bar enters coordinate mode. It shows the current mouse position in the plot. 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. 23) 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 start and end points for the shift.

## 8.2 Setting slice plot properties

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

**SET SLICE PLOT STYLE**

This command brings up the dialog of Fig. 24. The *Region* plot style is a cross-section view



Figure 24: Slice plot style dialog.

of the mesh element divisions color-coded by region. In contrast to the logical plane plot of **MetaMesh**, **PhiView** attempts to resolve the exact mesh structure in the plane. In the *Filled contour* style, the program determines 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. When the *Element outline* box is checked, **PhiView** includes facets in the *Region* and *Element* modes. In comparing relative advantages, the *Filled contour* plot provides the most attractive and accurate display in extended field regions. Use the *Element* plot for the best view of discrete field changes at the boundaries of electrodes and dielectric materials. The *Contour* plot is useful for checking field uniformity over a volume.

## PLOT QUANTITY

Use this command to set the quantity for color-coding in *Filled contour* and *Element* plots and line values in *Contour* plots. The choices depend on entries in the currently-loaded configuration file.

## PLOT LIMITS

This command opens the dialog of Fig. 25. In the *Autoscale* mode, **PhiView** chooses defaults for the minimum and maximum values for colors and contours based on the range of values of the current plot quantity in the current slice. Deactivate *Autoscale* to set the values manually. The manual mode is useful for comparing absolute field distributions in different slices. In either case, **PhiView** uses a uniform interval between the limits based on the number of contour lines. For plots in the *Contour* style, you can set specific values for the contour lines. Applications include logarithmic intervals or the boundary of a region above a critical electric field. To use this feature, open a text document using the internal editor or an external program. Make a list of the values you want, using any valid numerical format. There should be one entry per line and the entries should appear in ascending order. Save the document if you want to compare a reference set. Highlight the values and copy them to the Windows clipboard. Then return to the plot limits dialog and check the *Values from clipboard* box. **PhiView** displays an error

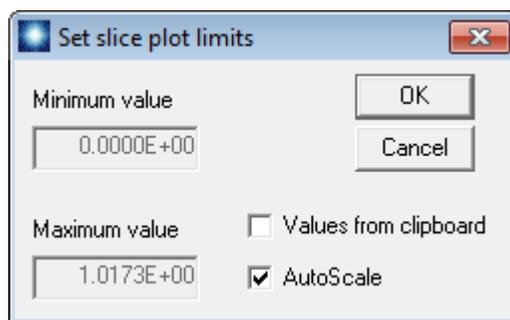


Figure 25: Dialog to set limits of the plotted quantity.

message if the number set is invalid. Otherwise, the program reads the data, exits the dialog and updates the plot. Note that you will see a change only if the *Contour* style is displayed. Other plot styles are not affected.

## RESET PLOTS TO DEFAULT

Return to the default view that applied when the solution was loaded

## TOGGLE GRID

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

## GRID INTERVALS

This command brings up a dialog to set intervals for grid lines in the horizontal and vertical directions. Deactivate the *AutoGrid* box and enter real numbers in the boxes. The values should be specified in the length units set by the *DUnit* parameter.

## NUMBER OF CONTOUR LINES

Change the number of lines for *Filled contour* and *Contour* plots.

## 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.

## SNAP DISTANCE

Change the value of *DSnap* from the default value determined by the program.

## 8.3 Analyses in a slice

You can determine field values at points and along scan lines with the commands of the *Analysis* popup menu.

### POINT CALCULATION

This command is useful to make quick checks of fields in the solution volume. After you click the *Point calculation* command, move the mouse into the plot area. The mouse pointer changes to a cross-hair pattern and the status bar enters coordinate mode. Click the left button to specify a point or press the *F1* key to enter the coordinates from the keyboard. Note that mouse coordinates will shift between discrete values if snap mode is active. **PhiView** calculates potential and field values at the point in the normal plane given by the coordinates. The included quantities depend on the currently-loaded configuration file. The results are recorded if a data file is open.

### LINE SCAN

Line scans are one of the most useful **PhiView** capabilities. After clicking on the command, supply two points with the mouse to define a scan line (or press the *F1* key to enter coordinates manually). The snap mode is useful in this application (for example, you may want the scan to extend from 0.000 to 5.000 rather than 0.067 to 4.985.) The program computes a series of values of field quantities in the normal plane at equal intervals along the line. Complete information is recorded if a data file is open. The program also makes a screen plot of the currently-selected quantity versus distance along the scan and activates the *Scan plot* display (Fig. 26). **PhiView** adds fiducial lines to the plot using intelligent grid selection. This means that the plot is adjusted to fill the screen and grids are drawn at useful intervals (*i.e.*, 0.05 or 2.00).

### SET SCAN QUANTITY

With this command you can pick the quantity that will be displayed in screen and exported plots of line scans. Pick the quantity from the list box and click *OK*. The listed quantities depend on the entries in the *INTERPOLATION* section of the configuration file. This setting has no effect on the data file listing which includes all requested field quantities.

### SET NUMBER OF SCAN POINTS

This command sets the number of line scan points in the screen plot and data file listing. The default value is 150 and the maximum number is 500.

## 8.4 Scan menu

As shown in Fig. 26, the current scan quantity is plotted as a function of distance from the start point. The coordinates of the start and end points are listed above the graph. Coordinates are plotted in units set by the *DUnit* parameter. In the default mode, **PhiView** sets limits so that the plot fills the screen. The program includes grid lines at even intervals. The status bar at the bottom lists the solution file, the data file (if open) and the grid intervals. The *Export plot*

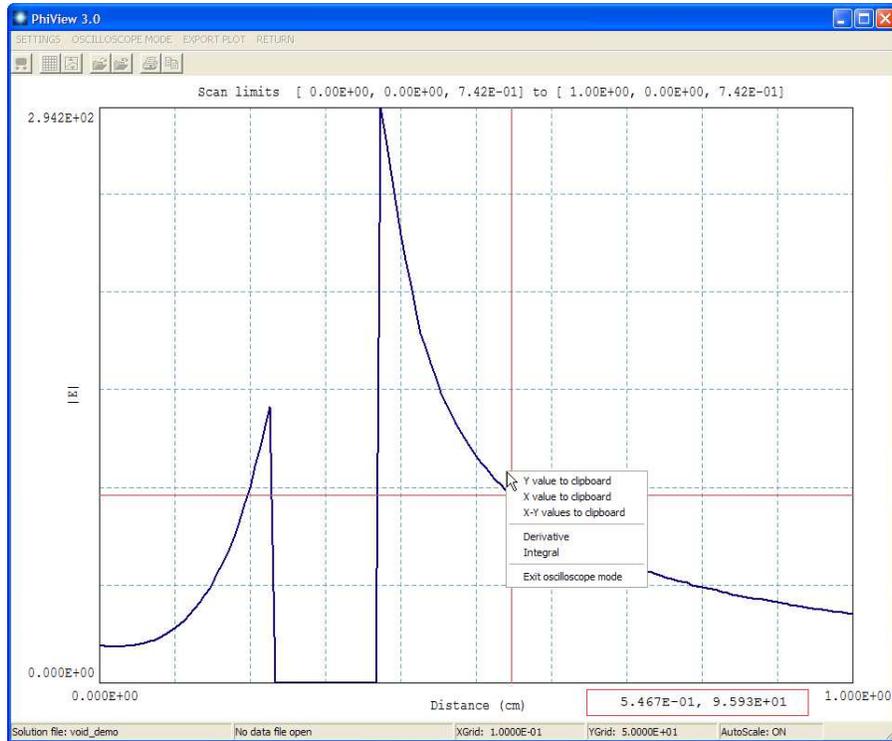


Figure 26: Scan plot display.

menu includes that standard set of commands described in the previous chapter. The *Settings* menu contains the following commands.

### SYMBOL DISPLAY

Toggle the plot display to either a thick line or a set of symbols at calculated points connected by a thin line.

### VERTICAL LIMITS

Override automatic selection and set vertical limits for the current quantity. To set manual limits, uncheck the *AutoScale* box and enter real-numbers values for the lower and upper limits of the plotted quantity. Check the *Autoscale* box to return to the default mode.

### TOGGLE GRID

Suppress or activate the display of grid lines.

### GRID INTERVALS

Override automatic selection of grid intervals. To set values for the horizontal and vertical grid intervals, uncheck the *AutoGrid* box and enter real-number values. For the horizontal direction, enter the interval in length units set by the *DUnit* parameter. Check the *Autogrid* box to return to the default mode.

## OPEN DATA FILE

In the oscilloscope mode, you can record a variety of numerical values for the scan. The information will be written to a data file if already opened. Use this command to open a new data file.

## CLOSE DATA FILE

Close the current data file. This command should be invoked before attempting to use the file in another program.

## OSCILLOSCOPE MODE

In response to this command, the scan plot becomes interactive with many of the features of a digital oscilloscope. The commands and tools of the *Scan plot* menu are deactivated and **PhiView** displays the red cross-hair lines shown in Fig. 26. Values of the distance and plot quantity at the cross-hair intersection are displayed in the red box below the plot. The intersection point follows the mouse when the cursor is inside the plot area. Note that the intersection moves discontinuously between the data points. To check points at the end of the interval, move the mouse slowly toward the left or right-hand boundaries.

If a data file is open, **PhiView** records values at the current intersection point when you click the left mouse button:

```
Scan point calculation
  Distance:  9.506666E-01 (cm)
  Phi:      0.384693E+04
```

Clicking the right button activates the pop-up menu shown in Fig. 26. The menu contains the following commands:

## Y VALUE TO CLIPBOARD

Copy the value of the plotted quantity at the cross-hair intersection to the Windows clipboard. You can then paste the number into a text document, spreadsheet or calculator program.

## X VALUE TO CLIPBOARD

Copy the value of the distance (in units set by *DUnit*) at the cross-hair intersection to Windows clipboard.

## XY VALUES TO CLIPBOARD

Copy the values of the distance and plotted quantity to the clipboard. The *x* value is on the first line and the *y* value on the second.

## DERIVATIVE

Calculate and record the numerical derivative at the intersection point. The dimensions are determined by the *DUnit* parameter. For example, if the scan quantity is potential in V and the distance is shown in centimeters, the derivative has units V/cm. **PhiView** writes the value to the clipboard and also displays the results in a dialog. If a data file is open, the program makes a record like the following one:

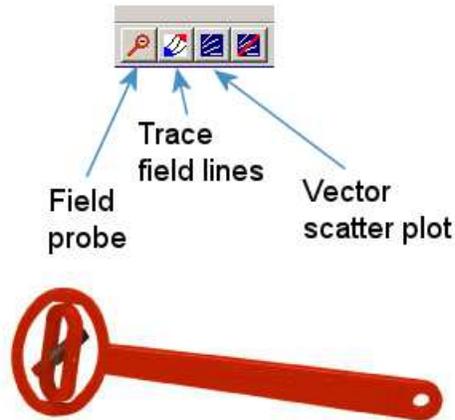


Figure 27: Tools to display vector information in the slice plot menu.

Derivative

Location: 1.694667E+00 (cm)  
 Phi/(cm): 3.654382E+05

**INTEGRAL**

Calculate and record the definite integral of the plotted quantity from the start point of the scan to the cross-hair intersection point. The dimensions are determined by the *DUnit* parameter. For example, if the scan quantity is the electric field in V/cm and the distance is shown in centimeters, the integral has units (V/cm)-cm = V. **PhiView** writes the value to the clipboard and also displays the results in a dialog. If a data file is open, the program makes a record like the following one:

Definite integral

Location: 1.908000E+00 (cm)  
 Phi-(cm): 9.679801E+03

**EXIT OSCILLOSCOPE MODE**

Activate commands and tools and return to the standard *Scan plot* mode. You can also press the *Esc* key to exit the oscilloscope mode.

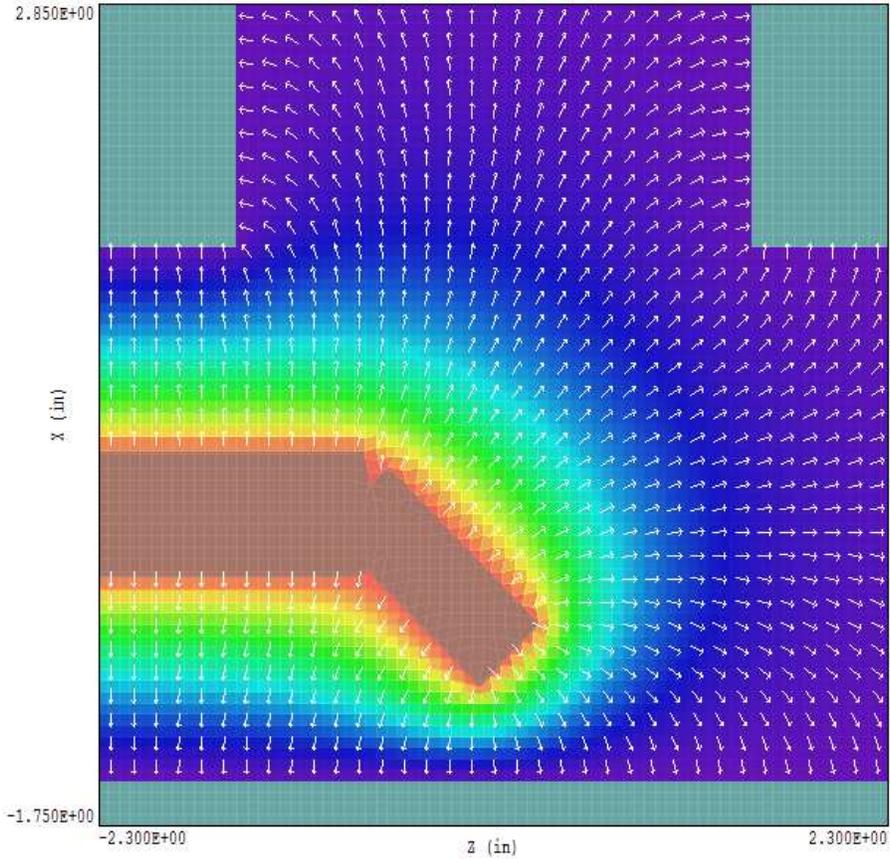


Figure 28: Element plot with electric field vectors.

## 8.5 Slice vector tools

**PhiView** has useful tools to display the direction of the vector quantities  $\mathbf{E}$  and  $\mathbf{j}$  in slice plots. Figure 27 shows the corresponding entries on the toolbar.

### **E PROBE**

This feature was inspired by the familiar Magnaprobe illustrated in Fig. 27. When you click on the tool and move the cursor into the slice-plot area, it changes to a semi-transparent probe that rotates about a pivot point to show the direction of the electric field. The direction also applies to the current density which is parallel to  $\mathbf{E}$  in isotropic media. The probe functions in all plot styles and may be combined with field-line traces and scatter plots. The status bar shows the coordinates and  $|\mathbf{E}|$  at the pivot point.

### **FIELD LINES AT POINTS**

Use this command to add lines of  $|\mathbf{E}|$  projected to the slice plane. The program enters coordinate entry mode when you click the command. Move the mouse to a point in the solution volume and click the left button. **PhiView** calculates the three-dimensional path of an electric field line that passes through the point and plots the projection in the slice plane. You can continue to add any number of lines. Click the right mouse button or press *ECS* to exit coordinate mode.

The lines are not included in hardcopy or plot file exports and disappear if you change the slice view. Use a screen capture utility to record them. It is important to recognize the nature of the plots. The lines are three-dimensional curves projected to the slice plane. They may be difficult to interpret if the line does not lie close to the slice plane. For full three-dimensional field line plots, see the *Field line plot file* command in Chap. 9.

### **VECTOR SCATTER PLOT**

Superimpose a uniform distribution of vector arrows pointing in the direction of  $\mathbf{B}$  and  $\mathbf{H}$  in the slice plane (Fig. 28). Arrows may be added to any of the plot types, including *Region*. They are included in hardcopy and plot file export. The arrows are preserved and adjusted if you change the slice view.

### **REMOVE VECTORS**

Use this command to turn off the vector scatter plot mode and to remove vector tool displays from the current plot.

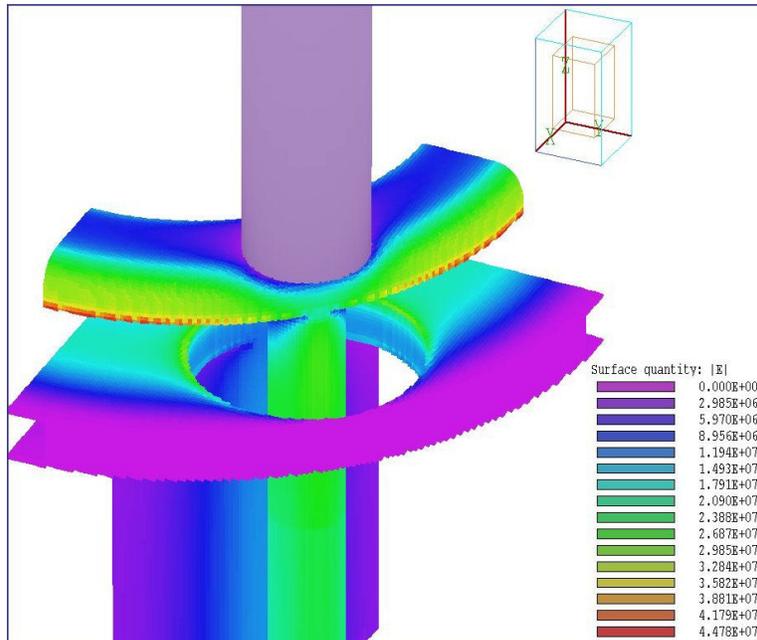


Figure 29: Three-dimensional surface view of region boundaries with color coding by electric field magnitude.

## 9 PhiView – surface plots

Surface plots are three-dimensional views of the solution space (Fig. 29). Three types of information may be displayed: 1) boundaries of regions color coded by region number and shaded with distance from the view, 2) region boundaries color coded by the values of computed quantities on the surface and 3) computed quantities in a slice plane normal to one of the Cartesian axes. Slice plane plots may be superimposed on either of the region boundary plot types. Surface plots are created from the conformal mesh and preserve true spatial scaling.

The method to control the three-dimensional display with the mouse is the same as that in **MetaMesh**. Figure 30 shows the active areas of the screen. The cursor changes as you move between zones. 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. You can control additional aspects of the three-dimensional view with the commands of the *Adjust view* popup menu.

### SET VIEW DIALOG

This command brings up a dialog (Fig. 31) where you can set specific view angles, displacements and the relative distance to the viewpoint,  $DView$ . The parameter  $DView$  controls perspective. Set it to a large value ( $DView \gg 1.0$ ) for an orthographic view. The minimum value is 1.5.

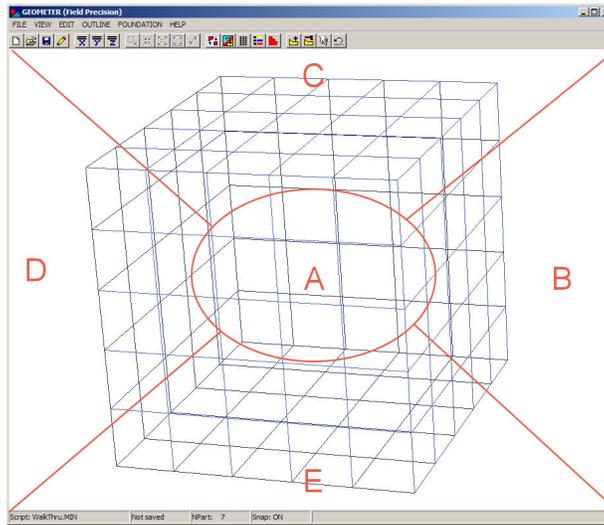


Figure 30: Active areas for mouse control of the 3D view.

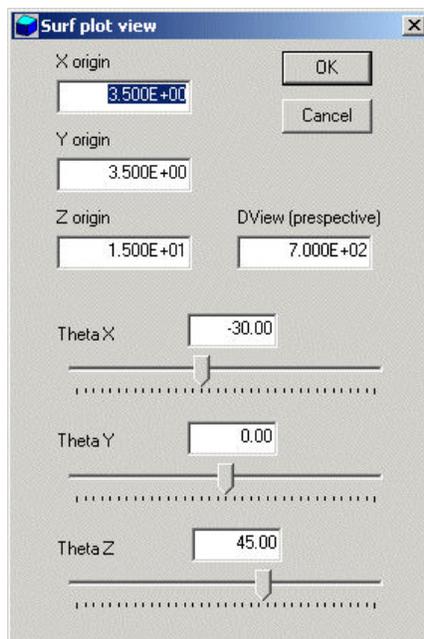


Figure 31: Surface plot control dialog.

## RESTORE DEFAULT VIEW

This command is useful if you lose your orientation after several rotations and translations. The view is returned to the default:  $\theta_x = -30^\circ$ ,  $\theta_y = 0^\circ$  and  $\theta_z = 45^\circ$  with the origin at the center.

## +X VIEW

## +Y VIEW

## +Z VIEW

Rotate to views from the  $+x$ ,  $+y$  or  $+z$  directions. Origin shifts are not affected

## CENTER VIEW

Remove shifts by setting the origin to the center of the solution volume.

## ORTHOGONAL/PERSPECTIVE VIEW

By default, **PhiView** includes perspective correction in surface plots. Click on this command to move the view point to infinity. Click the command a second time to restore the view point.

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

## SURFACE PLOT CONTROL

This command brings up the dialog of Fig. 32. The group of commands on the left-hand side controls the plot style. A three-dimensional plot is constructed from color-coded facets. There are two types of facets: 1) element facets on the boundary of a region with color coding by region number or the selected computed quantity and 2) rectangular facets comprising a slice plane with color-coding by the values of the computed field quantity. Slice plane and region boundary information may be superimposed. The *Facet style* radio buttons control whether the facets of region boundaries are plotted as solid plates (hidden surface) or wireframe outlines. The *Include facet boundaries* check box determines whether the boundaries of region facets are plotted in the hidden-surface mode. (Note that the facets of the slice plane are always plotted as solids with no outline). **PhiView** plots a reference grid along the boundaries of the solution volume when the *Include reference grid* box is checked. The group of commands on the right-hand side controls the slice plane. The *Include slice plane* box determines whether a slice plot is included. The radio buttons determine the axis normal to the slice plane. You may move the plane along the chosen axis by entering values in the box or moving the slider.

## DISPLAYED REGIONS

The command brings up the dialog of Fig. 33 where you can pick region boundaries to include in the plot. Check a box in the *Region* column to activate a region. The buttons in the column marked *Quantity* determine the presentation style for boundary facets. If the box is unchecked, **PhiView** colors facets by region number. If a box is checked in the *Quantity* column, color coding is by the computed field quantity. The information window shows the correspondence between color and the quantity. Note that the field quantity is always calculated at a point near

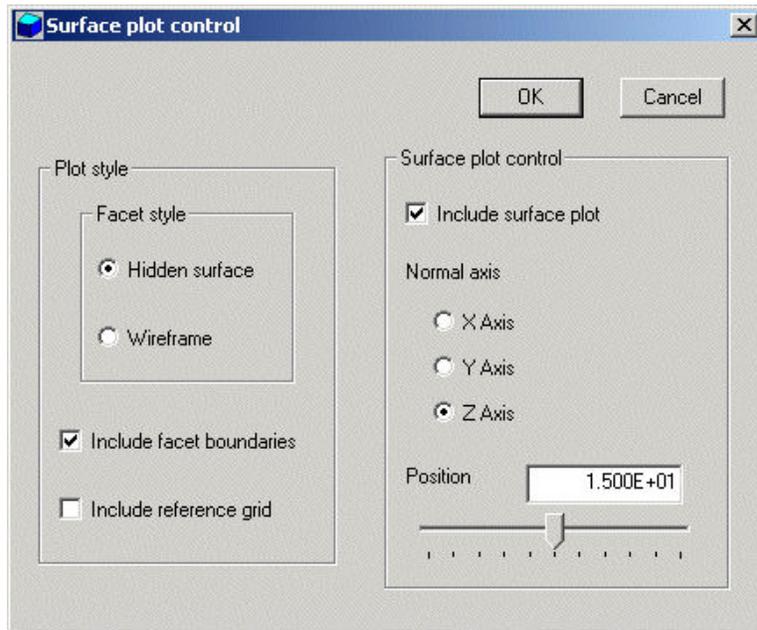


Figure 32: Dialog to set the three-dimensional view.

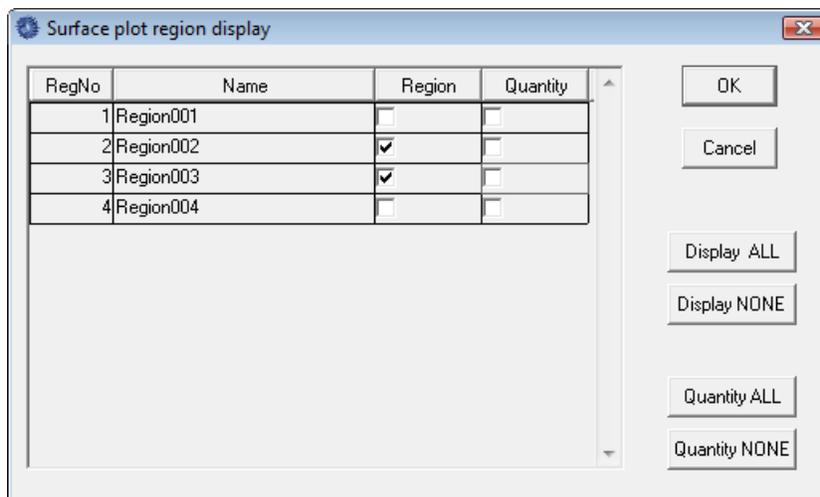


Figure 33: Region display dialog.

the facet **outside** the chosen region. If a boundary separates regions with different materials properties, the displayed quantity values may depend on which region is chosen for display.

### SET CUT PLANES

In a hidden surface plot, internal details may be obscured by surrounding parts or regions. This command brings up a dialog that allows you to adjust the displayed area along the  $x$ ,  $y$  and  $z$  axes. **PhiView** does not display facets that lie outside the limits. With this feature you can create cutaway views. The default is that cut limits are set equal to the dimensions of the solution volume so that all facets are included.

### SET SURFACE PLOT QUANTITY

Set the quantity to be plotted on a slice plane or region facets. The choices depend on the those defined in the currently-loaded configuration file.

### SET PLOT LIMITS

By default the limits of the plotted quantity are automatically computed by **PhiView**. With this command you can set limits manually. Facets where the quantity is outside the set range are plotted in gray. The feature is useful for emphasizing a certain range of values (*i.e.*, portions of an electrode surface where  $|\mathbf{E}|$  exceeds a value). The command is essential on occasions when a distorted element on a region surface leads to an inaccurate field interpolation, skewing the plot range.

### FIELD LINE PLOT FILE

You can add three-dimensional electric field lines to three-dimensional plots with this command. To use it, you must prepare a text file that contains any number of point coordinates. **PhiView** reads the file and adds field lines that pass through each point. The file consists of data lines, where each line contains three real numbers ( $x, y, z$ ) separated by any of the standard delimiters. Enter the coordinates in units set by *DUnit*. The file may contain comment lines that start with an asterisk and should end with the *EndFile* command. Lines are not plotted for target points within fixed-potential regions or outside the solution volume.

## RECORD FIELD LINE PLOT

Use this command to record coordinates of calculated field lines. **PhiView** prompts for the name of an output file with a name of the form `FNAME.FLP`. A file extract is shown below. The command functions only if a file of starting points has been opened with the *Field line plot file* command.

```
Electric field line plots for file: walkthru.HOU
  DUnit(unit conversion factor):  1.00000E+02
```

```
Start point
  X:  8.00000E+00
  Y:  2.50000E-01
  Z:  3.00000E+00
```

```
Forward
 8.00000E+00  2.50000E-01  3.00000E+00
 8.00599E+00  2.50385E-01  3.00003E+00
 8.01197E+00  2.50768E-01  3.00006E+00
 8.01796E+00  2.51151E-01  3.00009E+00
 8.02395E+00  2.51532E-01  3.00011E+00
 8.02994E+00  2.51912E-01  3.00014E+00
```

## CLOSE FIELD LINE PLOT

Close the field line plot file and remove field lines from the current plot.

---

## 10 Customizing PhiView

You can create configuration files to customize the operation of **PhiView** for your application. You can then switch between configurations for different types of analyses. **PhiView** is supplied with two default configuration files in the same directory as the executable program:

`phiview_dielectric.cfg` (analysis of dielectric solutions)

`phiview_conductive.cfg` (analysis of conductive solutions)

`phiview_force.cfg` (force and torque in dielectric solutions)

We recommend that you do not change these files. Instead, make a copy to use as a template and rename it. You can put your new configuration file anywhere, but we suggest you store all configurations in the same directory as `phiview.exe`. Note that **PhiView** remembers the last configuration used and attempts to reload it at the next run.

When you have created a template file, open it in an editor. Note that it is divided into the ten sections shown in Table 10. With few exceptions, do not make changes to the sections `HEADER`, `STOREDQUANT`, `REGIONQUANT`, `PROGPARAM`, `RUNPARAM` and `REALTIMEPARAM`. Entries in these sections coordinate the operation of **HiPhi** and **PhiView**. Modifications may cause file load errors.

### 10.1 Defining calculated quantities

You can define custom quantities for display and analysis in the last four sections of the configuration file:

**Interpolation.** Scalar quantities for multiple uses: 1) displays in slice, plane and surface plots, 2) results of point and scan calculations and 3) values written to matrix output files.

**Vector.** Vector quantities for the display of field or flux lines in slice and surface plots.

**Volume.** Scalar quantities for volume integrals over regions.

**Surface.** Vector quantities for surface integrals over the boundaries of region sets.

Run speed is a critical issue in **PhiView**. The generation of a single slice plot may require over 50,000 interpolations. To ensure fast operation, expressions for calculated quantities in the configuration file are written in reverse Polish notation (RPN). **PhiView** parses the function strings once while loading the configuration file and saves them in a coded form. Thereafter, operations are performed at the speed of compiled code. The following section reviews the basics of RPN notation. Section 10.3 covers parameters and variables that may appear in expressions. Section 10.4 reviews the form of the data sections.

Table 7: General organization of the **PhiView** configuration file

```
HEADER
  Control parameters (Required)
END
STOREDQUANT
  Stored quantities in the solution file (Required)
END
REGIONQUANT
  Physical properties of regions (Optional)
END
PROGPARAM
  Program parameters (Optional)
END
RUNPARAM
  Parameters set by the solution program (Optional)
END
REALTIMEPARAM
  Parameters set by the user (Optional)
END
INTERPOLATION
  Calculated quantities for plots and data output (Required)
END
VECTOR
  Calculated quantities for vector display (Optional)
END
VOLUME
  Calculated quantities for volume integrals (Optional)
END
SURFACE
  Calculated quantities for surface integrals (Optional)
END
ENDFILE
```

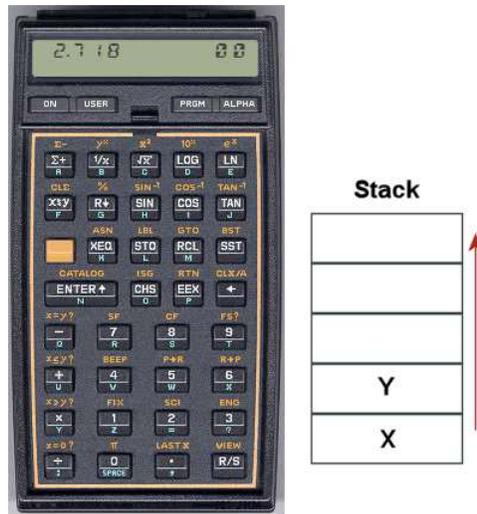


Figure 34:

## 10.2 Introduction to RPN

Consider the expression

$$5.0 \times 4.5^3 + [9.2 + 0.6 \times 0.9^{(2+0.67)}]. \quad (16)$$

Equation 16 could be represented in algebraic notation by the string:

5.0 \* 4.5^3 + (9.2 + 0.6 \* 0.9 ^ (2 + 0.67))

The rules for parsing such a string are involved, requiring an interpreter with recursive logic. The equation has the following form in RPN:

0.9 0.67 2 + ^ 0.6 \* 9.2 + 4.5 3 ^ 5.0 \* +

The string is parsed in strict order from left to right for expressions of any degree of complexity. A calculator with RPN logic uses the stack shown in Fig. 34. Numbers are pushed and popped at the bottom. The first two stack registers have the special names *X* and *Y*. Three simple rules govern the evaluation of RPN expressions:

If the entry is a number, push it on the stack.

If the entry is a unary operator (exp, ln, sin,...), apply it to the number in the *X* register.

If the entry is a binary operator (+, \*, ^, ...) combine the numbers in the *X* and *Y* registers and move the stack down.

The final result is given by the number remaining in the *X* register. By convention, the binary operators act in the following way:

Addition(+):  $Y + X$

Subtraction (-):  $Y - X$

Multiplication (\*):  $Y \times X$

Division (/):  $Y/X$

Exponentiation (^):  $Y^X$

Reverse Polish notation eliminates all issues with regard to parsing order. Furthermore, it is simple to compile expressions. Entries are represented either by variables or operator codes.

### 10.3 Variables, operations and parameters

Expressions to define quantities in **PhiView** may contain numbers, unary and binary operations, standard variables, standard calculated quantities and parameters. The standard variables are

`$X`, `$Y`, `$Z`, `$NREG`, `$DUNIT`

Note that the name of variables and parameters must start with a dollar sign. Calls to defined functions for plots, line scans and other operations are always made at a specific location in the solution space. **PhiView** sets the current position [`$X`,`$Y`,`$Z`] before any calls are made. To illustrate, the following expression gives the distance from the origin in meters:

```
RDIST = $X 2 ^ $Y 2 ^ $Z ^ 2 + + @SQRT
```

The standard variable `$DUNIT` is a dimension conversion factor passed from the solution program. For example, suppose dimensions in **MetaMesh** were defined in centimeters and converted to meters for use in the solution program. In this case, an entry in the header of the solution file would set `$DUNIT = 100.0`. The following expression returns the distance from the origin in centimeters:

```
RDIST = $X 2 ^ $Y 2 ^ $Z ^ 2 + + @SQRT $DUNIT *
```

Unary operators have names that begin with '@'. Table 8 lists the available set. The following considerations apply with respect to the `@CHS` operator:

You may include negative numbers (such as `-3.1456` and `-8.9E-09`) in expressions. The number `-22.56` is equivalent to `22.56 @CHS`.

Expressions like `-$Epsi0` and `-&grady[3]` that mix the binary operator with a variable are invalid. Instead, use forms like `$Epsi0 @CHS`.

The following quantities represent values from the electrostatic solution calculated at the current position:

```
&Phi  
&Ex  
&Ey  
&Ez  
&EMag  
&Q[N]
```

Table 8: Unary operators

Name	Operation	Comments
@SIN	$\sin(X)$	Angle in radians
@COS	$\cos(X)$	Angle in radians
@TAN	$\tan(X)$	Angle in radians
@ASIN	$\sin^{-1}(X)$	Returns angle in radians
@ACOS	$\cos^{-1}(X)$	Returns angle in radians
@ATAN	$\tan^{-1}(X)$	Returns angle in radians
@EXP	$e^X$	
@LN	$\ln X$	
@LOG	$\log_{10}(X)$	
@SQRT	$\sqrt{X}$	
@ABS	$ X $	
@EXCH	$X \rightleftharpoons Y$	
@OVERX	$1/X$	
@XSQ	$X^2$	
@ENTER	$X \Rightarrow X, Y$	Copy $X$ and push on stack
@CHS	$X = -X$	

The names must begin with an ampersand. They are always followed by an integer in box brackets giving the number of the stored quantity. The quantity  $\&Q[n]$  is the interpolated value of stored quantity  $n$  at the current position  $[X, Y, Z]$ . For a dielectric solution,  $\&Q[1]$  is the same as  $\&phi$ ,  $\&Q[2]$  equals the relative dielectric constant  $\epsilon_r$  and  $\&Q[3]$  is the space-charge density  $\rho$  in coulombs/m<sup>3</sup>. For a conductive solution,  $\&Q[2]$  is the conductivity  $\sigma$  in S/m and  $\&Q[3]$  is undefined. The program performs the interpolation and replaces the symbol with its numerical value before evaluating the expression. The interpolation method depends on whether the stored quantity is of type *Node* or *Element*. **PhiView** performs detailed interpolations of *Node* quantities. In the **LIN** mode, the program interpolates within the target element using shape functions and values of the stored quantity at the eight nodes. In the **LSQ** mode, **PhiView** collects data from surrounding elements of the same region type and makes a least-squares fit to a three-dimensional, second-order function.

Expressions may also include parameters defined in the **PROGRARAM** section. The section in the file `phivie3w_dielectric.cfg` has this form:

```
PROGRARAM
$EPSIO = 8.85E-12
END
```

The following expression for the electrostatic field energy in the interpolation section uses the parameter:

```
EngDens = &EMag 2 ^ $Epsi0 * &Q[2] * 0.5 *
```

You can define up to nine additional parameters in the **PROGRARAM** section. The name must begin with a dollar sign and may include up to 12 characters. A value in any valid real-number format follows the equal sign.

## 10.4 Setting up the calculated-quantity sections

The final sections of the configuration file define quantities for plots and calculations. There are four sections:

```
INTERPOLATION
  QuantityName = Definition
END
VECTOR
  QuantityName = Definition1;Definition2;Definition3
END
VOLUME
  QuantityName = Definition
END
SURFACE
  QuantityName = Definition1;Definition2;Definition3
END
```

The INTERPOLATION section may contain from zero to 50 quantity definitions. The VECTOR, VOLUME and SURFACE sections may contain a maximum of 20 quantity definitions each.

A definition line for a scalar quantity in the *Interpolation* section consists of a variable name, an equal sign and an RPN expression using the components discussed in the previous section. The following rules apply to quantity names in all sections:

A name has a maximum length of 12 characters (the length is limited so that quantity names will fit in plots and formatted output lists).

Names may include characters, numbers and the underscore symbol. Names may not include spaces and other punctuation marks.

The RPN expression is a string (maximum length of 300 characters) with entries separated by spaces. Here are some examples of quantity definitions:

```
EPERP = &EY @XSQ &EY @XSQ + @SQRT
LOGEXVCM = &EX 100.0 / @LOG
DX = &EX $Epsi0 * &Q[2] *
FIELDANGLE = &EY &EX / @ATAN $RADTODEG *
```

The last expression requires that the parameter

```
$RADTODEG = 57.2958
```

was defined in the PROGPARAM section. Note that the result of the division operator in the expression is 0.0 if  $\&EX = 0.0$ .

The order in which quantities appear in the INTERPOLATION section affects some operations in **PhiView**.

The first listed quantity is the default choice for *Slice* plots. You can change the choice by entering a value in the *SliceDisplay* command in the HEADER. The integer is the order of the default plot quantity as listed in the INTERPOLATION section.

Table 9: Naming conventions in RPN expressions

Name form	Functions
\$NAME	Program parameters
\$NAME	Standard variables ( <i>e.g.</i> , \$X)
&NAME	Special quantities calculated from the solution ( <i>e.g.</i> , &Ex)
@NAME	Unary operators in RPN expressions

In dialogs for picking a plot quantity, the options are listed in the same order that they appear in the configuration file. Therefore, you should put important quantities near the beginning and specialized quantities near the end.

The same order is followed in listings created by the *Matrix file* command. If you are exporting tables for use in **OmniTrak**, be sure that the first three quantities are  $E_x$ ,  $E_y$  and  $E_z$ .

Each quantity in the *Vector* section requires three expressions for the  $x$ ,  $y$  and  $z$  components. The format for a definition line is

```
Name = (Expression x);(Expression y);(Expression z)
```

The line may contain a maximum of 500 characters. The component expressions are separated by semicolons. The following example defines the current-density vector in a conductive solution:

```
JVector = &Ex &Q[2] *;&Ey &Q[2] *;&Ez &Q[2] *
```

The volume integral of a quantity  $Q$  over a region  $N_R$  is given by

$$\int dx \int dy \int dz Q(x, y, z), \quad (17)$$

where only those elements with region number equal to  $N_R$  are included. You can define up to 20 quantities in the *Volume* section for integrals. The expressions should define quantities with dimensions of the form (*quant/m<sup>3</sup>*). For example, the power dissipated in a conductive solution in units of W/cm<sup>3</sup> is

```
Power = &EMag 2 ^ &Q[2] * $DUnit 3 ^ /
```

Volume integrals use internal program coordinates in units of meters. Note that the quantity \$DUNIT is used in **PhiView** only for display labels.

The surface integral of a vector quantity  $\mathbf{S}$  over a region or a set of regions is given by

$$\int \int dA \mathbf{S} \cdot \mathbf{n}, \quad (18)$$

where  $dA$  is an element of area on the surface of the region set and  $\mathbf{n}$  is a unit vector normal to the surface pointing out of the region set. You can define up to 20 quantities in the **SURFACE**

section for integrals. The data line for each quantity should contain the name, an equal sign, and three RPN expressions separated by semicolons for the vector components. The expressions should define quantities with dimensions of the form (*quant/m<sup>2</sup>*). As an example, a surface integral of the following quantity would give the induced charge on the surface of a region representing an electrode in an electrostatic solution:

```
Charge = &Ex $Epsi0 * &Q[2] *;&Ey $Epsi0 * &Q[2] *;&Ez $Epsi0 * &Q[2] *
```

Note that quantity names must begin with special symbols if they are used in RPN expressions. Symbols for different quantity types are listed in Table 9. Therefore, quantity names in the `PROGPARAM` section must include the dollar sign. Quantity names that are defined only for display purposes need not include a special symbol. The includes quantities defined in the `INTERPOLATION`, `VECTOR`, `VOLUME` and `SURFACE` sections.

## 11 Force calculations in PhiView

### 11.1 Surface integrals for force and torque

If we knew the distribution of space-charge density  $\rho(x, y, z)$  and electric field  $\mathbf{E}(x, y, z)$  over an object, we could find the force on the object from the expression:

$$\mathbf{F} = \int \int \int dV \rho \mathbf{E}. \quad (19)$$

Volume integrals are not practical for calculating forces on electrodes and dielectric bodies. Here, the charge distribution is concentrated in an infinitely thin surface layer (compared to the element size) and the electric field changes discontinuously at the surface. For this situation, the best approach is to perform an integral of the Maxwell stress tensor over a surface that encloses the body. This process yields good numerical accuracy and can be performed automatically using the surface-integral capabilities of **PhiView**.

The reference S. Humphries, **Field Solutions on Computers** (CRC Press, Boca Raton, 1998), Sect. 10.5 discusses the physical basis of electromagnetic force calculations. The integral of force density over a volume can be converted to the following integral over the bounding surface to give the total force  $\mathbf{F}$ ;

$$\mathbf{F} = \int \int \mathbf{S} \cdot \mathbf{n} dA. \quad (20)$$

where the quantity  $\mathbf{S}$  is the tensor:

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix}. \quad (21)$$

The integral of Eq. 20 can be implemented in **PhiView** by adding the following vector definitions to the **SURFACE** section of the configuration file:

$$\begin{aligned} F_{xs} &= [S_{11}, S_{12}, S_{13}], \\ F_{ys} &= [S_{21}, S_{22}, S_{23}], \\ F_{zs} &= [S_{31}, S_{32}, S_{33}]. \end{aligned} \quad (22)$$

Expressions for torque are more complex. A differential element of torque is defined by

$$d\mathbf{t} = \mathbf{r} \times d\mathbf{F}. \quad (23)$$

Here, the vector  $\mathbf{r}$  points from a torque origin  $[x_t, y_t, z_t]$  to the current position:

$$\mathbf{r} = [r_x, r_y, r_z] = [(x - x_t), (y - y_t), (z - z_t)]. \quad (24)$$

The total torque corresponding to the force on a body may be written in terms of the Maxwell stress tensor as:

$$\mathbf{t} = \iint \mathbf{r} \times (\mathbf{S} \cdot \mathbf{n}) dA. \quad (25)$$

To add torque calculations in the **PhiView** configuration file, we need to determine a torque tensor  $\mathbf{T}$  such that the torque vector is given by a surface integral,

$$\mathbf{t} = \iint \mathbf{T} \cdot \mathbf{n} dA. \quad (26)$$

We can compute  $\mathbf{T}$  by expanding the right-hand side of Eq. 25 in component form and collecting terms with factors of  $n_x$ ,  $n_y$  and  $n_z$ . The procedure gives the following results:

$$\mathbf{T} = \begin{bmatrix} r_y S_{31} - r_z S_{21} & r_y S_{32} - r_z S_{22} & r_y S_{33} - r_z S_{23} \\ r_z S_{11} - r_x S_{31} & r_z S_{12} - r_x S_{32} & r_z S_{13} - r_x S_{33} \\ r_x S_{21} - r_y S_{11} & r_x S_{22} - r_y S_{12} & r_x S_{23} - r_y S_{13} \end{bmatrix}. \quad (27)$$

## 11.2 Expressions for electrostatic forces

The configuration file `PHIVIEW_FORCE.CFG` includes definitions for force and torque calculations using both volume and surface integral methods. Volume integrals are useful for regions with non-zero space charge  $\rho$ . The routines return zero values for dielectric bodies and electrodes. For reference, the volume-integral expressions for torque are:

$$\begin{aligned} \tau_{xv} &= \iiint dV \rho (r_y E_z - r_z E_y), \\ \tau_{yv} &= \iiint dV \rho (r_z E_x - r_x E_z), \\ \tau_{zv} &= \iiint dV \rho (r_x E_y - r_y E_x). \end{aligned} \quad (28)$$

The Maxwell stress tensor has the following form for electrostatic forces:

$$\mathbf{S} = \epsilon_0 \begin{bmatrix} \epsilon_0(E_x^2 - E^2/2) & \epsilon_0(E_x E_y) & \epsilon_0(E_x E_z) \\ \epsilon_0(E_y E_x) & \epsilon_0(E_y^2 - E^2/2) & \epsilon_0(E_y E_z) \\ \epsilon_0(E_z E_x) & \epsilon_0(E_z E_y) & \epsilon_0(E_z^2 - E^2/2) \end{bmatrix}. \quad (29)$$

The quantity  $E^2$  in Eq. 29 is the field magnitude:

$$E^2 = E_x^2 + E_y^2 + E_z^2. \quad (30)$$

Note that all field values are evaluated on the air/vacuum side of the object surface. Surface integrals of force are calculated with the vector expressions of Eq. 22 while torque components are determined using the expressions of Eq. 27.

Figure 35 illustrates the logic of surface integrals in **PhiView**. If we have an assembly of dielectrics and electrodes surrounded by an air volume, then we can take the integral of Eq. 20 over any enclosing surface. One choice is the outer boundary of the assembly components (designated surface  $A$  in Fig. 35a). In this case, we evaluate electric field values in the air elements near the surface to ensure that the integral encloses all material charge  $\rho_m$ . We could also define an arbitrary surface by enclosing the assembly inside a special diagnostic air region ( $\epsilon = \epsilon_0$ ). The integral over the surface marked  $B$  in Fig. 35a gives the same result (to within

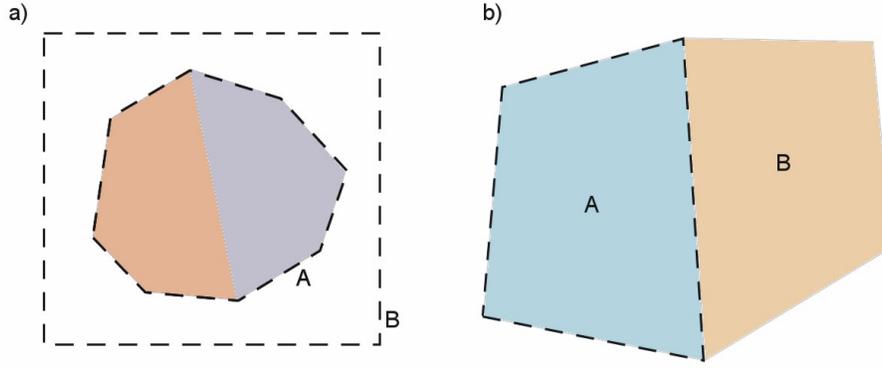


Figure 35: Surface integral of the electric stress tensor. *a)* Valid surfaces for an integral of electrically-active materials. A: region surfaces bounded by air elements. B: surface of an air region that encloses the electric materials. *b)* Invalid surface for an integral. The enclosed material current is undefined on the common boundary.

the numerical accuracy of field interpolations). The *B* surface (where the fields vary smoothly) is the better choice when the object has sharp edges. Figure 35*b* shows a case where the stress tensor integral does not give the correct result. In this case the integral extends over the surface of a dielectric/electrode region *A* that is in contact with another dielectric/electrode region (*B*). The field values inside region *B* along the common boundary include the effects of the surface charge of both regions. The calculation gives the force on region *A* plus an indeterminate portion of the force on region *B*. In summary, the following rules apply to electric force calculations:

- Electric forces act on electrodes and dielectric objects with  $\epsilon \neq \epsilon_0$ .
- The surface integral of Eq. 20 over a region boundary gives the correct forces only if the region is surrounded by air/vacuum elements ( $\epsilon = \epsilon_0$ ).
- Any integration surface in air/vacuum that encloses the region (or assembly of regions) yields the same net force.

### 11.3 Commands for force calculations

For a force calculation, load the configuration file `PHIVIEW_FORCE.CFG` and click on the command *Surface integral* of the *Analysis* menu. In the dialog, choose the region or set of regions of interest. Note that if one region is inside another, you must pick both regions so that the integral is performed only over the outer surface. **PhiView** employs the following procedure to find the force on a single region. The program locates all elements within the target region number and checks the six bounding facets. If the element on the opposite side of facet has a different region number, then the facet lies on the region surface. (Facets on the boundary of the solution volume or adjacent to an element with region number 0 are excluded.) For each facet, **PhiView** performs a  $3 \times 3$  Gaussian-quadrature integral using normal coordinates. At each point, the program calculates electric field values at a nearby location  $(x, y, z)$  in the adjacent element and determines the components of  $\mathbf{S}$ . The program sets an error flag if any adjacent element has  $\epsilon \neq \epsilon_0$ . **PhiView** constructs a unit vector normal to the facet that points out of the target region. The program then forms the dot product for Eq. 20. Each segment

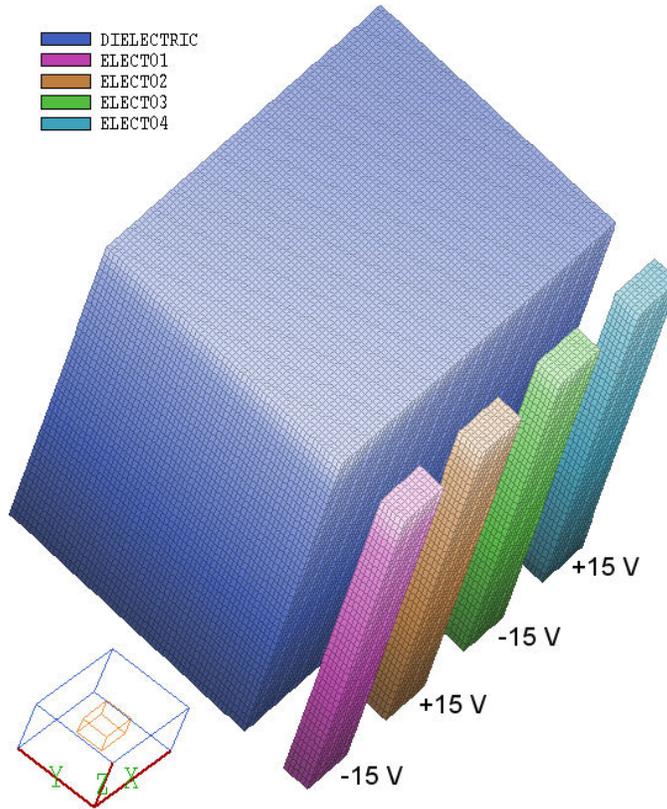


Figure 36: Geometry of example FORCETEST. The diagnostic region is not shown.

of the integral gives an increment of force  $d\mathbf{F}$ . The sum of  $d\mathbf{F}$  over the segments of each facet and over all facets of the region surface gives the total force. **PhiView** can also find the net force on assemblies of adjacent electrically-active regions. In this case, the program identifies all elements of the region set. Surface integrals are performed only over facets that border on elements with region numbers that are not part of the set. If the assembly is surrounded by air elements, the procedure gives the total force. To calculate force from an analysis script, be sure that the proper configuration file is loaded and use the **SurfaceInt** command described in Sect. 7.3.

## 11.4 Example

Figure 35 shows an example to illustrate an electrostatic force calculation. We want to find the force between the four electrodes and the dielectric block ( $\epsilon_r = 3.1$ ). The dielectric has width in  $x$  of  $70.0 \mu\text{m}$  and height in  $y$  of  $60.0 \mu\text{m}$ . The electrodes have width  $10.0 \mu\text{m}$  and height  $6.0 \mu\text{m}$ . The gap in  $y$  between the electrodes and dielectric is  $10.0 \mu\text{m}$ . All objects have length  $100.0 \mu\text{m}$  along  $z$ . A special diagnostic region (not shown in Fig. 35) has been included. It surrounds the electrodes and has the properties of vacuum ( $\epsilon_r = 1.0$ ). The function of the region is to provide a surface removed from the electrodes to perform the integral of the stress tensor (see path  $B$  in Fig. 35a).

The input files for the example are **FORCETEST.MIN**, **FORCETEST.HIN** and **FORCETEST.SCR**. Figure 37 shows the calculated equipotentials in a plane normal to  $z$  at the midplane of the

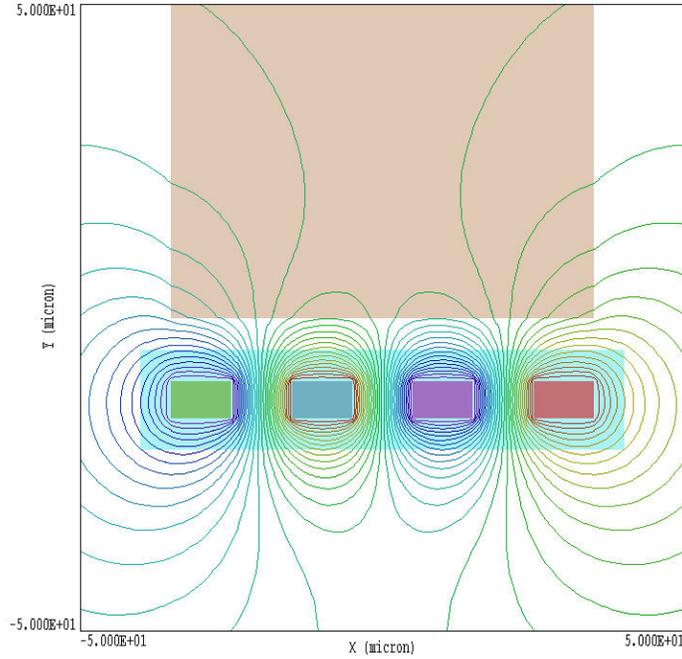


Figure 37: Calculated equipotential lines for the FORCETEST example in a plane normal to  $z$  at  $z = 0.0 \mu\text{m}$ . Dielectric: tan. Diagnostic region: cyan.

assembly ( $z = 0.0 \mu\text{m}$ ). The analysis script is listed in Table 10. The command SURFACEINT 3 signals a stress tensor integral over the surface of the dielectric body. In response to the command SURFACEINT 2 4 5 6 7, the stress tensor integral is taken over the outer boundary of the diagnostic region (Reg 2), excluding the internal boundaries with the electrodes (Reg 4, 5, 6 and 7).

The results are recorded in the file FORCETEST.DAT. The surface integral over the dielectric region yields  $F_y = -2.199 \times 10^{-8}$  and the diagnostic region is  $F_y = 2.308 \times 10^{-8}$  newton. As expected, forces in the  $x$  and  $z$  directions are effectively zero while the forces along  $y$  on the dielectric and electrode have approximately equal magnitude and opposite directions. The second calculation is more accurate because field values are continuous through the surface of the diagnostic region.

Table 10: File FORCETEST.SCR

```
* File FORCETEST.SCR
* NReg      RegName
* =====
*   1      VACUUM
*   2      DIAGNOSTIC
*   3      DIELECTRIC
*   4      ELECT01
*   5      ELECT02
*   6      ELECT03
*   7      ELECT04
*   8      BOUNDARY

CONFIGURATION = C:\FIELDP\AMAZE\PhiView_Force.CFG
INPUT = FORCETEST.HOU
OUTPUT = FORCETEST.DAT
* Dielectric
SURFACEINT = 3
* Electrodes and diagnostic region
SURFACEINT = (2 4 5 6 7)

ENDFILE
```

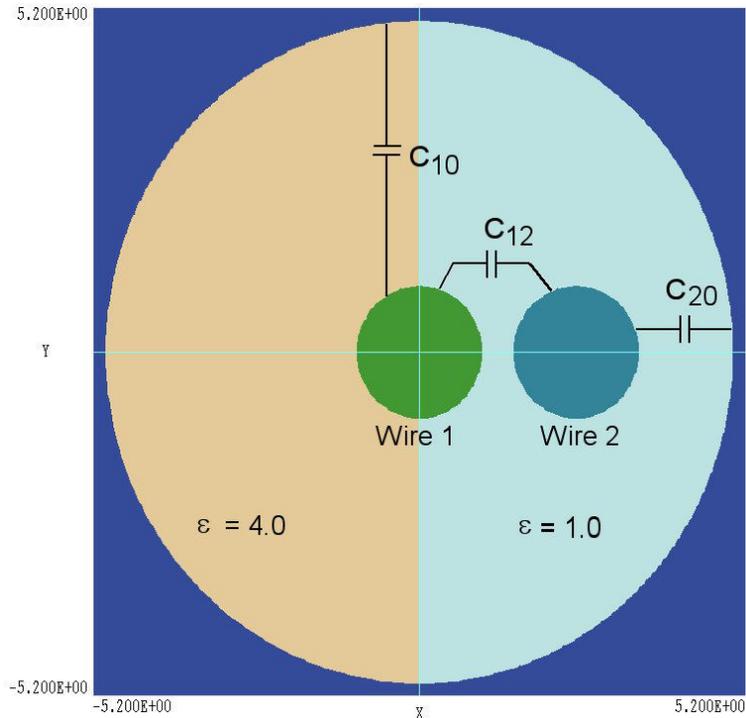


Figure 38: Benchmark example – geometry and definition of quantities (system is uniform in  $z$  with a length  $L_z = 0.02$  m).

## 12 Calculating mutual capacitance

Finite-element programs for electrostatics are useful for calculating mutual capacitance in a multi-electrode system. **PhiView** has several automatic calculation routines to facilitate the process. Because the user must exercise some care to ensure accuracy and physical validity, we shall review available methods in this chapter and work through an example. Figure 38 shows a benchmark calculation that gives good accuracy and runs quickly (input files `MUTUALCAP.MIN` and `MUTUALCAP.HIN`). The geometry consists of two circular wires inside a grounded outer conductor with a circular inner radius. The enclosed volume is divided into two dielectric regions with  $\epsilon_r = 4.0$  and  $\epsilon_r = 1.0$ . In this case, there are three unknown capacitance quantities:  $C_{10}$  (capacitance between Wire 1 and ground),  $C_{20}$  (capacitance between Wire 2 and ground) and  $C_{12}$  (mutual capacitance between Wires 1 and 2). The assembly is uniform along the  $z$  direction with length  $L_z = 0.02$  m.

One approach is to use the automatic surface integral routines in **PhiView** to find induced charge. To perform an integral, make sure the configuration file `PHIVIEW_DIELECTRIC.CFG` is loaded and open a data file in the main menu. Then click on the command `Surface integral` in the *Analysis* menu. Specify the region set over which integrals will be performed. **PhiView** lists results on the screen and writes a record in the data file.

To illustrate, consider the capacitance quantities in the example of Fig. 38. We create Solution 1 with Wire 1 set to  $V_1 = 1.0$  V and Wire 2 to  $V_2 = 0.0$  V (Fig. 39). In **PhiView**

Table 11: Calculated quantities for example MUTUALCAP.

Sol	V1	V2	Q1	Q2	Q0	U
1	1.0	0.0	$2.13 \times 10^{-12}$	$-6.09 \times 10^{-13}$	$-1.53 \times 10^{-12}$	$1.05 \times 10^{-12}$
2	0.0	1.0	$-6.09 \times 10^{-13}$	$1.27 \times 10^{-12}$	$-6.67 \times 10^{-13}$	$6.27 \times 10^{-13}$
3	1.0	1.0	$1.52 \times 10^{-12}$	$6.64 \times 10^{-13}$	$-2.20 \times 10^{-12}$	$1.08 \times 10^{-12}$

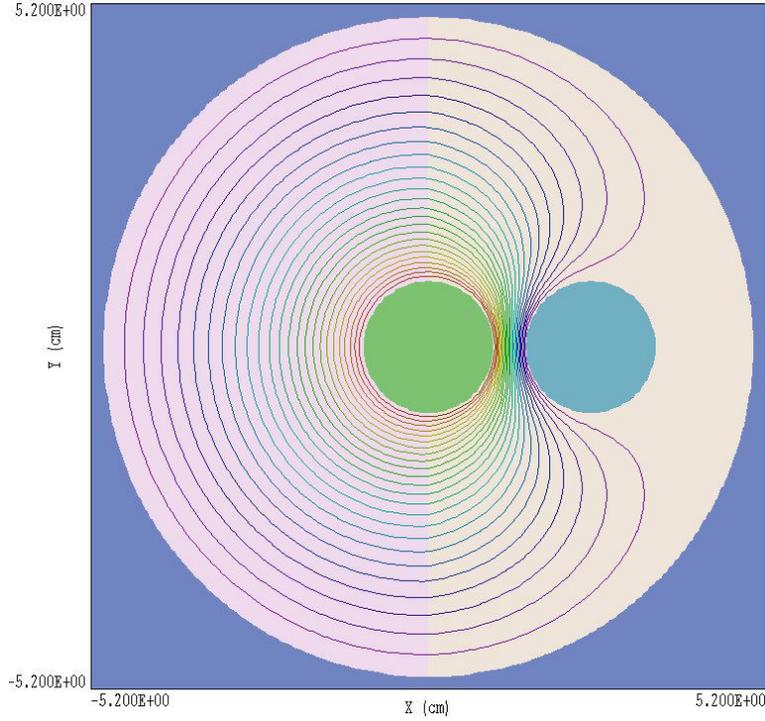


Figure 39: Calculated equipotential lines for Solution 1.

we use the *Surface integral* command for Regions 1 (ground electrode), 4 (Wire 1) and 5 (Wire 2). Table 11 shows the calculated surface charge values in coulombs. In the table,  $Q_1$  is the surface charge on Wire 1,  $Q_2$  is the charge on Wire 2, and  $Q_0$  is the charge on the inside of the ground electrode. Because we applied a 1.0 V drive, the numbers also equal the capacitance in farads. From the table, we can determine that  $C_{12} = 6.09 \times 10^{-13}$  F and  $C_{10} = 1.05 \times 10^{-12}$  F. To check accuracy, we can verify that  $Q_1 + Q_2 + Q_0 \cong 0.0$ . We next generate Solution 02 with  $V_1 = 0.0$  V and  $V_2 = 1.0$  V. The results (shown in the second data row of Table 11) imply that  $C_{12} = 6.09 \times 10^{-13}$  F and  $C_{20} = 6.65 \times 10^{-13}$  F.

The surface integral method works well in the example because we have used a fine mesh for good resolution of the curved electrode surfaces. The accuracy of the method declines significantly when electrodes have sharp corners where the electric field is effectively undefined. For these situations we can get better results with volume integral methods that are less sensitive to local field interpolation errors.

For a simple two-electrode system, the volume integral of field energy is related to the capacitance by

Table 12: Applied voltages for a three-electrode system with ground.

Solution	V1	V2	V3
1	1.0	0.0	0.0
2	0.0	1.0	0.0
3	0.0	0.0	1.0
4	1.0	1.0	0.0
5	1.0	0.0	1.0
6	0.0	1.0	1.0

$$U_e = \frac{CV^2}{2}. \quad (31)$$

We can use the *Volume integral* command in the *Analysis* menu of to perform a field energy integral. In response, **PhiView** lists the global energy and components associated with different regions. It is not difficult to extend this concept to multiple electrodes. For an applied voltage of 1.0 V, the global electrostatic energy  $U_1$  for Solution 1 is related to the capacitance values by

$$U_1 = \frac{1}{2} (C_{12} + C_{10}). \quad (32)$$

Similarly, the field energy for Solution 02 is

$$U_2 = \frac{1}{2} (C_{12} + C_{20}). \quad (33)$$

We perform a third solution where a 1.0 V potential is applied to both Wires 1 and 2. Here, the field energy is related to capacitance values by

$$U_3 = \frac{1}{2} (C_{10} + C_{30}). \quad (34)$$

Using the three equations, we can solve for the capacitance values in terms of the calculated values of field energy:

$$C_{12} = U_1 + U_2 - U_3, \quad (35)$$

$$C_{10} = U_1 - U_2 + U_3, \quad (36)$$

$$C_{20} = -U_1 + U_2 + U_3. \quad (37)$$

Table 11 shows field energy values for the three solutions. The calculated values of capacitance are  $C_{12} = 5.98 \times 10^{-13}$  F,  $C_{10} = 1.51 \times 10^{-12}$  F and  $C_{20} = 6.65 \times 10^{-13}$  F. These values are close to those determined by the surface integral method.

There are six unknown quantities for a system with three electrodes and a ground reference:  $C_{12}$ ,  $C_{23}$ ,  $C_{31}$ ,  $C_{10}$ ,  $C_{20}$  and  $C_{30}$ . We can generate a set of six equations similar to Eqs. 37 with the set of voltages shown in Table 12.

---

## 13 HiPhi file formats

**HiPhi** creates output files in binary or text format. The mode is controlled by the *Format* command (Sect. 3.1).

- The default binary mode should be used for most applications. It gives the minimum file size and preserves the full accuracy of the double precision numbers used in **HiPhi**.
- Use the text mode to port information to your own programs or if you want to inspect results with an editor.

As in **MetaMesh**, nodes are referenced with the indices  $[I, J, K]$  where  $I$  (the index along the  $x$  axis) extends from 0 to  $I_{max}$ ,  $J$  ( $y$  axis) from 0 to  $J_{max}$ , and  $K$  ( $z$  axis) from 0 to  $K_{max}$ . The number of elements is approximately equal to the number of nodes. A single element (in the direction of positive  $x$ ,  $y$  and  $z$ ) is associated with each node for storage. Nodes and elements have integer region numbers to associate them with structures in the solution space.

The format of the **HiPhi** binary output file is simple and compact. The FORTRAN code extract shown in Table 13 comprises the entire output algorithm. The header contains the following elements:

- The 6-character string *BINARY*.
- The number of regions in the solution space, *NReg* (4-byte integer).
- A total of *NReg* 20-character strings giving the names of regions obtained from the **MetaMesh** input file.
- A total of *NReg* single-character strings, either *I* or *E*. The character *I* designates a standard material region that should be included in a PhiView analysis and *E* indicates a fixed-potential that should be excluded.
- The mesh size parameters  $I_{max}$ ,  $J_{max}$  and  $K_{max}$  (4-byte integers).
- The quantity *DUnit*, the unit conversion factor associated with the solution (double precision, 8-byte real).

The next step is to record information for each node. The quantities *RegNo* (the region number of the node) and *RegUp* (the region number of the adjacent element in the direction of increasing  $I$ ,  $J$  and  $K$ ) are 4-byte integers. The coordinates  $X, Y$  and  $Z$  (in meters) are 8-byte real numbers as are the following stored quantities:

- The electrostatic potential  $\phi$  at the node in units of volts.
- The relative dielectric constant  $\epsilon_r$  associated with the upper element.
- The space charge density  $\rho$  (in coulombs/m<sup>3</sup>) associated with the upper element.

Table 13: Code to create the **HiPhi** binary output file

```

! File type
  Character6 = 'BINARY'
  WRITE (OutField) Character6
! NReg
  WRITE (OutField) NRegMax
! Region names
  DO N=1,NRegMax
    WRITE (OutField) Reg(N).RName
  END DO
! Region status
  DO N=1,NRegMax
    IF (Reg(N).Fixed) THEN
      Character1 = 'E'
    ELSE
      Character1 = 'I'
    ENDIF
    WRITE (OutField) Character1
  END DO
! Mesh size
  WRITE (OutField) IMax,JMax,KMax
! DUnit
  WRITE (OutField) DUnit
! Node quantities
  DO K=0,KMax
    DO J=0,JMax
      DO I=0,IMax
        IF (.NOT.ElementEpsi(I+1,J+1,K+1,EpsiWrite)) &
          EpsiWrite = DZero
        IF (.NOT.ElementRho(I+1,J+1,K+1,RhoWrite)) &
          RhoWrite = DZero
        MC = M(I,J,K)
        RegNoWrite = C(MC).RegNo
        RegUpWrite = C(MC).RegUp
        WRITE (OutField) &
          RegNoWrite,RegUpWrite,C(MC).x,C(MC).y,C(MC).z, &
          C(MC).Phi,EpsiWrite,RhoWrite
      END DO
    END DO
  END DO
ENDIF

```

Table 14 shows the header and initial node data lines from an output file in text format (note that the node data lines have been broken for readability). Three region names are shown in the example. The names are strings up to 20 characters in length with no spaces. Each name is followed by the character *I* (included) or *E* (excluded). The order of nodes is the same as that of Table tab:outputcode. Real numbers are recorded in format (1P,E14.6).

Files created in response to the **PhiView** *Matrix file* are in text format and have a structure similar to that of Table 14. The files are useful to transfer information to your own analysis programs and they are compatible with **OmniTrak** and **FEVision**. There are two differences from the HOU file in text format. First, calculations are performed over a regular mesh at locations

$$X(I) = X_{min} + I\Delta x, \quad (38)$$

$$Y(J) = Y_{min} + J\Delta y, \quad (39)$$

$$Z(K) = Z_{min} + K\Delta z. \quad (40)$$

where

$$\Delta x = (X_{max} - X_{min})/I_{max}, \quad (41)$$

$$\Delta y = (Y_{max} - Y_{min})/J_{max}, \quad (42)$$

$$\Delta z = (Z_{max} - Z_{min})/K_{max}. \quad (43)$$

Second, the set of calculated quantities depends on the choice in the the *Recorded quantities* command. The set is  $[E_x, E_y, E_z]$  for the *Field components* option, where the electric field is in V/m. The quantities included under the *Full set* option depend on the **PhiView** configuration file. The following quantities are included when operating under `phiview_dielectric.cfg`:  $[E_x, E_y, E - z, |E|, \phi, U_e, \epsilon_r, \rho]$ . The units are volts for electrostatic potential  $\phi$ , J/m<sup>3</sup> for field energy density and coulombs/m<sup>3</sup> for space-charge density.

Table 14: Header and data extract showing the form of the **HiPhi** HOU file under the **Text** option and the **PhiView** matrix file

```

TEXT
NReg:    4
OUTERBOUND E
OUTERDIELECTRIC I
INNERDIELECTRIC I
INNERCONDUCTOR E
IMax:    102
JMax:    102
KMax:    102
DUnit:   1.000000E+02

  RegNo RegUp      X          Y          Z
=====
    1    1 -5.100000E-02 -5.100000E-02 -5.100000E-02
    1    1 -5.000000E-02 -5.100000E-02 -5.100000E-02
    1    1 -4.900000E-02 -5.100000E-02 -5.100000E-02
    ...

          Phi          EpsiR          Rho
=====
    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
    ...

```

# Index

- batch file, [51](#)
- binary file format, [87](#)
- calculated quantities
  - special, [74](#)
  - standard variables, [73](#)
- capacitance, [47](#), [84](#)
- charge conservation, [15](#)
- command line, [51](#)
- conductive solution
  - criterion, [14](#)
- conductivity, [23](#)
- conductors and ideal dielectrics, [14](#)
- configuration file, [70](#)
  - calculated quantities, [70](#)
  - order of quantities, [75](#)
  - sections, [70](#)
  - surface section, [77](#)
  - volume section, [76](#)
- current density, [15](#)
- cutaway view, [68](#)
- derivative of scan, [60](#)
- dielectric solution, [12](#)
  - criterion, [14](#)
- Dirichlet boundary condition, [14](#), [16](#), [24](#)
- electric force, [79](#)
- electric stress tensor, [79](#)
  - valid surface integrals, [80](#)
- Eprobe, [62](#)
- equipotential plot, [10](#)
- field energy, [47](#)
- floating electrode, [15](#)
- Geometer, [5](#), [17](#)
- grid control, [57](#)
- HiPhi
  - batch file control, [35](#)
  - features, [4](#)
  - file format, [21](#), [87](#)
  - file types, [17](#)
  - large meshes, [36](#)
  - learning, [4](#)
  - memory use, [36](#)
  - output file, [17](#)
  - reducing mesh size, [36](#)
  - running in background, [35](#)
  - running the program, [33](#)
  - script, [17](#), [18](#)
  - script dialog, [17](#)
  - setup dialog, [8](#)
- HiPhi commands
  - Boundary, [8](#), [25](#)
  - DUnit, [8](#)
  - Edit file, [33](#)
  - Edit script, [33](#)
  - Epsi, [22](#), [29](#)
  - Format, [21](#)
  - InitVal, [21](#)
  - Interp, [22](#)
  - Manual, [35](#)
  - MaxCycle, [8](#), [21](#)
  - Mesh, [20](#)
  - NCheck, [21](#)
  - Omega, [9](#), [21](#)
  - Parallel, [21](#)
  - Pause run, [34](#)
  - Physical constants, [35](#)
  - Potential, [22](#), [28](#)
  - ResTarget, [20](#)
  - Restarget, [8](#)
  - Rho, [23](#), [29](#)
  - Setup, [34](#)
  - Sigma, [23](#), [29](#)
  - Start run, [34](#)
  - Stop run, [35](#)
  - Superposition, [8](#), [27](#)
  - Void, [23](#)
- insulator, perfect, [23](#)
- integral of scan, [61](#)
- Laplace equation, [14](#)
- logarithmic intervals, [57](#)
- manual plot limits, [57](#)

- mathematical functions, [30](#)
- MetaMesh
  - clipping, [8](#)
  - index conventions, [87](#)
  - output file, [17](#), [20](#)
  - script, [5](#), [17](#)
- names of quantities, rules, [75](#), [77](#)
- Neumann boundary condition, [5](#), [16](#), [23](#)
- oscilloscope mode, [60](#)
- PhiView
  - analysis script, [10](#), [17](#), [38](#)
  - batch file control, [51](#)
  - configuration file, [70](#)
  - data file, [39](#), [46](#), [47](#)
  - force calculations, [79](#)
  - interpolation methods, [46](#)
  - line scans, [46](#)
  - making plot files, [43](#)
  - matrix file, [48](#)
  - numerical analysis, [10](#), [46](#), [58](#)
  - plane plot type, [41](#)
  - plot types, [10](#)
  - printing plots, [43](#)
  - script control, [50](#)
  - slice plot type, [54](#)
  - surface plot controls, [64](#)
  - surface plot type, [64](#)
- PhiView command
  - Save reflected solution, [37](#)
- PhiView commands
  - Close data file, [39](#)
  - Copy to clipboard, [43](#)
  - Create script, [39](#)
  - Default printer, [43](#)
  - Derivative, [60](#)
  - Displayed regions, [66](#)
  - Edit data file, [39](#)
  - Edit file, [39](#)
  - Edit script, [39](#)
  - Eprobe, [62](#)
  - Expand view, [55](#)
  - Field line plot file, [68](#)
  - Field lines at points, [62](#)
  - Global view, [55](#)
  - Grid intervals, [57](#), [59](#)
  - Instruction manual, [39](#)
  - Integral, [61](#)
  - Jump, [55](#)
  - Line scan, [46](#), [58](#)
  - Load named view, [44](#)
  - Load solution file, [37](#)
  - LSQ/LIN interpolation, [49](#)
  - Matrix file, [48](#)
  - Number of contour lines, [57](#)
  - Open data files, [39](#)
  - Orthogonal/perspective, [66](#)
  - Pan, [55](#)
  - Plot file, [43](#)
  - plot limit, [56](#)
  - Plot limits, [42](#)
  - Plot quantity, [42](#), [56](#)
  - Plot style, [42](#)
  - Point calculation, [46](#), [58](#)
  - Record field line plot, [69](#)
  - Recorded quantities, [49](#)
  - Remove vectors, [63](#)
  - Reset to default, [57](#)
  - Restore default view, [66](#)
  - Rotate plot, [43](#)
  - Run script, [38](#)
  - Save named view, [44](#)
  - Scan quantity, [49](#), [58](#)
  - Set cut plane, [68](#)
  - Set number of scan points, [49](#), [58](#)
  - Set plane, [41](#)
  - Set plot limits, [68](#)
  - Set slice plane properties, [54](#)
  - Set slice plot style, [55](#)
  - Set surface plot quantity, [68](#)
  - Set view dialog, [64](#)
  - Slice normal, [55](#)
  - Snap distance, [57](#)
  - Solution file information, [37](#)
  - Step, [55](#)
  - Surface plot control, [66](#)
  - Symbol display, [59](#)
  - Toggle grid, [57](#), [59](#)
  - Toggle snap mode, [57](#)
  - Values to clipboard, [60](#)
  - Vector scatter plot, [63](#)

- Vertical limits, [59](#)
- Zoom in, [55](#)
- Zoom window, [55](#)
- PhiView file command
  - Create script, [50](#)
  - Edit script, [50](#)
  - Run script, [50](#)
- PhiView script command
  - Configuration, [51](#)
  - GenScan, [52](#)
  - Input, [51](#)
  - Interpolation, [53](#)
  - Matrix, [53](#)
  - NScan, [53](#)
  - Output, [51](#)
  - Plot, [53](#)
  - Point, [52](#)
  - Record, [53](#)
  - Scan, [52](#)
  - SurfaceInt, [52](#)
  - VolumeInt, [52](#)
- Plot views, saving, [44](#)
- Poisson equation, [12](#), [31](#)
- quantities
  - order, [75](#)
  - vector, [76](#)
- region, [5](#)
  - material properties, [22](#)
  - spatial variations, [28](#)
- relative dielectric constant, [12](#), [22](#)
- resistance, [48](#)
- resistive relaxation time, [14](#)
- reverse Polish notation, [72](#)
- rpn
  - binary operators, [72](#)
  - parsing rules, [72](#)
- script control, [50](#)
- set contour line values, [57](#)
- small-scale solutions, [24](#)
- space-charge density, [12](#), [23](#)
- spatial variations, [28](#)
  - functions, [30](#)
  - tables, [30](#)
- special material types, [15](#)
- surface integral, [76](#)
- symmetry boundary, [37](#)
- text file format, [87](#)
- vector tools, [62](#)
- volume integral, [76](#)