EStat 8.0

Finite-element Electrostatics

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

Contents

1 Introduction

1.1 Program functions

EStat is a versatile numerical tool to find electrostatic fields in complex two-dimensional geometries. Simulated systems may include electrodes, conductors, dielectrics, and space-charge. The unitized, self-contained package addresses all aspects of the problem: mesh generation, finite-element solution, analysis and plotting. **EStat** employs finite-element methods on variable-resolution conformal triangular meshes for high accuracy and speed. The mesh size is limited only by the installed memory on your computer. The program handles three-dimensional cylindrical problems (symmetry in θ) and two-dimensional rectangular problems (arbitrary variations in x and y with infinite extent in z). Analysis functions include a wide variety of ϕ and **E** plots as well as automatic calculation of Gaussian surface integrals, electrostatic energy and induced charge.

The intuitive graphical-user-interface makes it easy to learn **EStat** and to perform quick application setups. In contrast to other field solution software, **EStat** features advanced capabilities (such as support for anisotropic materials) and complete data transparency. Input operations are automatically recorded in text scripts that provide documentation of your work. Scripts make it easy to reconstruct solutions and to share setups with colleagues.

EStat performs electrostatic solutions in two limits:

- **Dielectric solutions**. Here materials are insulators ($\sigma = 0.0$) with different values of relative dielectric constant ϵ_r and (optionally) space-charge density ρ.
- **Conductive solutions**. Materials are conductors with $\sigma > 0.0$. In the static limit the dielectric constant does not affect the electric field distribution and $\rho = 0.0$.

A *dielectric* solution applies to systems with perfect insulators or when short voltage pulses are applied to good insulators. An electroplating apparatus is an example of a *conductive* solution. In some cases, it is possible to model a mixed system of ideal insulators and conductors (see Sect. 1.5).

1.2 Learning EStat

The size of this manual reflects the broad capabilities of the **Mesh**/**EStat** package. It is not necessary to read the entire document to address most applications. We have organized the chapters to help you get started quickly. You can investigate advanced topics as you gain more experience.

- Browse the following sections in this chapter to review some basics of electostatics. You will probably return to the material to resolve issues (such as boundary conditions) as you get more involved in the programs.
- Scan the section on **TC** (the **TriComp** program launcher) in the **Mesh** manual.

- Be sure to read Chapter 2 of the **Mesh** manual, which reviews essential concepts for conformal meshes in finite-element solutions.

- Chapter 3 of the **Mesh** manual follows a walkthrough example to introduce construction of a solution geometry with the interactive drawing editor. The exercise introduces fundamental tools you will need for you own simulations. Chapter 5 of the **Mesh** manual describes how to convert the script into a conformal mesh.

- Be sure to read Chapter 2. The walkthrough example covers the steps in creating and analyzing a finite-element solution with **EStat**.

- Run some of the prepared examples supplied with the package. The examples give insights into solution options and may serve as templates for your own work.

- As you gain experience you will want to to take advantage of the full range of **Mesh**/**EStat** capabilities. Chapter 4 of the **Mesh** manual is a comprehensive reference on the drawing editor. Chapter 5 covers processing, plotting and repairs of meshes. Chapter 3 in this manual is a complete reference on **EStat** control scripts, including advanced commands for non-uniform and anisotropic materials. Chapter 5 gives information on the analysis functions common to all the **TriComp** programs. Capabilities include a variety of plots, interpolations of field values and integrals of field energy and other quantities. Chapter 6 reviews the calculated quantities defined by the two standard **EStat** configuration files. Finally, Chap. 7 describes how to build a customized postprocessor that includes your own calculated quantities.

- Additional chapters in the **Mesh** manual cover advanced techniques. Chapters 6 and 7 show how to make direct entries in scripts to invoke advanced control features. Chapter 8 of the **Mesh** manual describes how to create meshes directly from photographic and data images. This feature is useful to model complex or irregular systems that are difficult to describe with simple geometric specifications.

1.3 Finite-element solution procedure

An in-depth understanding of finite-element numerical methods is not necessary to use **EStat**. Nonetheless, it is important to have a clear idea of fundamental concepts to create effective solutions. This section describes background material to understand the steps in an **EStat** solution. The textbook **Finite-element Methods for Electromagnetics** (included with the program distribution) gives a detailed presentation of the underlying physics and numerics of **EStat**.

The term *field* indicates a quantity (scalar or vector) defined over a region of space. Examples of fields include the vector electric field $\mathbf{E}$ in an electrostatic solution, electric and magnetic fields in an electromagnetic solution and the scalar temperature T in a thermal solution. Variations of field quantities are usually described by continuous partial differential equations, such as the Poisson equation (Sect. 1.5). These equations can be solved directly by analytic methods if the system geometry and material properties are simple (for example, dielectric layers with uniform ϵ_r between concentric cylinders). Analytic solutions are extremely difficult in systems with asymmetric structres or non-linear materials. Furthermore, closed-form results

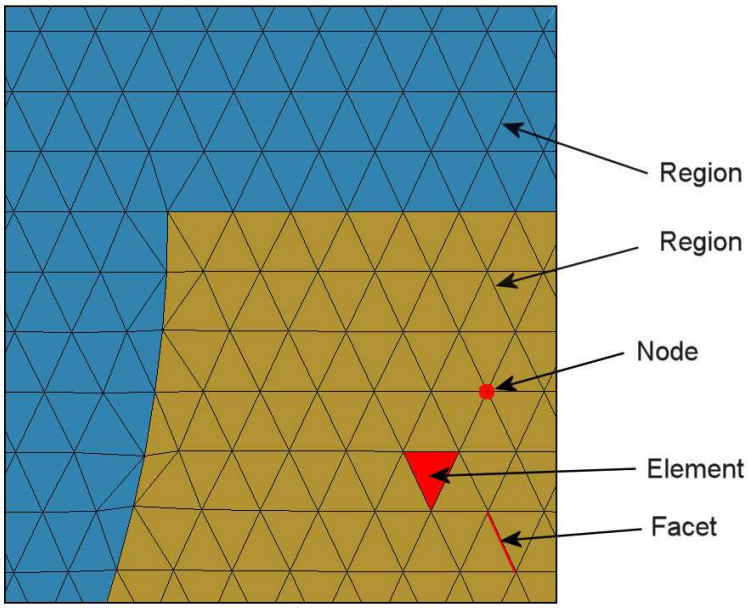

Figure 1: Conformal triangular mesh – definition of terms.

are often expressed in terms of series expansions that must be evaluated numerically. For all but the simplest problems, it is usually quicker and more accurate to employ a direct numerical approach.

The fundamental issue in numerical field solutions is that digital computers cannot directly solve continuous equations. On the other hand, computers are well suited to solving large sets of coupled linear equations. The goal of all numerical field methods (finite-difference, finite-element or boundary-element) is to convert the governing differential equations into a set of coupled linear equations. The solution of the linear equations approaches the results of the differential equation when the set becomes large.

The basis of the finite-element approach is to divide the full solution volume into a number of small volumes, or *elements*. Here, the term *small* indicates that element dimensions are much less than the scale length for variations of field quantities. The division of the volume is called the *computational mesh*. Figure 1 shows the type of mesh used for the two dimensional solutions of **EStat**. The figure defines three terms that will be used throughout this manual:

- **Element**. Volume divisions of the system.
- **Nodes**. Points where element sides intersect.
- **Facets.** Surfaces between two elements.

The elements in Fig. 1 have triangular cross-sections. In a planar solution, the cross-section lies in the x-y plane and the elements extend an infinite distance in the z direction. In a cylindrical solution, an element is a figure of revolution about the axis with a triangular cross-section in the z-r plane.

The mesh in Fig. 1 has the important property of *conformality*. The term means that the triangles have been specially shaped to conform to the boundaries between materials (*regions*). As a result, each element has an unambiguous material identity. The finite-element method is

based on two approximations: 1) material properties in an element are uniform and 2) elements are small enough so that the field quantities may be approximated by simple interpolation functions. With these assumptions, the governing differential equation can be integrated over elements surrounding a node to yield a linear equation. This equation relates the field quantity at the node to those at the surrounding nodes. The coupled set has one linear equation for each node in the mesh. In **EStat**, solution of the set representing the Poisson equation gives the electrostatic potential at each node. We can then perform two-dimensional interpolations to find potential at intervening points, or take numerical spatial derivatives to find the components of the electric field.

With this background, we can understand the steps in an **EStat** solution:

1. You define the boundaries of material objects in a solution volume for your application. The task is usually accomplished with the drawing editor of **Mesh**. You can also specify control information such as the target element sizes that may affect the accuracy and run time of the solution. The result is a text record (*script*) with sets of line and arc vectors that outline electrodes and dielectrics.

2. **Mesh** analyzes the boundary specifications and automatically generates a set of conformal triangles such as those of Fig. 1. The program creates an output file that lists the locations of nodes and the identities of elements.

3. You define the material properties of regions in the solution volume. Usually, this task is performed through an interactive dialog in **EStat**.

4. **EStat** reads the mesh geometry and applies the material parameters to generate the linear equation set.

5. **EStat** solves the coupled equations using iterative methods and records the node potentials and coordinates in an output file. This file serves as a permanent record of the solution that can be re-loaded for latter analysis.

6. You can use the interactive graphical environment of **EStat** to explore the solution. The program creates a wide variety of plots and performs quantitative calculations of field quantities. You can also write scripts for automatic control of complex analyses.

1.4 Scripts and data files

Mesh and **EStat** read and generate several types of files. For a typical solution it is not necessary for your to deal directly with the files – the user interface takes care of data organization. On the other hand, if you generate a large amount of data it's a good practice to archive application results in individual folders. Furthermore, there are advanced features of **Mesh** and **EStat** that require direct entries to the input scripts. For convenience, all input scripts and output data files are in text format. Both **Mesh** and **EStat** feature integrated text editors. Table 1 lists the file types and functions in the **EStat** package. Note that the suffix indicates the file function.

Table 1: **EStat** files

Name	Function
`MName.MIN`	**Mesh** input script (definition of foundation mesh and region outlines)
`MName.MLS`	**Mesh** diagnostic listing
`MName.MOU`	**Mesh** output (node locations and element identifies)
`EName.EIN`	**EStat** input script (run control and material properties)
`EName.ELS`	**EStat** diagnostic listing
`EName.EOU`	**EStat** output file (node locations and electrostatic potential)
`AName.SCR`	**EStat** script for automatic data analysis

1.5 Theoretical background

The text **Finite-element Methods for Electromagnetics** (included with the package) gives detailed information on the theory of electrostatics and the application of finite-element methods in **EStat**. This section summarizes some basic concepts that may be helpful in preparing solutions.

In a region of ideal dielectrics and space charge, the electrostatic potential ϕ is determined by the Poisson equation:

$$\nabla \cdot \epsilon_r \nabla \phi = -\frac{\rho}{\epsilon_0}. \tag{1}$$

In Eq. 1, the quantity ϵ_r is the relative dielectric constant and ρ is the space-charge density (in coulombs/m^3). In general, values of ϕ, ϵ_r and ρ vary with position. In the absence of space charge, Eq. 1 has the form of the Laplace equation:

$$\nabla \cdot \epsilon_r \nabla \phi = 0. \tag{2}$$

The electric field is related to the potential by:

$$\mathbf{E} = -\nabla \phi. \tag{3}$$

Equation 1 holds when the distribution of charges on electrodes and dielectric surfaces is influenced mainly by the flow of displacement current. **EStat** may also be used to model steady-state flow of real current in conductive media. In conductors, the current density is related to the gradient of potential by

$$\mathbf{j}_r = \sigma \mathbf{E} = -\sigma \nabla \phi. \tag{4}$$

where σ is the electrical conductivity in S/m. Conservation of charge in the medium implies that

$$\nabla \cdot \mathbf{j}_r = 0. \tag{5}$$

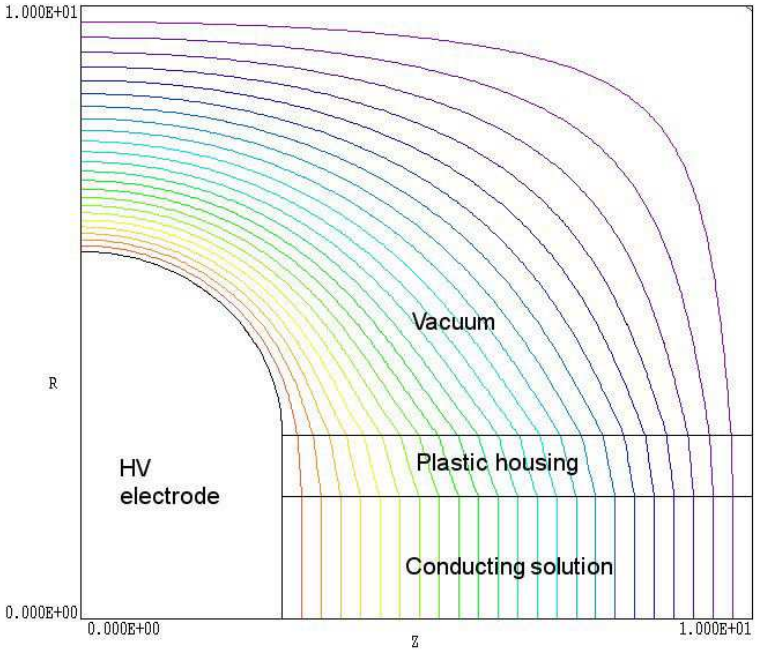

Figure 2: Illustration of a mixed dielectric-conductive solution.

or

$$\nabla \cdot \sigma \nabla \phi = 0. \tag{6}$$

Equation 6 has the same form as Eq. 2 with the association:

$$\epsilon \Rightarrow \sigma. \tag{7}$$

The implication is that the same routines may be used in **EStat** to find either dielectric or conductive solutions.

We can derive a criterion to decide when dielectric or conductive solutions are appropriate. In the presence of an RF electric field oscillating at frequency f, the displacement current density amplitude in a dielectric medium is given by:

$$|\mathbf{j}_d| = 2\pi f \epsilon_r \epsilon_0 |\mathbf{E}|. \tag{8}$$

The real current amplitude is given by Eq. 4. The dielectric limit applies when $|\mathbf{j}_d| \ll |\mathbf{j}_r|$, or

$$\sigma \ll 2\pi f \epsilon_r \epsilon_0 \cong \frac{\epsilon_r \epsilon_0}{\Delta t}. \tag{9}$$

The quantity Δt in Eq. 9 represents an approximate time scale for anharmonic field variations. For example, with a 100 ns voltage pulse we can treat de-ionized water as an ideal dielectric if the conductivity satisfies the condition $\sigma \ll 7.2 \times 10^{-3}$ S/m or if the volume resistivity is much higher than 140 Ω-m. A electrostatic solution is conductive when

$$\sigma \gg 2\pi f \epsilon_r \epsilon_0 \cong \frac{\epsilon_r \epsilon_0}{\Delta t}. \tag{10}$$

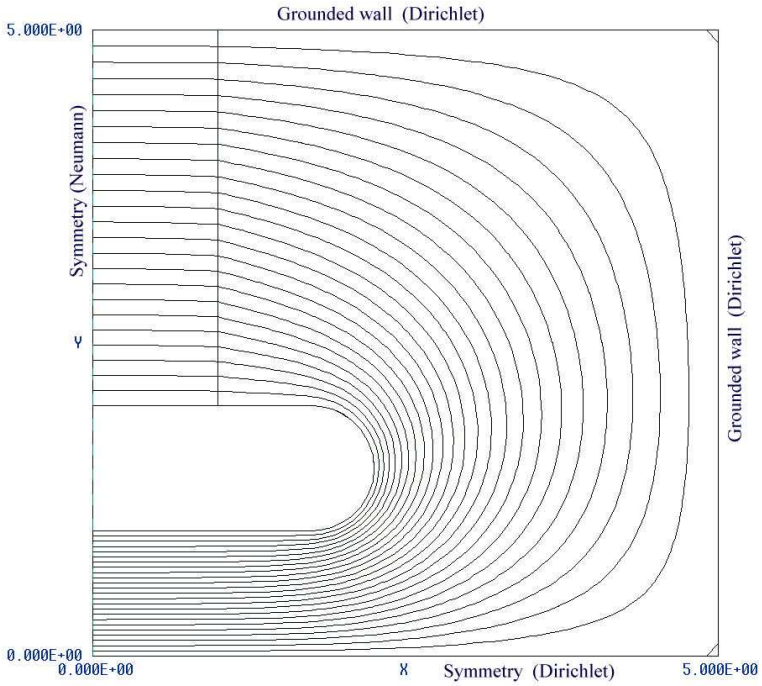

Figure 3: Application of Neumann boundaries to a symmetric system – bipolar electrodes at the top and bottom. The solution applies to one quarter on the solution volume. The specialized Neumann condition applies on the left-hand side and the Dirichlet condition $\phi = 0.0$ applies on the bottom boundary.

EStat does not handle general solutions where both real and displacement currents have significant roles. The Field Precision **RFE2** package addresses this regime. **EStat** may be useful to model systems with a mixture of materials where some satisfy the limit of Eq. 9 and others satisfy Eq. 10. For example, Fig. 2 shows a resistive probe for pulsed voltages. The conductive solution inside the probe has been adjusted to satisfy Eq. 10. The plastic housing and vacuum space have $\sigma = 0.0$ and therefore satisfy Eq. 9. In the dielectric solution of Fig. 2 the plastic and vacuum regions have relative dielectric constants $\epsilon_r = 2.7$ and $\epsilon_r = 1.0$. The conductive region was assigned the high value $\epsilon_r = 1.0 \times 10^4$. The electrostatic solution in the resistor was therefore determined by the shape of the conductive medium with little dependence on the surrounding dielectrics. Note that the electric field in the solution has uniform amplitude and points mainly in the axial direction. The solution within the conductor provides the proper boundary condition for the field solution in the surrounding insulators.

To conclude, we shall review boundary conditions in electrostatic solutions. The *boundary* is the outer edge of the finite solution volume. Nodes on the boundary may assume one of two conditions:

- *Dirichlet* boundary points have a fixed potential that does not change as the **EStat** relaxation proceeds. A region of uniform potential nodes represents an electrode (equipotential volume). Electric field lines are normal to such a surface.

- A *Neumann* boundary is one where the normal derivative of the potential is specified. The boundaries in **EStat** are limited to the specific case $\partial\phi/\partial n = 0.0$. 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 set as fixed potentials automatically satisfy the special Neumann condition. The condition applies even if the boundaries are slanted or curved.

Neumann and Dirichlet boundaries are often employed to reduce computation time for symmetric systems. For example, Fig. 3 shows an equipotential plot for a system with bipolar planar electrodes on the top and bottom. The conditon $\phi = 0.0$ is satisfied in the midplane at $y = 0.0$ and the condition $E_\perp = 0.0$ holds along $x = 0.0$. The solution time can be reduced by a factor of four by limiting the solution volume to the region $x \geq 0.0$, $y \geq 0.0$ and applying Neumann and Dirichlet conditions on the left and bottom boundaries respectively.

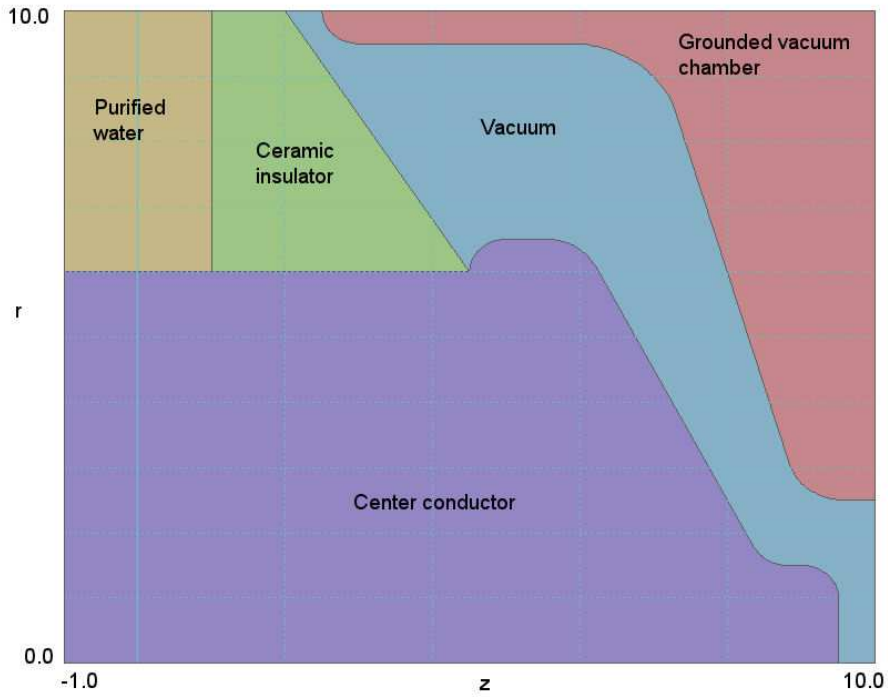

Figure 4: **EStat** example – high-current electron beam diode

2 Creating and analyzing a solution with EStat

2.1 Mesh generation

As a quick introduction to **EStat** techniques and capabilities, we shall work through the example shown in Fig. 4. The solution volume represents the output section of a high-current, pulsed-power generator. The system has cylindrical symmetry and the dimensions are in inches[1]. A voltage pulse from a coaxial water-filled transmission line travels through a vacuum insulator to a high-current electron-beam load. The 100 ns voltage pulse is long enough to ensure that an electrostatic solution is a good approximation but short enough so that the effects of conductivity in the water can be neglected. The **Mesh** script `ElectronDiode.MIN` (supplied in the example library) is required for the solution. Copy the file to a working directory and be sure that the *Data folder* in the **TC** program launcher is set correctly.

Run **Mesh** and load `ElectronDiode.MIN`. Pick the *Edit script/graphics* command to enter the drawing editor. Here, you can use the display capabilities to confirm the following region assignments:

[1]In a *z*-*r* plot, the physical system is a rotation of the plotted regions around the lower axis in Fig. 4.

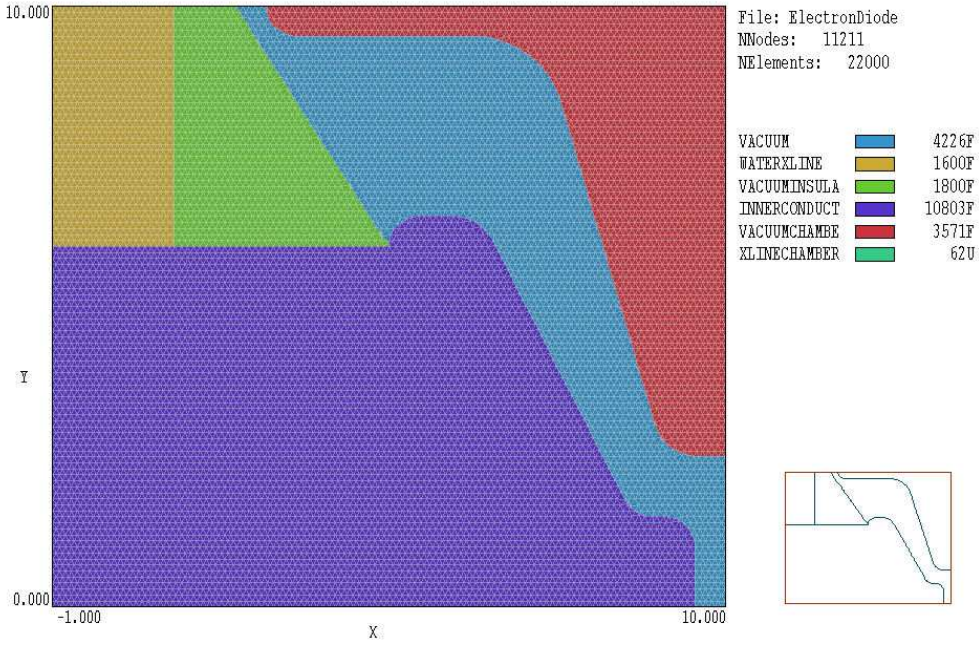

Figure 5: Completed mesh for the electron diode example.

1. Vacuum (Filled)
2. Water (Filled)
3. Vacuum insulator (Filled)
4. Shaped high-voltage inner conductor (Filled)
5. Shaped section of grounded vacuum chamber (Filled)
6. Ground condition applied along top and right sides of the solution region not covered by Region 5.

Abandon the drawing and return to the main menu. Click the *Process* command. The completed mesh is shown in Fig. 5. Choose the *Save mesh (MOU)* command. You can now close or minimize **Mesh**.

2.2 Creating the EStat control script

Run **EStat** from **TC**. The screen is initially blank and the status bar indicates that the program is waiting for input. Note the prominent tools marked *1*, *2* and *3*. The notation is a reminder of the three steps in an electrostatic solution:

1. Set up program controls and material properties.
2. Generate and solve the finite-element equations.
3. Analyze the solution.

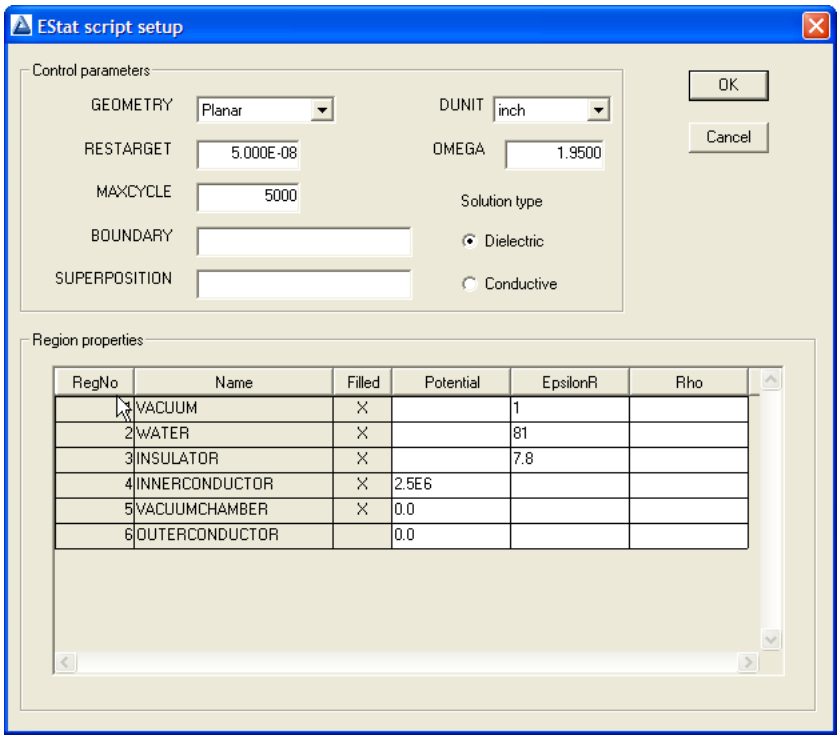

Figure 6: Dialog to set **EStat** control parameters and region properties.

To start the first operation, we must identify the **Mesh** output file that defines the system geometry. Click the *1* tool or the *Setup* menu command. In the dialog, choose the file `ElectronDiode.MOU`. **EStat** loads the information and displays the dialog of Fig. 6. Note that the grid contains a row for each mesh region with a notation of the *Fill* status. In the default dielectric mode, there are three columns in the grid where you can enter values: *Potential* (fixed voltage), *EpsilonR* (relative dielectric constant) and *Rho* (optional space-charge density in units of coulomb/m^3).

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

- GEOMETRY. The symmetry of the solution: *planar* or *cylindrical*. A planar solution varies in x-y and has infinite length in z, while a cylindrical solution has symmetry in θ.
- 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.3.5.
- DUNIT. Choose an entry to set a factor to convert the units used for coordinates in the **Mesh** file to meters. The factor is the number of mesh units per meter: 39.37 for inches, 100.0 for cm.
- OMEGA. A parameter in the range 0.0 to 2.0 to control the iterative matrix solution.

- SOLUTION TYPE. The options are *Dielectric* and *Conductive*. The solution types were described in Sect. 1.5.

The column options in the region grid box depend on the solution type: relative dielectric constant ϵ_r and space-charge density ρ for a dielectric solution and electrical conductivity σ for a conductive solution. The entered values shown in Fig. 5 define the following characteristics:

- Region 1. Dielectric (vacuum): $\epsilon_r = 1.0$.
- Region 2. Dielectric (water): $\epsilon_r = 81.0$.
- Region 3. Dielectric (ceramic): $\epsilon_r = 7.8$.
- Region 4. Fixed potential (inner conductor): $\phi = 2.5 \times 10^6$ V.
- Region 5. Fixed potential (vacuum chamber): $\phi = 0.0$ V.
- Region 6. Fixed potential (boundary): $\phi = 0.0$ V.

Set up the dialog with the values shown in Fig. 6 and click *OK*. **EStat** uses the information in the dialog to create the script `ElectronDiode.EIN` shown in Table 2. Chapter 3 reviews the script format and advanced program capabilities.

2.3 Finite-element solution

The next step is to generate and to solve the finite-element equations. Click on the *Solve/Solution display* to display the status of the matrix solution. Then choose the tool marked *2* or the menu command *Solve*. Accept the default entry of `ElectronDiode.EIN` in the dialog by clicking `OK`. The screen shows the progression of equipotential lines as the solution accuracy improve, while the status bar reports the progress of the operation. **EStat** creates the output file `ElectronDiode.EOU` which contains full information on the mesh as well as values of the electrostatic potential at each node. Section 3.7 describes the file format.

2.4 Solution analysis

We can now use the information in `ElectronDiode.EOU` to create plots and to perform quantitative analyses. Choose either the tool marked *3* or the *Analyze* menu command. In the dialog, accept the default choice of solution file. **EStat** loads the file, sets up the analysis menu and toolbar, and creates the default contour plot shown in Fig. 7. Take a moment to inspect the organization of the working environment. The status bar at bottom lists the following information: solution file name, data file name, plot quantity, plot type, interpolation method and mouse snap status. The data file gives you the option to record the results of analysis operations in text format. The plot window is divided into four areas. The main plot is at the upper left and the plot legend is at the upper right. The orientation area at the lower right is a miniature representation of the solution volume with an outline showing the current view. The information area at the bottom left appears in response to analysis commands such as *Point calculation* or *Region properties*.

Table 2: **EStat** solution script for the electron diode example

```
* File: ElectronDiode.EIN
Mesh = ElectronDiode
Geometry = Cylin
DUnit =  3.9370E+01
ResTarget =  5.0000E-08
Omega =  1.9500E+00
MaxCycle =    5000
Epsi(1) =  1.0000E+00
Epsi(2) =  8.1000E+01
Epsi(3) =  7.8000E+00
Potential(4) =  2.5000E+06
Potential(5) =  0.0000E+00
Potential(6) =  0.0000E+00
EndFile
```

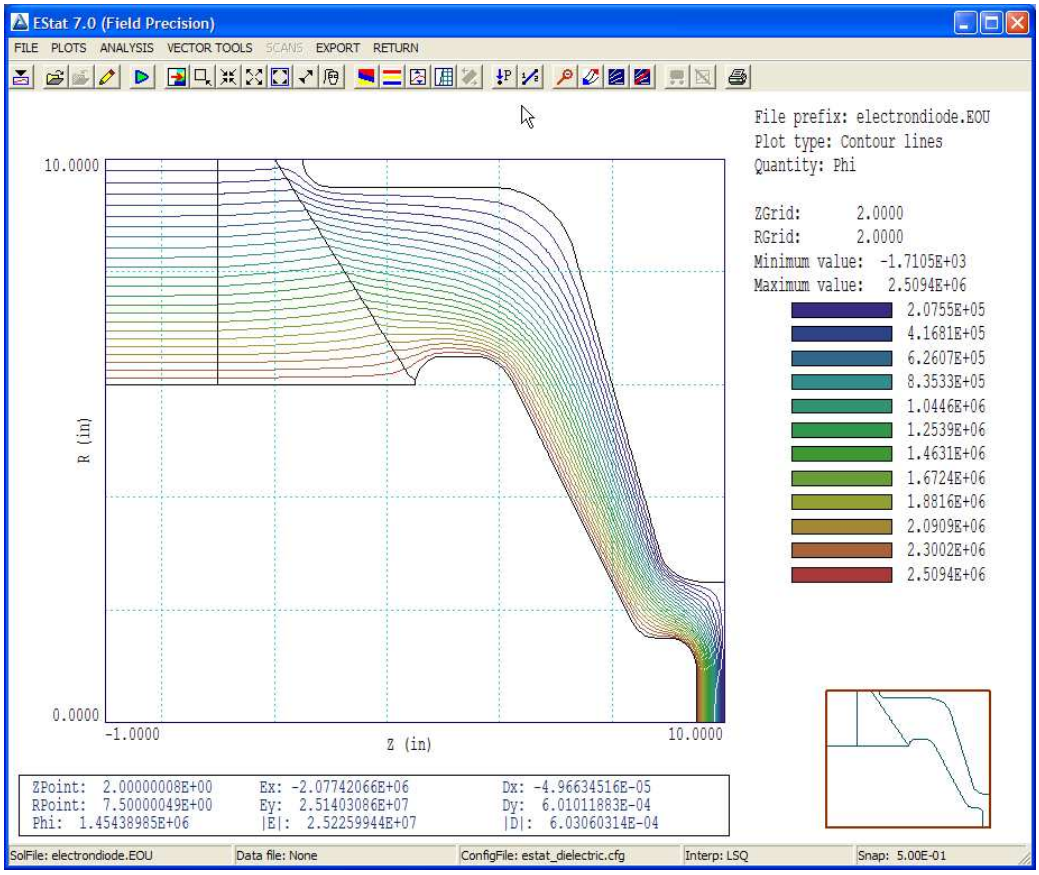

Figure 7: Working environment of **EStat** in the analysis mode.

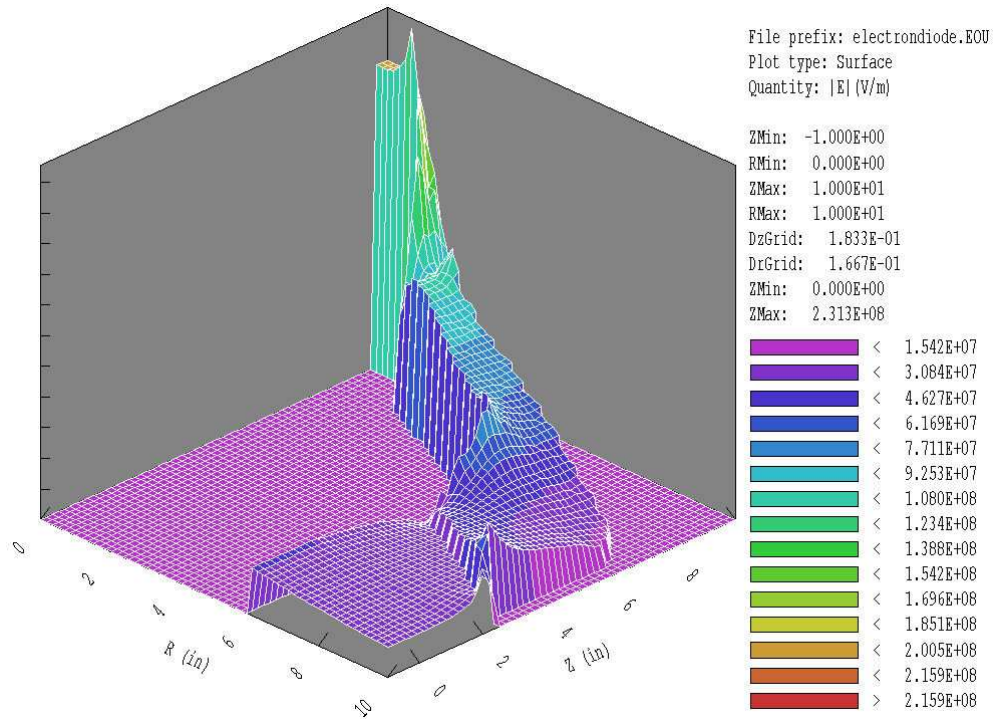

Figure 8: Surface plot – height shows $|\mathbf{E}|$ as a function of (z,r).

Table 3: **EStat** analysis script `ElectronDiode.SCR`.

```
INPUT ElectronDiode.EOU
OUTPUT ElectronDiode
VOLUMEINT
NSCAN 25
SCAN 9.9999 0.0000  9.9999 2.5000
ENDFILE
```

Table 4: Entry in the data file created by `ElectronDiode.SCR`.

```
 --- Volume Integrals ---
Energy:   1.681E+03 J
Power:   1.899E+14 W
Epeak:   2.474E+08 V/m
ZPeak:   9.482E+00
RPeak:   1.238E+00

    Integrals by region
NReg   Volume     Energy      Power       PeakE     PeakZ      PeakR
        (m3)       (J)        (W)        (V/m)
=========================================================================
   1  1.372E-02  1.439E+02  1.625E+13  2.474E+08  9.482E+00  1.238E+00
   2  6.590E-03  1.401E+03  1.582E+14  3.164E+07 -9.344E-01  6.033E+00
   3  7.073E-03  1.366E+02  1.543E+13  3.849E+07  2.014E+00  9.931E+00
   4  1.583E-02  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
   5  1.342E-02  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00
```

The following chapter gives detailed reference material on analysis function of **EStat**. You can experiment with the program capabilities to create a wide variety of plots, such as the three-dimensional surface plot of Fig. 8. To complete the session, we shall run a script that controls automatic calculations. To inspect the file content, click on *Edit script* in the *File* menu and pick the file `ElectronDiode.SCR`. Table 3 shows the file content. The first command ensures that the file `ElectronDiode.EOU` is loaded while the second command opens the data file `ElectronDiode.DAT`. The *VolumeInt* command initiates automatic volume integrals of electrostatic field energy. The *Scan* command writes a set of 25 data lines listing the potential and electric field components over the anode plane. Table 4 shows an excerpt from the listing. The field energy U_e in Region 1 (the vacuum diode) is 143.9 joules. Using the formula $U_e = CV^2/2$ and taking $V = 2.6 \times 10^6$, the capacitance of the diode region is approximately 44.36 pF.

Table 5: Setting material models from entries in the **EStat** dialog

Material type	Potential	EpsilonR	Rho	Sigma
Fixed-potential electrode	X			
Isotropic dielectric		X		
Dielectric/space charge		X	X	
Isotropic conductor				X

3 EStat solution reference

3.1 EStat script format

You can create a control script for **EStat** interactively using the dialog described in Sect. 2.2 or by writing the commands directly with a text editor. You can easily alter any script with an editor. This chapter gives a detailed description of the syntax and functions of script commands. Section 3.2 covers commands that control program operation. These commands are created by the entries in the *Control parameters* section of the dialog (Fig. 6). Section 3.3 reviews commands to set simple material properties. These commands are created by entries in the *Region properties* grid of the dialog. Table 5 summarizes how entries in the dialog control the type of material associated with a region. The remaining sections describe advanced capabilities of **EStat** as well as program operation and output file format.

The **EStat** script is a text file composed of data lines that contain commands and parameters. You can construct a script using the *Setup* dialog in **EStat**. You can also write and modify scripts with an editor. Direct script editing is required for some advanced **EStat** capabilities.

The script must end with the *EndFile* command. The entries on a line may be separated by the standard delimiters introduced in the **Mesh** manual:

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

Any number of delimiters may be used in a line. Blank lines and comment lines are ignored. Comment lines begin with an asterisk (*). **EStat** accepts commands in any order. The following example illustrates a complete script:

```
* File CYLPROBE.EIN
GEOMETRY = Cylin
DUNIT = 100
* Region 1: Set to EpsiR = 1.0
EPSI(1) = 1.0
* Region 2: Dielectric sheath
EPSI(2) = 1.0E-6
* Region 3: Probe
POTENTIAL(3) = 1.0
* Region 4: Chamber wall
POTENTIAL(4) = 0.0
ENDFILE
```

There are two classes of commands: program control and region properties. A control command contains a keyword and a value. Region commands set the physical properties associated with elements and nodes. They have the format:

```
Keyword  RegNo  Value
```

Here, the integer *RegNo* is the region number defined in the **Mesh** input file. The string *Keyword* specifies the physical property. The value may be one or more numbers. As an example, the command

```
POTENTIAL 2 -5500.0
```

sets nodes with region number 2 to the fixed potential -5.5 kV.

3.2 Program control commands

In the following sections, commands are written symbolically and as they might appear in the **EStat** script:

MESH MPrefix
MESH SparkGap
Specify the **Mesh** output file that defines the geometry of the simulation. The file must have a name of the form `MPrefix.MOU` and be available in the working directory. If this command does not appear in a script with the name `EPrefix.EIN`, then **EStat** will seek the default mesh file `EPrefix.MOU`.

DUNIT Unit
DUNIT = micrometer
DUNIT = 1.0E4
You can use any convenient distance units in **Mesh**. This command defines a factor to convert coordinates supplied by **Mesh** to the standard distance units of meters used in **EStat**. The Unit argument may be one of the following strings for common units: `angstrom`, `nanometer`, `micrometer`, `mil`, `mm`, `cm`, `inch`, `foot`, `yard`, `meter`, `kilometer` or `mile`. For custom units,

enter a real number equal to the number of **Mesh** units per meter. For example, if the **Mesh** dimensions are entered in microns, set $Unit = 1.0 \times 10^6$. The default value is 1.00. (**Note**. Spatial quantities recorded in the output file `FPrefex.EOU` are always in meters. In an analysis session with **EStat**, spatial quantities in graphs and listing files are scaled to the **Mesh** units. For example, if the **Mesh** dimensions are in cm and $Unit = 100.0$, the spatial quantities in plots will be in cm.)

GEOMETRY [Rect, Cylin]
GEOMETRY = Cylin
EStat handles problems in rectangular or cylindrical geometries. Rectangular systems have variations in x and y with infinite length in z. Cylindrical systems have variations in r and z with azimuthal symmetry. The parameter options are *Rect* and *Cylin*. In cylindrical solutions the program takes the z axis along the **Mesh** x direction and the r axis along y. In this case the program issues an error message if any node has a y coordinate less than 0.0.

OMEGA Omega
OMEGA = 1.96
This command controls the relaxation factor for the iterative matrix solution, a number in the range 0.0 to 2.0. Generally, higher values give faster convergence. Reduce the relaxation factor if the solution does not converge. If this command does not appear, the program uses the Chebyshev acceleration method to pick optimal values.

MAXCYCLE MaxCycle
MAXCYCLE = 500
The maximum number cycles for the iterative matrix solution. The default value is 2500.

RESTARGET ResTarget
RESTARGET = 5.0E-6
EStat calculates the relative error in the magnitude of the potential during the iterative matrix solution. The program stops when the error falls below *ResTarget*. Solutions generally have sufficient accuracy for *ResTarget* less than about 1.0×10^{-6}. The default value is 5.0×10^{-8}.

INTERP [Linear, Spline]
INTERP = Spline
Set the interpolation method used when spatial variations of ϕ, ϵ_r, σ and/or ρ are defined from data tables. Use the *Linear* interpolation method for noisy or discontinuous data. The default is *Spline*.

PARALLEL [NProc]
PARALLEL = 4
Use multiple threads for the matrix inversion. The integer parameter *NProc* is the number of threads to invoke. If omitted, the number of threads equals to maximum number on the

machine (8 for a Core i7 processor). This command functions only for multi-core or multi-processor machines running the 64-bit version of the software. Parallel processing may be useful for huge-mesh solutions that run several minutes. In typical runs, the computational overhead may actually increase the run time.

3.3 Commands for material properties

The following four commands 'define the properties of isotropic materials. They may be generated with the *Setup* dialog in **EStat** or directly with a text editor.

POTENTIAL RegNo Pot
POTENTIAL(5) = 3500.0
The keyword *Potential* designates that the potential of the region nodes is fixed and will not change during the matrix relaxation. The command sets the amplitude of the potential in volts. The default value for all fixed regions is 0.0 V. The command applies to both dielectric and conductive type solutions.

EPSI RegNo EpsiR
EPSI(3) = 5.8
Set the relative dielectric constant, $\epsilon_r = \epsilon/\epsilon_0$, in dielectric regions. The default value is $\epsilon_r = 1.0$. This command applies only to dielectric type solutions.

RHO RegNo Rho
RHO(7) = 3.6E-3
Sets the space-charge density in coulombs/m^3. Because the quantity is not physically meaningful in problems with conductive materials, the command functions only in dielectric solutions. In this case both the *Epsi* and *Rho* commands may apply to the same dielectric region.

SIGMA Regno Sigma
SIGMA(2) = 0.145
Sets the conductivity in S/m. The conductivity is equal to the inverse of the volume resistivity ρ in units of Ω-m. This command applies only to conductive type solutions.

In a single solution, you can use *Sigma* commands or the *Epsi*/*Rho* commands. **EStat** issues on error message if dielectric and conductive quantities are mixed.

3.4 Anisotropic materials

EStat handles anisotropic dielectrics and conductors with different values of ϵ_r or σ along two normal axes.

EPSI RegNo Epsi1 Epsi2 Theta
EPSI(5) = (5.8, 2.6, 45.0)
Define an anisotropic dielectric material with different values of relative dielectric constant along two normal axes. The first real-number value (ϵ_1) following the region number is the relative dielectric constant along Axis 1, and the second number (ϵ_2) is the dielectric constant along Axis 2. The default values are $\epsilon_1 = \epsilon_2 = 1.0$. The third number ($\theta$) is the angle in degrees of Axis 1 relative to the x (or z) axis. The angle of Axis 2 is $\theta + 90^o$. For cylindrical solutions, it is important to note that all structures must have cylindrical symmetry. Therefore, if $\theta \neq 0.0^o$, then Axis 1 and Axis2 have the shape of cones. Because it is unlikely to encounter such a material, practical cylindrical simulations are generally limited to the choice $\theta = 0.0^o$ (different relative dielectric constants along z and r).

SIGMA RegNo Sigma1 Sigma2 Theta
SIGMA(8) = (5.0, 0.1, 45.0)
Define an anisotropic conductor. The first real number (σ_1) following the region number is the electrical conductivity in S/m along Axis 1, and the second number (σ_2) is the conductivity along Axis 2. The third number (θ) is the angle of Axis 1 relative to the x (or z) axis in degrees.

The analysis functions of **EStat** include the effects of material anisotropy in calculations of field energy, resistive power dissipation, induced charge and current density. The following formulas apply to planar solutions or to cylindrical solutions with the substitution x $\rightarrow$ z, y $\rightarrow$ r. For anisotropic conductive materials, the local components of current density are related to the electric field by

$$\begin{bmatrix} j_x \\ j_y \end{bmatrix} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{bmatrix} \begin{bmatrix} E_x \\ E_y \end{bmatrix} . \tag{11}$$

where

$$\sigma_{xx} = \sigma_1 \cos^2\theta + \sigma_2 \sin^2\theta,$$

$$\sigma_{xy} = \sigma_{yx} = (\sigma_1 - \sigma_2)\cos\theta \sin\theta,$$

$$\sigma_{yy} = \sigma_1 \sin^2\theta + \sigma_2 \cos^2\theta. \tag{12}$$

Given the electric field and current density, the resistive power density is

$$p \;=\; \frac{1}{2}\;[\; j_x\; E_x \;+\; j_y\; E_y\;]\;. \tag{13}$$

Anisotropic dielectrics have different values of relative dielectric constant (ϵ_1 and ϵ_2) along two normal axis. We shall write equations in terms of the applied field $\mathbf{E}_0$ (the total electric field minus contributions from dielectric materials). The applied field is related to the displacement vector by

$$\mathbf{D} \;=\; \epsilon_0\; \mathbf{E}_0\;. \tag{14}$$

The applied field is related to the total electric field by

$$\begin{bmatrix} E_{0x} \\ E_{0y} \end{bmatrix} = \begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} \\ \epsilon_{yx} & \epsilon_{yy} \end{bmatrix} \begin{bmatrix} E_x \\ E_y \end{bmatrix} . \tag{15}$$

where

$$\epsilon_{xx} = \epsilon_1 \cos^2\theta + \epsilon_2 \sin^2\theta ,$$

$$\epsilon_{xy} = \epsilon_{yx} = (\epsilon_1 - \epsilon_2) \cos\theta \sin\theta ,$$

$$\epsilon_{yy} = \epsilon_1 \sin^2\theta + \epsilon_2 \cos^2\theta . \tag{16}$$

The electrostatic field energy density is given by

$$u = \frac{\epsilon_0}{2} [E_{0x} E_x + E_{0y} E_y] . \tag{17}$$

The induced charge density on the surface of a conductor may be expressed in terms of the electric field component normal to the surface. The *total* surface charge density (in coulomb/m^2) is given by

$$\rho = \epsilon_0 E_\perp . \tag{18}$$

The total charge includes contributions from free charge on the electrode and displaced charge in the adjacent dielectric. The *free* charge density is given in terms of the applied field by

$$\rho = \epsilon_0 E_{0\perp} . \tag{19}$$

The analysis routines of **EStat** calculate both dielectric and resistive quantities. The user decides which ones are relevant to a particular calculation. The operations in **EStat** are identical for both types of solutions. Therefore, in a dielectric solution the quantity reported as current density is equal to the applied field.

3.5 Boundary and superposition commands

EStat can perform accurate calculations of fields near small features in a large solution space. Suppose we wanted a precise calculation near a field-emission tip. The radius of the tip is much smaller than the scale size of the of the supporting electrodes. We need to perform 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. A limitation to this technique arises from the structured mesh used in **TriComp** - the region of small elements must extend the full length of the solution volume.

Figure 9 illustrates an alternate approach. We create a second solution that covers a small subregion of the original solution (green outline). The microscopic solution includes electrodes that penetrate the subregion. The difference is that the electrode shapes may be resolved in greater detail. The challenge is how to include the macroscopic fields correctly into the

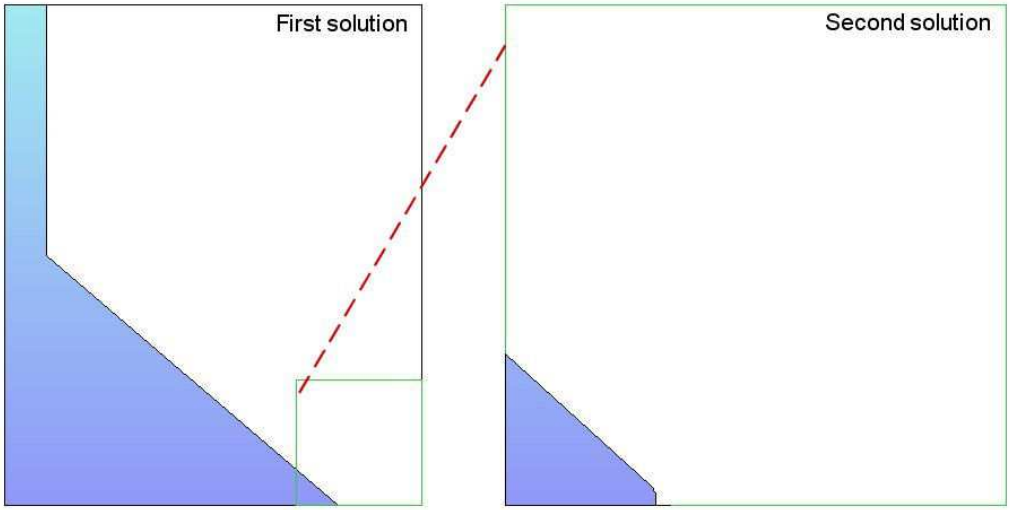

Figure 9: Interpolated Dirichlet boundary for a microscopic solution.

microscopic solution. The approach is **EStat** is to enclose the second solution in a variable-potential Dirichlet boundary. Values of ϕ 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 not close to the boundary. The variable potential boundary is controlled by the following command in the control script of Solution 2:

BOUNDARY FPrefix [BndScale]
BOUNDARY FEmitMacro
The string quantity *FPrefix* is the prefix of the output file for Solution 1. The optional quantity *BndScale* (real) is a scaling factor applied to potential values transferred from Solution 1 to Solution 2. The default is *BndScale* = 1.0.

EStat issues an error message under the following conditions:

- The output file for Solution 1 (`FPrefix.EOU`) is not available in the working directory.
- Solution 2 does not fit completely inside Solution 1.
- Solution 1 and Solution 2 have different symmetries.
- An interpolation failed.

The program makes no further validity checks. You must ensure that the geometry of Solution 2 represents a correct section of Solution 1.

A node in Solution 2 is on the boundary under two conditions:

- It has indices $K = 1$, $K = K_{max}$, $L = 1$ or $L = L_{max}$. In solutions with cylindrical symmetry, points on the axis ($L = 1$ and $r = 0.0$) are not set to the Dirichlet condition.

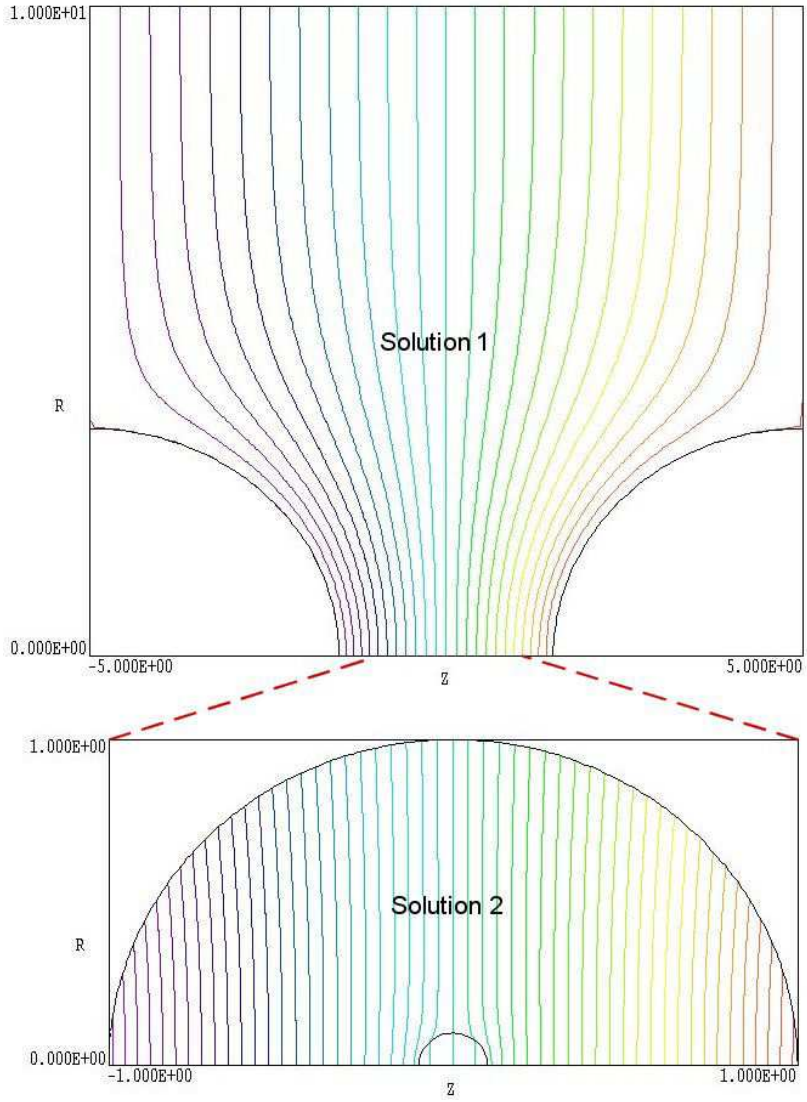

Figure 10: Example of a two-stage solution using the *Boundary* command..

- One of the adjacent elements has $RegNo = 0$. This condition means that you can use non-rectangular boundaries for Solution 2.

Figure 10 illustrates the boundary point criteria. Solution 1 represents electric fields between two spherical-section electrodes separated by a 3.00" gap. We want to find the fields on a small dielectric body inserted at the center of the gap. Solution 2 is a spherical region (radius 1.00") centered at the midpoint. The microscopic solution contains a dielectric sphere of radius 0.10". Note the assignment of potential values on the circular outer boundary of Solution 2. The lower boundary of the cylindrically-symmetric solution has not been set to the Dirichlet condition; therefore, the on-axis potential can adjust to the presence of the dielectric. Input scripts for this example are included in the **EStat** example library under the names *BoundaryVal01* and *BoundaryVal02*.

The *Superposition* command directs **EStat** to superimpose values from a large-scale solution (Solution 1) on a small-scale solution (Solution 2). The following statement appears in the

command script for Solution 2.

SUPERPOSITION FPrefix [SScale]
SUPERPOSITION UniField
The quantity *FPrefix* (string) is the prefix of the output file for Solution 1. The optional quantity *SScale* (real) is a scaling factor applied to potential values transferred from Solution 1 to Solution 2. The default is $SScale = 1.0$.

In response to the command, **EStat** opens the file `FPrefix.EOU` after Solution 2 has been completed. The program performs an interpolation in the space of Solution 1 to determine the potential ϕ_1 at each node location in Solution 2 and adjusts potential values according to

$$\phi_2' = \phi_2 + SScale\ \phi_1, \tag{20}$$

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.

3.6 Running EStat

EStat can run as an interactive program in a window or as a background task. To run the program from the Command Prompt, use a command of the form:

```
[ProgPath\]ESTAT [DataPath\]FPrefix.EIN <ENTER>
```

where the file `FPrefix.EIN` and the appropriate **Mesh** output file are available in the data directory. With this capability, you can set up extended autonomous **EStat** runs using a DOS batch file or a Python script.

The remainder of this section discusses commands in the main menu when **EStat** runs in the interactive mode. The program enters this mode when launched from **TC** or run with no input file prefix. The following commands appear in the *File* popup menu:

EDIT SCRIPT (EIN)
EDIT LISTING (ELS)
EDIT FILE
These commands call the internal editor to inspect or to modify **EStat** input and output files. Choosing a file from an alternate directory does not change the working directory. The *Edit script (EIN)* command shows a list of all files with names of the form `FPREFIX.EIN` while *Edit listing (ELS)* displays files with names `FPREFIX.ELS`. Note that you must exit the editor to resume normal operation of the program.

RUN ANALYSIS SCRIPT
An analysis script allows you to perform complex or repetitive analyses on a set of similar solutions. This command displays a dialog listing files with the suffix `SCR`. Pick a file and

click *OK*. The script can load data files, open and close data records, and perform any of the quantitative analysis functions described in this chapter. The script command language is described in Sect. 5.7. Note that the analysis script must be in the same directory as the data files.

SETUP

The function of this command is to create an **EStat** script to control an electrostatic solution. The program first prompts for a **Mesh** output file to define the system geometry. The prefix of the file will be used as the argument of the *Mesh* script command. The program then displays the dialog shown in Fig. 6. The number of regions in the dialog is determined by the **Mesh** file. The functions of the control parameters in the upper box are described in Sect. 3.2. You can enter basic physical properties of regions in the grid box (see Sect. 3.3). You must edit the script directly to invoke advanced functions like anistropic materials.

The *Solve* popup menu includes the following three commands.

RUN

Pick an input file (such as `FPrefix.EIN`) to start a solution. The working directory is changed if you pick a file from an alternate directory. The run begins if the file `FPrefix.MOU` or the file specified in the *Mesh* command is present. During the solution, the screen color is blue and the progress is shown in the status bar.

STOP

This command terminates **EStat** and saves the current state of the relaxation solution. For example, you may want to stop a relaxation solution at a moderate value of convergence to check whether the problem has been correctly defined.

SOLUTION DISPLAY

When *solution display* is active, **EStat** plots equipotential lines during the matrix inversion to show solution convergence. The number of relaxation steps per plot is initially small and increases as the solution nears its final state. This feature is useful to verify convergence or to identify problem areas. (Note that the plot operations increase the run time.)

ANALYZE

Pick a file of the type `FPrefix.EOU` and call up the analysis menu for plotting and numerical analysis.

ESTAT MANUAL

Display this manual in your default PDF display sofware. The file `estat.pdf` must be available in the same directory as `estat.exe`.

PHYSICAL CONSTANTS

Display a list of physical constants using the default PDF viewer. The file `PHYSCONS.PDF` must be in the same directory as `estat.exe`.

DIELECTRIC CONSTANTS

Display a list of dielectric constants for common materials using the default PDF viewer. The file `DIELECTRIC_CONSTANTS.PDF` must be in the same directory as `estat.exe`.

3.7 Format of the EStat output file

The EStat output file `FPrefix.EOU` is in text format. The file has three sections:

- Header with general information on the run
- Node and element information
- Region information

The header section consists of a title line and 10 data lines:

```
--- Run parameters ---
XMin: -2.54000508E-02
XMax:  2.54000508E-01
KMax:    111
YMin:  0.00000000E+00
YMax:  2.54000508E-01
LMax:    101
DUnit:  3.93700000E+01
NReg:   6
ICylin:  1
CondFlag: 0.0
```

Lines 2 and 3 list x_{min} and x_{max}, the limits along horizontal axis (x or z) of the solution volume. Dimensions are given in meters. The quantity k_{max} in Line 4 is the number of nodes along the horizontal direction. Lines 5-7 describe the vertical axis (y or r). Line 8 contains the quantity $DUnit$, the conversion factor from dimensions used in **Mesh** to meters. Line 9 gives the number of regions in the solution, while Line 10 specifies the symmetry (0: planar, 1: cylindrical). The final entry is 0.0 for a dielectric solution and 1.0 for a conductive solution.

The node section consists of 4 title lines and $k_{max} \times l_{max}$ data lines, one for each node of the solution space.

```
--- Nodes ---
   k     l   RgNo  RgUp  RgDn         x               y
=============================================================
     1     1     4     4     0 -2.54000508E-02  0.00000000E+00
     2     1     4     4     0 -2.27829181E-02  0.00000000E+00
     3     1     4     4     0 -2.01708561E-02  0.00000000E+00
     4     1     4     4     0 -1.75655857E-02  0.00000000E+00
     5     1     4     4     0 -1.49681144E-02  0.00000000E+00
    ...

       Phi            EpsiUp          EpsiDn           RhoUp           RhoDn
=============================================================================
 -5.00000000E+05  1.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00
 -5.00000000E+05  1.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00
 -5.00000000E+05  1.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00
 -5.00000000E+05  1.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00
 -5.00000000E+05  1.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00
```

Each data line contains the following quantities:

- The indices of the node (K,L)
- The region number of the node (*RgNo*) and region numbers for two associated elements (*RgUp* and *RgDn*). The upper element lies above the line between nodes (K,L) and (K+1,L) and the lower element lies below the line.
- The coordinates of the node in meters, (x,y) or (z,r).
- The electrostatic potential ϕ at the node in volts.
- For a dielectric solution, the final four quantities represent the relative dielectric constant and space-charge density in the upper and lower elements. For a conductive solution, the first two quantities represent the electric conductivity in the upper lower elements while the second two dummy quantities are set equal to zero.

The region section consists of four title lines following by *NReg* data lines, one for each region. For a dielectric solution, the region section has the following appearance:

```
--- Region properties ---
RegNo  Fix  Aniso                        Epsilon1      Space charge     Potential
=================================================================================
     1     0     0     0     0     0  1.00000000E+00  0.00000000E+00  0.00000000E+00
     2     0     0     0     0     0  8.10000000E+01  0.00000000E+00  0.00000000E+00
     3     0     0     0     0     0  2.70000000E+00  0.00000000E+00  0.00000000E+00
    ...

         Epsilon2         Angle          SinAngle        CosAngle
       ================================================================
        0.00000000E+00  0.00000000E+00  0.00000000E+00  1.00000000E+00
        0.00000000E+00  0.00000000E+00  0.00000000E+00  1.00000000E+00
        0.00000000E+00  0.00000000E+00  0.00000000E+00  1.00000000E+00
       ...
```

The region section for a conductive solution is as follows:

```
RegNo  Fix  Aniso                          Sigma1
=============================================================================
     1     0     0     0     0     0  1.00000000E+00  0.00000000E+00
     2     0     0     0     0     0  8.10000000E+01  0.00000000E+00
     3     0     0     0     0     0  2.70000000E+00  0.00000000E+00
```

```
   ...

     Potential       Sigma2          Angle          SinAngle        CosAngle
================================================================================
  0.00000000E+00 0.00000000E+00  0.00000000E+00  0.00000000E+00  1.00000000E+00
  0.00000000E+00 0.00000000E+00  0.00000000E+00  0.00000000E+00  1.00000000E+00
  0.00000000E+00 0.00000000E+00  0.00000000E+00  0.00000000E+00  1.00000000E+00
       ...
```

A entry of one in the second column designates a fixed-potential region (electrode), while the third column designates an anisotropic materials. Values of the final four real-number quantities are non-zero only for anisotropic materials. Finally the program records the names of regions inherited from **Mesh**:

```
--- Region names ---
VACUUM
WATER
INSULATOR
INNERCONDUCTOR
VACUUMCHAMBER
OUTERCONDUCTOR
```

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

4.1 Program capabilities

Section 3.3 covered commands to define potential values or material properties (ϵ_r, σ and/or ρ) that are uniform throughout a region. This section describes how to represent quantities that vary continously 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 + $r∧2))
Set values of ϕ (in volts) at 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 space: $f(x, y)$ for a planar solution and $f(z, r)$ for a cylindrical solution. The parser uses the Perl standard for variables where `$x` designates the variable x, `$y` stands for y, and so forth. The potential at a node equals the value of the function evaluated at the node position. The following conditions apply to the command:

- There is no limit on the number of functions – any region may be associated with a function.
- Positions are passed to the function in units set by *DUnit* (cm, inches, μm,...).
- You can model discontinuous functions by dividing a volume into two or more regions.
- Variations may be assigned to both filled regions and contiguous or disconnected line regions.

POTENTIAL RegNo TABLE [x,y,z,r] TabName
POTENTIAL(3) = TABLE R ZUpBoundary.DAT
Set values of ϕ (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 x, y and r may appear in planar solutions. In this case, the variable r is interpreted as

$$r = \sqrt{x^2 + y^2}. \tag{21}$$

The options z and r may appear in cylindrical solutions. 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.

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 ϵ_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. **EStat** issues an error message if the value of the function in any element gives $\epsilon_r \leq 0.0$. Note that functions may not be used in a solutions with materials with anisotropic dielectric constant.

EPSI RegNo TABLE [x,y,z,r] TabName
EPSI(5) = TABLE Z GradedDielectric.DAT
Set values of the relative dielectric constant ϵ_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. **EStat** issues an error message if the returned value in any element gives $\epsilon_r \leq 0.0$. Note that tables may not be used in a solutions with materials with anisotropic dielectric constant.

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. **EStat** issues an error message if the value of the function in any element is $\sigma < 0.0$. Note that functions may not be used in a solution where other materials have an anisotropic electrical conductivity.

SIGMA RegNo TABLE [x,y,z,r] TabName
SIGMA(5) = TABLE X SwitchBreakdown.DAT
Set values of the conductivity σ 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 σ in units of S/m. **EStat** issues an error message if the returned value in any element gives $\sigma < 0.0$. Note that tables may not be used in a solution where other materials have an anisotropic electrical conductivity.

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^3.

RHO RegNo TABLE [x,y,z,r] TabName
RHO(2) = TABLE Y PulsedBeam.DAT
Set values of the space-charge density ρ 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 ρ in units of coulomb/m^3

4.2 Function syntax and table format

EStat 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`) for planar solutions and (`$z`, `$r`) for cylindrical solutions.
- 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 logorithm), `exp` (normal exponent) and `sqt` (square root).
- Up to 20 sets of parentheses to any depth.
- Any number of space delimiters.

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

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

corresponds to

$$1 - \exp\left[-\left(\frac{z^2 + r^2}{24}\right)\right]. \tag{22}$$

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^3). The entries may be in any valid floating point format and may be separated by any of the standard delimiters listed in Sect. 3.1. You may include comment lines that start with an asterisk and text in any format after the *EndFile* command.

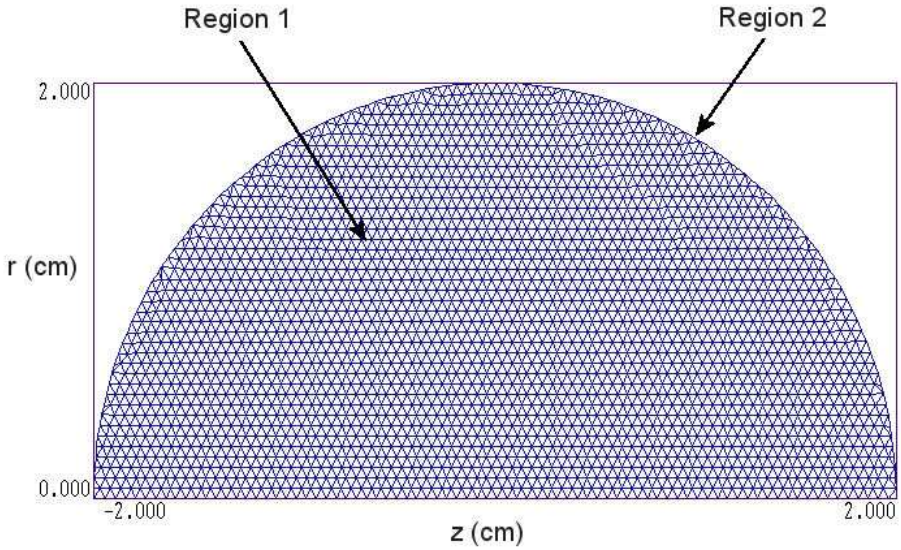

Figure 11: Geometry for the `CHARGEFUNC` example showing the element size. Dimensions in cm.

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.

4.3 Benchmark example

The `CHARGEFUNC` example (Fig. 11) 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:

$$\frac{1}{R^2}\,\frac{d}{dR}\,R^2\,\frac{d\phi}{dR} = -\frac{\rho(R)}{\epsilon_0}. \tag{23}$$

If the space-charge density has the uniform value ρ_0, solution of Eq. 23 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^3 and $R_0 = 0.02$ m, the potential is $\phi_0 = 7.529$ V. If the space-charge density varies with radius as:

$$\rho(R) = \rho_0\,\left[1 - \left(\frac{R}{R_0}\right)^2\right], \tag{24}$$

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

Figure 11 shows the geometry for the numerical solution in cylindrical coordinates. The element size is approximately 0.05 cm. Region 1 is a dielectric with $\epsilon_r = 1.0$ that fills the solution volume and Region 2 is a set of fixed-potential nodes on the boundary with $\phi = 0.0$

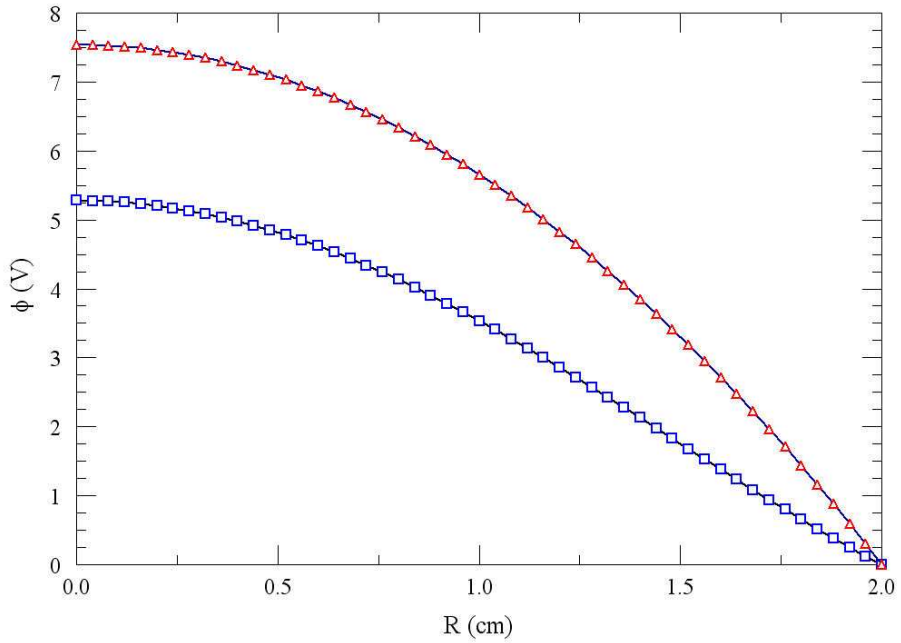

Figure 12: Variation of potential with R for the `CHARGEFUNC` example. Red: uniform space-charge density. Blue: non-uniform space-charge density.

V. To create a solution with uniform charge density $\rho_0 = 1.0 \times 10^{-6}$ C/m^3, the **EStat** script contains the following entry:

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

The red curve in Fig. 12 shows the calculated radial variation of potential. The potential at the center is $\phi_0 = 7.541$ V, within 0.16% of the theoretical value. To introduce the variable space-charge density of Eq. 24, we simply change the above line to

```
Rho(1) > 1.0E-6*(1.0 - 0.25*($r^2+$z^2))
```

The resulting variation of ϕ is plotted as the blue curve in Fig. 12. The calculated central potential is $\phi_0 = 5.280$ V, within 0.17% of the theoretical value.

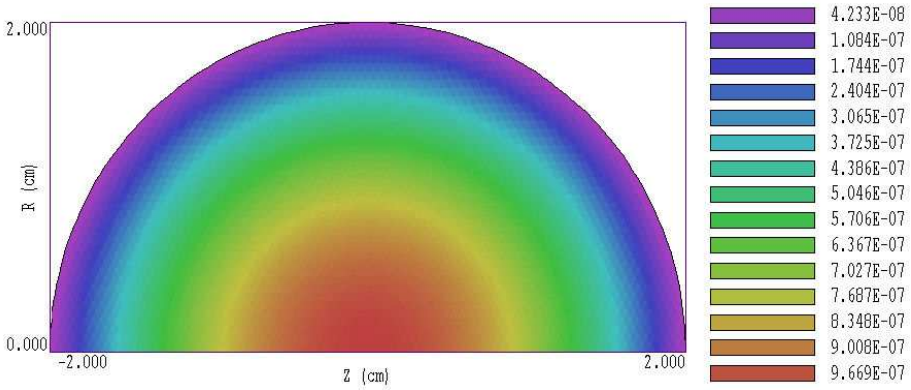

Figure 13: Element plot of $\rho(r, z)$ for the `CHARGEFUNC` example.

You can use the plot and analysis functions of **EStat** to check the validity of function expessions. The program can display spatial variations of ϕ, ϵ_r, σ and ρ. Figure 13 is an element plot of $\rho(z, r)$. The plot confirms that $\rho(r, z) = 1.0 \times 10^{-6}$ at $(z = 0.0, r = 0.0)$ and follows a parabolic variation with R.

5 TriComp analysis functions

To create plots and to perform numerical calculations, click the *Analyze* command in the main menu and choose a solution file. The *Analysis* menu contains the following main entries: *File*, *Plots*, *Analysis*, *Scans*, *Export* and *Return*. The commands of the *Export* menu (which generate hardcopy output and plot files) are similar to those in **Mesh**. The *Return* command restores the main menu where you can run additional solutions. This chapter gives a general description of post-processing capabilities for all **TriComp** programs. The following chapter covers specific plot quantities and analysis capabilities of **EStat**.

5.1 File menu commands

LOAD SOLUTION FILE

Read a different solution file for analysis without returning to the main menu. Pick a new file in the dialog. Changing the directory in the dialog changes the program working directory.

LOAD CONFIGURATION FILE

A configuration file is a text file of information that controls operation in the *Analysis* menu. The file defines quantities for plots, interpolations, volume integrals and surface integrals. The program loads a default configuration the first time you run it. This file contains a broad spectrum of useful quantities for the solution type. Some programs may automatically switch between two default configurations, depending on the solution type (*e.g.*, dielectric or conductive solutions in **EStat**). The default quantities are sufficient for most applications. Use this command if you want to load a specialized configuration or one that you have prepared. Chapter 7 gives detailed information on the format of configuration files and how to prepare them.

LOAD FROM SERIES

Initial-value programs like **TDiff** may produce several solutions files in a run at different simulation times. Use this command to load other files in a series of solutions. The program displays the dialog of Fig. 14 which lists the full set of output files created by the run along with the simulation times. Pick the desired file and click *OK* to load the solution. Note that this command does not appear in the menus of boundary-value programs like **EStat**.

OPEN DATA RECORD

Commands such as *Point calculation* and *Line scan* generate quantitative information. You can automatically record the data generated during an analysis session by opening a data file. Pick a file in the dialog or accept the default. The text-format data file will be stored in the working directory. You can use an editor to view the file or to extract information to send to mathematical analysis programs or spreadsheets. The suggested suffix for data records is `DAT`.

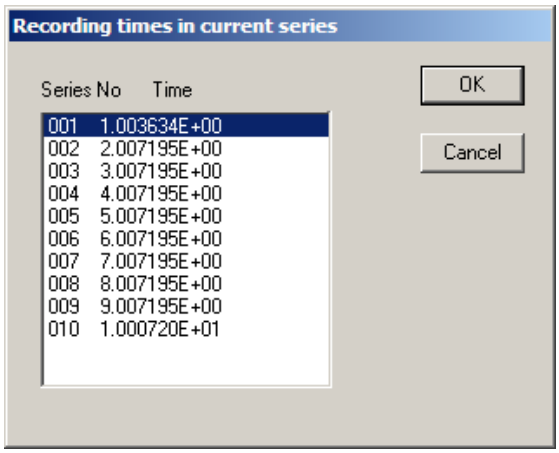

Figure 14: Dialog to pick a solution in a series generated by an initial-value program.

CLOSE DATA RECORD

Close the current data file. Use this command if you want to start a new file. You must close the data file before opening it with the internal editor.

RUN SCRIPT

A analysis script allows you to perform complex or repetitive calculations on a set of similar solutions. This command displays a dialog listing files with the suffix `SCR`. Pick a file and click *OK*. The script can load data files, open and close data records, and perform any of the quantitative analysis functions described in this chapter. The script command language is described in Sect. 5.7. Note that the analysis script must be in the same directory as the data files.

CREATE SCRIPT

Use this command to create an analysis script with the internal text editor. Supply a file prefix *SPrefix* in the dialog – the resulting script will be saved with the name `SPrefix.SCR`. The program opens the file in the editor and writes a reference list of allowed commands. The list follows the *EndFile* command and will be ignored by the script parser. Enter commands above the *EndFile* command.

EDIT FILES

Use this command to view or to modify an existing file with the internal program editor. Use the dialog to choose a text file. Changing directories in the dialog does not change the working directory of the program. You must exit the editor to resume normal operation of the program.

EXIT

Exit the program.

5.2 Plot menu commands

Spatial plots show variations of quantities over the two-dimensional space of the simulation. The following plot types are available:

- **Mesh**. Element facets of the computational mesh.
- **Region**. Computational mesh with elements color-coded by region number.
- **Filled contour**. Discrete bands of color coded according to values of the current plot quantity.
- **Contour lines**. Lines that follow constant values of the current plot quantity.
- **Element**. Elements of the solution space color-coded by values of the current plot quantity.
- **Surface**. A three-dimensional plot representing the current quantity as height over a region in the x-y or z-r plane. The spatial limits of the plot correspond to the current view window for *Mesh*, *Region*, *Filled contour*, *Contour* or *Element* plots.

The *Settings* popup menu contains the following commands.

PLOT TYPE

Change the plot type.

PLOT QUANTITY

A dialog shows a list of the available quantities defined in the `INTERPOLATION` section of the current configuration file.

PLOT LIMITS

In the default *Autoscale* mode the program picks limits in *Filled contour*, *Contour lines*, *Element* and *Surface* plots that span the full range of the current quantity. With this command you can set specific limits. In the dialog uncheck the *Autoscale* box and supply minimum and maximum values. Note that the program does not check the consistency of the values. Check the box to return to *Autoscale* mode.

TOGGLE GRID DISPLAY

Use this command to activate or to suppress the display of grid lines in *Mesh*, *Region*, *Contour lines*, *Filled contour* and *Element* plots. Grid lines corresponding to the axes ($x = 0.0$ or $y = 0.0$) are plotted as solid lines.

GRID CONTROL

This command displays the dialog of Fig. 15 to set properties of the grid. In the default *Autoscale* mode, the program automatically chooses intervals and positions so that lines occur at convenient values of x or y (for example, 0.01 rather than 0.01153). The grid intervals change as the view is zoomed. To set the grids manually, uncheck the *Autoscale* box and enter values for the intervals in x and y.

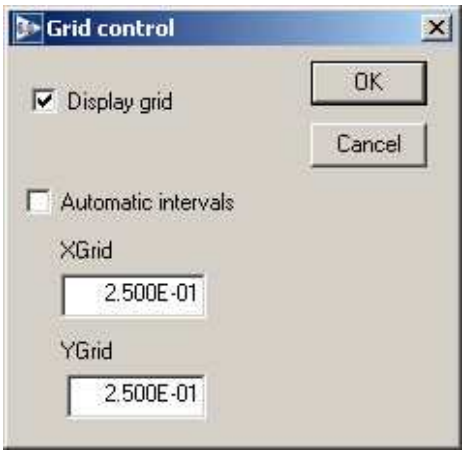

Figure 15: Grid control dialog

MOUSE/KEYBOARD

By default the program uses mouse entry of coordinates for commands like *Line scan* and *Zoom*. This command toggles between mouse and keyboard input. Enter keyboard coordinates in the distance units used in **Mesh**. In other words, if the solution program has $DUnit = 1.0 \times 10^6$, then enter dimensions in microns. You can temporarily switch to keyboard entry when entering coordinates with the mouse by pressing the *F1* key.

TOGGLE SNAP MODE

When snap mode is active, the mouse returns the coordinate values closest to an integer multiple of the quantity *DSnap*. In other words, if $DSnap = 0.5$ and the mouse position is (5.4331, -2.6253), the returned coordinates are (5.500, -2.500). Exercise caution when using commands like *Point calculation*. If snap mode is active, the calculation is performed at the snap point rather than the current location of the mouse cursor. You can also toggle the snap mode by pressing the *F2* key during coordinate entry.

SNAP DISTANCE

Set the distance scale *DSnap* for the mouse snap mode.

TOGGLE ELEMENT OUTLINE

This command determines whether element facets are displayed.

TOGGLE FIXED POINT DISPLAY

In the default mode, the programs generate contour line, filled contour and element plots from element information. Therefore, isolated nodes (representing structures like fixed-potential grids or sheets) do not appear. In response to this command, the program plots circles at fixed-potential nodes that are surrounded by material elements.

CONTOUR STYLE

This command is active only when the current plot type is *Contour lines*. There are four

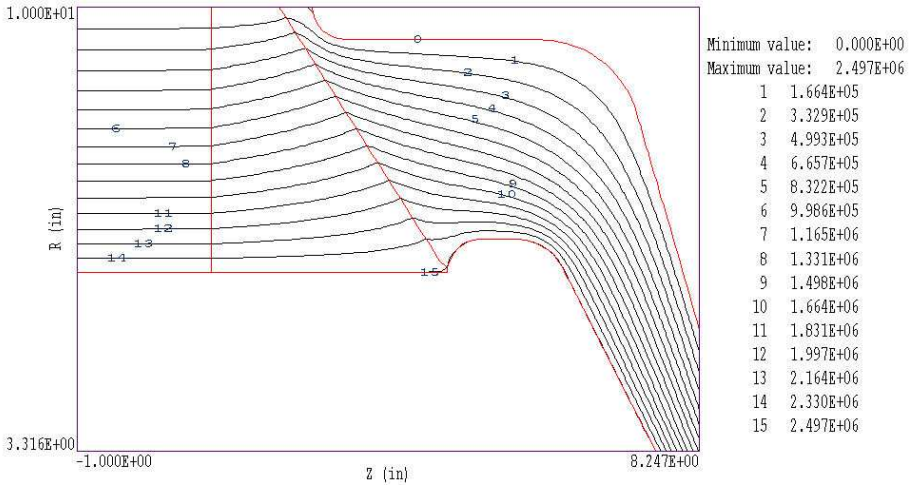

Figure 16: Monochrome contour plot with labels

choices: monochrome, monochrome with labels, colored and colored with labels. In the colored mode, the lines are color-coded according to the value of the plotted quantity. A legend is included in the information window to the right of the plot. In the labeled modes, contour lines are numbered according to their values (Fig. 16). Overlapping labels on closely-spaced lines may look better in a zoomed view.

NUMBER OF CONTOURS

Change the number of plotted contour lines. This command is active only when the current plot type is *Filled contour* or *Contour lines*.

TOGGLE NUMBER FORMAT

The program automatically chooses real-number notation for axis labels when the range of dimensions is close to unity. Use this command to switch to scientific notation.

The following commands (described in the **Mesh** manual), change the view limits in *Mesh*, *Region*, *Filled contour*, *Contour line* and *Element* plots. The current view limits of the two-dimensional plots are used when creating three-dimensional *Surface* plots.

ZOOM WINDOW
ZOOM IN
EXPAND VIEW
GLOBAL VIEW
PAN

The following commands control the appearance of *Surface* plots. The commands are active only when a *Surface* plot is displayed.

ROTATE 3D IMAGE

Rotate the *Surface* plot by 90° in the spatial plane.

VIEW ANGLE 3D
Set the elevation angle for the view point.

SET GRID 3D
Change the resolution of the surface plot. To create a *Surface* plot, a quantity is mapped to a rectangular grid with dimensions $N_x \times N_y$. The numbers also determine the total number of grid lines in the *Surface* plot. The default values are $N_x = N_y = 75$.

5.3 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 zoom limits, plotted quantity and plot type (spatial or scan) are saved. 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 either a spatial or scan plot.

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

```
Program: TriComp
PlotStyle: Spatial
Outline: ON
Grid: ON
Scientific: OFF
FixedPoint: OFF
Vectors: OFF
XGMin: -1.000000E+00
XGMax:  1.000000E+01
YGMin:  0.000000E+00
YGMax:  1.000000E+01
PlotType: Elem
NPQuant:    1
...
```

If a specific solution file is loaded, the plot will be restored exactly. The saved view feature has two useful features if a different mesh is 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 plot type or quantity) are applicable to any solution, but others (like zoom limits) depend on the geometry. The program 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, the programs use 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.

5.4 Analysis menu commands

The commands in the *Analysis* popup menu generate numerical data. Most of the functions require coordinate input from the user, usually through the mouse. Therefore, the analysis menu is active only when a *Mesh*, *Region*, *Contour*, *Element* or *Vector* plot is displayed.

POINT CALCULATION

Click on the command and move the mouse cursor to a position inside the solution volume. Current coordinates are displayed at the bottom of the screen. Press *F2* if snap mode is active and you want to specify an intermediate point. Press *F1* to type in the coordinates. The program writes a subset of quantities to the information area below the plot. If a data file is open, the program also records complete information on all quantities defined in the *Interpolation* section of the configuration file.

LINE SCAN

After clicking on the command, supply two points with the mouse or keyboard to define a scan line. The snap mode is useful in this application (for example, you may want a 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 at equal intervals along the line. The program makes a screen plot of the currently-selected quantity versus distance along the scan and activates the commands in the *Scan plot* menu (Sect.5.5). The information is recorded if a data file is open. The default is to include all computed quantities. Use the *Set recorded quantities* command to limit the information.

VOLUME INTEGRALS

Determine volume integrals of quantities defined in the `VOLUME` section of the configuration file and record the results. The program will prompt if a data file is not currently open. The program first records the global and regional volumes (or areas for planar solutions) and then integrals of defined quantities organized by region.

SURFACE INTEGRALS

Calculate surface integrals of quantities defined in the `SURFACE` section of the configuration

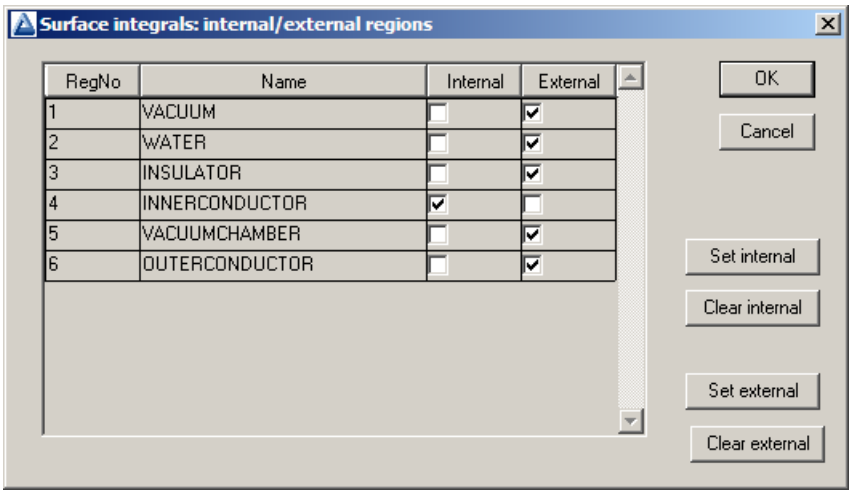

Figure 17: Surface integral dialog

file over region boundaries. In response to the command, the program displays the dialog of Fig. 17. Pick one or more regions as internal regions and bounding regions as external. The program computes the integral over the surfaces of the internal region(s) in contact with the external regions.

LINE INTEGRALS

Take integrals of vector quantities specified in the `SURFACE` section of the current configuration file over a line in the solution volume. Specify the line with the mouse or keyboard as in the *Line scan* command. The program reports both parallel and normal integrals in the data file (*i.e.*, the vector quantity is taken both parallel and normal to the differential line segments).

MATRIX FILE

The program can write matrix files of field values to help you create your own analysis routines. Although information is available in the output file of the solution program, it may be difficult to deal with the conformal triangular mesh. The *Matrix file* command uses the interpolation capabilities of the program to create a text data file of field quantities on a rectangular grid in x-y or z-r. The command displays the dialog of Fig. 18. Here you can set the matrix file name, the dimensions of the box and the number of intervals along x and y (or z and r). The program creates the file in the current directory. The default is to include all computed quantities. Use the *Set recorded quantities* command to limit the information.

The *Analysis settings* popup menu contains the following entries.

INTERPOLATION METHOD

The default interpolation method for plots and analysis commands such as *Point calculation* and *Line scan* is a second-order least-squares fit with a selective choice of data points. For example, only points on the side of a material boundary that contains the target point are included to give the correct field discontinuity. The least-squares fit may fail in very small

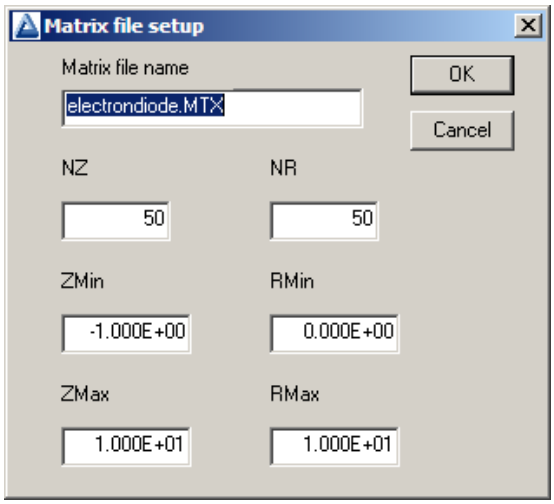

Figure 18: Dialog to generate a matrix file.

regions or enclosed areas if the program cannot identify enough data points. In this case, toggle to the linear mode. Here, field values are determined by a first order fit in the element that contains the target point. The status bar reports the current interpolation type.

SCAN PLOT QUANTITY

With this command you can choose the quantity to display in the screen plots of line scans. Pick the quantity from the list box and click *OK*. All quantities defined in the `INTERPOLATION` section of the current configuration file are available. This setting has no effect on the data file records which may include all field quantities.

NUMBER OF SCAN POINTS

This command sets the number of line scan points in plots and data file records. The default is 50 and the maximum number is 500.

SET RECORDED QUANTITIES

The default for data records of line scans and matrix files is to include all quantities defined in the `INTERPOLATION` section of the current configuration file. This state may result in large files with unnecessary information. Use this command to set the quantities that will be included. In the dialog of Fig. 19, activate the check box to include a quantity. There are a few rules:

- Quantity selection is available when the program runs interactively in a window. It does not apply when the program is run in the background. In this case, you can limit recorded quantities by editing the `INTERPOLATION` section of a custom configuration file.
- In the interactive mode, quantity selection applies to direct calls to the *Line scan* or *Matrix file* commands and indirect calls from an analysis script.
- All quantities are active when the program starts.
- The state of active quantities is preserved when a new data file is loaded.

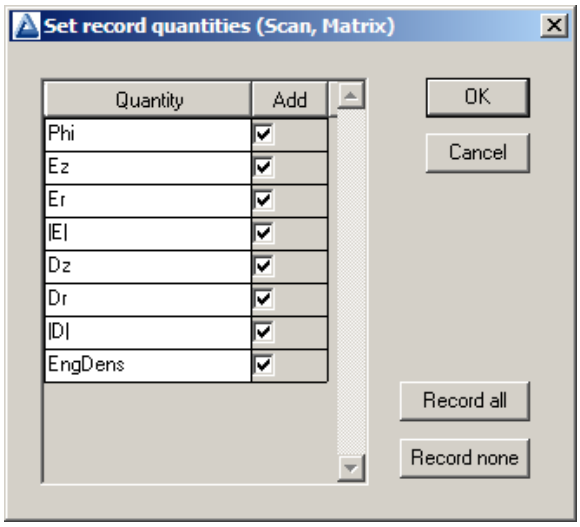

Figure 19: Dialog to set quantities recorded in response to the *Line scan* and *Matrix commands*.

5.5 Scan plot menu

The commands of the *Scan* menu become active following the *Line scan* command.

OSCILLOSCOPE MODE

In the oscilloscope mode, a scan plot assumes characteristics of a digital oscilloscope (Fig. 20). The program superimposes a cross-hair pattern on the graph. Plot values at the intersection are displayed in the information window. Move the marker along the plot by moving the mouse. If you click the left mouse button at a point, the program displays the plot values along with the numerical derivative and integral of the curve. The definite integral is taken from the left-hand side of the plot to the current point. Values are displayed on the screen and written to the data file if open. Press the right mouse button to exit the oscilloscope mode.

TOGGLE SCAN SYMBOLS

The setting determines whether plot symbols are added to the scan plot showing calculated points.

TOGGLE GRID

The setting determines whether grid lines are added to the screen and hardcopy scan plots.

CLOSE SCAN PLOT

Close the scan plot plot and return to the *Analysis* menu.

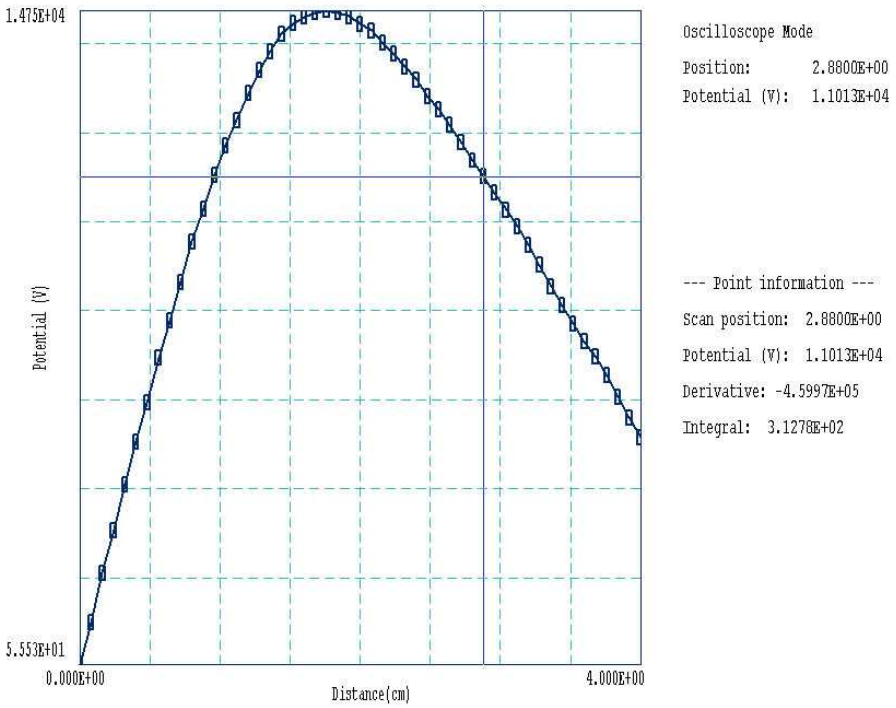

Figure 20: Scan plot in the oscilloscope mode

5.6 Vector tools

The analysis menu includes useful tools to display directional quantities in the VECTOR section of the current configuration file. Figure 21 shows the corresponding entries on the toolbar.

PROBE

This feature was inspired by the familiar Magnaprobe illustrated in Fig. 21. When you click on the tool and move the mouse cursor into the plot area, it changes to a semi-transparent probe that rotates about a pivot point to show the local direction of the vector quantity. The status bar shows the coordinates and the magnitude of the quantity.

FIELD LINES AT POINTS

Use this command to add lines of the vector quantity to any two-dimensional plot. The program enters coordinate mode when you click the command. Move the mouse to a point in the solution volume and click the left button. The program calculates and plots the path of the vector line that passes through the point. You may continue to add any number of lines. Click the right mouse button or press *Esc* to exit coordinate mode.

VECTOR SCATTER PLOT

Superimpose a uniform distribution of vector arrows pointing in the direction of the vector quantity. Vectors may be added to any of the two-dimensional plot types, including *Region*.

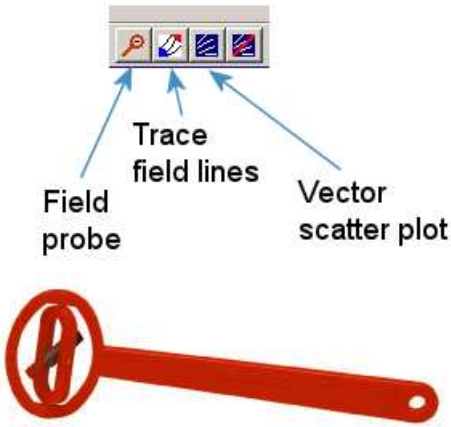

Figure 21: Tools to display vector information in the slice plot menu.

REMOVE VECTORS

Use this command to turn off the vector display. The program removes vector arrows and calculated field lines from the current plot.

PICK VECTOR QUANTITY

Choose a quantity defined in the `VECTOR` section of the current configuration file for display with the vector tools.

5.7 Analysis script commands

Scripts to control analysis sessions have a name of the form `FPREFIX.SCR`. They should be in the same directory as the data files. Scripts are text files that follow the **TriComp** syntax conventions. The program ignores blank lines and indentations. Data lines use the standard delimiters and comment lines begin with an asterisk [*]. Processing ends when the *EndFile* command is encountered.

To run a script, choose the *Run script* command in the *File* menu. The program shows a list of available scripts. Pick a file and click *OK*. The script operates on the presently-loaded data file or you can load other files from within the script. You may also sequentially open one or more data output files.

The program can perform analyses autonomously under script file control from the command prompt. If the file `GTEST.SCR` is in the data directory, then use a command of the form:

```
[ProgPath]\ProgName GTEST.SCR <Enter>
```

In the command prompt mode, the analysis functions of the program may be invoked from batch files, Perl or Python scripts or you own programs. The main application of the command prompt mode is to generate data files and to perform extended analyses under batch file control.

The following commands may appear in a script:

INPUT FileName
INPUT Switch1.EOU
Close the current data file and load a file for analysis. The parameter is the full name of the data file. You can load several files for sequential analysis.

OUTPUT FileName
OUTPUT SW02.DAT
Close the current data file and open an output file `FileName`.

CONFIGURATION [datapath\] FileName
CONFIGURATION = C:\FieldP\TriComp\ESTAT_FORCE.CFG
Load a new 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 during the last run.

SCAN Xs Ys Xe Ye
SCAN Zs Rs Ze Re
SCAN = (0.00, 0.00) (10.00, 0.00)
Write the results of a line scan between the specified points to the data file. The four real number parameters are the starting and end coordinates in **Mesh** units.

GENSCAN
Perform any number of calculations along an arbitrary path in space. At each point, write values of 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
  xp2 yp2
  ...
  xpn ypn
END
```

Each data line contains two real numbers separated by spaces to define a point in the solution volume (x-y or z-r). Enter coordinates in **Mesh** units.

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 current data file. If a region number does not appear, integrals are taken over all regions in the solution volume. Otherwise, the integral extends over elements with region number *NReg*. If quantities in the `VOLUME` section of the configuration file have units of C/m^3, then the output quantities will have units of C for cylindrical solutions and C/m for planar calculations.

SURFACEINT Reg1 Reg2 ... RegN
SURFACEINT 5 7 -9 -12
Perform a surface integral of vector quantities defined in the `SURFACE` section of the configuration file and write the results to the data file. Positive integer values define the set of *Internal* regions and negative values define the *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 the program does not include facets on the boundary of the solution volume in surface integrals.

MATRIX FName Nx Ny Xs Ys Xe Ye
MATRIX FPrefix Nz Nr Zs Rs Ze Re
MATRIX = Switch1 (10, 20) (0.00, 0.00, 5.00, 10.00)
Open a matrix file and record values. The command requires seven parameters: 1) The name of the matrix file (string), 2) the number of intervals along the x or z direction (integer). 3) the number of intervals along the y or r direction (integer), 4-7) coordinates of the corners of a box in the solution volume (real). The suggested suffix for matrix files is `MTX`.

NSCAN NScan
NSCAN = 150
Set the number of points in a line scan. The default is 50 and the maximum number is 500.

INTEPOLATION [LSQ,LINEAR]
INTERPOLATION = Linear
Set the interpolation method for subsequent *Point*, *Line scan* and *Matrix* commands. The options are *LSQ* (least-squares fit) and *Linear*.

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. 5.3). 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.*, 1024×768).

ENDFILE

Terminate execution of the script. You may add descriptive text in any format after this command.

The following is an example of a script to compare field values along the axes of four different solutions and to write the results to the file `COMP.DAT`.

```
NSCAN 200
OUTPUT COMP.DAT
INPUT SWITCH01.EOU
SCAN 0.00 -50.00 0.00 50.00
INPUT SWITCH02.EOU
SCAN 0.00 -50.00 0.00 50.00
INPUT SWITCH03.EOU
SCAN 0.00 -50.00 0.00 50.00
INPUT SWITCH04.EOU
SCAN 0.00 -50.00 0.00 50.00
ENDFILE
```

Table 6: Quantities for dielectric solutions (`estat_dielectric.cfg`).

Section	Quantity	Units
Interpolation	ϕ (electrostatic potential)	V
	E_x or E_z (electric field)	V/m
	E_y or E_r (electric field)	V/m
	$\|\mathbf{E}\|$ (electric field magnitude)	V/m
	D_x or D_z (electric displacement)	C/m^3
	D_y or D_r (electric displacement)	C/m^3
	$\|\mathbf{D}\|$ (electric displacement magnitude)	C/m^3
	u (field energy density)	J/m^3
Vector	**E** (electric field)	V/m
	D (electric displacement)	coulomb/m^3
Volume	U (field energy density)	J or J/m
	Q (charge)	C or C/m
Surface	Q (charge)	C or C/m

6 Standard EStat calculated quantities

The **EStat** package includes two standard configuration files for the analysis of dielectric solutions (`estat_dielectric.cfg`) and conductive solutions (`estat_conductive.cfg`). The program automatically switches to match the loaded solution if one of the standard files is currently active. For example, if `estat_conductive.cfg` is the current configuration and a dielectric solution is loaded, **EStat** automatically loads the dielectric configuration. Table 6 shows quantities defined in the file `estat_dielectric.cfg` for plots and scan, vector tools, volume integrals and surface integrals. Table 7 lists quantities for conductive solutions (`estat_conductive.cfg`). The **EStat** package also includes a custom configuration, `estat_force.cfg`, for calculating electrostatic forces and torques on assemblies of regions by volume and surface integral methods. The volume method applies only to regions with non-zero space-charge density (ρ). The surface integral method can be applied to dielectric and fixed-potential regions. The theory of force calculations using the Maxwell stress tensor is reviewed in the tutorials:

`http://www.fieldp.com/documents/stresstensor.pdf`
`http://www.fieldp.com/documents/forceandtorque.pdf`

6.1 EStat variables

EStat supplies values of the following field quantities at the current location for use in expressions of the `INTERPOLATION`, `VECTOR`, `VOLUME` and `SURFACE` sections of the configuration file. Values are calculated using the current interpolation method (least-squares fit or linear).

Table 7: Quantities for conductive solutions (`estat_conductive.cfg`).

Section	**Quantity**	**Units**
Interpolation	ϕ (electrostatic potential)	V
	E_x or E_z (electric field)	V/m
	E_y or E_r (electric field)	V/m
	$\|\mathbf{E}\|$ (electric field magnitude)	V/m
	J_x or J_z (current density)	A/m^2
	J_y or J_r (current density)	A/m^2
	$\|\mathbf{J}\|$ (current density magnitude)	A/m^2
	p (power density)	W/m^3
Vector	$\mathbf{E}$ (electric field)	V/m
	$\mathbf{J}$ (current density)	A/m^2
Volume	P (power)	W or W/m
Surface	I (current)	A or A/m

- `&Phi`: the electrostatic potential in V.
- `&Exz`: the x (planar) or z (cylindrical) component of electric field in V/m.
- `&Eyr`: the y (planar) or r (cylindrical) component of electric field in V/m.
- `&EMag`: the magnitude of the electric field in V/m.
- `&Exz0`: the x or z component of the normalized electric displacement ($\mathbf{D}/\epsilon_0$ in V/m) in dielectric solutions. In conductive solutions, this variable equals the current density j_x or j_z in A/m^2.
- `&Eyr0`: the y or r component of the normalized electric displacement in V/m for dielectric solutions. In conductive solutions, this variable equals the current density j_y or j_r in A/m^2.
- `&EMag0`: The magnitude of the normalized electric displacement in V/m or the magnitude of current density in A/m^2.
- `&EpsSig1`: The relative dielectric constant (dielectric solutions) or the electrical conductivity (conductive solutions in S/m) for isotropic materials or the value along the primary axis for anisotropic materials.
- `&EpsSig2`: The relative dielectric constant (dielectric solutions) or the electrical conductivity (conductive solutions in S/m) along the normal axis for anisotropic materials. The value equals `$EpsSig1` for isotropic materials.
- `&Rho`: the space-charge density for dielectric solutions in coloumb/m^3.

The following region variables are available for use in expressions. The value corresponds to the region that contains the current location.

- `$RegEps1`: the constant value of relative dielectric constant or electrical conductivity (in S/m) along the primary axis of anisotropic materials
- `$RegRho`: the space-charge density of uniform regions in dielectric solutions (in coulombs/m^3).
- `$RegPot`: the voltage (in V) of fixed-potential regions.
- `$RegEps2`: the constant value of relative dielectric constant or electrical conductivity (in S/m) along the normal axis of anisotropic materials
- `$RegAng`: the angle (in degrees) of the primary axis (relative to the x or z axis) for anisotropic materials.
- `$RegSinAng`: the sine of the angle of the primary axis for anisotropic materials.
- `$RegCosAng`: the cosine of the angle of the primary axis.

EStat records one run parameter in solution files: $CondFlag. It assumes the value 0.0 for dielectric solutions and 1.0 for conductive solutions.

6.2 Dielectric analysis

The configuration file `estat_dielectric.cfg` (Table 8) is used to analyze dielectric-type electrostatic solutions. The single program parameter is the value of ϵ_0. Several quantities in the `INTERPOLATION` section are simply the special variables generated by **EStat** ($\phi, E_x, ...$). Note the command form for defining electric field components:

```
Ex/Ez = &Exz
```

The value is given by the variable `&Exz`. The name that will be displayed in plots has two forms separated by a slash. The first form is used if the solution has planar symmetry while the second form applies to cylindrical solutions.

The electric displacement components and magnitude are defined by

$$\mathbf{D} = \epsilon_0 \mathbf{E}_0. \tag{25}$$

The expression for the field energy density,

$$u = \frac{\mathbf{E} \cdot \mathbf{D}}{2}, \tag{26}$$

holds for anisotropic as well as isotropic materials. The vector tools can display either **E** or **D**. Note that the plots will differ only if the solution includes anisotropic materials. The volume integral of the field energy density gives the total energy of regions, a useful quantity to determine the capacitance of two-electrode systems. The volume integral of charge applies only to regions with $\rho \neq 0.0$. The surface integral of charge applies to any assembly of dielectric and fixed-potential regions surrounded by air ($\epsilon_r = 1.0$). Induced charge is a useful quantity for calculating mutual capacitance.

Table 8: Contents of the file `estat_dielectric.cfg`.

```
PROGPARAM
  $Epsi0 = 8.854187E-12
END
RUNPARAM
  $CondFlag = 0.0
END
REALTIMEPARAM
END
INTERPOLATION
  Phi = &Phi
  Ex/Ez = &Exz
  Ey/Er = &Eyr
  |E| = &EMag
  Dx/Dz = $Epsi0 &Exz0 *
  Dy/Dr = $Epsi0 &Eyr0 *
  |D| = $Epsi0 &EMag0 *
  EngDens = &Exz &Exz0 * &Eyr &Eyr0 * + 0.5 * $Epsi0 *
END
VECTOR
  EVect = &Exz;&Eyr
  DVect = &Exz0 $Epsi0 *;&Eyr0 $Epsi0 *
END
VOLUME
  Energy = &Exz &Exz0 * &Eyr &Eyr0 * + 0.5 * $Epsi0 *
  Charge = &Rho
END
SURFACE
  Charge = &Exz0 $Epsi0 *;&Eyr0 $Epsi0 *
END
ENDFILE
```

Table 9: Contents of the file `estat_conductive.cfg`.

```
PROGPARAM
END
RUNPARAM
  $CondFlag = 1.0
END
REALTIMEPARAM
END
INTERPOLATION
  Phi = &Phi
  Ex/Ez = &Exz
  Ey/Er = &Eyr
  |E| = &EMag
  Jx/Jz = &Exz0
  Jy/Jr = &Eyr0
  |J| = &Exz0 &Exz0 * &Eyr0 &Eyr0 * + @SQRT
  PDens =  &Exz &Exz0 * &Eyr &Eyr0 * +
END
VECTOR
  EVect = &Exz;&Eyr
  JVect = &Exz0;&Eyr0
END
VOLUME
  Power = &Exz &Exz0 * &Eyr &Eyr0 * +
END
SURFACE
  Current = &Exz0;&Eyr0
END
ENDFILE
```

6.3 Conductive analysis

The configuration file `estat_conductive.cfg` (Table 9) is used to analyze conductive-type electrostatic solutions. In the `INTERPOLATION` section, the definitions of potential and electric-field quantities are the same those for dielectric solutions. The difference is that the current density $\mathbf{j}$ (in A/m^2) replaces the electric displacement. The resistive power density (in W/m^3) is given as

$$p = \mathbf{j} \cdot \mathbf{E}. \tag{27}$$

The vector tools can display either $\mathbf{E}$ or $\mathbf{j}$. Vector directions will differ only if the solution includes anisotropic materials. The volume integral gives the total resistive power dissipation over regions. The surface integral gives the total current I flowing out out of an assembly of conductive and fixed-potential regions.

6.4 Special EStat analysis commands

The function of the *Analysis/Equiline tool* command is to create a file listing of an organized set of vectors that follows an equipotential surface. **EStat** adds the vector list to the currently-opened data file. If no file is open, the program prompts for a file prefix and opens the file `FPrefix.DAT`. The next dialog prompts for ϕ (the value for the equipotential line) and the integer parameter $NMSkip$. When you click *OK*, **EStat** finds the vectors, organizes them, and closes the data file if it was not previously open.

The methods and purpose of the *Equiline tool* are best illustrated with an example. Suppose we want to design a pulsed CO_2 laser that requires a stable gas discharge between a ground plane and a long electrode inside a grounded box. There are two conditions on the variation of electric field amplitude: 1) $|\mathbf{E}|$ should be approximately uniform over the space between the electrodes and 2) $|\mathbf{E}|$ should decrease monotonically along the surface of the electrode moving away from the symmetry axis. The second condition is necessary to avoid non-uniform discharges or sparks. A simple radius on the outer edge of the electrode gives an enhanced surface field. Although the Rogowski method [W. Rogowski and H. Rengier, Arch. Elekt. **16**, 73 (1926)] provides some guidance on electrode shape, it applies to electrodes of infinite width in free space. Therefore, we shall use a direct numerical approach.

Figure 22 shows results of a primary calculation (`UNIFIELD01`) where the generating electrode is a thin plate of width 8.0 cm at a height 1.5 cm above the ground plane. Because the system is symmetric about $x = 0.0$, we can limit the solution to the region $x \geq 0.0$ with a Neumann boundary on the left-hand side. Figure 22*a* shows equipotential lines while Fig. 22*b* shows contours of $|\mathbf{E}|$ for an electrode potential of 1.0 V. The shaded region is the zone where the electric field amplitude is approximately equal to that at the center of the gap. Consider the equipotential line that passes through the point (4.00, 0.95), marked by the red arrow in Fig. 22*b*. This line stays within the $|\mathbf{E}|$ contours to the left and moves to lower contours to the right. Therefore, the surface field along an electrode that follows the potential contour would satisfy the second condition.

Using the *Point calculation* command, we find that the potential at the point equals 0.504 V. We then use the *Equiline tool* to generate a set of vectors along the potential surface, saving the results in `UNIFIELD01.DAT`. The file contains two lists. The first contains the full set of 209 vectors calculated on a scale length comparable to the element size:

```
Full vector set along potential contour
   Potential:   5.0400E-01
================================================
      0.0000      3.1997      0.0002      3.1997
      0.0002      3.1997      0.1124      3.1995
      0.1124      3.1995      0.1128      3.1995
      0.1128      3.1995      0.2151      3.1990
      0.2151      3.1990      0.2161      3.1990
      0.2161      3.1990      0.3167      3.1983
      ...
```

The full vector list might be useful as input to a computer-controlled milling machine. In the second list, the number of vectors has been reduced by a factor of 3, the default setting for $NMSkip$:

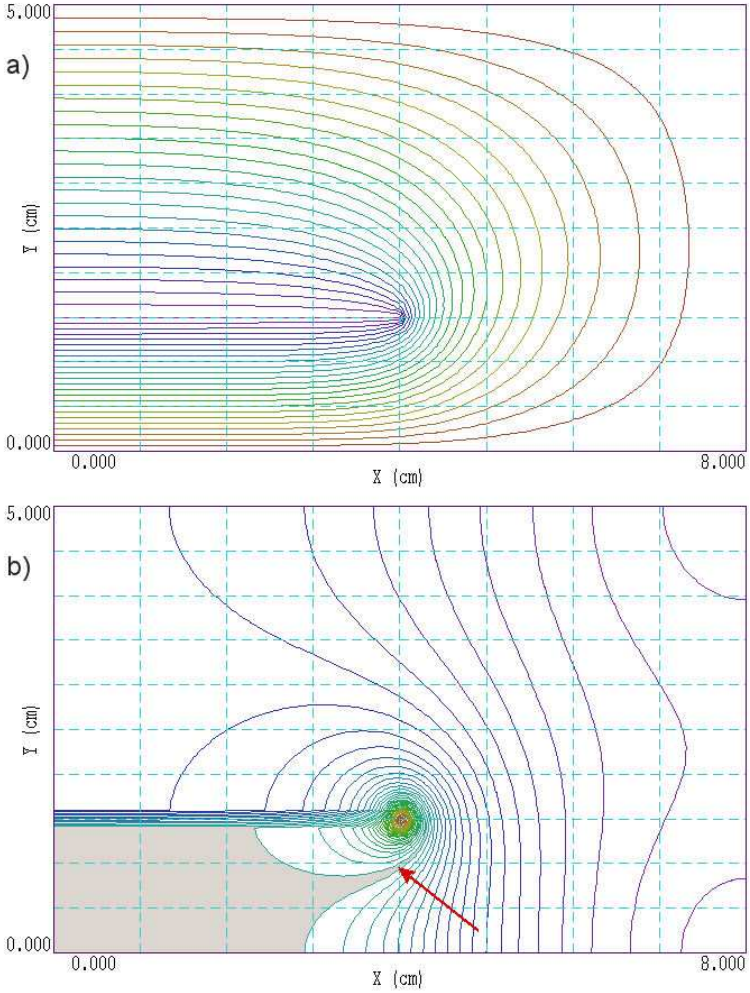

Figure 22: Initial solution, `UNIFIELD01`. *a*) Equipotential contours. *b*) Contours of $|\mathbf{E}|$.

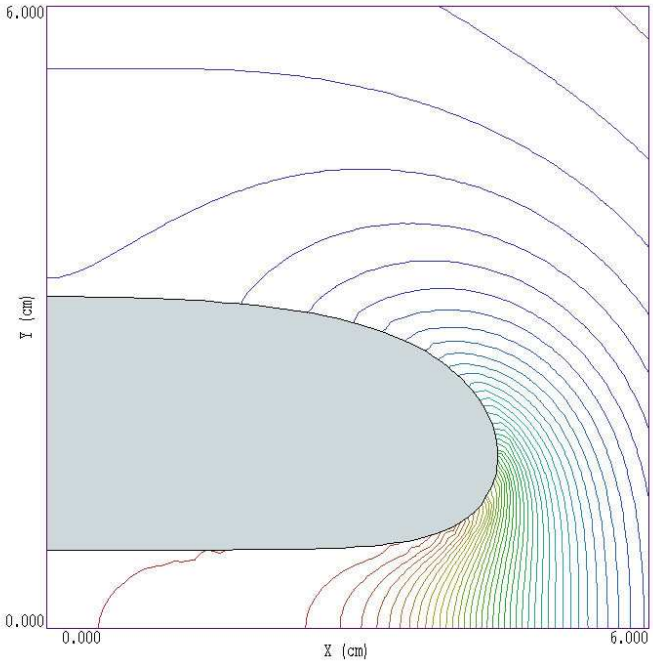

Figure 23: Secondary solution `UNIFIELD02`, showing lines of constant $|\mathbf{E}|$. Electrode has the shape of the 0.504 V equipotential line of `UNIFIELD01`.

```
Reduced vector set for Mesh along potential contour
   Potential:   5.0400E-01
=====================================================
  L         0.0000      3.1997      0.1128      3.1995
  L         0.1128      3.1995      0.3167      3.1983
  L         0.3167      3.1983      0.4206      3.1971
  L         0.4206      3.1971      0.6181      3.1940
  ...
```

The longer vectors are appropriate as input to **Mesh**, where line segments should be larger than the element size. Note that **EStat** has added the letter L at the beginning of each line so that the vector list can be incorporated directly into the boundary definition of a **Mesh** region. The listing was used to create the file `UNIFIELD02.MIN`. It was necessary to add an additional vector along the left-hand boundary to complete the filled region. Figure 23 shows a detailed view of $|\mathbf{E}|$ contours near the electrode. As expected, the field amplitude on the surface decreases moving away from the center line of the electrode. The design could be modified in two ways:

- The arbitrary choice of the generating electrode geometry leads to a gap of 0.7561 cm in the secondary solution. We could change the parameters of the generating electrode to achieve different gap or electrode widths.
- To reduce machining costs, we could replace the complex surface in the low-field region at the top with a simple flat.

EStat has powerful sorting features to arrange vectors in a continuous line. The procedure works well for the relatively simple shape of Fig. 23, but may encounter problems when lines

are complex (*i.e.*, saddle points) or when there are multiple disconnected lines. In this case, it may be necessary to edit the vector list.

The *Analysis/Region surface field* command generates a list of electric field values on the facets that constitute the surface of a specified region. One application is the determination of the maximum field on an electrode. The tool writes values to the data listing file and prompts to open a file if necessary. Supply the number of the target region. The program collects all facets of the region surface adjacent to a dielectric or conductive element. The facets are ordered using the same methods as the *Equiline tool*. For each facet, the program calculates the field at the facet center with a small displacement into the element of the adjacent region. Here is an example of the listing:

```
Ordered listing of |E| on the surface of Region   4
    Identified  146 facets
    Maximum field value of    4.6319E+07
    occurs at z =    9.4667E+00  r =    1.0014E+00
    on the boundary with Region No   1

       Z            R            D           |E|           Ez           Er      NBorder
  ================================================================================================
   9.4515E+00   0.0000E+00   0.0000E+00   3.9420E+07  -3.9420E+07   0.0000E+00     1
   9.4649E+00   2.0000E-01   0.0000E+00   3.9467E+07  -3.9467E+07   0.0000E+00     1
   9.4649E+00   2.0000E-01   2.9333E-07   3.9420E+07  -3.9420E+07   0.0000E+00     1
   9.4665E+00   4.0002E-01   0.0000E+00   3.9622E+07  -3.9622E+07   0.0000E+00     1
   9.4665E+00   4.0002E-01   1.8333E-08   3.9467E+07  -3.9467E+07   0.0000E+00     1
   9.4670E+00   6.0007E-01   0.0000E+00   4.0123E+07  -4.0123E+07   0.0000E+00     1
```

The quantity D is the total distance along the region surface from the start point. The quantity $NBorder$ is the region number of the adjacent element.

7 Building custom analysis configurations

You can create configuration files to customize analysis operations for your application. You can then switch between configurations for different types of solutions. To build a configuration, copy and rename one of the standard files supplied with the program to act as a template. You can put your new configuration file anywhere, but we suggest you store all configurations in the same directory as the executable programs (**c:\fieldp \tricomp**). Note that the program remembers the last configuration used and attempts to reload it at the next run. This chapter describes the format and function of configuration files and how you can define calculated quantities.

7.1 Configuration file structure

An analysis configuration is a text file with a name of the form `FPREFIX.CFG`. Table 7.1 shows the contents of `estat_dielectric.cfg`. The file includes seven sections with the following functions:

- `PROGPARAM`: define fixed numerical parameters to use in the expressions of the `INTERPOLATION`, `VECTOR`, `VOLUME` and `SURFACE` sections.
- `RUNPARAM`: numerical parameters passed by the program (such as the RF frequency in **REF2** or the dump time in TDiff). **Do not change this section**. Doing so could cause errors reading solution files.
- `REALTIMEPARAM`: numerical parameters used in the expressions of the `INTERPOLATION`, `VECTOR`, `VOLUME` and `SURFACE`. The difference from a `PROGPARAM` is that these values may be changed by the user while the program is running. An example is the reference phase in **RFE2**.
- `INTERPOLATION`: definition of scalar quantities for plots and scans.
- `VECTOR`: definition of vector quantities for use with the vector tools.
- `VOLUME`: definition of scalar quantities used for volume integrals.
- `SURFACE`: definition of vector quantities for surface integrals.

You may define custom quantities for display and analysis in the last four sections. In plotting, run speed is a critical issue. The generation of a single element 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). The program 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 7.3 covers parameters and variables that may appear in expressions.

Table 10: Configuration file `estat_dielectric.cfg`.

```
PROGPARAM
  $Epsi0 = 8.854187E-12
END
RUNPARAM
  $CondFlag = 0.0
END
REALTIMEPARAM
END
INTERPOLATION
  Phi = &Phi
  Ex/Ez = &Exz
  Ey/Er = &Eyr
  |E| = &EMag
  Dx/Dz = $Epsi0 &Exz0 *
  Dy/Dr = $Epsi0 &Eyr0 *
  |D| = $Epsi0 &EMag0 *
  EngDens = &Exz &Exz0 * &Eyr &Eyr0 * + 0.5 * $Epsi0 *
END
VECTOR
  EVect = &Exz;&Eyr
  DVect = &Exz0 $Epsi0 *;&Eyr0 $Epsi0 *
END
VOLUME
  Energy = &Exz &Exz0 * &Eyr &Eyr0 * + 0.5 * $Epsi0 *
  Charge = &Rho
END
SURFACE
  Charge = &Exz0 $Epsi0 *;&Eyr0 $Epsi0 *
END
ENDFILE
```

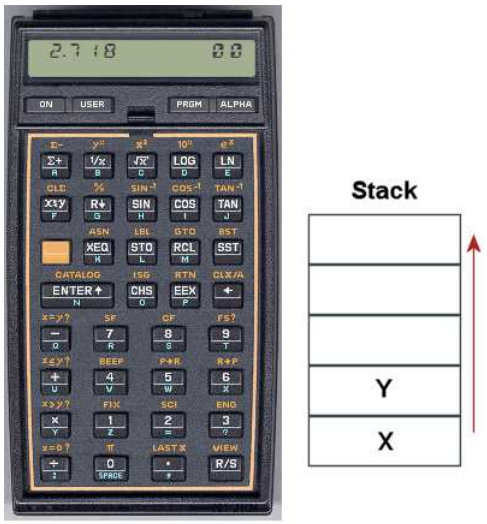

Figure 24:

7.2 Introduction to RPN

Consider the expression

$$5.0 \times 4.5^3 + \left[9.2 + 0.6 \times 0.9^{(2+0.67)}\right]. \tag{28}$$

Equation 28 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. 24. 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(+): $Y + X$

Table 11: 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`	$X = -X$	

- 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.

7.3 Operators, parameters and variables

Expressions to define quantities may contain numbers, unary and binary operations, parameters, standard variables and special variables for the program. Unary operators have names that begin with '@'. Table 11 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 `-$Epsi0` and `-&grady[3]` that mix a minus sign with a variable are invalid. Instead, use forms like `$Epsi0 @CHS`.

You may define up to 10 constants in the `PROGPARAM` section that can be used in your expressions. For example:

```
PROGPARAM
  $Epsi0 = 8.854187E-12
  $Epsi0Inv = 1.1294E11
END
```

A data line contains a parameter name, an equal sign and a value in any valid real-number formal. A parameter name must begin with a dollar sign ($) and may contain a maximum of 14 characters. Depending on the program, additional parameters may be defined in the `RUNPARAM` section:

```
RUNPARAM
  $CondFlag = 0.0
END
```

Here, a data line consists of a name, an equal sign and a default value. The actual value depends on the properties of the finite-element solution and is recorded in the output solution file. The parameter is set when a data file is loaded. Although you may use run variables in your expressions, the form of the `RUNPARAM` section should not be changed. Be sure that the form in a custom configuration is that same as that in the standard files.

The standard variables are

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

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

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

The special program variables give interpolated values of field quantities at the current location. As an example, the following quantities are available for use in expressions in **EStat**: `&Phi` (electrostatic potential), `&Exz` (the x or z component of electric field, `&Eyr` (the y or r component of electric field), `&EMag` (magnitude of the electric field), `&Exz0` (the x or z component of the normalized electric displacement, $\mathbf{D}/\epsilon_0$). `&Dyr` (the y or r component of the normalized electric displacement), `&EMag0` (the magnitude of the normalized electric displacement), `&EpsSig1` and `&EpsSig2` (the relative dielectric constant or the electrical conductivity), and `&Rho` (the space-charge density for dielectric solutions). In addition, region variables may be defined. These are quantities that have fixed values over a region. The current value corresponds to the region that contains the current location. As an example, EStat supports the following region variables:

`$RegEps1` (the relative dielectric constant or electric conductivity along the primary axis for anisotropic materials), `$RegRho` (the space-charge density for uniform materials in dielectric solutions), `$RegPot` (the voltage of a fixed-potential region), `$RegEps2` (the relative dielectric constant or electric conductivity along the normal axis for anisotropic materials), `$RegAng` (the angle of the primary axis relative to the x or z axis for anisotropic materials), `$RegSinAng` and `$RegCosAng` (the sine and cosine of the angle of primary axis).

7.4 Building expressions

You can add your own expressions to the `INTERPOLATION`, `VECTOR`, `VOLUME` and `SURFACE` sections of the configuration file, one expression per line. Lines for a scalar quantities in the `INTERPOLATION` and `VOLUME` sections have three components:

- A name for display in plots and listings.
- An equal sign (=).
- A valid RPN expression.

For a quantity that has the same name in planar and cylindrical solutions, the name is a string up to 14 characters in length. Alternatively, the name may have the form:

```
RectName/CylinName
```

Here, two strings (each up to 14 characters in length) are separated by a slash (/). The string *RectName* is used for labels when a planar solution has been loaded, and the string *CylinName* is used for cylindrical solutions. For example:

```
FVolX/FVolZ = $RegCurrent &Byr * $RegArea / @CHS
```

Expression lines in the `VECTOR` and `SURFACE` sections have the following form:

```
NAME = EXPRESSION01;EXPRESSION02
```

A line consists of a name, an equal sign and two valid RPN expressions separated by a semicolon. The first expression gives the x or z component of the vector, while the second expression gives the y or r component.

It is best to build a configuration one expression at a time. Modify and save the file, then use the *Load configuration file* command. The routine will report the location of the first syntax error encountered. If the configuration file loads successfully, test the new expression using plot and point calculation commands. At shutdown, the program stores the current configuration file name in the registry and attempts to load it at the next session. If the file is missing or has a syntax error, the program displays the *Load configuration* dialog so that you can pick an alternative. An error message is displayed if the alternate file is invalid.

Index