

Email: techinfo@fieldp.com

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

Contents

1	Introduction	4
1.1	Program functions	4
1.2	Learning Pulse	5
1.3	Finite-element solution procedure	6
1.4	Scripts and data files	8
1.5	Theoretical background	8
1.6	Vector potential and boundary conditions	10
1.7	Physical units	11
2	Creating and analyzing a solution with Pulse	12
2.1	Mesh generation	13
2.2	Creating the Pulse script	13
2.3	Finite-element solution	15
2.4	Solution analysis	17
3	Pulse solution reference	20
3.1	Organizing Pulse calculations	20
3.2	Pulse script format	21
3.3	Control commands	22
3.4	Picking a time step	23
3.5	Avoiding numerical instabilities	24
3.6	Region properties	26
3.7	Nonlinear materials	27
3.8	Time variations of current or vector potential	27
3.9	Diagnostic commands for dynamic solutions	30
3.10	Running Pulse	32
3.11	Format of the Pulse output files	34
4	TriComp analysis functions	37
4.1	File menu commands	37
4.2	Plot menu commands	39
4.3	User specified contours	42
4.4	Saving and loading views	43
4.5	Analysis menu commands	45
4.6	Scan plot menu	48
4.7	Vector tools	49
4.8	Analysis script commands	50
5	Standard Pulse calculated quantities	54
5.1	Pulse variables	54
6	Building custom analysis configurations	57
6.1	Configuration file structure	57
6.2	Introduction to RPN	59
6.3	Operators, parameters and variables	60

6.4	Building expressions	62
7	Probe – history file plot utility	63
7.1	Introduction	63
7.2	Loading data files	64
7.3	Plot settings	64
7.4	Plot functions	66
7.5	Information	68

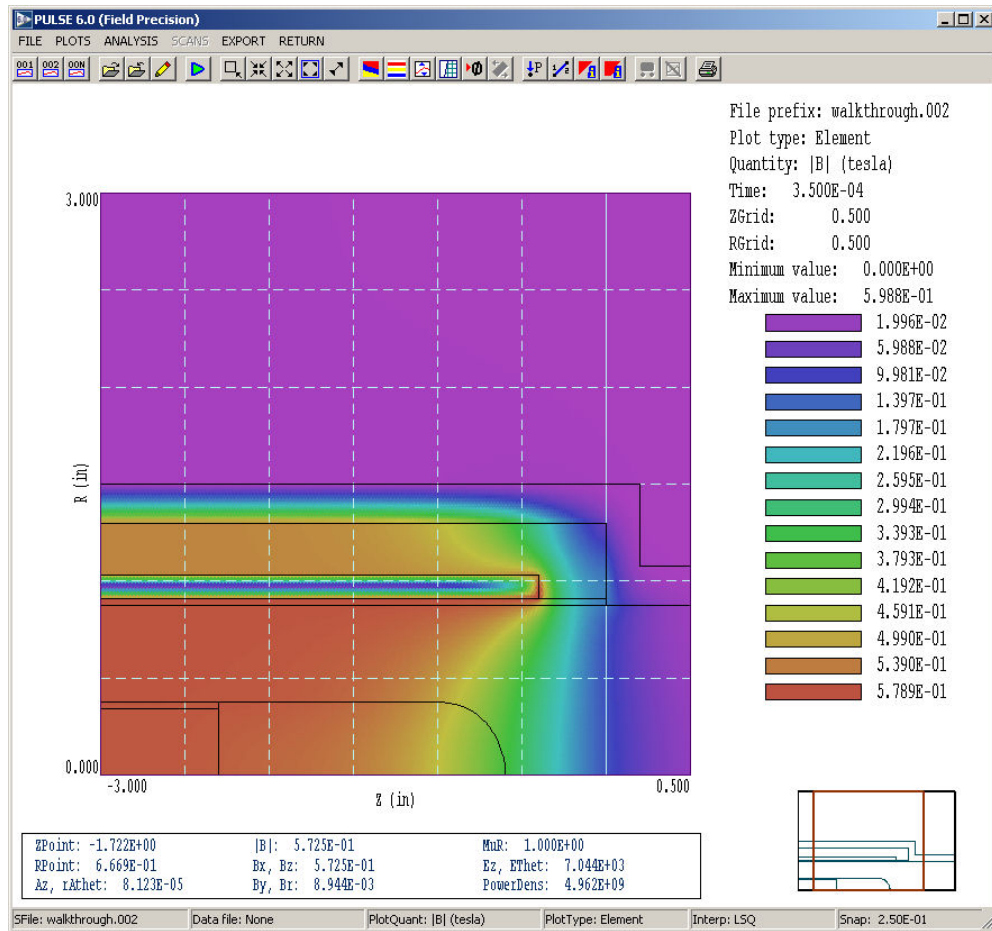


Figure 1: **Pulse** solution analysis working environment.

1 Introduction

1.1 Program functions

Pulse calculates time variations of magnetic fields in the presence of eddy currents. The program solves the two-dimensional magnetic diffusion equation in cylindrical and rectangular geometries. **Pulse** utilizes finite-element methods on variable-resolution conformal triangular meshes. The mesh size is limited only by the installed memory of the computer. You can define up to 250 regions to represent materials or coils. Multiple coils may be assigned arbitrary current waveforms using either mathematical functions or a flexible system of tabular function input. The program also handles non-linear magnetic materials with user-defined $\mu_r(|\mathbf{B}|)$ tables. **Pulse** produces multiple data files at specified times and history files of magnetic and induced electric field at given locations. Data files may be analyzed in an interactive graphical environment. The program provides full information on the fields and magnetic permeability. Other features include automatic calculation of stored energy, peak field values, magnetic forces and torques. The program creates a wide variety of screen and hardcopy plots (Fig. 1).

History files may be analyzed and plotted with the utility program **Probe**. **Pulse** has applications to transformers, magnetic recording devices, high-field pulsed magnets, linear induction accelerators, high-frequency inductors and magnetic coupling in microelectronic devices.

The intuitive graphical-user-interface makes it easy to learn **Pulse** and to perform quick application setups. In contrast to other field solution software, **Pulse** features 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.

1.2 Learning Pulse

The size of this manual reflects the extensive capabilities of the **Pulse** 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. The first step is to follow the walkthrough example in Chapter 2. This will give you an overview of the package organization and the solution process. Then, 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. Here are suggestions for additional activities:

- Browse the following sections in this chapter to review some basics of pulsed magnetic fields in conductive media. 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 **FPController** (the component 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 **Mesh** drawing editor. The exercise reviews fundamental tools you will need for your own simulations. Chapter 5 describes how to convert a script into a conformal mesh.
- As you gain experience you will want to take advantage of the full range of **Mesh/Pulse** 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 **Pulse** control scripts, including advanced commands for non-uniform and anisotropic materials. Chapter 4 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 5 reviews the calculated quantities defined by the standard **Pulse** configuration file. Finally, Chap. 6 describes how to build a customized postprocessor that includes your own calculated quantities.
- Additional chapters cover advanced techniques in **Mesh**. Chapters 6 and 7 of the **Mesh** manual show how to make direct entries in scripts to invoke advanced control features. Chapter 8 describes how to create meshes from photographic and data images. This

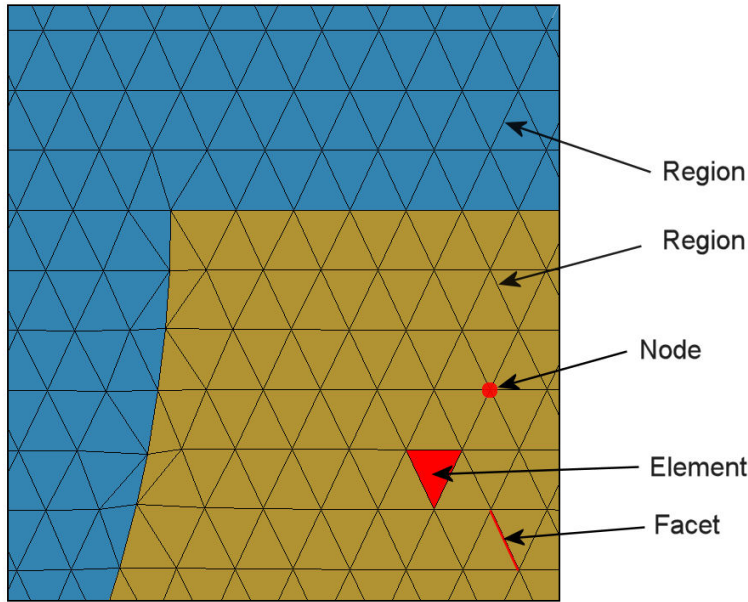


Figure 2: Conformal triangular mesh – definition of terms.

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 **Pulse**. 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 **Pulse** solution.

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 diffusion equation. These equations can be solved directly by analytic methods if the system geometry and material properties are simple (for example, a ferromagnetic cylinder inside a solenoid of infinite length). Analytic solutions are extremely difficult in systems with asymmetric structures or non-linear materials. Furthermore, closed-form results 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 2 shows the type of mesh used for the two dimensional solutions of **Pulse**. The figure defines three terms that will be used throughout this manual:

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

The elements in Fig. 2 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. 2 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 can 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 **Pulse**, solution of the set representing the magnetic diffusion equation gives the vector potential (A_z) or stream function (rA_θ) at each node. We can then perform two-dimensional interpolations to potential values at intervening points, or take numerical spatial derivatives to find the components of the magnetic and induced electric fields.

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

- 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.
- **Mesh** analyzes the boundary specifications and automatically generates a set of conformal triangles such as those of Fig. 2. The program creates an output file that lists the locations of nodes and the identities of elements.
- You define the material properties of regions in the solution volume. Usually, this task is performed through an interactive dialog in **Pulse**.
- **Pulse** reads the mesh geometry and applies the material parameters to generate the complex-value linear equation set.
- **Pulse** 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.

Table 1: **Pulse** file types

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)
PName.PIN	Pulse input script (run control and material properties)
PName.PLS	Pulse diagnostic listing
DName.POT	Data table, vector potential versus time
DName.CUR	Data table, drive current versus time
DName.MUR	Data table, magnetic permeability versus $ \mathbf{B} $
PName001.POU	Pulse data dump sequence
PName.P01	Pulse probe history
SName.SCR	Pulse script for automatic data analysis

- You can use the interactive graphical environment of **Pulse** 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 **Pulse** read and generate several types of files. For a typical solution it is not necessary for you 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 **Pulse** that require direct entries to the input scripts. For convenience, all input scripts and output data files are in text format. Both **Mesh** and **Pulse** feature integrated text editors. Table 1 lists the file types and functions in the **Pulse** package. Note that the suffixes indicate the file function.

1.5 Theoretical background

Although you can use **Pulse** without a detailed knowledge of magnetic diffusion, it is essential to understand some basics to run the code effectively. The program calculates magnetic field variations by solving the Maxwell equations in the limit that displacement currents (generated by changing electric fields) are small compared to the other currents in the problem. The currents treated by **Pulse** include specified currents in coils and the currents induced in surrounding conducting materials by inductive electric fields. Elimination of displacement currents is equivalent to neglecting the effects of electromagnetic radiation.

The reduced set of the Maxwell equations is

$$\nabla \times \left(\frac{\mathbf{B}}{\mu} \right) = \mathbf{j}_o + \mathbf{j}_r, \quad (1)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}. \quad (2)$$

Equation 1 is Ampere's law. The quantity μ is the local value of *magnetic permeability*. Including it in the equation automatically introduces the effects of atomic currents in materials. In the following discussions, we use the *relative magnetic permeability*, defined by $\mu_r = \mu/\mu_o$. In **Pulse** calculations, we assume that all magnetic materials are soft and isotropic, so that the magnetic permeability is a single-valued function of the magnitude of the local magnetic field, $\mu_r(|\mathbf{B}|)$. The non-atomic current density on the right-hand side of Eq. 1 is divided into two parts. The quantity \mathbf{j}_o represents the current density in coils. This current density is specified by the user and does not depend on the field solution. The quantity \mathbf{j}_r is the *eddy current density* induced in surrounding conductors by changing magnetic fields. If the conductors have an isotropic electrical conductivity σ , the eddy current density is related to the local value of electric field by

$$\mathbf{j}_r = \sigma \mathbf{E}. \quad (3)$$

Equation 2 is Faraday's law, giving the electric field in terms of the changing magnetic field. We assume that there is no space charge so that only inductive electric fields are present. The condition $\nabla \cdot \mathbf{B} = 0$ implies that the magnetic field can be written as the curl of a vector quantity, the vector potential,

$$\mathbf{B} = \nabla \times \mathbf{A}. \quad (4)$$

Combining Eqs. 2 and 4, we can show that the induced electric field is given by the time derivative of the vector potential

$$\mathbf{E} = -\frac{\partial \mathbf{A}}{\partial t}, \quad (5)$$

or

$$\mathbf{j}_r = -\sigma \frac{\partial \mathbf{A}}{\partial t}. \quad (6)$$

Substituting Eqs. 4 and 6 into Eq. 1 gives the magnetic diffusion equation

$$\sigma \mu_o \frac{\partial \mathbf{A}}{\partial t} = \nabla \cdot \left(\frac{1}{\mu_r} \nabla \mathbf{A} \right) + \mu_o \mathbf{j}_o. \quad (7)$$

The strategy is to solve Eq. 7 for the vector potential using appropriate boundary conditions and then to derive the magnetic fields from Eq. 4. The procedure is straightforward in two dimensional geometries where there is only one component of vector potential. Rectangular geometries have variations in x and y and are uniform in z . The conditions of symmetry imply that the applied currents, the vector potential, and the inductive electric field point in the z direction and that the magnetic field components B_x and B_y are non-zero. Cylindrical systems are symmetric in θ and have field components B_r and B_z . The current density, vector potential

and induced electric field are directed in θ . **Pulse** does not directly solve the differential form of Eq. 7. Instead, the finite-element code solves the integral form of Ampere's law applied to the triangular elements of the mesh. In this approximation, values of A_z and $\partial A_z/\partial t$ are defined at the nodes of the mesh triangles. The material properties σ and μ_r as well as the field components B_x and B_y are associated with the triangle areas. To apply the finite element method to cylindrical problems, **Pulse** uses the *stream function*, rA_θ , rather than the vector potential as the main solution variation. A complete review of the physical basis of **Pulse** is given in S. Humphries, **Field Solutions on Computers** (CRC Press, Boca Raton, 1997), Sect. 11.5.

1.6 Vector potential and boundary conditions

A boundary condition is the specification of the calculated field quantity on the outer edge of the solution region. To understand boundaries, it is important to remember that **Pulse** determines values for the magnetic vector potential at the vertex points. The routines of the Analysis menu then take spatial derivatives of these values to determine the magnetic fields. The boundary conditions therefore apply to the vector potential.

In rectangular problems (variation in x and y with no variation in z) **Pulse** determines the vector potential A_z created by currents that move in the z direction. The magnetic fields are given by

$$B_y = -\frac{\partial A_z}{\partial x}, \quad (8)$$

$$B_x = \frac{\partial A_z}{\partial y}. \quad (9)$$

Equations 8 and 9 imply that changes in values of the vector potential are related to the flux of magnetic field lines (the surface integral of magnetic field). For example, in a rectangular problem with boundaries at y_1 and y_2 , the difference in the boundary values of the vector potential at position x_0 is given by

$$A_z(x_0, y_2) - A_z(x_0, y_1) = \int_{y_1}^{y_2} B_x(y', x_0) dy'. \quad (10)$$

A similar equation holds for integrals in the x direction. Therefore, if we set A_z equal to a constant value (usually zero) around the complete solution boundary, all magnetic flux must be trapped inside the region because the integral of B_x or B_y over any plane is zero. This is equivalent to generating the magnetic field inside a perfectly conducting metal box. Another way to see this is to note that because $\mathbf{B} = \nabla \times \mathbf{A}$, the magnetic field must be parallel to a surface of constant \mathbf{A} .

For problems in cylindrical coordinates, **Pulse** calculates the quantity rA_θ , where A_θ is the vector potential created by azimuthal currents. In this case, the magnetic fields are given by

$$B_r = -\frac{1}{r} \frac{\partial(rA_\theta)}{\partial z}, \quad (11)$$

$$B_z = \frac{1}{r} \frac{\partial(rA_\theta)}{\partial r}. \quad (12)$$

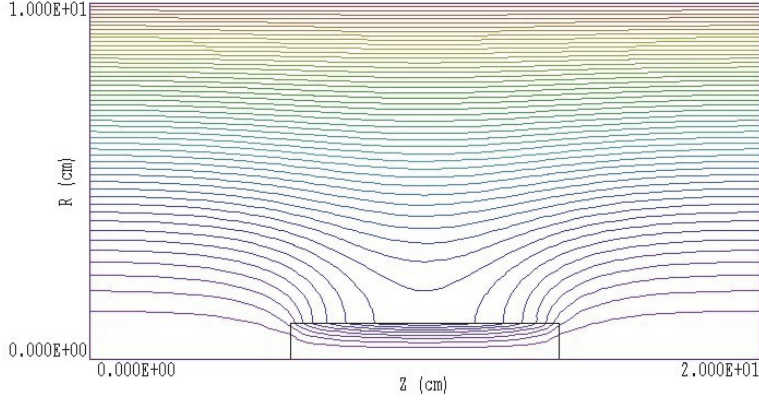


Figure 3: Setting boundary values of the vector potential to create a region of uniform flux.

The symmetry of magnetic field lines dictates that the axis ($r = 0$) must always be a line with $rA_\theta = 0.0$. **Pulse** automatically sets the condition for cylindrical problems that include the axis by adding a constant potential line region on the axis. The value of rA_θ on the outer boundary at the outer solution radius is related to the flux of axial field by

$$(rA_\theta)_{r_0} = \frac{1}{2\pi} \int_0^{r_0} /2\pi r' dr' B_z(r', z) = \frac{Flux}{2\pi}. \quad (13)$$

As an example, Fig. 3 shows the magnetic field lines around an iron cylinder immersed in a uniform solenoid field of 0.1 tesla. The uniform field was created by setting $rA_\theta = 5.0 \times 10^{-4}$ tesla-m² on the outer boundary at $r_0 = 0.10$ m.

With this background, we can now discuss the implications of the two types of boundary conditions used in **Pulse**:

- **Dirichlet**

Dirichlet boundary points follow a prescribed value of vector potential as the Pulse solution proceeds. A line region of uniform vector potential represents a perfectly conducting metal surface with magnetic field lines parallel to the boundary. An internal volume region at fixed vector potential represents a metal body that excludes the magnetic field.

- **Neumann**

The normal derivative of the vector potential is specified on a Neumann boundary. **Pulse** handles the special case $\partial \mathbf{A} / \text{partial} n = 0$. This condition implies that the magnetic field is normal to the boundary. An advantages of the finite-element method is that boundaries that are not fixed automatically satisfy the special Neumann condition, even if they are slanted or curved. Neumann boundaries are useful to reduce computation time by modeling portions of symmetric systems.

1.7 Physical units

Magnetic field quantities, time and length are in the standard SI units shown in Table 2. Coordinates entered in alternative units are converted to meters through the *DUnit* parameter. The quantity *DUnit* is the number of unit lengths per m. For example, if dimensions are in cm, set *DUnit* = 100.0.

Table 2: **Pulse** physical units

Quantity	Unit
Length	meters (internal)
Time	seconds
Vector potential	tesla-m
Magnetic fields	tesla
Current	amperes
Current density	amperes/m ²
Electrical conductivity	S/m

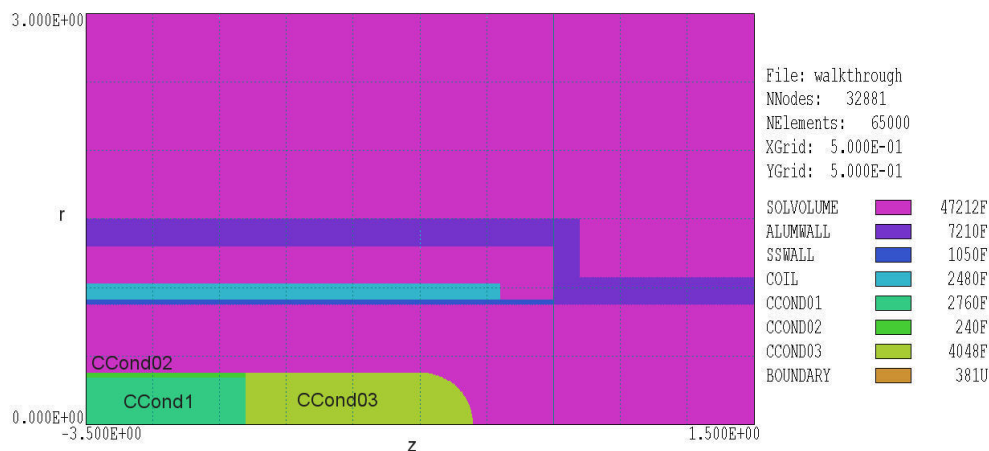


Figure 4: Region definitions for the WALKTHROUGH example.

2 Creating and analyzing a solution with Pulse

This chapter reviews a complete solution to illustrate **Pulse** procedures. We shall consider a system where the magnetic field generated by a pulsed coil penetrates surrounding metal structures. In preparation, move the files `WALKTHROUGH.MIN` and `CRITDAMP.CUR` from the example library to a working directory. The file `WALKTHROUGH.MIN` contains geometry information for the mesh generator and `CRITDAMP.CUR` contains a table that defines a curve for a normalized, critically-damped pulse. Start the **TriComp** program launcher `tc.exe` and make sure that the program points to the working directory using the *Set data directory* button.

Figure 4 shows the geometry of the cylindrical system. Note that the figure is a z - r plot as opposed to the a longitudinal section. The physical system is a figure of revolution about the z axis and only values $r \geq 0.0$ are defined. The device is a miniature magnetically-insulated transmission line to carry pulsed voltages in the megavolt range to a high-current electron beam diode. The assembly is contained inside an aluminum vacuum chamber (indigo). The aluminum wall also confines the pulsed magnetic field, reducing inductance and minimizing external fringe fields. The pulsed solenoid coil (cyan) has a uniform winding density that we approximate by

current density distributed uniformly over the cross-section. A thin layer of stainless steel (dark blue) acts as the outer conductor of the coaxial transmission line. The inner conductor is divided into three parts so we can study the effect of changing its material composition. In the baseline configuration, the regions CCOND1, CCOND02 and CCOND03 are assigned the properties of aluminum to represent a solid assembly. For an alternate calculation, CCOND1 is vacuum, CCOND02 is a thin stainless steel tube and CCOND03 is a stainless steel cap. The remaining elements of the solution volume (light violet) represent air or vacuum ($\mu_r = 1.0$). There is an additional region (BOUNDARY) that is not displayed in Fig. 4. The region consists of the nodes on the bottom, right and top sides of the solution volume to which we shall apply a special condition (discussed below).

2.1 Mesh generation

Run **Mesh** and load WALKTHROUGH.MIN. Pick the *Edit script/graphics* command to enter the drawing editor. Here, you can use the display capabilities to confirm the shapes of the regions shown in Fig. 4. Abandon the drawing and return to the main menu. Click the *Process* command. Click on the *Plot-Repair* command to inspect the completed mesh. Before proceeding, you can check some of the **Mesh** plot capabilities by clicking on *Plot/Repair*. Note how the program has flexed triangles to create accurate and smooth surfaces. The mesh contains 32,881 nodes and 65,000 elements. Exit the plot menu and choose the *Save mesh (MOU)* command to create the file WALKTHROUGH.MOU. You can now close or minimize **Mesh**.

2.2 Creating the Pulse script

Run **Pulse** 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 a magnetic-field solution:

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

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 WALKTHROUGH.MOU. **Pulse** loads the information and displays the dialog of Fig. 5. Note that the grid contains a row for each mesh region with a notation of the *Fill* status. There are four columns in the grid where you can enter values: *Sigma* (electrical conductivity in S/m), *MuR* (relative magnetic permeability), *Current* (total current I_z or I_θ through the region cross section in amperes) and *Fixed* (specified value of A_z in tesla-m or rA_θ in tesla-m²).

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 .

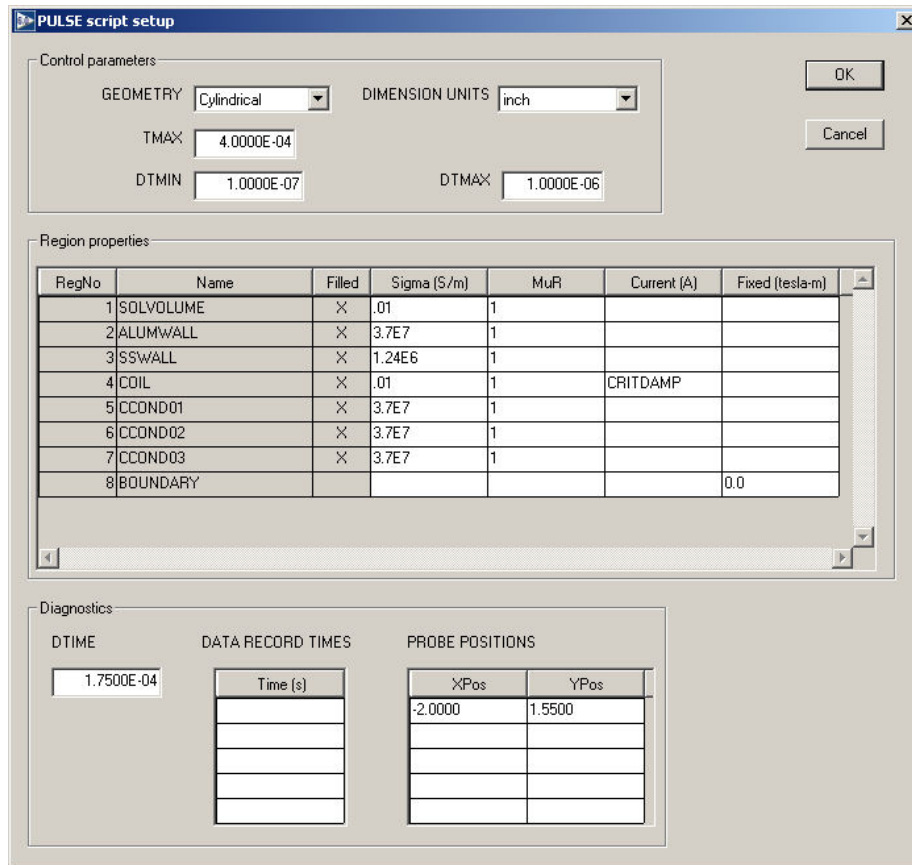


Figure 5: Dialog to set **Pulse** control parameters and region properties.

- **DIMENSION UNITS.** Choose the units that were used for the mesh definition. Coordinates will be converted to meters for the **Pulse** calculation. Choose Custom if the units do not appear in the list. In this case, the program includes a dummy entry in the script that you can modify with an editor.
- **TMAX.** Sets the stop time for the run. Enter the value in seconds. All runs begin at $t = 0.0..$
- **DTMIN.** Sets the minimum time interval, over-riding the automatic step selection. Enter the value in seconds. If no value is specified, **Pulse** uses a default of 1.0×10^{-9} s.
- **DTMAX.** Sets the maximum time interval, over-riding the automatic step selection. Enter the value in seconds. The default is infinity.

Set up the dialog with the values shown in Fig. 5 and click *OK*. **Pulse** uses the information in the dialog to create the script WALKTHROUGH.PIN shown in Table 3. Chapter 3 reviews the script format and advanced program capabilities.

In the region quantities of Fig. 5, note that the vacuum/air region (1) has $\mu_r = 1.0$ and $\sigma = 0.01$ S/m. The conductivity is much smaller than the metal regions but not zero. In a diffusion solution, information must propagate at a finite velocity. The speed of light does not enter into the magnetic diffusion equations discussed in Sect. 1.5. Therefore, the value $\sigma = 0.0$ would give an infinite mesh velocity leading to numerical instability. The choice $\sigma = 0.01$ ensures that changes of magnetic field move rapidly through the air/vacuum regions compared to the rise-time of the drive current pulse. The entry for region 8 sets $rA_\theta = 0.0$ for the nodes on the bottom, right and top boundaries of the solution. The condition constrains the magnetic field to be parallel to these surfaces. It is clear that the condition holds on the bottom boundary (axis of rotation), but some explanation is needed for the other two. The field will not be completely contained by the aluminum vacuum chamber. Some flux leaks out, and one of our goals is to find the magnetic field amplitude outside the tube. In a finite-element calculation of magnetic-field diffusion, it is not possible to define an ideal free-space boundary. In **Pulse**, there are two choices for the boundary condition: 1) parallel **B** and 2) normal **B**. To approximate the free-space condition, we include an isolating air region around the system and apply the condition that **B** is parallel. With no specification, the left-hand boundary automatically assume the natural boundary conditions for a finite-element magnetic-field solution (normal **B**). Because there is a long, uniform section of transmission line to the left of the solution volume, the condition that $B_r = 0.0$ is approximately correct.

2.3 Finite-element solution

Before we can carry out the solution, we must make a small modification to the script of Table 3. The current in the normalized data file CRITDAMP.CUR reaches a maximum value of 1.0 A at $t = 1.0$ s. We shall add scaling factors so that the peak current of 70 kA occurs at 350 μ s. In the **Pulse** main menu, click on the command *File/Edit script* and choose WALKTHROUGH.PIN. The program loads the file in the internal editor. The original form of Line 32 is

```
Current(4) = TABLE CRITDAMP.CUR
```

Table 3: File WALKTHROUGH.PIN

```

* ---- CONTROL ----
Mesh = walkthrough
Geometry = Cylin
DUnit = 3.9370E+01
TMax = 4.0000E-04
DtMin = 1.0000E-07
DtMax = 1.0000E-06
* ---- REGIONS ----
* Region 1: SOLVOLUME
Sigma(1) = 1.0000E-02
Mu(1) = 1.0000E+00
* Region 2: ALUMWALL
Sigma(2) = 3.7000E+07
Mu(2) = 1.0000E+00
* Region 3: SSWALL
Sigma(3) = 1.2400E+06
Mu(3) = 1.0000E+00
* Region 4: COIL
Sigma(4) = 1.0000E-02
Mu(4) = 1.0000E+00
Current(4) = TABLE CRITDAMP.CUR
* Region 5: CCOND01
Sigma(5) = 3.7000E+07
Mu(5) = 1.0000E+00
* Region 6: CCOND02
Sigma(6) = 3.7000E+07
Mu(6) = 1.0000E+00
* Region 7: CCOND03
Sigma(7) = 3.7000E+07
Mu(7) = 1.0000E+00
* Region 8: BOUNDARY
Fixed(8) = 0.0000E+00
* ---- DIAGNOSTICS ----
DTime = 1.7500E-04
History = -2.0000E+00 1.5500E+00
EndFile

```

The command instructs the program to calculate the time-dependent current of region 4 from the table contained in the file. Use the editor to change the line to:

```
Current(4) = TABLE CRITDAMP.CUR 350.0E-6 7.00E4
```

The first scaling factor applies to values of time and the second to values of current. **Pulse** adjusts quantities using the scaling factors when they are loaded from the table. Save the file and exit the editor.

To carry out the solution, click on the tool marked 2 or the menu command *Solve/Run* and choose `WALKTHROUGH.PIN`. The program reads all the information in the script and then starts the solution. Initial activities include loading the mesh, reading the data file for the current waveform and setting up the finite element equations. **Pulse** then proceeds with the initial-value solution. The status bar at the bottom of the window shows the progress. The screen assumes a blue color during the solution and returns to tan at the end. In response to the commands of the *Diagnostics* section, the program creates two data dumps (`WALKTHROUGH001.POU` and `WALKTHROUGH002.POU`) at times 175.0012 μs and 350.0025 μs . The times are not exactly equal to those specified in the script because the data dump must be performed between variable time steps. The program also creates the file `WALKTHROUGH.P01`, a record of quantities as a function of time at the probe location.

2.4 Solution analysis

From the **Pulse** main menu, click on the tool marked 3 or the *Analyze* menu command. The program changes the menu and tool selection. Click on *File/Load solution file* and choose `WALKTHROUGH001.POU`. The program loads the file and creates the default magnetic field-line plot of Fig. 6a. Click on *File/Load from series* and pick the solution at the peak of the drive current (350 μs). At this point, you can experiment with the extensive plot and analysis functions of **Pulse**. Figure 6b shows modified field lines when the conductivity for the center-conductor regions is changed to the value for stainless steel. Figure 7 plots the variations of B_z along the left boundary of Fig. 6. Values for the plot were automatically calculated using the *Analysis/Line scan* command. Chapters 4 and 5 give detailed descriptions of analysis capabilities.

To conclude, launch **Probe** from **TC**. Click on *Load probe file* and choose `WALKTHROUGH.P01`. The program displays a summary of file information – click *OK* to proceed. **Probe** creates the default plot shown in Fig. 8. As expected, the external field rises exponentially with time, reaching a peak value of about 33 G at $t = 350 \mu\text{s}$. Within the program you can change quantities, modify the plot limits or use the digital oscilloscope features to find specific values. Chapter 7 describes program capabilities in detail. Note that probe files are in text format so it is easy to port information to your own analysis programs.

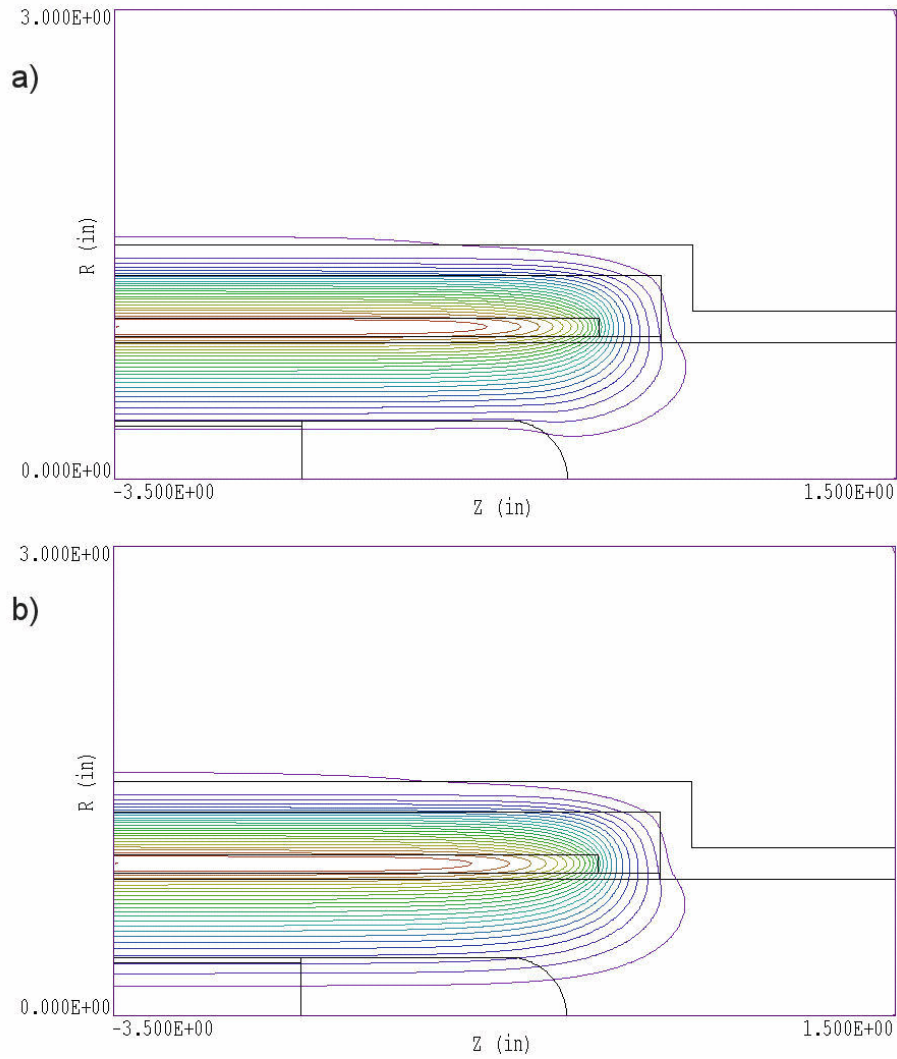


Figure 6: Magnetic field lines for the WALKTHROUGH example at $t = 350 \mu\text{s}$. Note that lines are separated by equal intervals of flux so that the spacing appears larger near the axis. *a)* Solid aluminum center-conductor. *b)* Center-conductor comprised of stainless steel tube and cap.

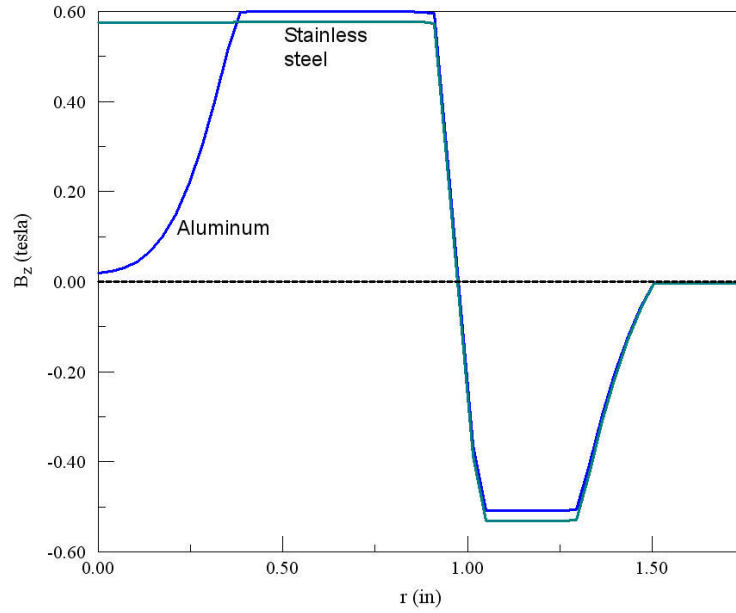


Figure 7: Variation of B_z along the left boundaries of Figs. 6a and 6b.

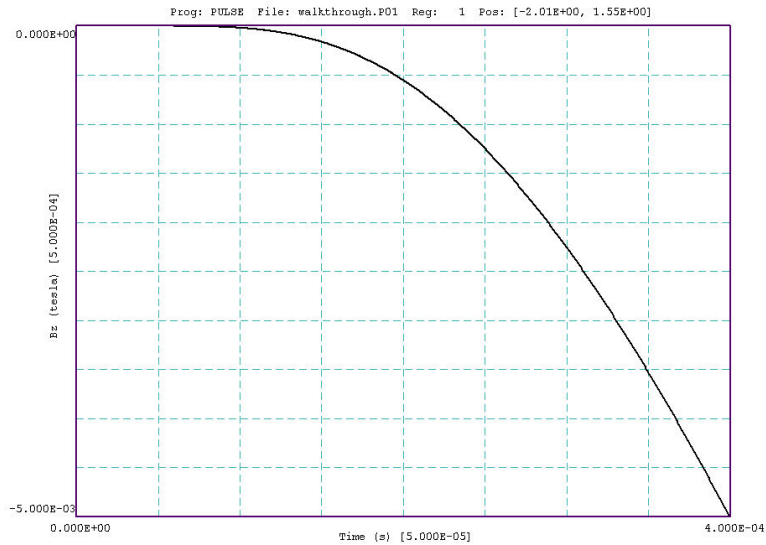


Figure 8: **Probe** plotting capabilities for the WALKTHROUGH example. Calculated magnetic field $B_z(t)$ just outside the aluminum vacuum tube at $z = -2.00$ in.

3 Pulse solution reference

3.1 Organizing Pulse calculations

The program `pulse.exe` can run in two modes: interactively in a window or autonomously in the background with the option for batch file control. The autonomous modes allow automatic processing of large or repetitive data sets.

Pulse requires two input files.

- A **Mesh** output file that describes the conformal triangular mesh. The file contains node coordinates and the region numbers of elements and vertices.
- A command script that sets control parameters for the solution program and describes the physical properties associated with region numbers.

Runs with time-dependent current and/or non-linear magnetic materials may require additional files to define the variations.

The mesh output file always has a name of the form `MPREFIX.MOU`, where `MPREFIX` is a valid file prefix (1 to 32 characters). The script must have a name of the form `FPREFIX.PIN`. **Pulse** issues an error message if both input files are not available in the working directory. To organize data, the resulting output files have names that begin with `FPREFIX`.

A **Pulse** run consists of three activities that involve different programs: mesh generation, field solution, and analysis. The programs communicate through data files. Sometimes, you may calculate several solutions from the same mesh by changing the properties of materials. Each run includes the following steps.

- Prepare a **Mesh** input script with a name of the form `FPREFIX.MIN` following the instructions in Chaps. 3 and 5 of the **Mesh** manual.
- Run **Mesh** either interactively from the **TC** program launcher or from the Windows Command Prompt to create the file `MPREFIX.MOU`.
- Prepare a command script (`FPREFIX.PIN`) using the *Setup* command in **Pulse** or a text editor. The allowed file commands are described in this chapter.
- Optionally, prepare data files to define time-dependent current or characteristics $\mu(|\mathbf{B}|)$ of a nonlinear material. The text files may consist of experimental data and may be created with a text editor or spreadsheet.
- Run **Pulse** to create one or more output files. Files with names of the form `FPREFIX.001` are data dumps at specific times that can be used as input in the *Analysis* menu. The files are in text format and contain information on the mesh geometry, the physical properties of regions, and values of computed quantities at the nodes. In magnetic diffusion solutions the recorded region properties are relative magnetic permeability (for linear materials) and the volume resistivity. The code stores node values of the vector potential (or stream

function) and its time derivative. The latter quantity is used to display inductive electric field in the *Analysis* menu. Values of relative magnetic permeability are also stored to plot variations in non-linear materials. Files with names of the form `FPREFIX.P01` are probe records of temperature at specific locations. The contents of these files can be plotted with the **Probe** program.

- Analyze the solution using the *Analysis* menu of **Pulse** or **Probe**. You can also transfer the information in data dumps or history files to your own analysis programs.

3.2 Pulse script format

You can create a control script for **Pulse** 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. Direct script editing is required for some advanced **Pulse** capabilities. This chapter gives a detailed description of the syntax and functions of script commands.

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 (*). **Pulse** accepts commands in any order. Table 3 illustrates a complete script.

There are three classes of commands: program control, region properties and diagnostics. 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

```
SIGMA 2 2.5E5
```

assigns electrical conductivity $\sigma = 2.5 \times 10^5$ to elements with region number 2. (**Note:** For back-compatibility, **Pulse** recognizes command formats from Version 1.0 through 5.0 where control commands start with the keyword *Set*, region commands with *Region* and diagnostic commands with *Diag*.)

3.3 Control commands

This section discusses commands that control program operation. Note that commands are written symbolically and as they might appear in the 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 **FPrefix.TIN**, then **Pulse** will seek the default mesh file **FPrefix.MOU**.

GEOMETRY [Rect, Cylin]

GEOMETRY = Cylin

Pulse 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.

DUNIT DUnit

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 **Pulse**. The quantity $DUnit$ equals the number of **Mesh** units per meter. For example, if the **Mesh** dimensions are entered in microns, set $DUnit = 1.0 \times 10^6$. The default value is 1.00. (**Note.** Spatial quantities recorded in the output files **FPrefix.001** are always in meters). In an analysis session with **Pulse**, spatial quantities in graphs and listing files are scaled to the **Mesh** units. For example, if the **Mesh** dimensions are in cm and $DUnit = 100.0$, the spatial quantities in plots will be in cm.)

TMAX TMax

TMAX = 105.0

Set the stop time for the run. Enter the value in seconds. All runs begin at $t = 0.0$. (Required input)

DTMIN DtMin

DTMIN = 0.0001

Set the minimum time interval, over-riding the automatic step selection. Enter the value in seconds. The default value is $\Delta t_{min} = 1.0 \times 10^{-9}$ s.

DTMAX DtMax

DTMAX = 0.25

Set the maximum time interval, over-riding the automatic step selection. Enter the value in seconds. The default is $\Delta t_{max} = \infty$.

SAFETY Sf

SAFETY = 35.0

Set a safety factor for the automatic time step adjustment. The range of typical values is $S_f = 2.5$ to 25 . Higher values improve accuracy at the expense of longer computation times. The default value is $S_f = 15.0$.

NCHECK Nc

NCHECK = 10

Set the number of time increments between a recalculation of the automatic time step routine. The default value is $N_c = 5$.

NSTEP Ns

NSTEP = 5000

Sets the maximum number of time steps in the solution. The program stops at either T_{max} or N_s , depending on which occurs first. This command can prevent solutions controlled by a batch file from running out of hand. The default is $N_s = 100,000$.

INTERP [Linear, Spline]

INTERP = Linear

Set the interpolation mode for quantities where a time or temperature variations is described by a table of values (Sect. 3.7). The default cubic spline method gives the highest accuracy for smooth data (continuous value and slope). Use linear interpolation when data points are noisy or discontinuous.

3.4 Picking a time step

Pulse follows magnetic diffusion by advancing in small time increments. Because time scales may vary several orders of magnitude in numerical solutions of diffusion problems, it is essential to use a variable time step. For example, consider setting an instantaneous non-zero value of vector potential on a boundary. Initially, the full change in vector potential occurs across a single mesh element. If the solution space is about 100 elements on a side, the initial time step must be 10,000 times shorter than the time step necessary to model magnetic diffusion in the full solution space. To perform this large adjustment, **Pulse** has a routine that estimates a good time step. The procedure relies on a calculation of the root-mean-squared value for the second derivative of vector potential over the full solution region. Although the automatic time step feature generally works well, some user intervention may be necessary because of the wide diversity of problems that **Pulse** can handle. Section 3.3 introduced three control commands that allow you to modify the time step: *Safety*, *DtMin* and *DtMax*. We will discuss each in turn.

SAFETY = 35.0

The automatic time-step routine calculates the maximum allowed time increment and then divides by the factor S_f . Generally, S_f should exceed unity. Higher values improve accuracy but lengthen the computation time. The numerical methods used in **Pulse** should be stable for all choices of time step. If you observe unstable behavior, the most likely cause is the mesh distortion effect discussed in the next section.

DTMIN = 5.0E-7

The automatic time step feature of **Pulse** can cope with difficult solutions where there are initial discontinuities in vector potential. Depending on the geometry, **Pulse** may have to use extremely short time steps to achieve accuracy early in time. Sometimes, details of the initial relaxation of the magnetic field are not important. To speed the calculation, you can over-ride automatic time selection by setting a minimum time step with the *DtMin* command. Once the estimated time step rises above Δt_{min} , the automatic routine takes over. The *DtMin* command is essential in solutions with time-dependent currents that start from zero. In this case, the vector potential is initially zero everywhere. Because the second derivative is zero, **Pulse** cannot determine a time step. In this case, the program assigns a very small default increment. You can set a better value using the *DtMin* command. In picking a time step, note that a good solution usually require about 1000 or more steps.

DTMAX = 1.0E-3

Occasionally, the automatic routine in **Pulse** may pick a time step that is too long for good accuracy. This situation occurs if there is a local temperature variation in a large solution volume at approximately uniform temperature. In this case, you can clamp the maximum value of time increment using the *DtMax* command. Again, a good solution should have about 1000 steps or more.

Ironically, one disadvantage of a stable numerical procedure is that it always gives an answer, even if it is quite inaccurate. You can use the *DtMax* command to check the accuracy of important solutions. First, run the problem using automatic time step selection. Note the final time step and use the *Analysis* menu to find field values at critical positions. Run the solution again, setting Δt_{max} to a fraction (0.25 - 0.50) of the final time. Check for significant changes in the predicted magnetic field.

3.5 Avoiding numerical instabilities

Pulse uses the time-centered Dufort-Frankiel method to advance the diffusion equation in time. The explicit method gives fast calculations on large meshes and is stable for all choices of time step on a simple rectangular finite-difference mesh. A numerical instability may occur if the method is applied on conformal meshes that have triangular elements that are compressed in the y or r directions. For a discussion of the theory and conditions of the instability, see S. Humphries, **Field Solutions on Computers** (CRC Press, Boca Raton, 1997), Sect. 12.4. The simplest way to avoid problems is to use uniform element dimensions in Mesh (with $\Delta x = \Delta y$) constructing the foundation mesh. If it is necessary to define regions along the axes with different resolution, you can invoke a special feature in Mesh for automatic correction of element shapes. To signal corrections, you must include the command

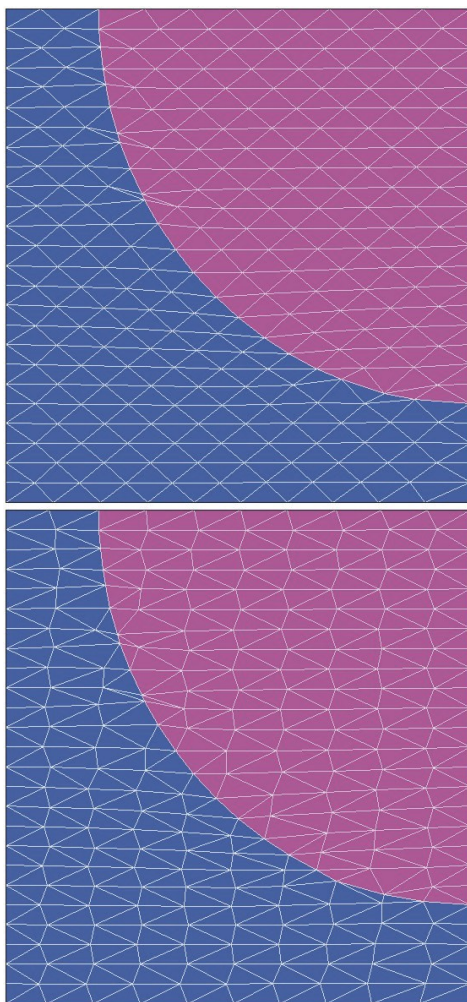


Figure 9: Effective of the DCORRECT command in a **Mesh** script.

DCORRECT

(for diffusive correction) in the *Global* section of the **Mesh** script.

Figure 9 shows the effect for a mesh with elements where $\Delta y = 0.4\Delta x$. In the top plot with no correction most of the triangles have an internal angle greater than 90° , the condition for an instability. With the diffusive correction (bottom plot), **Mesh** automatically adjusted the shapes of mesh elements so that there are no contiguous blocks of elements with obtuse internal angles. The following global section was used for the bottom plot:

```
GLOBAL
  XMESH
    0.0  5.0  0.5
  END
  YMESH
    0.0  5.0  0.2
  END
  DCORRECT
END
```

As a rule, always include `DCORRECT` in `Mesh` scripts for `Pulse` solutions when variable resolution is used.

3.6 Region properties

The `Pulse` command set for defining the material properties of regions is relatively simple. In this section, we limit attention to commands to set values that remain constant through the solution. The following section describes alternative forms to define variable quantities.

FIXED RegNo A

FIXED(4) = 0.0

Set the region to a fixed value of vector potential that does not change during the solution. For planar solutions, enter the values of A_z in tesla-m. In cylindrical solutions, specify rA_θ in tesla-m². Because the stream function always equals zero at $r = 0$, `Pulse` automatically creates a fixed potential region with $rA_\theta = 0$ on the axis.

The following commands refer to electrically-conductive materials. Multiple commands may be necessary for a full region definition.

SIGMA RegNo Sigma

SIGMA(6) = 1.4E6

Set the electrical conductivity of the region in S/m.

MU RegNo MuR

MU(8) = 500.0

Sets a uniform value for the relative magnetic permeability. The default is $\mu_r = 1.0$. To model a floating ideal conductor with a constant value of vector potential that is not known in advance, set $\mu_r \ll 1.0$.

CURRENT RegNo Ir

CURRENT(6) = -10000.0

Sets a constant current for the region in amperes. The current is divided between elements to maintain uniform current density. To model variations in current density, divide a coil cross section in multiple regions.

In a magnetic diffusion solution, all areas in the solution that do not have a fixed value of vector potential are treated as electrical conductors. You must take care in solutions with vacuum or air regions where the conductivity is zero. Diffusion through such a medium is instantaneous, so that numerical solutions may exhibit inaccuracies or oscillations. A conservative approach is to set the conductivity in insulators to a lower (but non-zero) value compared to that of the conductive media. For example, in a solution for field penetration into a stainless steel structure ($\sigma = 1.24 \times 10^6$ S/m, you can usually obtain a good solution by taking $\mu_r = 1.0$ and $\sigma = 10^{-2}$ S/m in coil and air regions. In this case, the rapid field diffusion through the insulators is distributed over several time steps.

3.7 Nonlinear materials

A powerful feature of **Pulse** is the ability to handle field-dependent magnetic permeability. You can define any material using *tabular functions*. A tabular function is a text file consisting of up to 256 entry lines of values for an independent and dependent variable. To define a field-dependent permeability, each line has a value of field magnitude $|\mathbf{B}|$ (in tesla) followed by the corresponding relative permeability. You can prepare tabular function files with a text editor or spreadsheet. You can also use published data or digitized experimental traces. As an example, Table 4 shows the tabular function supplied with the code for soft iron. Note that the file syntax conforms to the same rules as the **Pulse** command file. The free-form parser accepts real numbers in any format with a choice of delimiters. You can add documenting comment lines starting with an asterisk (*). The end of the data is marked with the *EndFile* command.

Although the data lines of the example are ordered by increasing value of the independent variable, this ordering is not required. **Pulse** sorts the list before use and records the final order in the listing file, `FPREFIX.PLS`. Furthermore, the independent variable intervals need not be uniform. A table requires a minimum of 5 entries. You must ensure that the tabular functions extend over the full range of field amplitude that will be encountered in the solution. The interpolation routines return 0.0 for values of the independent variable out of the table range.

Use the following command to apply a field-dependent magnetic permeability to a region.

MU RegNo TABLE TabName
MU(4, TABLE) = SOFTIRON.DAT

Specify the magnetic permeability in a region as a function of the magnetic field amplitude. The values are contained in a data file with the name `TabName` available in the working directory. Each data line in the file contains a value of $|\mathbf{B}|$ (in tesla) and $\mu_r(|\mathbf{B}|)$. By default, **Pulse** uses cubic splines for interpolation of the tables. This method minimizes recalculation time and gives smooth interpolations that aid convergence. It is important to note that the quality of the interpolation depends on the nature of the numerical data. The dependent quantity and its first derivative should vary smoothly over the table range. For noisy data, use the `Interp` command to set linear interpolation. You can check the fidelity of the interpolations by inspecting the file `FPREFIX.PLS`. After listing the sorted table, **Pulse** records a sample set of interpolated values.

3.8 Time variations of current or vector potential

Time variations of current over a region or vector potential along a fixed boundary may be defined by either a tabular function or a mathematical expression. The following command forms employ tabular functions.

FIXED RegNo TABLE TabName [Tm, Am]
FIXED(5, TABLE) = FLUX.POT

Set the vector potential of a fixed region to a prescribed function of time rather than a constant value. For rectangular problems, the numerical values are contained in the named file. Each data line contains two entries: 1) the time in seconds and 2) the value of A_z (tesla-m) or rA_θ (tesla-m²). Note that the values of the independent variable must extend from $t = 0.0$ to

Table 4: Table to define the properties of soft iron

```
* Generic soft iron table
*  B(tesla)  Mu(relative)
  0.0000    4075.45105
  0.8944    3768.28312
  1.2000    3166.80993
  1.4000    2380.64690
  1.5000    1595.42303
  1.5500    997.827296
  1.6000    674.163054
  1.6500    473.573054
  1.7000    319.425385
  1.7500    221.384065
  1.8000    173.209366
  1.8500    139.973170
  1.9000    113.066781
  1.9500    90.6346946
  2.0000    73.9635734
  2.0500    60.9549021
  2.1000    51.1983450
  2.1250    44.2497995
  2.1500    38.2874802
  2.1750    32.4062541
  2.2000    25.9244242
  2.2500    19.6677855
  2.2797    15.2170140
  2.3069    12.3583893
  2.3443    10.5837506
  2.3996    9.15722611
  2.4905    7.34265734
  2.5627    5.93589744
  2.6706    4.85174825
  2.8498    3.94592075
  3.2074    3.15384615
ENDFILE
```

$t > T_{max}$. The optional real-number parameters T_m and A_m are multiplication factors applied to values of time and vector potential when the table is loaded. The default values are $T_m = 1.0$ and $A_m = 1.0$.

Time-dependent vector potential quantities are useful in problems where you want to specify a time variation of magnetic flux. In planar solutions, the difference in A_z between two surfaces equals the integral of normal magnetic field along a line connecting the surfaces. In other words, the difference in vector potential equals the magnetic flux per length (in z) between the surfaces. In cylindrical problems, the quantity $rA_\theta/2\pi$ equals the enclosed axial magnetic flux. The time-dependent fixed boundary condition is useful for inductor problems. Setting constant values of dA_z/dt or $d(rA_\theta)/dt$ is equivalent to applying a constant voltage to the structure.

CURRENT RegNo TABLE TabName [Tm Im]
CURRENT(2, TABLE) = ALPHA.CUR (2.5E-6, 15000.0)

Define the total region current as a function of time from the named tabular function. Each file data line contains the time t in seconds and $I(t)$ in amperes. The optional real-number parameters T_m and I_m are multiplication factors applied to values of time and current when the table is loaded. The default values are $T_m = 1.0$ and $I_m = 1.0$. Using multiplication factors, you can create a library of normalized waveforms and scale them for different solutions. Four normalized tables are supplied with Pulse: **LINEAR.DRV** (independent and dependent variables rise linearly from 0.0 to 1.0), **SINE.DRV** (independent variable t in the range 0.0 to 1.0 with the dependent variable given by $f(t) = \sin(t)$), **SMOOTH.DRV** (the dependent variable rises smoothly from 0.0 to 1.0 over the range $0.0 \leq t \leq 0.1$ and remains constant thereafter) and **CRITDAMP.DRV** (critically-damped pulsed with a peak value of 1.0 at $t = 1.0$).

Alternatively, you can set temporal variations of vector potential or current from mathematical functions.

FIXED RegNo > Function
FIXED(5) > 1.50E-4*cos(\$t/20.5) + 1.3E-5

Set A_z (tesla-m) or rA_θ (tesla-m²) according to a specified function of time. 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 below. The function defines a variation in time: $f(t)$. The parser uses the Perl standard for variables where $\$s$ designates the variable t .

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

- The independent variable $\$t$ equal to the time in seconds.
- Real and/or integer numbers in any valid format (*e.g.*, 3.1415, 476, 1.367E23, 6.25E-02, 8.92E+04,...). Integers are converted to real numbers for evaluation.
- Binary operations: + (addition), - (subtraction), * (multiplication), / (division) and ^ (exponentiation).

- Functions: **abs** (absolute value), **sin** (sine), **cos** (cosine), **tan** (tangent), **ln** (normal logarithm), **log** (base 10 logarithm), **exp** (normal exponent) and **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*($t^2/24))`

corresponds to

$$1 - \exp \left[- \left(\frac{t^2}{24} \right) \right]. \quad (14)$$

3.9 Diagnostic commands for dynamic solutions

Diagnostic commands control data output from **Pulse** in the dynamic mode. The program produces three types of data, all in text format:

- From 1 to 999 data files with the names of the form **FPREFIX001.POU**, **FPREFIX002.POU**, Each file contains a complete record of spatial information (the mesh geometry and field distribution) at a specified time. These files may be used for plots and calculation in the analysis mode of **Pulse**.
- A listing file with the name **FPREFIX.PLS** that contains a variety of information about the run. This information is useful mainly to check the validity of input parameters and to diagnose problems.
- Optional history files with names **FPREFIX.P01**, **FPREFIX.P02**, Here, the magnetic and electric fields at one or more probe locations is recorded at each time step of the solution. The information can be inspected with a text editor or the utility program **Probe** (Chap. 7).

The first three commands control the times for spatial data dumps. The last command sets probe positions for the history file.

DTIME DTime **DTIME = 0.2**

For many runs a convenient way to make spatial data dumps is at uniform time intervals. This command sets the approximate interval (in seconds) between dumps. Because **Pulse** uses variable time steps, it is not possible to ensure that the dump occurs at an exact time. Instead, the program writes data as soon as possible after passing the desired time. For example, **Pulse** will write dump 010 when $t \geq 10 \times DTime$. The default value is $DTime = \infty$.

SETTIME STime
SETTIME = 3.45

Sometimes you may want to inspect data at certain critical times or at the end of a run. For this situation, you can set up to 100 time markers using multiple *SetTime* commands. **Pulse** will write data as soon as possible after passing each marked time. Note that multiple *SetTime* commands must be in order of increasing time.

Note that the two commands for data dumps can work concurrently. Be careful in setting the parameters. You may generate enough data to fill a hard disk.

HISTORY XPos YPos
HISTORY ZPos RPos
HISTORY = (5.00, 9.85)

Pulse opens history files `FPREFIX.P01`, `FPREFIX.P02`,... when it detects a History command. The parameters in the command are the spatial location of a probe where the time and temperature are recorded. The location is given as (x,y) or (z,r) . Enter the coordinates in the original units used in **Mesh**. **Pulse** locates the closest mesh node and records the magnetic and electric fields at that point. You can define up to 10 probe points with multiple *History* commands. The history file contains information on the actual probe location.

NHSTEP Ns
NHSTEP = 10

In runs with a large number of time steps, the history files may be larger than necessary and difficult to export to other programs. Use this command to set the number of time steps per data record in the history files. The default value is $N_s = 1$.

MOVIE NFrame [Nx Ny] [QMin QMax]
MOVIE 100 800 600 0.0 110.0

Create a set of N_{frame} graphics files in BMP format separated by time interval t_{max}/N_{frame} . The plots show lines of magnetic flux density (contours of A_z or rA_θ) in the full solution volume. The number of field lines is determined by any previous setting in the *Analysis* menu. The integer parameters N_x and N_y give the horizontal and vertical image size in pixels. If they do not appear, the default is $N_x = 1024$ and $N_y = 768$. The two real-number parameters are the upper and lower values of A_z or rA_θ . Autoscaling is applied to each frame if they are absent. In addition to video editors, here are several free utilities to stitch image sequences into an MP4 file.

3.10 Running Pulse

Pulse 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\]PULSE [DataPath\]FPrefix <ENTER>
```

where the file `FPREFIX.PIN` and the appropriate **Mesh** output file are available in the data directory. With this capability, you set up extended autonomous **Pulse** runs using a DOS batch file or a Perl script.

The remainder of this section discusses commands in the main menu when **Pulse** 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 (PIN) **EDIT LISTING (PLS)** **EDIT FILE**

These commands call the internal editor to inspect or to modify **Pulse** input and output files. Choosing a file from an alternate directory does not change the working directory. The *Edit script (PIN)* command shows a list of all files with names of the form `FPREFIX.PIN` while *Edit listing (PLS)* displays files with names `FPREFIX.PLS`.

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. 4.8. 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 **Pulse** script to control a dynamic magnetic field 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. 5. 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. 3.3. You can enter basic physical properties of regions in the grid box (see Sect. 3.6). You must edit the script directly to invoke advanced functions like anisotropic materials.

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

RUN

Pick an input file (such as `FPREFIX.PIN`) 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

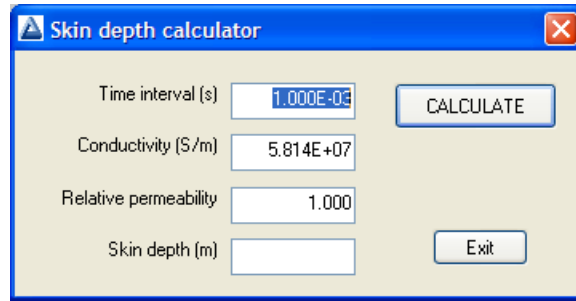


Figure 10: Skin depth calculator.

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 **Pulse**. The run output will include any data dumps that have been created and probe files up to the termination point.

SOLUTION DISPLAY

When *solution display* is active, **Pulse** plots the evolution of lines of magnetic flux density. The plot is updated each time a data file is created. The number of lines is controlled by the last setting in the *Analysis* menu.

ANALYZE

Go to the analysis menu.

PROBE MANUAL

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

SKIN DEPTH CALCULATOR

The *Skin depth calculator* (Fig. 10) in the **Pulse Help** menu is useful to estimate when there are strong contributions from eddy currents. It solves the equation:

$$\delta \cong \sqrt{\frac{2\Delta t}{\sigma\mu_r\mu_o}}, \quad (15)$$

where δ is the skin depth, σ is the electrical conductivity, Δt is the approximate time interval in seconds and μ_r is the relative magnetic permeability. To use the tool, fill in three known values and leave one field blank. **Pulse** fills the missing field when you press the *Calculate* button.

3.11 Format of the Pulse output files

Data dump files created by **Pulse** are 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: -8.89001778E-02  
XMax:  3.81000762E-02  
KMax:  251  
YMin:  0.00000000E+00  
YMax:  7.62001524E-02  
LMax:  131  
DUnit:  3.93700000E+01  
NReg:    8  
ICylin:  1  
Time:  1.75000000E-04
```

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). Line 11 lists the simulation time at which the data were recorded.

The node section consists of 4 title lines and $K_{max} \times L_{max}$ data lines, one for each node of the solution space. 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 nodal value of A_z in tesla-m or rA_θ in tesla-m².
- The values of relative magnetic permeability in the upper and lower associated elements.
- The time derivative of A_z or A_θ to calculate induced electric field.

The node section has the following appearance:

```

--- Nodes ---
  k      l  RgNo  RgUp  RgDn      x      y
=====
  1      1    9    5    0 -8.89001778E-02  0.00000000E+00
  2      1    9    5    0 -8.82810516E-02  0.00000000E+00
  3      1    9    5    0 -8.77849568E-02  0.00000000E+00
  4      1    9    5    0 -8.72739792E-02  0.00000000E+00
  ...

      A      MuUp      MuDn      dAdt
=====
  0.00000000E+00  1.00000000E+00  1.00000000E+00  0.00000000E+00
  0.00000000E+00  1.00000000E+00  1.00000000E+00  0.00000000E+00
  0.00000000E+00  1.00000000E+00  1.00000000E+00  0.00000000E+00
  0.00000000E+00  1.00000000E+00  1.00000000E+00  0.00000000E+00
  ...

```

The region section consists of four title lines following by *NReg* data lines, one for each region. An entry "1" in the second column designates a fixed region. The recorded region quantities are μ_r , current, fixed potential, area and electrical conductivity. The region section has the following appearance:

--- Regions ---

RegNo	Fix						Mu
1	0	0	0	0	0	1.00000000E+00	
2	0	0	0	0	0	1.00000000E+00	
3	0	0	0	0	0	1.00000000E+00	
4	0	0	0	0	0	1.00000000E+00	
5	0	0	0	0	0	1.00000000E+00	
6	0	0	0	0	0	1.00000000E+00	
7	0	0	0	0	0	1.00000000E+00	
8	1	0	0	0	0	1.00000000E+00	

...

Current	Vector Pot	Area	Sigma	
0.00000000E+00	0.00000000E+00	7.97628504E-03	1.00000000E-02	...
0.00000000E+00	0.00000000E+00	7.00001556E-04	3.70000000E+07	...
0.00000000E+00	0.00000000E+00	7.90324748E-05	1.24000000E+06	...
5.76908800E+04	0.00000000E+00	2.40000368E-04	1.00000000E-02	...
0.00000000E+00	0.00000000E+00	2.63226347E-04	3.70000000E+07	...
0.00000000E+00	0.00000000E+00	2.70968264E-05	3.70000000E+07	...
0.00000000E+00	0.00000000E+00	3.91796102E-04	3.70000000E+07	...
0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00	...

...

--- Region names ---

SOLVOLUME
ALUMWALL
SSWALL
COIL
CCOND01
CCOND02
CCOND03
BOUNDARY

4 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 **Pulse**.

4.1 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 6 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. 11 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**.

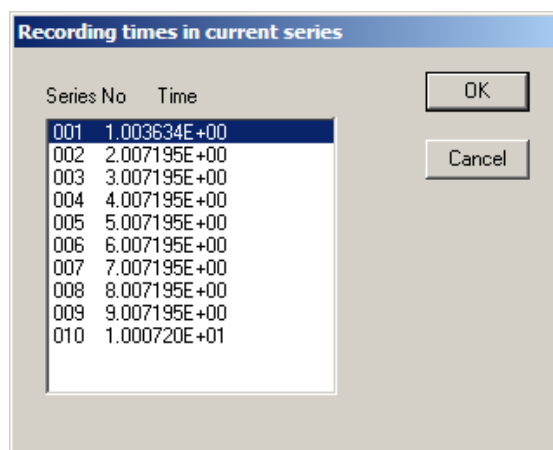


Figure 11: Dialog to pick a solution in a series generated by an initial-value program.

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

An 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. 4.8. 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.

4.2 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. 12 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 .

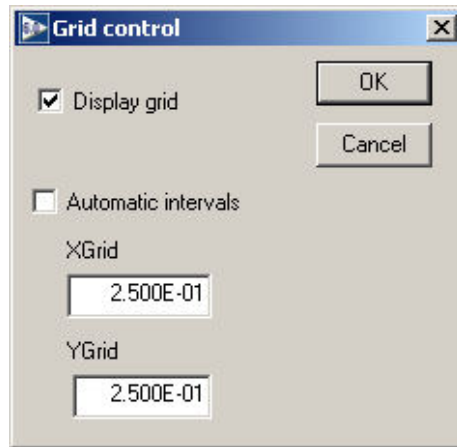


Figure 12: Grid control dialog

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

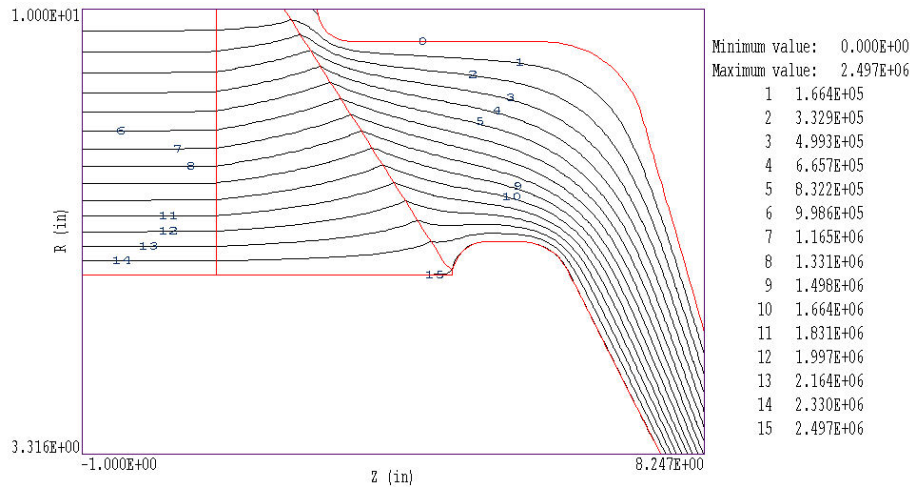


Figure 13: Monochrome contour plot with labels

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. 13). 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 $N_x = N_y = 75$.

4.3 User specified contours

In the default mode of *Contour line* and *Filled contour* plots, **Pulse** chooses a linearly-spaced set of plot values. The values depend on the specified number of lines and the limits. The minimum and maximum values are set by the program when *Autoscale* is active or by the user. Although this method is adequate for general use, there are circumstances where more control is desirable:

- Logarithmic intervals are useful when the plot quantity has a large variation.
- In applications like shielding, it may be necessary to plot a specific value of field magnitude.

Pulse features a simple but versatile method for user control of contour plots.

The first step is to prepare a text file of numerical values for contour lines. You can place one value per line or put all values on a single line separated by spaces. **Pulse** recognizes any valid real-number format (including scientific notation). Here is an example:

```
0.0000001
0.0000002
0.0000005
0.000001
0.000002
0.000005
0.00001
0.00002
0.00005
0.0001
```

The minimum number of entries is 1 and maximum is 100. The values must be arranged in ascending order. The plot listing is easiest to understand if the number is 15 or less. Save the file if you do extended work with standard intervals. Select the intervals you want to display and copy them to the clipboard.

Next, choose *Plot/Plot settings/Plot limits* (or click the *Plot limits* tool) to open the dialog of Fig. 14. Note the checkbox marked *From clipboard* at the bottom-right. When you check this box, **Pulse** performs the following actions:

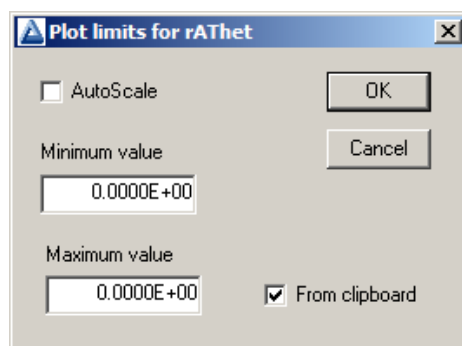


Figure 14: Plot limits dialog.

- Reads and analyzes the contents of the clipboard, checking that the entries are valid numbers in ascending order.
- Sets the desired contour values.
- Exits the dialog and updates the plot.

Uncheck the box to return to normal operation (linear intervals with automatic or user-specified limits). To enter a new set of values from the clipboard, uncheck and then recheck the box.

Note that the user-specified values affect only *Contour line* and *Filled contour* type plots (not plots of type *Element* and *Surface*). Figure 15 illustrates how the values are interpreted. In a *Contour line* plot (upper), each line is displayed with a corresponding listed value on the right if the number of lines is 15 or less. When the number is greater, the value list contains a partial set of selected values. In the *Filled contour* plot (lower), the list gives the minimum value for each colored interval. In other words, an interval covers the range from its listed value to the listed value of the next interval. For example, the dark violet band in the figure extends from 1.0×10^{-7} to 2.0×10^{-7} . Values below the first are plotted in the next lowest color (in this case, light violet) and values above the top value are plotted in white.

4.4 Saving and loading views

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

SAVE NAMED VIEW

Save the view parameters for the current plot. Quantities such as the 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:

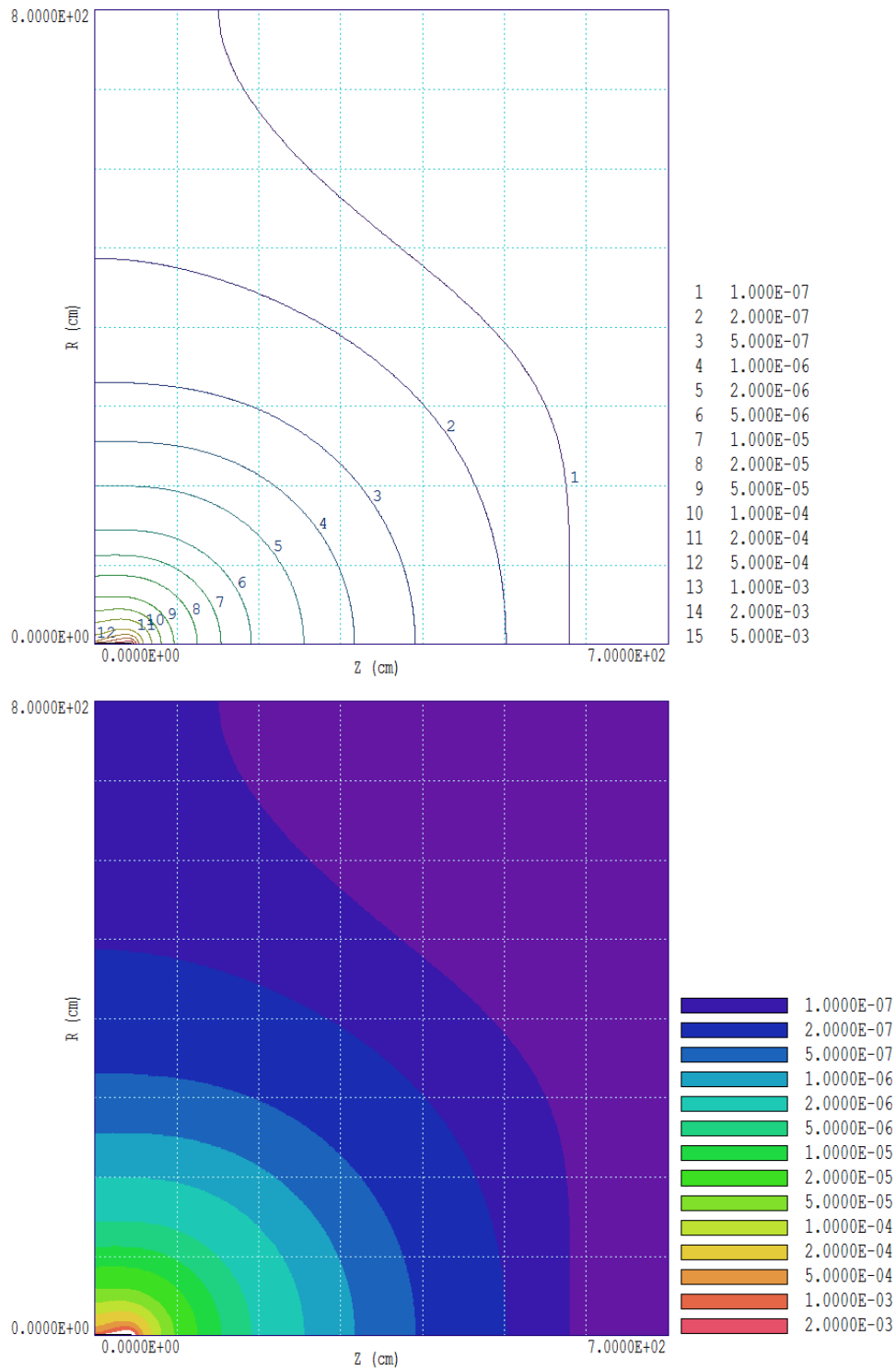


Figure 15: Display of user-specified contour intervals. Top: *Contour line* plot. Bottom: *Filled contour* plot.

```
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.

4.5 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.

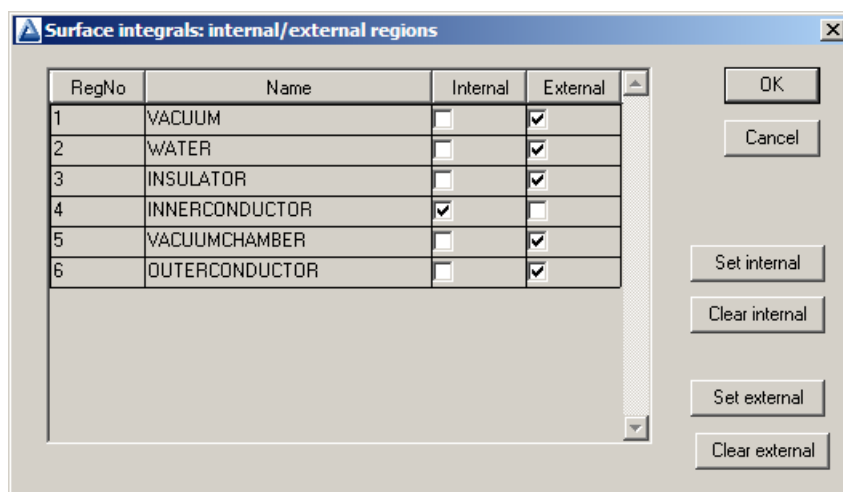


Figure 16: Surface integral dialog

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.4.6). 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 file over region boundaries. In response to the command, the program displays the dialog of Fig. 16. 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).

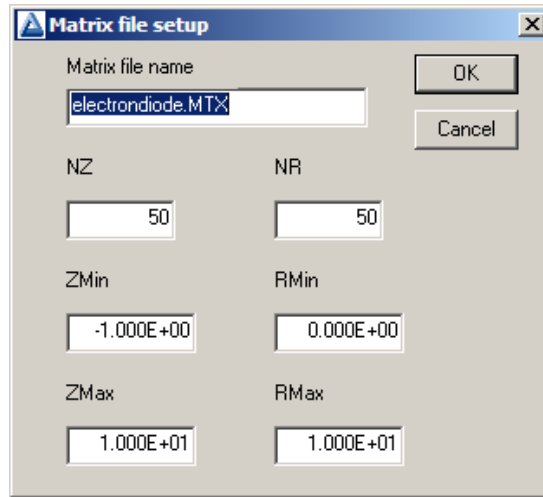


Figure 17: Dialog to generate a matrix file.

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. 17. 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 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.

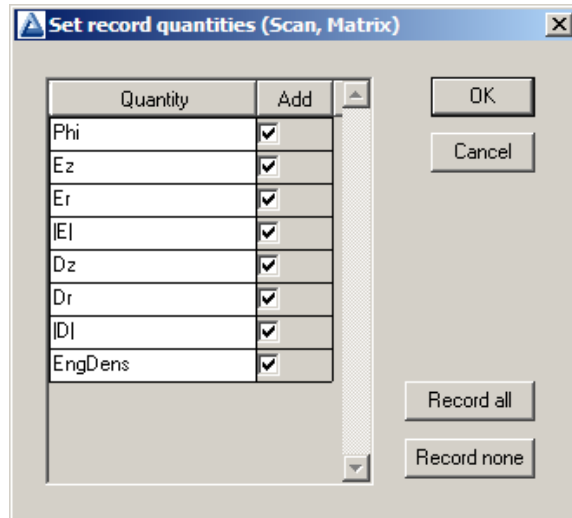


Figure 18: Dialog to set quantities recorded in response to the *Line scan* and *Matrix commands*.

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. 18, 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.

4.6 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. 19). 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.

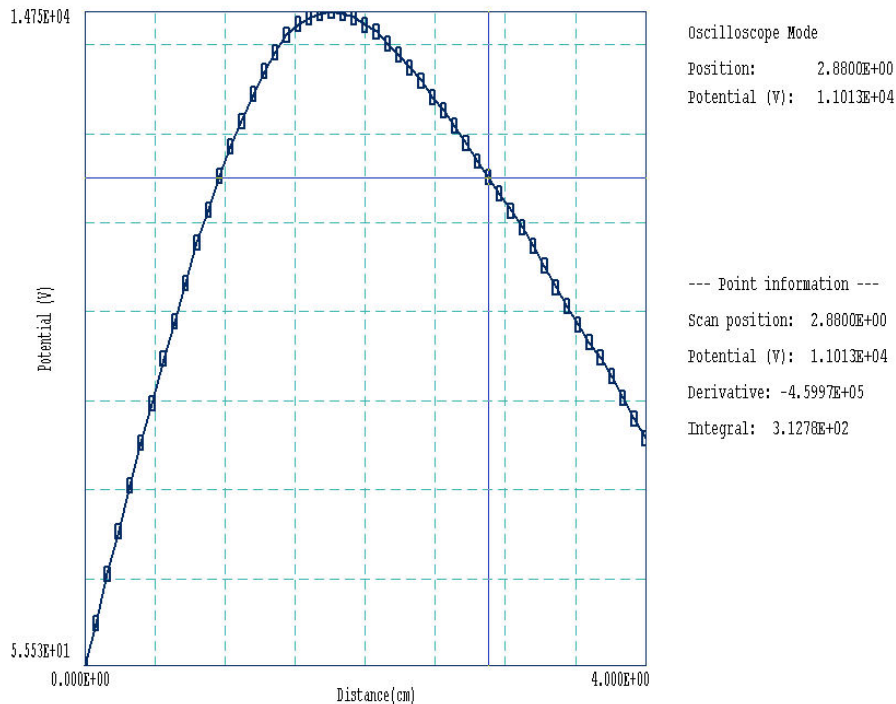


Figure 19: Scan plot in the oscilloscope mode

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.

4.7 Vector tools

The analysis menu includes useful tools to display directional quantities in the **VECTOR** section of the current configuration file. Figure 20 shows the corresponding entries on the toolbar.

PROBE

This feature was inspired by the familiar Magnaprobe illustrated in Fig. 20. 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

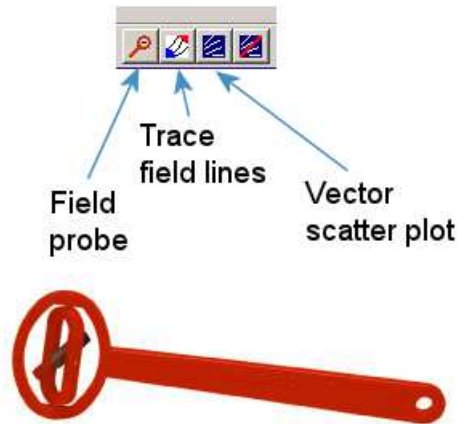


Figure 20: Tools to display vector information in the slice plot menu.

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*.

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.

4.8 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 [Append]

OUTPUT SW02.DAT

Open an output data file `FileName` to record the results of analysis operations. If the string `Append` does not appear, the program closes a currently opened file and starts new one. If the string appears, the program opens a file if no file is opened. Otherwise, it writes to the currently opened file.

CONFIGURATION [datapath\] FileName

CONFIGURATION = \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. 4.4). 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
```

5 Standard Pulse calculated quantities

The **Pulse** package includes a standard configuration file for the analysis of pulsed magnetic-field solutions: `pulse_standard.cfg`). Table 5 shows quantities defined in the file for plots and scans, vector tools, volume integrals and surface integrals. Note that contours of A_z (planar) and rA_θ follow lines of \mathbf{B} separated by equal increments of magnetic flux. Definitions for magnetic force calculations by surface integrals are also included in the file as comment lines. 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>

5.1 Pulse variables

Pulse 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).

- **&A**: the vector potential A_z (planar, tesla-m) or rA_θ (cylindrical, tesla-m²).
- **&Bxz**: the x (planar) or z (cylindrical) component of magnetic flux density in tesla.
- **&Byr**: the y (planar) or r (cylindrical) component of magnetic flux density in tesla.
- **&BMag**: the magnitude of the magnetic flux density in tesla.
- **&Hxz**: the x or z component of the normalized magnetic field in A/m.
- **&Hyr**: the y or r component of the normalized magnetic field in A/m.
- **&HMag**: The magnitude of the normalized magnetic field in A/m..
- **&Mu**: The relative magnetic permeability.
- **&Sigma**: The electrical conductivity in S/m.
- **&dAdt**: the derivative of the vector potential (dA/dt or dA_θ/dt) in V/m, equal to $-E_z$ or $-E_\theta$.

Table 5: Quantities defined in `pulse_standard.cfg`

Section	Quantity	Units
Interpolation	\mathbf{B} line (A_z/rA_θ)	tesla-m, tesla-m ²
	B_x, B_z (magnetic flux density)	tesla
	B_y, B_r (magnetic flux density)	tesla
	$ \mathbf{B} $ (flux density magnitude)	tesla
	E_z, E_θ (electric field)	V/m ³
	p (resistive power density)	W/m ³
	j (eddy current density)	A/m ²
	H_x, H_z (magnetic field)	A/m
	H_y, H_r (magnetic field)	A/m
	$ \mathbf{H} $ (magnetic field magnitude)	A/m
	μ_r (relative permeability)	
	σ (conductivity)	S/m
	Vector	\mathbf{B} (magnetic flux density)
\mathbf{H} (magnetic field)		A/m
Volume	P (power loss)	W/m, W
	U (field energy density)	J/m, J
Surface	F (magnetic flux)	weber/m, weber

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

- **RegMu**: the relative magnetic permeability for uniform regions.
- **RegCurrent**: the drive current assigned to a region, evaluated at the simulation time of the data file (in A).
- **RegVPot**: the value of A_z or rA_θ for fixed-potential regions, evaluated at the simulation time of the data file (in tesla-m or tesla-m²).
- **RegArea**: the cross-section area of the region in m².
- **RegSigma**: the electrical conductivity in S/m of a uniform region.

Pulse records one run parameter in solution files: $\$Time$. It equals the simulation time (in s) of the data file. Table 6 lists the full configuration file showing how the variables are used in expressions.

Table 6: Configuration file pulse_standard.cfg

```
PROGPARAM
  $Mu0 = 1.256637E-6
END

RUNPARAM
* Time of data dump for a dynamic solution
  $Time = 0.0
END

REALTIMEPARAM
END

INTERPOLATION
  BLine = &A
  Bx/Bz = &Bxz
  By/Br = &Byr
  |B| = &BMag
  Ez/EThet = &dAdt @CHS
  PDens = &dAdt &dAdt * &Sigma *
  J = &dAdt &Sigma * @CHS
  Hx/Hz = &Hxz
  Hy/Hr = &Hyr
  |H| = &HMag
  Mu = &Mu
  Sigma = &Sigma
END

VECTOR
  BVect = &Bxz;&Byr
  HVect = &Hxz;&Hyr
END

VOLUME
  Power = &dAdt &dAdt * &Sigma *
  FEnergy = &HMag &HMag * &Mu * $Mu0 * 0.5 *
END

SURFACE
  Flux = &Bxz;&Byr
END

ENDFILE
```

6 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\tricom`). 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.

6.1 Configuration file structure

An analysis configuration is a text file with a name of the form `FPREFIX.CFG`. Table 6.1 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 6.3 covers parameters and variables that may appear in expressions.

Table 7: 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
```

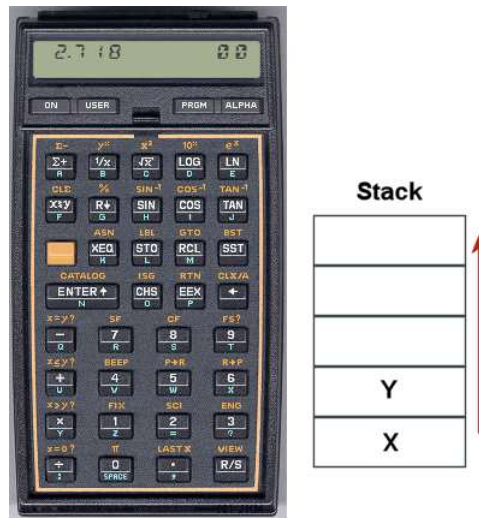


Figure 21:

6.2 Introduction to RPN

Consider the expression

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

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

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

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

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

The string is parsed in strict order from left to right for expressions of any degree of complexity. A calculator with RPN logic uses the stack shown in Fig. 21. 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 8: Unary operators

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

- 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.

6.3 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 8 lists the available set. The following considerations apply with respect to the @CHS operator:

- You may include negative numbers (such as -3.1456 and -8.9E-09) in expressions. The number -22.56 is equivalent to 22.56 @CHS.
- Expressions like -\$Epsi0 and -&grady[3] that mix 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}/ϵ_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).

6.4 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.

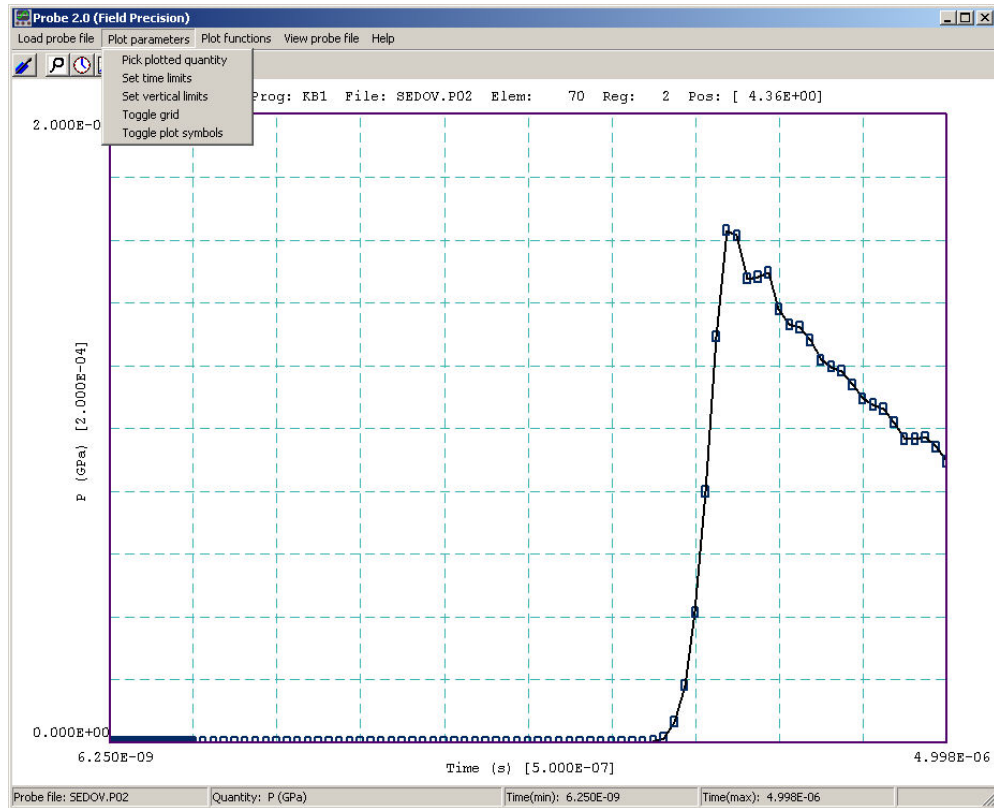


Figure 22: Probe screen shot.

7 Probe – history file plot utility

7.1 Introduction

Probe is the universal plotting program for all Field Precision initial-value solution codes. You can set from 1 to 20 probes by specifying positions in the solution program command script. The probes record quantities in an element or at a node as a function of time. The resulting text files have names of the form `FPREFIX.P01,...`, `FPREFIX.P12`, where `FPREFIX` is the run prefix.

Table 9 shows the standard probe file format. The first section is a header that contains the following information:

- Generating program name.
- Dimensionality of the generating program (1, 2 or 3).
- The spatial position of the probe (from 1 to 3 quantities).
- The index of the element sampled by the probe.
- The region number of the element.

- Conversion factors for the probe position and the recorded quantities.
- Labels for the recorded quantities.

Although the solution programs and their output files employ SI units (meters, kilograms,...), the graphical analysis displays often use practical units to make it easier to visualize results and to facilitate automatic grids. **Probe** multiplies file quantities by the conversion factors during the loading process. Note that the quantity *DConv* and conversion factors for positions are equal to *DUnit*, a variable used in many solution programs. After four lines of label information, the remainder of the file consists of data lines. Each line contains the time (in seconds) and one or more element or node quantities. Real numbers are recorded in E15.6 format.

7.2 Loading data files

When you start **Probe** the only active menu option is *Load probe*. Plotting and analysis functions become active when a probe file has been opened. The program displays a dialog showing all files with suffixes of the form P01,...,P12. Pick a file to analyze and click *OK*. Changing directories in the dialog will change the working directory of the program. If the load is successful, **Probe** creates a default plot of the data (Fig. 22).

The status bar at the bottom of the window contains the name of the probe file, the current plot quantity, and the temporal range of data. The default plot shows the first quantity recorded in the probe file over the full range of time. The horizontal and vertical scales are chosen so that the plot fits on the screen and the grid lines are automatically adjusted so that they lie on even values of the plotted quantity with easily recognized intervals (*e.g.*, 0.02, 0.05, 0.10, ...). The grid intervals are shown in parentheses next to the labels of the horizontal and vertical axes. The title line at the top of the plot shows the following information: generating program, probe file name, element number, region number and position. This information is recorded in hardcopy plots to help you archive your data.

7.3 Plot settings

The commands of the *Plotsettings* menu control the quantities, ranges and appearance of the plot. The screen plot automatically updates whenever you make a change.

PLOTTED QUANTITY

A dialog shows a list of element quantities included in the probe file. Highlight your choice and click *OK*.

TIME LIMITS

By default **Probe** shows the full time-span recorded. You can narrow the range by supplying values for the minimum and maximum time. Uncheck *Autoscale* in the dialog and supply maximum and minimum values. To return to the full range, check the *Autoscale* box.

VERTICAL LIMITS

In the default mode **Probe** picks a scale to display the full range of the plotted quantity. You

Table 9: Example of the **Probe** file format

Field Precision probe file

Program: KB1
 NDimen: 1
 XPosition: 4.357E-02
 ElementNo: 70
 RegionNo: 2
 NQuant: 6
 DConv: 100.0
 QConv1: 100.0
 QConv2: 0.001
 QConv3: 1.0E-9
 QConv4: 1.0
 QConv5: 1.0E-6
 QConv6: 0.001
 QLabel1: x or r (cm)
 QLabel2: Rho (gm/cm3)
 QLabel3: P (GPa)
 QLabel4: T (deg-K)
 QLabel5: U (MJ/kg)
 QLabel6: v (km/s)

Time (s)	Re (m)	Rho (kg/m3)	Press (Pa)
6.250000E-09	4.357041E-02	1.000000E-01	-1.238464E+02
1.250000E-08	4.357041E-02	1.000000E-01	-1.238464E+02
1.874999E-08	4.357041E-02	1.000000E-01	-1.238464E+02
2.499996E-08	4.357041E-02	1.000000E-01	-1.238464E+02
3.124995E-08	4.357041E-02	1.000000E-01	-1.238464E+02
...			

Temp (deg-K)	Epsi (J/kg)	V(Av) (m/s)
1.000000E-03	8.620001E+00	0.000000E+00
1.000000E-03	8.620001E+00	0.000000E+00
1.000000E-03	8.620001E+00	0.000000E+00
1.000000E-03	8.620001E+00	0.000000E+00
1.000000E-03	8.620001E+00	0.000000E+00
...		

can narrow or expand the range by supplying minimum and maximum values. The program returns to full range if you check the *Autoscale* box or if you change quantities using *Pick plotted quantity*.

TOGGLE GRID

Switch between grid and and fiducial lines in the plot.

TOGGLE PLOT SYMBOLS

Include or remove symbols to mark the recorded points.

7.4 Plot functions

The commands of the *Plot functions* menu activate the *Oscilloscope mode* of the program and also send plots to hardcopy devices or plot files.

OSCILLOSCOPE MODE

When you issue this command, **Probe** simulates a digital oscilloscope. As shown in Fig. 23, the mouse cursor changes to a cross-hair pattern when it is inside the plot window. The program adds movable fiducial lines to mark the current point. You can drag the markers along the time axis by moving the mouse. A box at the bottom of the plot shows values of the time and plotted quantity at the current position. If you press the left mouse button, the program displays a box with the following information about the current point:

- Time, t .
- Value of the plotted quantity, $V(t)$.
- Derivative of the plotted quantity, $dV(t)/dt$.
- Definite integral of the plotted quantity, $\int_0^t V(t')dt'$.

The definite integral is taken from the time of the first recorded value in the probe file to that of the current point. You can find integrals between points by subtracting values. The information is also written to the Windows clipboard. Other functions of the program are deactivated in the *Oscilloscope mode*. Press the right mouse button or the *Esc* key to return to normal program operation.

SMOOTH DISPLAY

Use this command one or more times to smooth the currently-displayed trace. Smoothing applies to the screen display and exported plots, but does not affect the data values in the probe file.

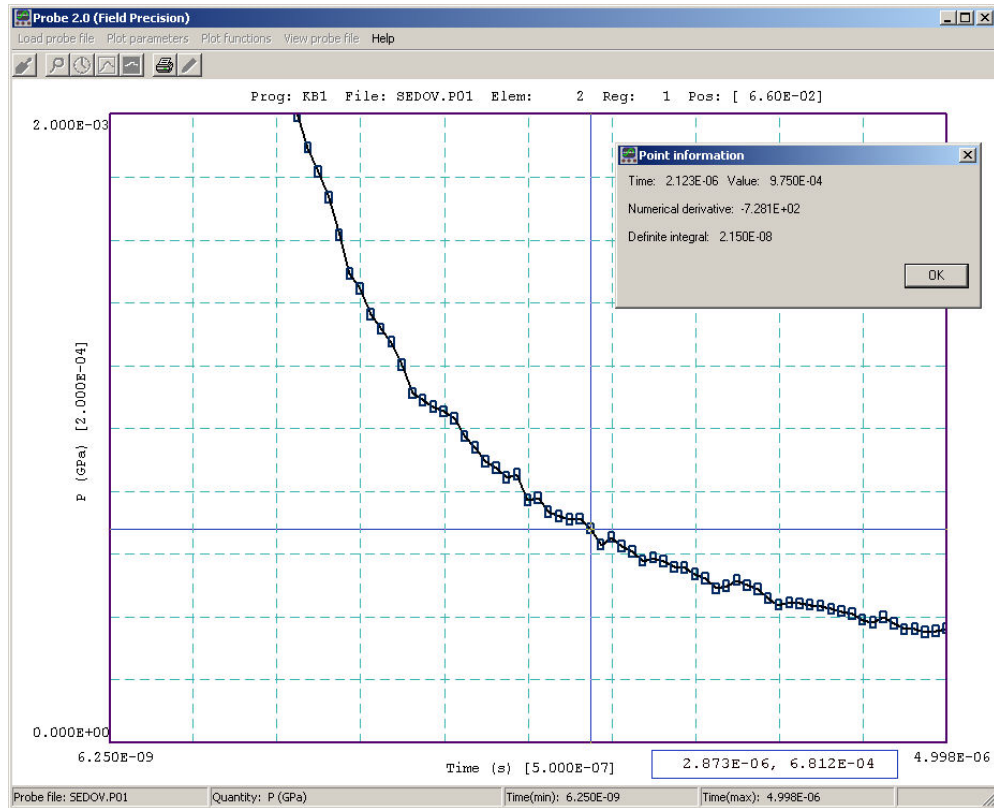


Figure 23: **Probe** in the *Oscilloscope mode*.

DEFAULT PRINTER

Probe can port copies of the plot to any installed Windows printer. The program sends output to the default printer, so be sure to select the correct device using the *Settings/Printer* function of Windows before making the plot.

PLOT FILE (EPS)

PLOT FILE (BMP)

PLOT FILE (PNG)

Send the plot to a file in the following formats: Encapsulated PostScript, Windows Bitmap or Portable Network Graphics. The program prompts for a file prefix and then creates a file with the names FPREFIX.EPS, FPREFIX.BMP or FPREFIX.PNG.

COPY TO CLIPBOARD

Copy the plot to the clipboard in in Windows MetaFile format.

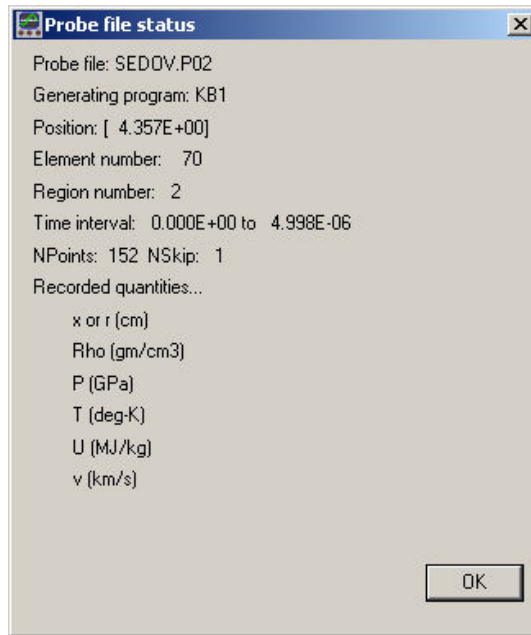


Figure 24: **Probe** file-information message box.

7.5 Information

PROBE FILE INFORMATION

Display information on the probe file in a message box (Fig 24). The quantity *NSkip* in line 7 is used for long files. There is no reason to store more than 1000 points for plots on typical screens and hardcopy devices. When there are less than 1000 data lines, **Probe** loads all points (*NSkip* = 1). When the file contains 1000 to 2000 data lines, the program loads every second point (*NSkip* = 2), and so forth. In this way the **Probe** can handle probe files of any length without exceeding memory limits.

VIEW PROBE FILES

Load a probe file into the internal editor so you can inspect the data directly. The editor runs in read-only mode so that you cannot change the file. Exit the editor to return to program operation.

PROBE MANUAL

Show this document in your default PDF viewer.

Index

- Ampere's law, 9
- Analysis
 - Standard quantities, 54
- Analysis commands
 - 3D angle, 42
 - 3D grid, 42
 - Close data record, 38
 - Contour style, 40
 - Create analysis script, 38
 - Field lines at points, 49
 - Grid control, 39
 - Interpolation method, 47
 - Line integrals, 46
 - Line scan, 46
 - Load configuration file, 37
 - Load from series, 37
 - Load named view, 43
 - Load solution file, 37
 - Matrix file, 47
 - Number display format, 41
 - Number of contours, 41
 - Number of scan points, 47
 - Open data record, 37
 - Oscilloscope mode, 48
 - Pick vector quantity, 50
 - Plot limits, 39, 42
 - Plot quantity, 39
 - Plot type, 39
 - point calculation, 45
 - Probe, 49
 - Remove vectors, 50
 - Rotate 3D, 41
 - Run analysis script, 38
 - Save named view, 43
 - Scan plot quantity, 47
 - Set recorded quantities, 48
 - Snap distance, 40
 - Surface integrals, 46
 - Toggle fixed point, 40
 - Toggle grid display, 39
 - Toggle keyboard, 40
 - Toggle outline, 40
 - Toggle snap mode, 40
 - Vector scatter plot, 50
 - Volume integrals, 46
- Analysis script
 - Configuration, 51
 - GenScan, 51
 - Input, 51
 - Interpolation, 52
 - Matrix, 52
 - NScan, 52
 - Output, 51
 - Plot, 52
 - Scan, 51
 - SurfaceInt, 52
 - VolumeInt, 52
- analysis script commands, 50
- analysis script example, 53
- Analysis script format, 50
- analysis, batch file control, 50
- analytic solutions, 6
- autoscale, 39
- boundary conditions, 10
- calculated quantities
 - standard variables, 61
- comment lines, 21
- computational mesh, 7
- configuration file, 57
 - calculated quantities, 57
 - expressions, 62
 - load errors, 62
 - structure, 57
- conformal mesh, 7, 13
- contour lines, user-specified, 42
- data export, 47
- delimiters, 21
- diffusion equation, 7
- diffusion, magnetic, 9
- Dirichlet boundary, 11
- eddy current, 9
- element, 7
- facet, 7

- Faraday's law, 9
- fiducials, 39
- field
 - definition, 6
 - equations, 6
- field line plots, 50
- field probe, 49
- finite-element method, 6
 - basis, 7
- function parser
 - rules, 29
- grids, 39
- induced electric field, 9
- interpolation methods, 47
- keyboard/mouse control, 40
- line integral, 46
- magnetic flux
 - relation to vector potential, 10, 29
- magnetic permeability, relative, 9
 - tables, 27
- mathematical functions, 29
- matrix file, 47
- Maxwell equations, 8
- Maxwell stress tensor, 54
- Mesh
 - dcorrect, 24
- mouse controls, 45
- Neumann boundary, 11
- node, 7
- node equations, 6, 7
- nonlinear materials, 27
- number notation, 41
- numerical instability, 24
- numerical solutions, 6
- plot control, 39, 42
- plot types, 39
- plot view control, 41
- Plot views, saving, 43
- Probe, 17, 31, 63
 - file format, 63
 - file information, 68
 - graphical environment, 64
 - loading files, 64
 - oscilloscope mode, 66
 - plot export, 67
 - plot settings, 64
 - smooth display, 66
- Pulse
 - accuracy, 24
 - choosing conductivity, 15
 - command prompt, 32
 - file types, 8, 17, 20, 30
 - instruction manual, 33
 - Movie, 31
 - output file format, 34
 - program function, 4
 - region properties, 13
 - running in a window, 32
 - running in the background, 20
 - script, 21
 - script dialog, 13, 32
 - Skin depth calculator, 33
 - Solution display, 33
 - solution procedure, 7
 - solution process, 17
 - stability, 24
 - steps in a solution, 13, 20
 - tables, 17
- Pulse analysis variables, 54
- Pulse commands
 - Current, 26, 29
 - DTime, 30
 - DtMax, 15, 23, 24
 - DtMin, 15, 22, 24
 - DUnit, 15, 22
 - Fixed, 26, 27
 - Geometry, 13, 22
 - History, 31
 - Interp, 23
 - Mesh, 22
 - Mu, 26, 27
 - NCheck, 23
 - NHStep, 31
 - NStep, 23
 - Run, 32
 - Run analysis script, 32
 - Safety, 23, 24

- script, 15
- SetTime, 31
- Setup, 32
- Sigma, 26
- Stop, 33
- TMax, 15, 22

recorded quantities, 46

recorded quantities, rules, 48

reverse Polish notation, 59

rpn

- binary operators, 59
- parsing rules, 59

scan plot menu, 48

script control of analyses, 38

skin depth, 33

snap mode, 40

standard analysis functions, 37

surface integral, 46

surface plot control, 41

symmetry

- cylindrical, 7
- planar, 7

tabular functions, 27

- scaling factors, 17

text editor

- internal, 32, 38

time variations

- mathematical functions, 29
- tables, 27

units, physical, 11

user-specified contour lines, 42

vector display, 50

vector potential, 9

vector tools, 49

volume integral, 46