



**Nelson 7.0**  
**Induction Heating – AC/RF**  
**Magnetic Fields with Eddy Current**  
(Copyright 2011)

**Field Precision LLC**  
PO Box 13595, Albuquerque, NM 87192 U.S.A.  
Telephone: +1-505-220-3975  
Fax: +1-617-752-9077  
E mail: [techinfo@fieldp.com](mailto:techinfo@fieldp.com)  
Internet: <http://www.fieldp.com>

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Program functions . . . . .	3
1.2	Learning <b>Nelson</b> . . . . .	3
1.3	Finite-element solution procedure . . . . .	4
1.4	Scripts and data files . . . . .	6
1.5	Theoretical background . . . . .	7
<b>2</b>	<b>Creating and analyzing a solution with Nelson</b>	<b>10</b>
2.1	Mesh generation . . . . .	10
2.2	Creating the <b>Nelson</b> script . . . . .	11
2.3	Finite-element solution . . . . .	13
2.4	Solution analysis . . . . .	13
<b>3</b>	<b>Nelson solution reference</b>	<b>17</b>
3.1	<b>Nelson</b> script format . . . . .	17
3.2	Program control commands . . . . .	18
3.3	Commands for material properties . . . . .	20
3.4	Running <b>Nelson</b> . . . . .	20
3.5	Format of the <b>Nelson</b> output file . . . . .	22
<b>4</b>	<b>Eddy current and energy loss in drive coils</b>	<b>25</b>
4.1	Theory . . . . .	25
4.2	Examples . . . . .	26
<b>5</b>	<b>TriComp analysis functions</b>	<b>31</b>
5.1	File menu commands . . . . .	31
5.2	Plot menu commands . . . . .	33
5.3	Analysis menu commands . . . . .	36
5.4	Scan plot menu . . . . .	39
5.5	Vector tools . . . . .	39
5.6	Analysis script commands . . . . .	41
<b>6</b>	<b>Standard Nelson calculated quantities</b>	<b>45</b>
6.1	Nelson variables . . . . .	45
6.2	Standard analysis configuration . . . . .	46
6.3	Additional configurations . . . . .	48
<b>7</b>	<b>Building custom analysis configurations</b>	<b>51</b>
7.1	Configuration file structure . . . . .	51
7.2	Introduction to RPN . . . . .	53
7.3	Operators, parameters and variables . . . . .	54
7.4	Building expressions . . . . .	56

---

# 1 Introduction

## 1.1 Program functions

**Nelson** is a finite-element tool to calculate AC or RF magnetic fields in conductive media. Applications include simulation of induction heating for material processing and the frequency response of transformers or audio devices. The program handles any cylindrical or rectangular geometry. You can define up to 250 regions to represent coils, conductors or ferrites. The amplitude and phase of harmonic currents may be set individually. Materials are characterized by values of the relative magnetic permeability and electrical conductivity. The graphical analysis functions of **Nelson** provide full information on the amplitude and phase of magnetic flux density, electric field and current density throughout the solution regions. The program automatically calculates and records power density and total current organized by regions. The information can be ported directly to the **TDiff** program for thermal calculations. **Nelson** makes screen and hardcopy plots of potential, field amplitude current density and resistive energy deposition.

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

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

- Browse the following sections in this chapter to review some basics of RF/AC magnetic fields with eddy currents. You will probably return to the material to resolve issues (such as boundary conditions) as you get more involved in the programs.
- Scan Chap. 1 of the **Mesh** manual to understand **TC**, the **TriComp** program launcher.
- 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 you own simulations. Chapter 4 describes how to convert a script into a conformal mesh.
- Be sure to read Chapter 2 of this manual. The walkthrough example covers the steps in creating and analyzing a finite-element solution with **Nelson**.

- Run some of the prepared examples supplied with the package. The examples give insights into solution options and may serve as templates for your own work.
- As you gain experience you will want to take advantage of the full range of **Mesh/Nelson** capabilities. Chapter 4 of the **Mesh** manual is a comprehensive reference on the **Mesh** drawing editor. Chapter 5 covers processing, plotting and repairs of meshes. Chapter 3 in this manual is a complete reference on **Nelson** control scripts. Chapter 5 gives information on the analysis functions common to all the **TriComp** programs. Capabilities include a variety of plots, interpolations of field values and integrals of field energy and other quantities. Chapter 6 reviews the calculated quantities defined by the standard **Nelson** configuration files. Finally, Chap. 7 describes how to build a customized post-processor 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 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 **Nelson**. 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 **Nelson** 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 complex-value Poisson equation (Sect. 1.5). These equations can be solved directly by analytic methods if the system geometry and material properties are simple (for example, ferromagnetic layers with uniform  $\mu_r$  between concentric cylinders). 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 1 shows the type of mesh used for the two dimensional solutions of **Nelson**. The figure defines three terms that will be used throughout this manual:

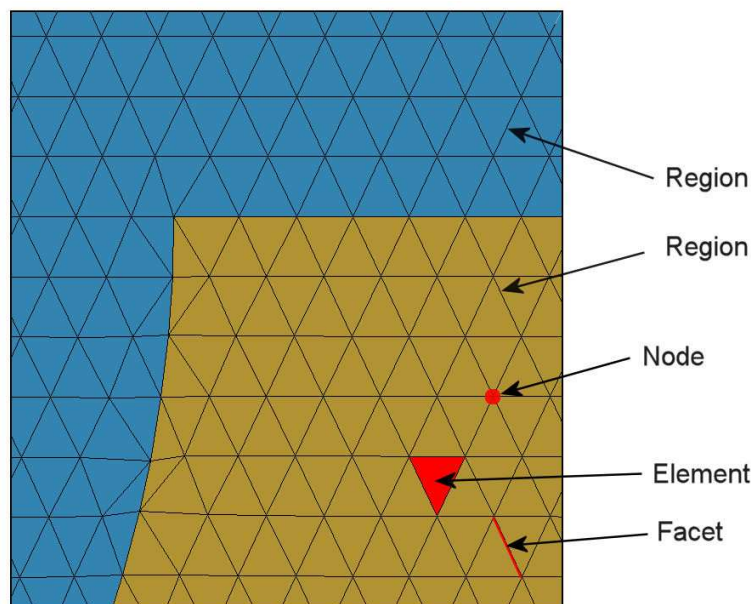


Figure 1: Conformal triangular mesh – definition of terms.

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

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

The mesh in Fig. 1 has the important property of *conformality*. The term means that the triangles have been specially shaped to conform to the boundaries between materials (*regions*). As a result, each element has an unambiguous material identity. The finite-element method is based on two approximations: 1) material properties in an element are uniform and 2) elements are small enough so that the field quantities 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 **Nelson**, solution of the set representing the complex-value Poisson equation gives the quasi-electrostatic potential at each node. We can then perform two-dimensional interpolations to find harmonic potential at intervening points, or take numerical spatial derivatives to find the components of the electric field.

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

1. You define the boundaries of material objects in a solution volume for your application. The task is usually accomplished with the drawing editor of **Mesh**. You can also specify control information such as the target element sizes that may affect the accuracy and run

Table 1: **Nelson** files

Name	Function
MName.MIN	<b>Mesh</b> input script (definition of foundation mesh and region outlines)
MName.MLS	<b>Mesh</b> diagnostic listing
MName.MOU	<b>Mesh</b> output (node locations and element identifies)
EName.NIN	<b>Nelson</b> input script (run control and material properties)
EName.NLS	<b>Nelson</b> diagnostic listing
EName.NOU	<b>Nelson</b> output file (node locations and electrostatic potential)
AName.SCR	<b>Nelson</b> script for automatic data analysis

time of the solution. The result is a text record (*script*) with sets of line and arc vectors that outline electrodes and dielectrics.

2. **Mesh** analyzes the boundary specifications and automatically generates a set of conformal triangles such as those of Fig. 1. The program creates an output file that lists the locations of nodes and the identities of elements.
3. You define the material properties of regions in the solution volume. Usually, this task is performed through an interactive dialog in **Nelson**.
4. **Nelson** reads the mesh geometry and applies the material parameters to generate the complex-value linear equation set.
5. **Nelson** solves the equations using iterative methods and records the node potentials and coordinates in an output file. This file serves as a permanent record of the solution that can be re-loaded for latter analysis.
6. You can use the interactive graphical environment of **Nelson** 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 **Nelson** read and generate several types of files. For a typical solution it is not necessary for your to deal directly with the files – the user interface takes care of data organization. On the other hand, if you generate a large amount of data it’s a good practice to archive application results in individual folders. Furthermore, there are advanced features of **Mesh** and **Nelson** that require direct entries to the input scripts. For convenience, all input scripts and output data files are in text format. Both **Mesh** and **Nelson** feature integrated text editors. Table 1 lists the file types and functions in the **Nelson** package. Note that the suffixes indicate the file function.

## 1.5 Theoretical background

The reference S. Humphries, **Field Solutions on Computers** (CRC Press, Boca Raton, 1997), Chap. 11 gives detailed information on the theory of non-radiative magnetic fields in conductive materials and the application of finite-element methods in **Nelson**. This section summarizes basic concepts that may be helpful in preparing solutions.

Static magnetic fields are created by applied currents in coils and atomic currents that result from domain alignment. Time-varying magnetic fields can generate a third type of current. The electric field associated with changing magnetic flux drives currents through conductive materials. Eddy currents are of particular concern in magnetic materials with high conductivity like iron. They can significantly reduce the effectiveness of transformers and other devices.

To derive approximate equations we assume that real currents in the solution volume are much larger than displacement currents. The condition is equivalent to

$$f \ll \frac{1}{2\pi\epsilon\rho}. \quad (1)$$

In ferrites with high volume resistivity, the limit of Eq. 1 holds for frequencies below about 1.0 MHz. The approximation holds for most frequencies of interest in iron. Dropping the displacement current, we can write Ampere's law as:

$$\frac{\nabla \times \mathbf{B}}{\mu} = \mathbf{J}_0 + \mathbf{J}_c. \quad (2)$$

In Eq. 2, the quantity  $\mathbf{J}_0$  is the applied current density in coils and  $\mathbf{J}_c$  is conductive current in metals and semiconductors. The magnetic flux density is related to the vector potential by:

$$\mathbf{B} = \nabla \times \mathbf{A}. \quad (3)$$

Using Eq. 3, we can express Faraday's law as

$$\nabla \times \mathbf{E} = -\frac{\partial}{\partial t}(\nabla \times \mathbf{A}). \quad (4)$$

If there are no free charges to create electrostatic-type fields, then Eq. 4 implies that the total electric field is related to the vector potential by

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

The conductive current is therefore given by

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

Combining Eqs. 2, 5 and 6, we find that the vector potential is given by:

$$\nabla \cdot \left( \frac{1}{\mu} \nabla \mathbf{A} \right) = -\mathbf{J}_0 - \sigma \frac{\partial \mathbf{A}}{\partial t} = -\mathbf{J}_0 - j\sigma\omega\mathbf{A}. \quad (7)$$

The final form applies to a system with steady-state harmonic drive currents at angular frequency. In two-dimensional solutions, there is only one non-zero component of vector potential. In this case, Eq. 7 has the form of the complex-number scalar Poisson equation. This equation

is easily solved with finite-element techniques. Given values of  $\mathbf{A}$ , the magnetic flux density  $\mathbf{B}$  and electric field  $\mathbf{E}$  can be determined from Eqs. 3 and 5.

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

$$B_x = \frac{\partial A_z}{\partial y}, \quad (8)$$

$$B_y = -\frac{\partial A_z}{\partial x}. \quad (9)$$

For problems in cylindrical coordinates, **Nelson** calculates the quantity  $rA_\theta$ , where  $A_\theta$  is the vector potential created by azimuthal currents. In this case the magnetic fields are given by

$$B_r = -\frac{1}{r} \frac{\partial}{\partial z} (rA_\theta), \quad (10)$$

$$B_z = \frac{1}{r} \frac{\partial}{\partial r} (rA_\theta). \quad (11)$$

The equations show that changes in values of the vector potential are related to the magnetic flux (the surface integral of the normal components of magnetic flux density). 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

$$A_z(x_0, y_2) - A_z(x_0, y_1) = \int_{y_1}^{y_2} B_x(y', x_0) dy'. \quad (12)$$

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 Eq. 3 implies that the magnetic flux density must be parallel to a surface of constant  $\mathbf{A}$ .

In cylindrical geometry, the symmetry of magnetic field lines dictates that the axis ( $r = 0.0$ ) must always be a line with  $rA_\theta = 0.0$ . **Nelson** automatically sets the condition for cylindrical problems that include  $r = 0.0$  by adding a constant potential region on the axis. The value of the stream function at the outer solution radius  $r_0$  is related to the flux of axial field by

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

For a uniform magnetic flux density  $B_0$ , Eq. 13 implies that  $r_0A_{\theta 0} = B_0r_0^2/2$ . As an example, Figure 2 shows the magnetic field lines around an iron cylinder immersed in a uniform solenoidal field of 0.1 tesla. A solenoidal field  $B_0 \cong 0.1$  tesla was created by setting  $rA_\theta = 5.0 \times 10^{-4}$  tesla-m<sup>2</sup> 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 **Nelson**:

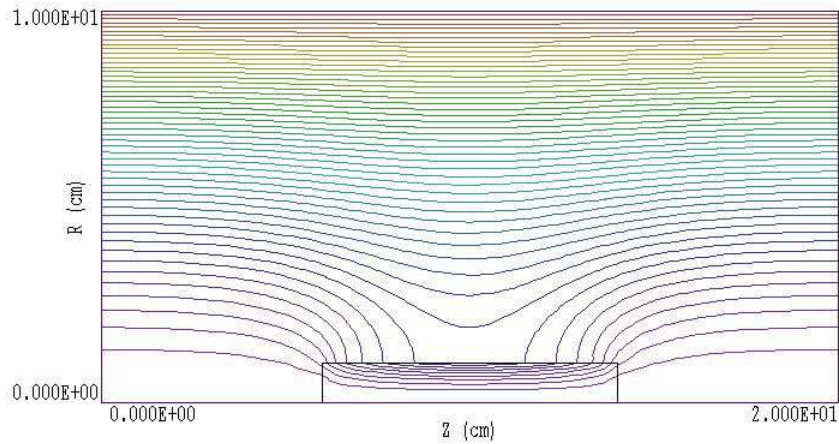


Figure 2: Setting boundary values of the vector potential to create a region of uniform flux.

- **Dirichlet.** Dirichlet boundary points have a fixed value of vector potential that does not change as the **Nelson** relaxation 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.** A Neumann boundary is one where the normal derivative of the vector potential is specified. The boundaries in **Nelson** are limited to the case where  $\partial \mathbf{A} / \partial \mathbf{n} = 0$ . The special Neumann condition implies that the magnetic field is normal to the boundary. One advantage of the finite-element method is that all boundaries that are not fixed automatically satisfy the special Neumann condition, even if they are slanted or curved. Neumann boundaries are often to reduce computation time by representing only half of a symmetric system.

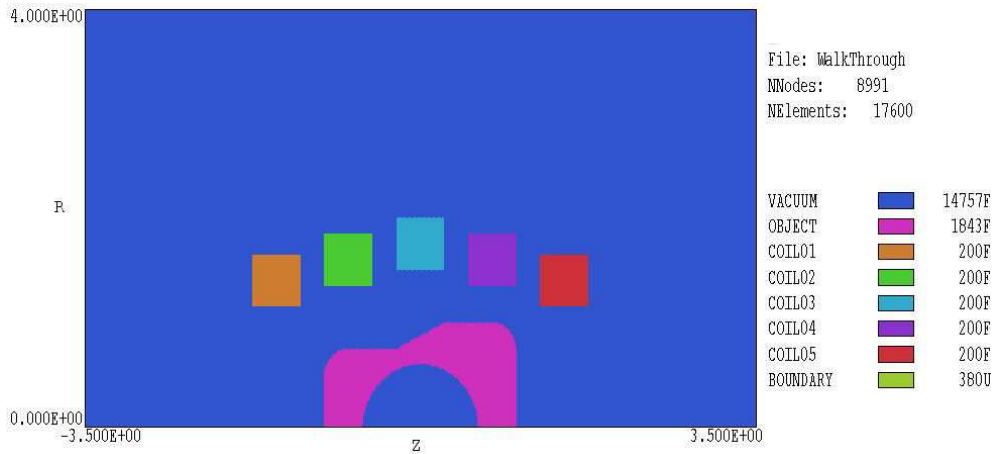


Figure 3: Region divisions for the WALKTHROUGH example, dimensions in cm.

## 2 Creating and analyzing a solution with Nelson

### 2.1 Mesh generation

As a quick introduction to **Nelson** techniques and capabilities, we shall work through the example shown in Fig. 3. The simulation addresses induction heating of the surface of a small metal object by a shaped coil. The system has cylindrical symmetry about the  $z$  axis (bottom). The five coils create a solenoid magnetic field at 2.0 kHz frequency. All coils are in phase and each coil carries a peak current of 1000 A. The hollow metal piece has relative magnetic permeability  $\mu_r = 3.0$  and conductivity  $\sigma = 8.33 \times 10^5$  S/m (volume resistivity  $\rho = 120.0 \mu\Omega\text{-cm}$ ). In a finite-element solution, a Neumann or Dirichlet condition must be specified along the boundaries of the solution volume. In this case, we use the condition  $rA_\theta = 0.0$ , equivalent to perfectly-conducting walls. The extra space around the assembly ensures that the boundary condition has a small effect on fields near the work piece. The goal of the calculation is to determine the distribution of electric and magnetic fields and the total volumetric resistive heating.

The **Mesh** script WALKTHROUGH.MIN (supplied in the **Nelson** example library) is required for the solution. Copy the file to a working directory and be sure that the *Data folder* in the **TC** program launcher is set correctly. Run **Mesh** and load WALKTHROUGH.MIN. Pick the *Edit script/graphics* command to enter the drawing editor. Here, you can use the display capabilities to confirm the following region assignments:

1. Air (Filled).
2. Ferromagnetic object (Filled).
3. Coil 1 (Filled).
4. Coil 2 (Filled).

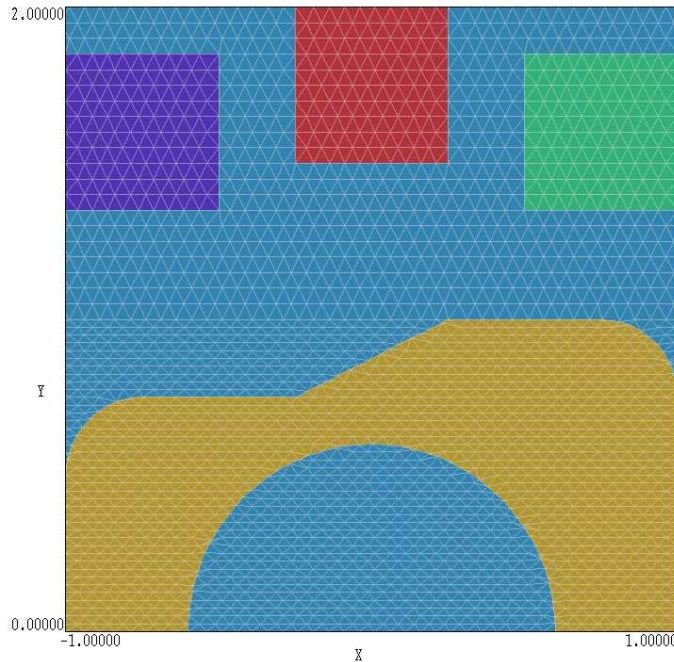


Figure 4: Completed mesh for the electron diode example.

5. Coil 3 (Filled).
6. Coil 4 (Filled).
7. Coil 5 (Filled).
8. Dirichlet boundary.

Abandon the drawing and return to the main menu. Click the *Process* command. A detail of the completed mesh is shown in Fig. 4. Note how the size of the elements varies for good resolution of skin-depth effects on the surface of the work piece. Choose the *Save mesh (MOU)* command. You can now close or minimize **Mesh**.

## 2.2 Creating the Nelson script

Run **Nelson** 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 magnetodynamic solution:

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

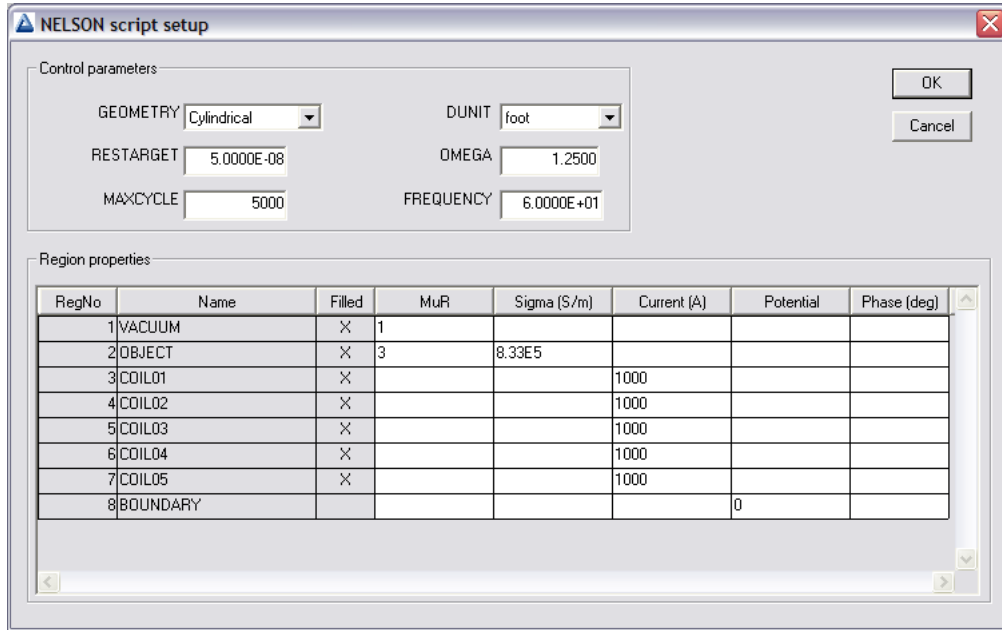


Figure 5: Dialog to set **Nelson** control parameters and region properties.

WALKTHROUGH.MOU. **Nelson** 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 five columns in the grid where you can enter values:  $\mu_r$  (relative magnetic permeability),  $\sigma$  (conductivity in S/m),  $I$  (total current through the region in amperes),  $V$  (vector potential  $A_z$  in tesla-m or stream function  $rA_\theta$  in tesla-m<sup>2</sup> for fixed regions) and  $\phi$  (phase of the potential or current in degrees).

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  $\theta$ .
- **FREQUENCY**. AC/RF frequency in Hz.
- **RESTARGET**. Accuracy tolerance for the iterative matrix solution of the finite-element equations.
- **MAXCYCLE**. Maximum number of cycles in the iterative solution.
- **DUNIT**. Choose an entry to set a factor to convert the units used for coordinates in the **Mesh** file to meters. The factor is the number of mesh units per meter: 39.37 for inches, 100.0 for cm.
- **OMEGA**. A parameter in the range 0.0 to 2.0 to control the iterative matrix solution.

The values entered in the region grid box (shown in Fig. 5) define the following characteristics:

- Region 1. Air region with  $\mu_r = 1.0$ .

- Region 2. Ferromagnetic object with  $\mu_r = 3.0$  and  $\sigma = 8.333 \times 10^5$  S/m.
- Regions 3-7. Coils with drive currents of 2000 A.
- Region 8. Fixed boundary with  $rA_\theta = 0.0$ .

Note that regions with a non-zero entry in the *MuR* column and no entry in the *Current* column are assigned zero drive current by default. Similarly, regions with non-zero current are assigned  $\mu_r = 1.0$  by default. Set up the dialog with the values shown in Fig. 5 and click *OK*. **Nelson** uses the information in the dialog to create the script `WALKTHROUGH.NIN` shown in Table 2. Chapter 3 reviews the script format and advanced program capabilities.

### 2.3 Finite-element solution

The next step is to generate and to solve the finite-element equations. Click the tool marked 2 or the menu command *Solve*. Accept the default entry of `WALKTHROUGH.NIN` in the dialog by clicking *OK*. The screen turns blue to indicate that the program is in solution mode and the status bar reports the progress of the operation. When complete, **Nelson** creates the output file `WALKTHROUGH.NOI` which contains full information on the mesh as well as values of the complex vector potential at each node. Section 3.5 describes the file format.

### 2.4 Solution analysis

We can now use the information in `WALKTHROUGH.NOI` to create plots and to perform quantitative analyses. Choose either the tool marked 3 or the *Analyze* menu command. In the dialog, accept the default choice of data file. **Nelson** loads the file, sets up the analysis menu and toolbar and creates the default contour plot shown in Fig. 6. Take a moment to inspect the organization of the working environment. The status bar at bottom lists the following information: solution file name, data file name, current analysis configuration, interpolation method and mouse snap status. The data file gives you the option to record the results of analysis operations in text format. The plot window is divided into four areas. The main plot is at the upper left and the plot legend is at the upper right. The orientation area at the lower right is a miniature representation of the solution volume with an outline showing the current view. Values appear in the information area at the bottom left in response to analysis commands such as *Point calculation*.

The contours in the plot of Fig. 6 lies along lines of constant  $rA_\theta$  at a reference phase of  $0.0^\circ$ . In a two-dimensional solution, the contours correspond to lines of  $\mathbf{B}$  (magnetic flux density). Because the lines are separated by equal intervals of magnetic flux, the spacing looks larger near the axis of a cylindrical solution. To plot another quantity, click on the command *Plots/Plot settings/Type* and choose the option *Filled contour*. Click on *Plots/Plot settings/Quantity* and choose  $|\mathbf{E}|$ , the amplitude of the toroidal electric field. If you zoom the view, you can reconstruct the plot of Fig. 7 which shows electric field in and near the work piece.

The following chapter gives detailed reference material on analysis functions of **Nelson**. You can experiment with the program capabilities to create a wide variety of plots. To complete the session, we shall run a script that controls automatic calculations. To inspect the file, click on *Edit script* in the *File* menu and pick the file `WALKTHROUGH.SCR`. Table 3 shows the file content. The first command ensures that the file `WALKTHROUGH.NOI` is loaded while the second command

Table 2: **Nelson** solution script for the WALKTHROUGH example

```
Mesh = walkthrough
Geometry = Cylin
DUnit = 3.2808E+00
ResTarget = 5.0000E-08
Omega = 1.2500E+00
MaxCycle = 5000
Freq = 6.0000E+01

* Region 1: VACUUM
Material(1) = 1.0000E+00 0.0000E+00
* Region 2: OBJECT
Material(2) = 3.0000E+00 8.3300E+05
* Region 3: COIL01
Material(3) = 1.0000E+00 0.0000E+00
Current(3) = 1.0000E+03 0.0000E+00
* Region 4: COIL02
Material(4) = 1.0000E+00 0.0000E+00
Current(4) = 1.0000E+03 0.0000E+00
* Region 5: COIL03
Material(5) = 1.0000E+00 0.0000E+00
Current(5) = 1.0000E+03 0.0000E+00
* Region 6: COIL04
Material(6) = 1.0000E+00 0.0000E+00
Current(6) = 1.0000E+03 0.0000E+00
* Region 7: COIL05
Material(7) = 1.0000E+00 0.0000E+00
Current(7) = 1.0000E+03 0.0000E+00
* Region 8: BOUNDARY
Potential(8) = 0.0000E+00 0.0000E+00

EndFile
```

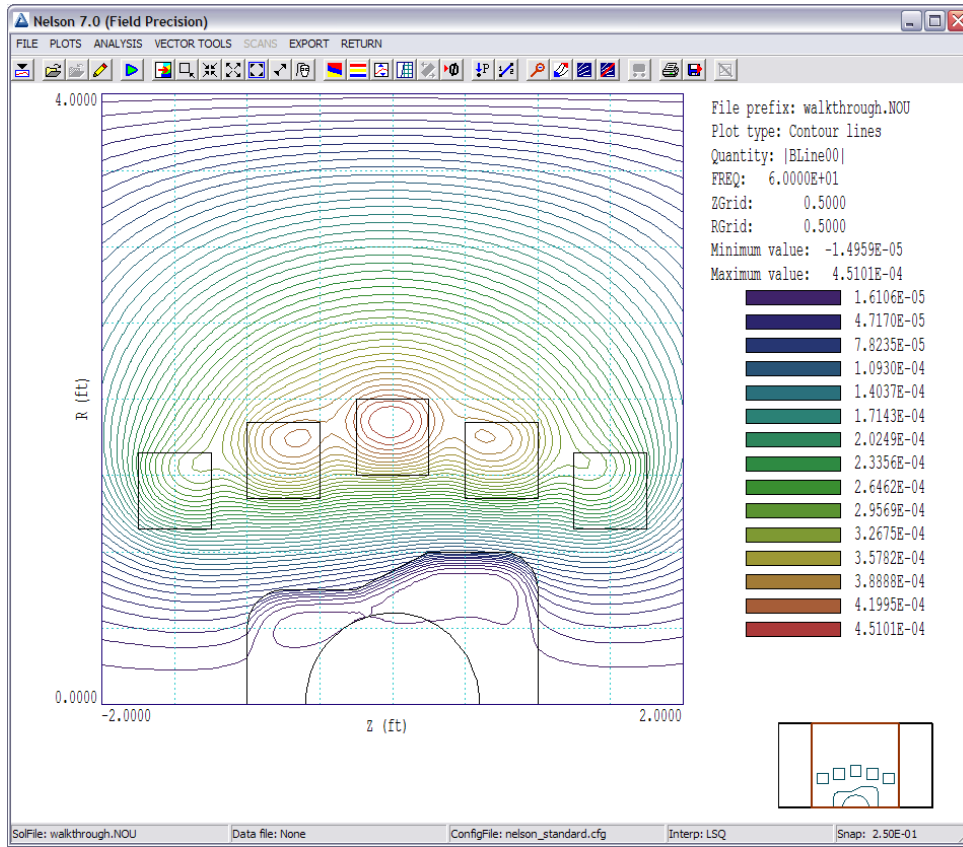


Figure 6: Working environment of **Nelson** in the analysis mode.

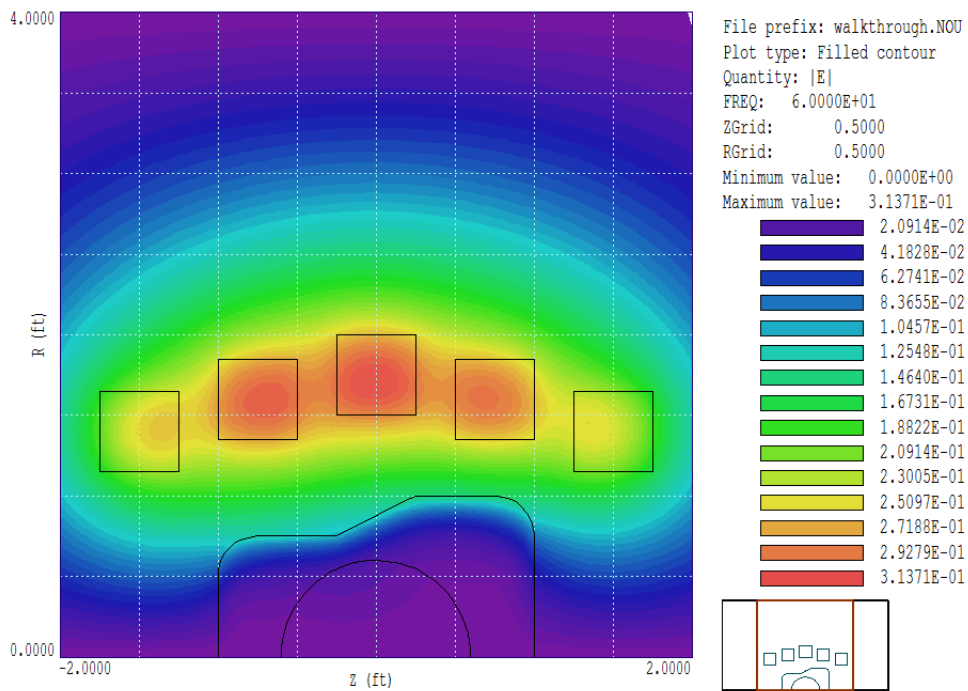


Figure 7: Amplitude of  $E_\theta$

Table 3: **Nelson** analysis script WAKLTHROUGH.SCR.

```
INPUT WalkThrough.NOI
OUTPUT WalkThrough.DAT
NSCAN 100
SCAN 0.0 0.0 0.0 1.5
VOLUMEINT 2
ENDFILE
```

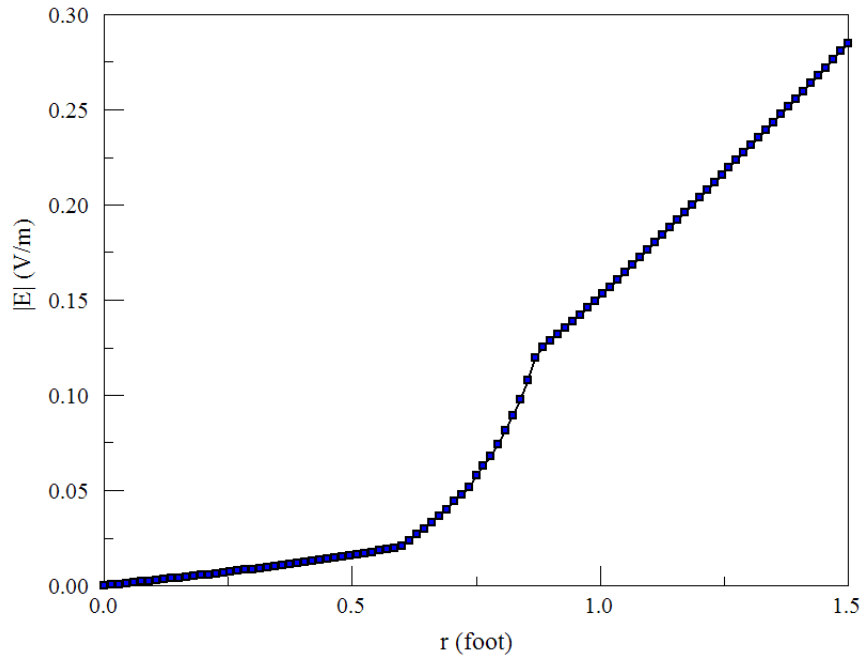


Figure 8: Amplitude of  $E_\theta$

opens the data file WALKTHROUGH.DAT. The program sets the number of scan points to 100 and computes a set of values at  $z = 0.0$  ft from  $r = 0.0$  ft to  $r = 1.5$  ft. Finally, **Nelson** analyzes Region 2 (the work piece) and closes the output file. It is easy to capture the values in the resulting text file and transfer them to other software. The plot of  $|\mathbf{E}|$  in Fig. 8 was created with PsiPlot. **Nelson** also has internal plot capabilities with digital oscilloscope features. The *VolumeInt* command initiates a volume integral and lists the resistive power dissipation as 167.5 W.

---

## 3 Nelson solution reference

### 3.1 Nelson script format

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

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

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

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

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

```

* File: REACTOR.WIN
* 1000.0 A per coil
* Skin depth: 0.15 m
* Omega = 8.52E7
* Rho = 1.208
Freq = 13.56E6
MaxCycle = 3000
Omega = 1.90
Geometry = Cylin
DUnit = 1000.0
* Solution volume
Material(1) = 1.0
* Dielectric chamber wall
Material(2) = 7.3, 0.0
* Plasma volume
Material(3) = 1.000, 1.208
* Right cap
Material(4) = 1.0, 0.0
* Mid Coil
Material(5) = 1.0, 0.0
Current(5) = 500.0, 0.0
* Right coil
Material(6) = 1.0, 0.0
Current(6) = 1000.0, 0.0
* Axis
Potential(7) = 0.0, 0.0
ENDFILE

```

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

```
Keyword  RegNo  Value
```

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

```
MATERIAL 2 50.0 3400.0
```

sets the properties of elements with region number 2 to  $\mu_r = 50.0$  and  $\sigma = 3400.0$  S/m. (**Note:** For back-compatibility, **Nelson** recognizes command formats from Version 1.0 through 5.0 where control commands start with the keyword *Set* and region commands start with *Region*.)

## 3.2 Program control commands

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

**MESH MPrefix****MESH PlasmaReactor**

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.NIN**, then **Nelson** will seek the default mesh file **FPrefix.MOU**.

**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 **Nelson**. 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 file **FPrefix.NOI** are always in meters. In an analysis session with **Nelson**, 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.)

**GEOMETRY [Rect, Cylin]****GEOMETRY = Cylin**

**Nelson** 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.

**FREQ Freq****FREQ = 3.45E6**

Specify the frequency (in Hz) for current and specific values of vector potential This command with a non-zero value is required for a solution.

**OMEGA Omega****OMEGA = 1.96**

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

**MAXCYCLE MaxCycle****MAXCYCLE = 500**

The maximum number cycles for the iterative matrix solution. The default value is 2500.

### **RESTARTGET ResTarget**

**RESTARTGET = 5.0E-6**

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

## **3.3 Commands for material properties**

The following three commands define the properties of regions. The commands may be generated with the *Setup* dialog in **Nelson** or directly with a text editor.

### **POTENTIAL RegNo VecPot [Phase]**

**POTENTIAL(3) = 0.0**

**POTENTIAL(5) = 5.0E-3 45.0**

The keyword *Potential* designates that the vector potential or stream function of the region nodes has fixed amplitude and phase and will not change during the matrix relaxation. The command sets the amplitude of the potential in tela-m or tesla-m<sup>2</sup> and the phase in degrees. The program uses a value 0.0° if the phase is omitted.

### **MATERIAL RegNo MuR [Sigma]**

**MATERIAL(5) = 1.0**

**MATERIAL(3) = 200.0 5000.0**

Set the properties of a ferromagnetic or air material. Enter the relative magnetic permeability ( $\mu_r = \mu/\mu_0$ ) and the conductivity  $\sigma$  in S/m. If the conductivity value is omitted, the default is  $\sigma = 0.0$

## **3.4 Running Nelson**

**Nelson** 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\]Nelson [DataPath\]FPrefix <ENTER>
```

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

The remainder of this section discusses commands in the main menu when **Nelson** 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 (NIN)**

**EDIT LISTING (NLS)**

**EDIT FILE**

These commands call the internal editor to inspect or to modify **Nelson** input and output files.

Choosing a file from an alternate directory does not change the working directory. The *Edit script (NIN)* command shows a list of all files with names of the form `FPREFIX.NIN` while *Edit listing (NLS)* displays files with names `FPREFIX.NLS`.

## RUN ANALYSIS SCRIPT

An analysis script allows you to perform complex or repetitive analyses on a set of similar solutions. This command displays a dialog listing files with the suffix `SCR`. Pick a file and click *OK*. The script can load data files, open and close data records, and perform any of the quantitative analysis functions described in this chapter. The script command language is described in Sect. 5.6. Note that the analysis script must be in the same directory as the data files.

## SETUP

The function of this command is to create a **Nelson** script to control an magnetodynamic 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. 3.2. You can enter basic physical properties of regions in the grid box (see Sect. 3.3).

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

## RUN

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

## STOP

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

## SOLUTION DISPLAY

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

## ANALYZE

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

## NELSON MANUAL

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

## SKIN DEPTH CALCULATOR

A handy utility to estimate the magnetic skin depth described in Sect. 4.2.

### 3.5 Format of the Nelson output file

The Nelson output file `FPrefix.NOU` is in text format. The file has three sections:

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

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

```
--- Run parameters ---  
XMin: -1.06681297E+00  
XMax:  1.06681297E+00  
KMax:  111  
YMin:  0.00000000E+00  
YMax:  1.21921483E+00  
LMax:   61  
DUnit:  3.28080000E+00  
NReg:   8  
ICylin:  1  
Frequency: 6.00000000E+01
```

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 gives the RF frequency in Hz.

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

--- Vertices ---

k	l	RgNo	RgUp	RgDn	x	y
1	1	9	1	0	-1.06681297E+00	0.00000000E+00
2	1	9	1	0	-1.03540759E+00	0.00000000E+00
3	1	9	1	0	-1.00406601E+00	0.00000000E+00
4	1	9	1	0	-9.72817581E-01	0.00000000E+00
5	1	9	1	0	-9.41696476E-01	0.00000000E+00
6	1	9	1	0	-9.10751862E-01	0.00000000E+00
7	1	9	1	0	-8.80057925E-01	0.00000000E+00
...						

A(Real)	A(Imag)	PowerUp	PowerDn
0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00

Each data line contains the following quantities:

- The indices of the node ( $K,L$ )
- The region number of the node ( $RgNo$ ) and region numbers for two associated elements ( $RgUp$  and  $RgDn$ ). The upper element lies above the line between nodes ( $K,L$ ) and ( $K+1,L$ ) and the lower element lies below the line.
- The coordinates of the node in meters, ( $x,y$ ) or ( $z,r$ ).
- The real and imaginary parts of the vector potential  $A_z$  at the node in tesla-m or the stream function  $rA_\theta$  in tesla-m<sup>2</sup>.
- The time-averaged resistive power density in the upper and lower elements at the node in W/m<sup>3</sup>.

The region section consists of four title lines following by  $NReg$  data lines, one for each region.

--- Regions ---

RegNo	Fix				VecPotAmp	VecPotPhase
1	0	0	0	0	0.00000000E+00	0.00000000E+00
2	0	0	0	0	0.00000000E+00	0.00000000E+00
3	0	0	0	0	0.00000000E+00	0.00000000E+00
...						
Mu	Sigma	CurrentReal	CurrentImag	Area		
1.25663700E-06	0.00000000E+00	0.00000000E+00	0.00000000E+00	2.37766263E+00		
3.76991100E-06	8.33000000E+05	0.00000000E+00	0.00000000E+00	1.07554126E-01		
1.25663700E-06	0.00000000E+00	4.30545946E+04	0.00000000E+00	2.32263249E-02		
...						

A one in the second column designates a fixed-potential region (electrode). Finally, **Nelson** records the region names passed from **Mesh**:

```
--- Region names ---  
VACUUM  
OBJECT  
COIL01  
COIL02  
COIL03  
COIL04  
COIL05  
BOUNDARY
```

---

## 4 Eddy current and energy loss in drive coils

### 4.1 Theory

In calculations of previous sections, drive coils were represented as regions with a specified current and zero conductivity. The solutions determined magnetic field penetration and energy losses in surrounding conductive structures. The underlying assumption was that regions carried either applied current density  $\mathbf{J}_0$  or induced conductive current density  $\mathbf{J}_c$  (see Eq. 2). In this limit, **Nelson** simply carries out a direct solution of Eq. 7.

In many applications, we want to find power loss in windings when the skin depth is comparable to or less than the width of the conductor. These solutions involve real coils with non-zero conductivity. In this case, regions carry both a virtual drive current  $I_0$  and a conductive current  $I_c$ . The quantities  $I_0$  and  $I_c$  are complex numbers that represent amplitude and phase. The total current that flows in a region is  $I_t = I_0 + I_c$ .

A self-consistent solution presents the following problem: although we must specify  $I_0$ , to initiate the calculation, the known quantity is  $I_t$  (the current supplied by the generator). The resolution is simple when the calculation includes a single conductor. We make an initial solution assigning a normalized virtual drive current  $I_0 = [1.0, 0.0]$ . The calculated vector potential gives the induced electric field and the conductive current density over the region. We can then perform an integral to find the total current for the normalized drive,  $i_{t1}$ . The total current generally has reduced amplitude and a different phase from the virtual drive current. Let the desired total current be represented by the complex number  $I_t$ . We could achieve the value by setting the virtual drive current to

$$I_0 = \frac{I_t}{i_{t0}}. \quad (14)$$

Calculations are more challenging with multiple drive coils. Here, the fields of one coil may induce conductive current in others. Because of the interactions, we cannot apply Eq. 14 individually to each region. Fortunately, solutions generated by **Nelson** are linear so we can use the principle of superposition. Suppose there are  $N$  coil regions with  $\sigma > 0.0$  and let  $I_{0n}$  represent the virtual drive current for coil  $n$ . The set of virtual drive currents can be represented as a column vector  $\mathbf{I}_0$  of dimension  $N$ . The drive currents gives rise to a set of total currents  $\mathbf{I}_t$ . Invoking the linearity of **Nelson** solutions, we can write the relationship as:

$$\mathbf{I}_t = \mathbf{A}_i \times \mathbf{I}_0, \quad (15)$$

where  $\mathbf{A}_i$  is a matrix with dimensions  $N \times N$ . We can invert Eq. 15 to find a relationship that gives the drive current necessary to achieve a given set of total currents:

$$\mathbf{I}_0 = \mathbf{A}_i^{-1} \times \mathbf{I}_t. \quad (16)$$

If the vector  $\mathbf{I}_t$  represents the set of known total currents supplied by the generator, then Eq. 16 determines the virtual drive currents necessary to achieve the solution.

The main effort is to find the matrix  $\mathbf{A}_i$ . To evaluate the components, **Nelson** performs  $N$  preliminary calculations with unity drive currents. For example, suppose that  $N = 3$ . The virtual drive current vector

$$I_{02} = \begin{bmatrix} (0.0, 0.0) \\ (1.0, 0.0) \\ (0.0, 0.0) \end{bmatrix}. \quad (17)$$

gives a set of normalized total currents that may be associated with the components in the second column of the matrix:

$$a_{12} = i_{t1}, \quad (18)$$

$$a_{22} = i_{t2}, \quad (19)$$

$$a_{32} = i_{t3}. \quad (20)$$

The program must perform  $N$  preliminary calculations to fill all columns of the matrix.

In summary, **Nelson** performs the following operations when it detects that the solution contains regions with drive current and non-zero conductivity:

1. Count and order the coil regions.
2. Perform  $N$  preliminary calculations with excitation by unit drive vectors to evaluate the  $N^2$  complex number matrix coefficients.
3. Invert the matrix and determine the virtual drive currents to achieve the target total currents.
4. Carry out the main solution with the calculated virtual drive currents and save the results.

All operations are performed automatically so that no additional information is required from the user. The calculation does requires more time. For example, a run with 10 conductive drive regions takes 11 times longer than a simple solution.

## 4.2 Examples

Two following two examples illustrate code features to find self-consistent current distributions in conductive coils. The first involves a single conductor, a copper bus bar that carries an AC current with peak amplitude 1000.0 A. The input file `BBAR.MIN` describes a long cylindrical with radius 1.0 cm in a large solution volume to approximate free space. Region 1 represents air, the bus bar is Region 2 and Region is the solution volume boundary with  $A_z = 0.0$

The *Skin depth calculator* (Fig. 9) in the **Nelson Tools** menu is useful to estimate regimes where we can expect strong contributions from eddy currents. It solves the equation:

$$\delta = \sqrt{\frac{1}{\pi f \sigma \mu_r \mu_o}}, \quad (21)$$

where  $\delta$  is the skin depth,  $\sigma$  is the electrical conductivity,  $f$  is the frequency and  $\mu_r$  is the relative magnetic permeability. To use the tool, fill in three known values and leave one field

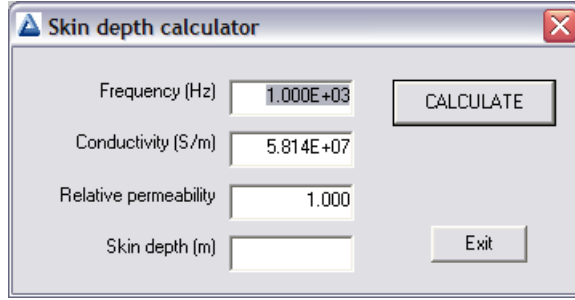


Figure 9: Skin depth calculator.

Table 4: Contents of the file BBAR.NIN

```

Mesh = BBar
Geometry = Rect
DUnit = 1.0000E+02
ResTarget = 2.0000E-07
MaxCycle = 5000
Freq = 175.0
Omega = 1.50
* Region 1: AIR
Material(1) = 1.0000E+00 0.0000E+00
* Region 2: COIL
Material(2) = 1.0000E+00 5.814E7
Current(2) = 1.0000E+03 0.0000E+00
* Region 3: BOUNDARY
Potential(3) = 0.0000E+00 0.0000E+00
EndFile

```

blank. **Nelson** fills the missing field when you press the *Calculate* button. In this case, we use the values  $\sigma = 5.814 \times 10^7$  S/m (copper),  $\mu_r = 1.0$  and  $\delta = 0.5 \times 10^{-2}$  m. The resulting transition frequency is 174.3 Hz. Accordingly, we shall investigate the frequency range 5.0 Hz to 250 Hz. Note that there is a practical upper limit on the frequency. If the skin depth is comparable to or less than the element width in Region 2 (0.05 cm), the calculation may not converge.

Table 4 shows the complete **Nelson** input script. Region 2 has the properties  $\mu_r = 1.0$  and  $\sigma = 1/\rho = 5.814 \times 10^7$ . The target total current is 1000.0 A at 0.0° phase. The code will automatically adjust the drive current to achieve this value. The calculation involves two matrix solutions: one to find the normalization factor and one to generate the normalized field distribution. We can check the validity of the code by comparing values at low frequency where the current density is approximately uniform with radius. At a current amplitude of 1000.0 A, we predict the following values: current density  $j_z = 3.183 \times 10^6$  A/m<sup>2</sup>, electrical field  $E_z = 0.0548$  V/m and power density  $p = 8.713 \times 10^4$  W/m<sup>3</sup>. The power loss integrated over the cross section is 27.37 W/m. An inspection of the listing at the end of the file BBAR.NLS shows the Nelson result at 2.0 Hz is 27.39 W/m.

At high frequency, the current density is forced toward the outer surface of the rod over a

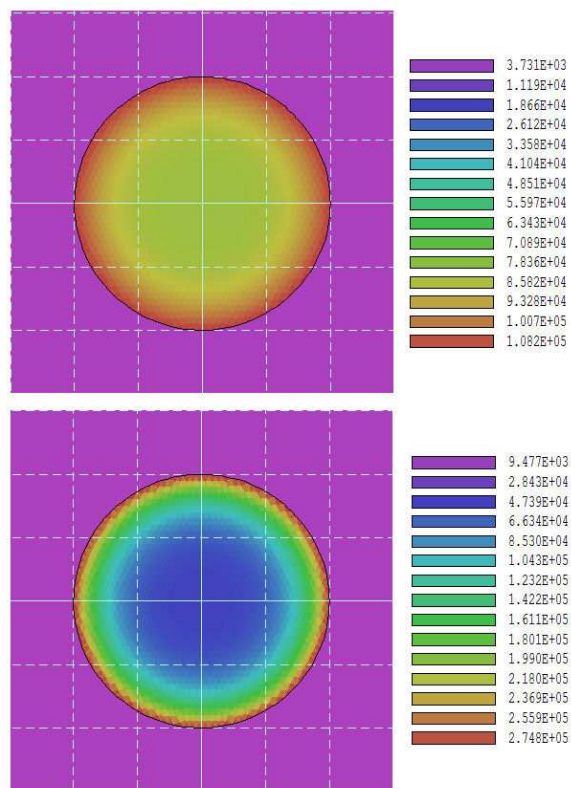


Figure 10: Resistive power density in the copper rod at different frequencies for example BBAR. Top: 75.0 Hz. Bottom: 175.0 Hz.

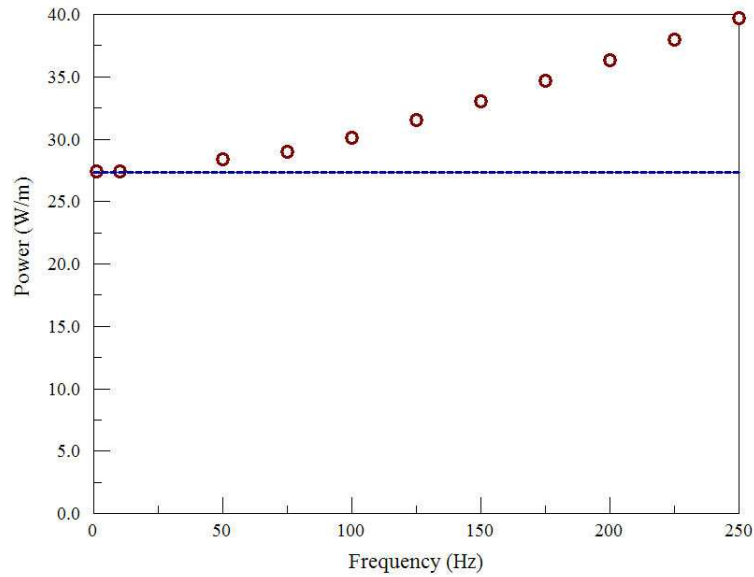


Figure 11: Power loss per length as a function of frequency in the copper rod of example BBAR. The dashed line is the predicted zero-frequency value.

width comparable to the skin depth. Figure 10 shows the resulting distribution of resistive power loss. The non-uniform current density also leads to an increase in the total power dissipation. Power loss per length in the bus bar as a function of frequency is plotted in Fig. 11.

The MULTICOIL example illustrates **Nelson** capabilities for in a more complex system. A 4-turn copper solenoid creates an axial magnetic field that acts on an enclosed stainless steel pipe. The target total current in each coil is 2000.0 A at 0.0° phase. Figure 12 shows the highly non-uniform power profile. As a result of eddy-current effects, the current density is highly enhanced at ends of the outer turns. At the end of the run, **Nelson** creates the following listing of information in the NLS file:

Global solution properties

Volume (m3): 1.41371669E-02  
 Power (W): 7.18692965E+02

Solution analysis

NReg	Drive current (A)		Total current (A)		Power (W)
	Amplitude	Phase	Amplitude	Phase	
2	0.0000E+00	( 0.00)	2.2830E+03	(-106.40)	2.6125E+02
3	1.7134E+04	( 78.31)	2.0000E+03	( 0.00)	1.1917E+02
4	2.0217E+04	( 78.80)	2.0000E+03	( 0.00)	1.0957E+02
5	2.0217E+04	( 78.80)	2.0000E+03	( 0.00)	1.0956E+02
6	1.7134E+04	( 78.31)	2.0000E+03	( 0.00)	1.1914E+02

The coils drive a current with amplitude 2.83 kA and -106.4° phase in the pipe. To achieve the desired total currents at the high frequency, the drive currents require strong adjustment. The drive current value in the outer coils is 17.13 kA at phase 78.31°.

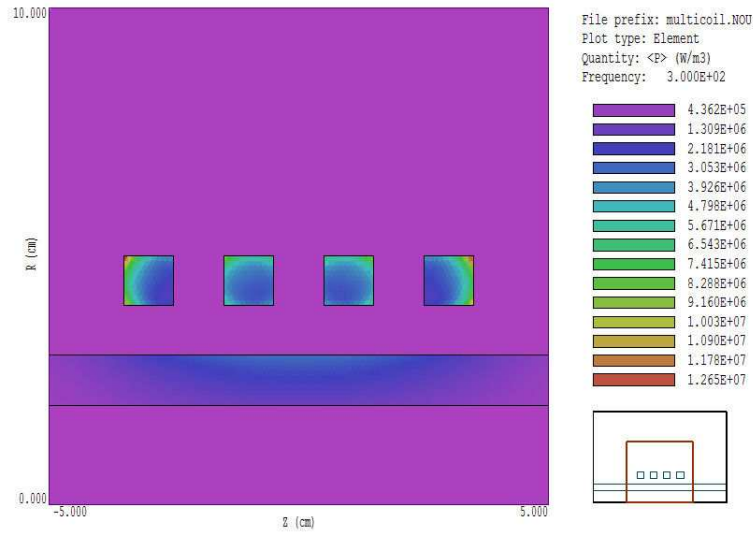


Figure 12: Geometry of the MULTICOIL example showing the power deposition distribution in the coil and workpiece at  $f = 300.0$  Hz.

Note: The following conventions are applied in the analysis routines described in the following chapter for regions that represent drives with non-zero conductivity. **Nelson** calculates *induced* electric field and current density from the vector potential using Eq. 5. To find the *total* electric field and current density, we must add the resistive components created by the drive current. The total values of electric field and current density are displayed for conductive coil regions in element, surface and scan plots. In these plots, the electric field appears to be discontinuous passing from a conducting coil to an insulator. The code does not compute resistive fields in insulators because they play no role in inductive coupling to other structures. Only the inductive components of  $\mathbf{E}$  and  $\mathbf{j}$  are displayed in contour plots.

---

## 5 TriComp analysis functions

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

### 5.1 File menu commands

#### LOAD SOLUTION FILE

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

#### LOAD CONFIGURATION FILE

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

#### LOAD FROM SERIES

Initial-value programs like **TDiff** may produce several solutions files in a run at different simulation times. Use this command to load other files in a series of solutions. The program displays the dialog of Fig. 13 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**.

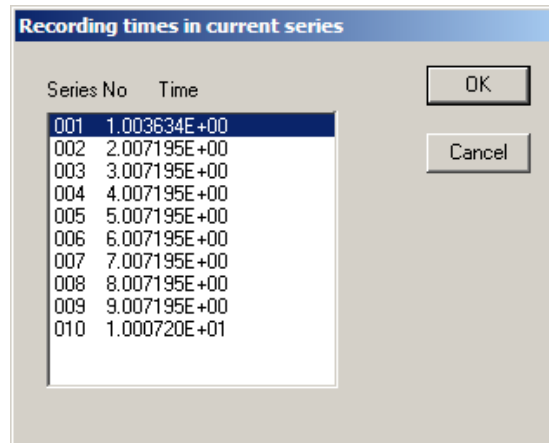


Figure 13: Dialog to pick a solution in a series generated by an initial-value program.

### CLOSE DATA RECORD

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

### RUN SCRIPT

A analysis script allows you to perform complex or repetitive calculations on a set of similar solutions. This command displays a dialog listing files with the suffix `SCR`. Pick a file and click *OK*. The script can load data files, open and close data records, and perform any of the quantitative analysis functions described in this chapter. The script command language is described in Sect. 5.6. 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 *S**Prefix* in the dialog – the resulting script will be saved with the name *S**Prefix*.`SCR`. The program opens the file in the editor and writes a reference list of allowed commands. The list follows the *EndFile* command and will be ignored by the script parser. Enter commands above the *EndFile* command.

### EDIT FILES

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

### EXIT

Exit the program.

## 5.2 Plot menu commands

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

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

The *Settings* popup menu contains the following commands.

### PLOT TYPE

Change the plot type.

### PLOT QUANTITY

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

### PLOT LIMITS

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

### TOGGLE GRID DISPLAY

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

### GRID CONTROL

This command displays the dialog of Fig. 14 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  $y$ .

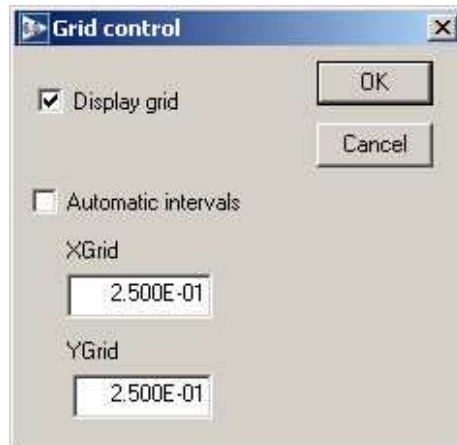


Figure 14: Grid control dialog

### 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

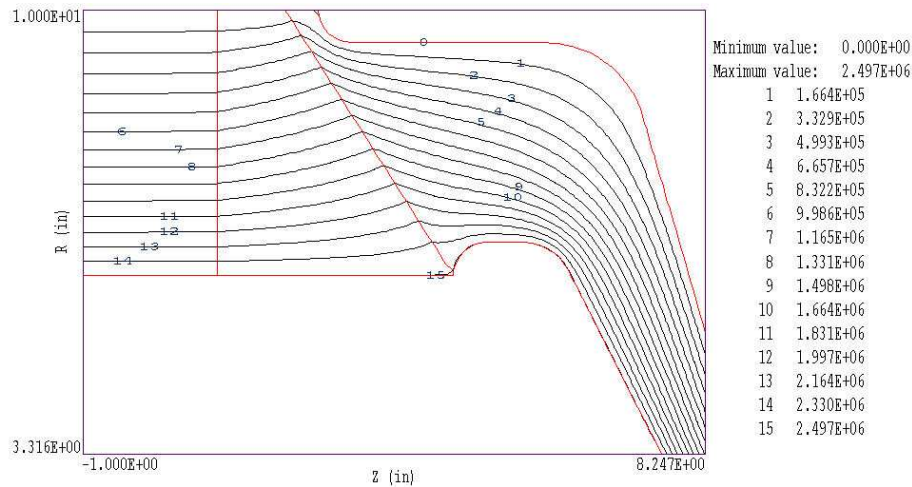


Figure 15: Monochrome contour plot with labels

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. 15). Overlapping labels on closely-spaced lines may look better in a zoomed view.

### NUMBER OF CONTOURS

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

### TOGGLE NUMBER FORMAT

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

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

### ZOOM WINDOW

#### ZOOM IN

#### EXPAND VIEW

#### GLOBAL VIEW

#### PAN

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

### ROTATE 3D IMAGE

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

## VIEW ANGLE 3D

Set the elevation angle for the view point.

## SET GRID 3D

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

## 5.3 Analysis menu commands

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

### POINT CALCULATION

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

### LINE SCAN

After clicking on the command, supply two points with the mouse or keyboard to define a scan line. The snap mode is useful in this application (for example, you may want a scan to extend from 0.000 to 5.000 rather than 0.067 to 4.985.) The program computes a series of values of field quantities at equal intervals along the line. The program makes a screen plot of the currently-selected quantity versus distance along the scan and activates the commands in the *Scan plot* menu (Sect.5.4). 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

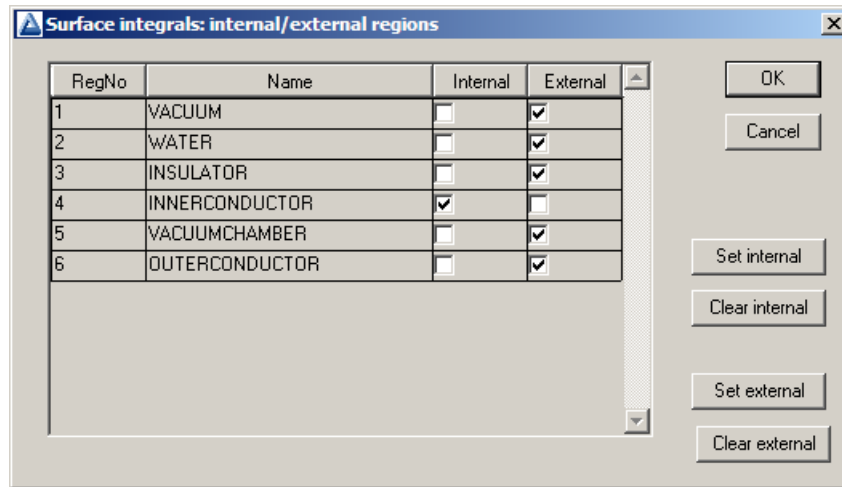


Figure 16: Surface integral dialog

program computes the integral over the surfaces of the internal region(s) in contact with the external regions.

## LINE INTEGRALS

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

## MATRIX FILE

The program can write matrix files of field values to help you create your own analysis routines. Although information is available in the output file of the solution program, it may be difficult to deal with the conformal triangular mesh. The *Matrix file* command uses the interpolation capabilities of the program to create a text data file of field quantities on a rectangular grid in  $x$ - $y$  or  $z$ - $r$ . The command displays the dialog of Fig. 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.

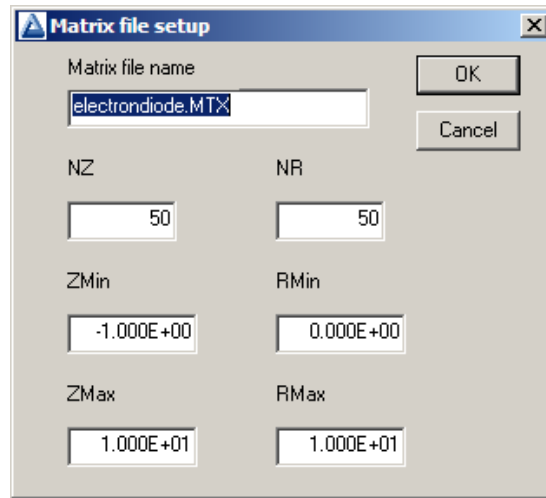


Figure 17: Dialog to generate a matrix file.

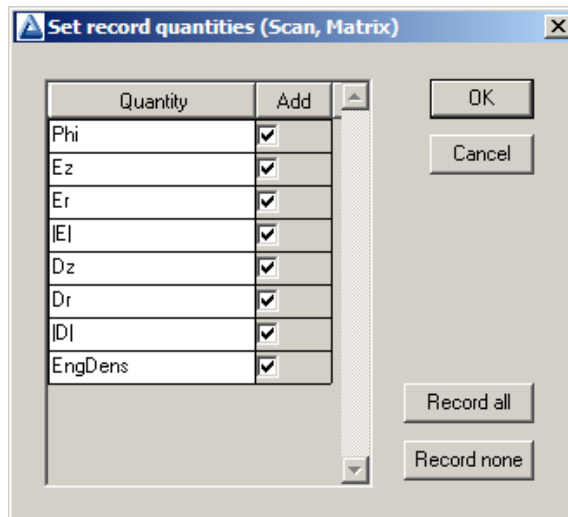


Figure 18: Dialog to set quantities recorded in response to the *Line scan* and *Matrix commands*.

### SCAN PLOT QUANTITY

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

### NUMBER OF SCAN POINTS

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

### SET RECORDED QUANTITIES

The default for data records of line scans and matrix files is to include all quantities defined in the **INTERPOLATION** section of the current configuration file. This state may result in large files

with unnecessary information. Use this command to set the quantities that will be included. In the dialog of Fig. 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.

## 5.4 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.

### 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 and return to the *Analysis* menu.

## 5.5 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.

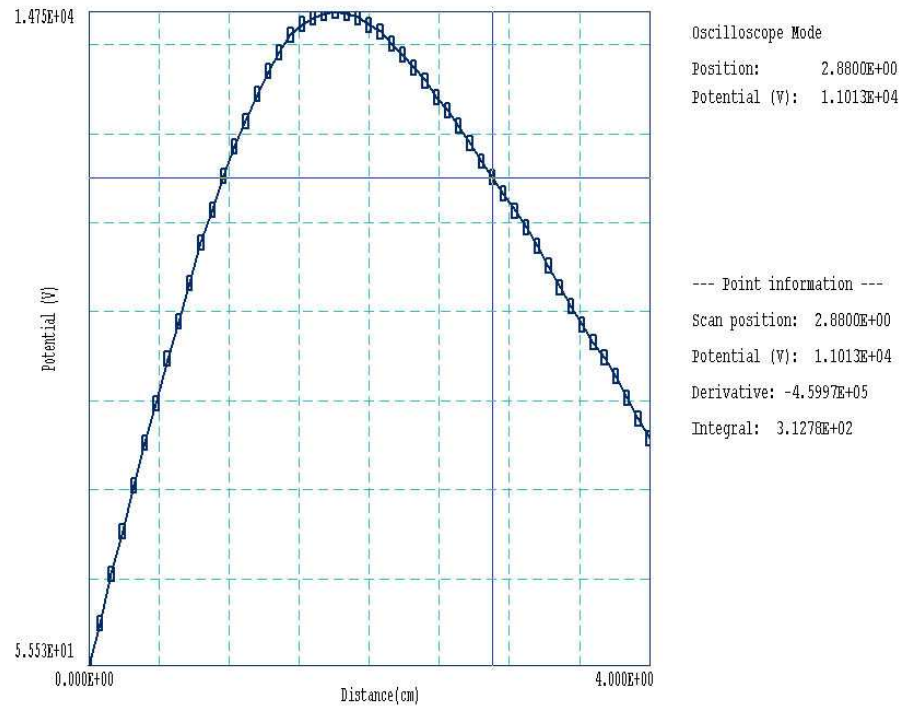


Figure 19: Scan plot in the oscilloscope mode

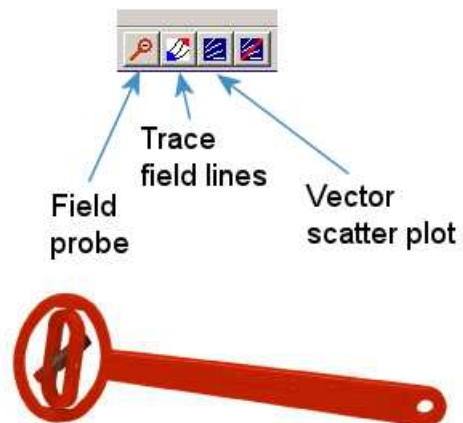


Figure 20: Tools to display vector information in the slice plot menu.

## 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 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.

## 5.6 Analysis script commands

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

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

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

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

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

The following commands may appear in a script:

**INPUT FileName****INPUT Switch1.EOU**

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

**OUTPUT FileName****OUTPUT SW02.DAT**

Close the current data file and open an output file **FileName**.

**CONFIGURATION [datapath\] FileName****CONFIGURATION = C:\FieldP\TriComp\ESTAT\_FORCE.CFG**

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

**SCAN Xs Ys Xe Ye****SCAN Zs Rs Ze Re****SCAN = (0.00, 0.00) (10.00, 0.00)**

Write the results of a line scan between the specified points to the data file. The four real number parameters are the starting and end coordinates in **Mesh** units.

**GENSCAN**

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

```
GENSCAN
  xp1 yp1
  xp2 yp2
  ...
  xpn ypn
END
```

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

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

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

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

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

**MATRIX FName Nx Ny Xs Ys Xe Ye****MATRIX FPrefix Nz Nr Zs Rs Ze Re****MATRIX = Switch1 (10, 20) (0.00, 0.00, 5.00, 10.00)**

Open a matrix file and record values. The command requires seven parameters: 1) The name of the matrix file (string), 2) the number of intervals along the  $x$  or  $z$  direction (integer). 3) the number of intervals along the  $y$  or  $r$  direction (integer), 4-7) coordinates of the corners of a box in the solution volume (real). The suggested suffix for matrix files is *MTX*.

**NSCAN NScan****NSCAN = 150**

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

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

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

**ENDFILE**

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

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

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

Table 5: **Nelson** standard units

Quantity	Unit
Spatial dimensions	meters or units set by <i>DUnit</i>
Vector potential	tesla-m
Phase	Degrees
Electric field	V/m (volts/meter)
Current	A (amperes)
Current density	A/m <sup>2</sup> (amperes per square meter)
Conductivity	S/m (Siemens/m)
Power density	W/m <sup>3</sup> (watts per cubic meter)

## 6 Standard Nelson calculated quantities

Table 5 shows the standard units used for **Nelson** analyses. The package includes four standard configuration files for the analysis of complex-number solutions of AC/RF magnetic fields:

- `nelson_standard.cfg`: amplitudes of quantities and instantaneous values at phases of 0.0° and 90.0°.
- `nelson_reference.cfg`: instantaneous quantities at a specified phase.
- `nelson_phase.cfg`: phases and amplitudes of field quantities
- `nelson_complex.cfg`: complex number representation of field quantities.

The default `nelson_standard.cfg` is loaded the first time you run the program. Thereafter, **Nelson** loads the configuration used in the last analysis session. The next section reviews **Nelson** variables that may be included in the definition of configuration quantities. Section 6.2 covers the content of `nelson_standard.cfg` while Sect. 6.3 summarizes the other configurations.

### 6.1 Nelson variables

**Nelson** supplies values of the following field quantities at the present 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).

- `&AR`: the real part of the quasi-static vector potential  $A_z$  (tesla-m, rectangular geometry) or  $rA_\theta$  (tesla-m<sup>2</sup>, cylindrical geometry) V.
- `&AI`: the imaginary part of the quasi-static vector potential:  $A_z$  (tesla-m, rectangular geometry) or  $rA_\theta$  (tesla-m<sup>2</sup>, cylindrical geometry) V.

- **&BxzR**: the  $x$  (planar) or  $z$  (cylindrical) component of the real part of the magnetic flux density in tesla.
- **&BxzI**: the  $x$  (planar) or  $z$  (cylindrical) component of the imaginary part of the magnetic flux density in tesla.
- **&ByrR**: the  $y$  (planar) or  $r$  (cylindrical) component of the real part of the magnetic flux density in tesla.
- **&ByrI**: the  $y$  (planar) or  $r$  (cylindrical) component of the real part of the magnetic flux density in tesla.
- **&ER**: the real part of the electric field  $E_z$  (planar) or  $E_\theta$  (cylindrical).
- **&EI**: the imaginary part of the electric field  $E_z$  (planar) or  $E_\theta$  (cylindrical).
- **&PDens**: the time-averaged resistive power dissipation in  $\text{W}/\text{m}^3$ .

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

- **\$RegAAmp**: the amplitude of  $A_z$  (planar, tesla-m) or  $rA_\theta$  (cylindrical, tesla-m<sup>2</sup>) for a region of fixed vector potential.
- **\$RegAPhase**: the phase of  $A_z$  (planar) or  $rA_\theta$  (cylindrical) for a region of fixed vector potential in radians.
- **\$RegMu**: the relative magnetic permeability of a material region.
- **\$RegSigma**: the electrical conductivity of a material region in  $\text{S}/\text{m}$ .
- **\$RegCDensR**: the real part of the current density assigned to a region in  $\text{A}/\text{m}^2$ .
- **\$RegCDensI**: the imaginary part of the current density assigned to a region in  $\text{A}/\text{m}^2$ .
- **\$RegArea**: the cross-section area of the region in  $\text{m}^2$ .

**Nelson** records a single run parameter in the solution files. The parameter **\$Freq** is the RF frequency in Hz.

## 6.2 Standard analysis configuration

The configuration file `nelson_standard.cfg` is shown in Table 6. It defines the program parameters  $\mu_0$  that may be used in expressions. The solution engine passes one run parameter, the AC/RF frequency. To avoid confusion, we must be careful to distinguish between the magnitude of a complex quantity (peak value during the harmonic variation) and the spatial magnitude of a vector quantity, such as:

$$|\mathbf{B}| = B_x^2 + B_y^2. \quad (22)$$

Table 6: Contents of the file `nelson_standard.cfg`

```

PROGPARAM
  $Mu0 = 1.256637E-6
END
RUNPARAM
  $Freq = 0.0
END
REALTIMEPARAM
END
INTERPOLATION
* Magnetic field lines at 0 deg
  |BLine00| = &AR
* Magnetic field lines at 90 deg
  |BLine90| = &AI @CHS
* Amplitude of magnetic field
  |B| = &BxzR 2 ^ &BxzI 2 ^ + &ByrR 2 ^ + &ByrI 2 ^ + @SQRT
* Amplitude of electric field
  |E| = &ER 2 ^ &EI 2 ^ + @SQRT
* Resistive current density
  |JRes| = &ER 2 ^ &EI 2 ^ + @SQRT $RegSigma *
* Drive current density
  |JDrive| = $RegCDensR 2 ^ $RegCDensI 2 ^ + @SQRT
* Time-averaged power density
  PowerDens = &PDens
END
VECTOR
  B00 = &BxzR;&ByrR
  B90 = &BxzI @CHS;&ByrI @CHS
END
VOLUME
  Power = &PDens
END
SURFACE
END
ENDFILE

```

Table 7: Reference-phase quantities in `nelson_reference.cfg`.

Section	Quantity	Units
Interpolation	$A_z$ or $rA_\theta$ (lines of $\mathbf{B}$ )	tesla-m/tesla-m <sup>2</sup>
	$B_x$ or $B_z$ (magnetic flux density)	tesla
	$B_y$ or $B_r$ (magnetic flux density)	tesla
	$ \mathbf{B} $ (flux density magnitude)	tesla
	$\mu_r$ (relative magnetic permeability)	
	$\sigma$ (electrical conductivity)	S/m
	$j_{zc}$ or $j_{\theta c}$ (eddy current density)	A/m <sup>2</sup>
Vector	$\mathbf{B}$ (magnetic flux density)	tesla

We shall use the following notations: 1) a quantity in bold type enclosed between single vertical lines represents a vector magnitude as in Eq. 22, 2) a non-vector quantity enclosed between vertical lines represents the amplitude of the harmonic variation and 3) a vector quantity enclosed between double vertical lines is the peak value of the vector magnitude.

The first quantities in the INTERPOLATION section are instantaneous vector potential values ( $A_z$  or  $rA_\theta$ ) at reference phases  $\phi_{ref} = 0.0^\circ$  and  $90.0^\circ$ . Contour lines of these quantities follow lines of magnetic field at the two phases. Note that lines of  $|A_z|$  or  $|rA_\theta|$  do not necessary define magnetic field lines.

The amplitude of the magnetic flux density is given by

$$\|\mathbf{B}\| = \sqrt{B_{xR}^2 + B_{xI}^2 + B_{yR}^2 + B_{yI}^2}. \quad (23)$$

The amplitude of the eddy current density is related to the amplitude of the electric field by

$$\|\mathbf{j}_c\| = \sigma \|\mathbf{E}\|. \quad (24)$$

The drive current density has non-zero values only in regions that represent coils.

The vector tools show the direction of magnetic flux density at phases of  $0.0^\circ$  and  $90.0^\circ$ . To plot field direction at other phases, load the configuration `nelson_reference.cfg`. The single volume-integral quantity is the total resistive power dissipation (in W/m for planar solutions and W for cylindrical solutions).

### 6.3 Additional configurations

The remaining configurations provide good illustrations of how to manipulate and to display complex-number quantities. You can inspect the text files with an editor. They can serve as useful templates for building your own expressions. Use `nelson_reference.cfg` to show snapshots of the harmonic quantities at a specified reference phase. The configuration file defines the real-time parameter `$Phi0`, the reference phase in degrees. The default is  $0.0^\circ$ . You can change the value during an analysis with the reference-phase tool or the command `Analysis/Analysis settings/Set reference phase`. Table 7 lists the available quantities.

The file `nelson_phase.cfg` illustrates how to display harmonic quantities by amplitude and phase. Use the file to plot or to calculate the amplitude and phase of components of the magnetic flux density,  $B_x/B_z$  and  $B_y/B_r$ . Finally, the file `nelson_complex.cfg` uses complex-number quantities directly. Table 8 lists the contents.

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

```
PROGPARAM
END

RUNPARAM
  $Freq = 0.0
END

REALTIMEPARAM
END

INTERPOLATION
  AzReal/rAtReal = &AR
  AzImag/rAtImag = &AI
  BxReal/BzReal = &BxzR
  BxImag/BzImag = &BxzI
  ByReal/BrReal = &ByrR
  ByImag/BrImag = &ByrI
  |B|Real = &BxzR 2 ^ &ByrR 2 ^ + @SQRT
  |B|Imag = &BxzI 2 ^ &ByrI 2 ^ + @SQRT
  EzReal/EThetReal = &ER
  EzImag/EThetImag = &EI
  JzReal/JThetReal = &ER $RegSigma *
  JzImag/JThetImag = &EI $RegSigma *
  JDriveReal $RegCDensR
  JDriveImag $RegCDensI
END

VECTOR
END

VOLUME
END

SURFACE
END

ENDFILE
```

---

## 7 Building custom analysis configurations

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

### 7.1 Configuration file structure

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

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

You may define custom quantities for display and analysis in the last four sections. In plotting, run speed is a critical issue. The generation of a single element plot may require over 50,000 interpolations. To ensure fast operation, expressions for calculated quantities in the configuration file are written in reverse Polish notation (RPN). The program parses the function strings once while loading the configuration file and saves them in a coded form. Thereafter, operations are performed at the speed of compiled code. The following section reviews the basics of RPN notation. Section 7.3 covers parameters and variables that may appear in expressions.

Table 9: 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

```

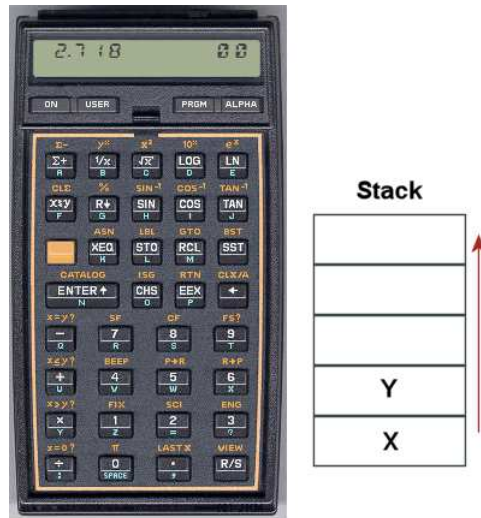


Figure 21:

## 7.2 Introduction to RPN

Consider the expression

$$5.0 \times 4.5^3 + \left[ 9.2 + 0.6 \times 0.9^{(2+0.67)} \right]. \quad (25)$$

Equation 25 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 10: Unary operators

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

- Subtraction (-):  $Y - X$
- Multiplication (\*):  $Y \times X$
- Division (/):  $Y/X$
- Exponentiation (^):  $Y^X$

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

### 7.3 Operators, parameters and variables

Expressions to define quantities may contain numbers, unary and binary operations, parameters, standard variables and special variables for the program. Unary operators have names that begin with '@'. Table 10 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, **D**/ $\epsilon_0$ ), **&Dyr** (the *y* or *r* component of the normalized electric displacement), **&EMag0** (the magnitude of the normalized electric displacement), **&EpsSig1** and **&EpsSig2** (the relative dielectric constant or the electrical conductivity), and **&Rho** (the space-charge density for dielectric solutions). In addition, region variables may be defined. These are quantities that have fixed values over a region. The current value corresponds to the region that contains the current location. As an example, **EStat** supports the following region variables:

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

## 7.4 Building expressions

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

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

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

`RectName/CylinName`

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

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

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

```
NAME = EXPRESSION01;EXPRESSION02
```

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

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

# Index

## Analysis commands

- 3D angle, [36](#)
- 3D grid, [36](#)
- Close data record, [32](#)
- Contour style, [34](#)
- Create analysis script, [32](#)
- Field lines at points, [41](#)
- Grid control, [33](#)
- Interpolation method, [37](#)
- Line integrals, [37](#)
- Line scan, [36](#)
- Load configuration file, [31](#)
- Load from series, [31](#)
- Load solution file, [31](#)
- Matrix file, [37](#)
- Number display format, [35](#)
- Number of contours, [35](#)
- Number of scan points, [38](#)
- Open data record, [31](#)
- Oscilloscope mode, [39](#)
- Pick vector quantity, [41](#)
- Plot limits, [33](#)
- Plot quantity, [33](#)
- Plot type, [33](#)
- point calculation, [36](#)
- Probe, [41](#)
- Remove vectors, [41](#)
- Rotate 3D, [35](#)
- Run analysis script, [32](#)
- Scan plot quantity, [38](#)
- Set recorded quantities, [38](#)
- Snap distance, [34](#)
- Surface integrals, [36](#)
- Toggle fixed point, [34](#)
- Toggle grid display, [33](#)
- Toggle keyboard, [34](#)
- Toggle outline, [34](#)
- Toggle snap mode, [34](#)
- Vector scatter plot, [41](#)
- Volume integrals, [36](#)

## Analysis script

- Configuration, [42](#)
- GenScan, [42](#)

Input, [42](#)

Interpolation, [43](#)

Matrix, [43](#)

NScan, [43](#)

Output, [42](#)

Scan, [42](#)

SurfaceInt, [43](#)

VolumeInt, [43](#)

analysis script commands, [41](#)

analysis script example, [43](#)

Analysis script format, [41](#)

analysis, batch file control, [41](#)

analytic solutions, [4](#)

autoscale, [33](#)

calculated quantities

standard variables, [55](#)

coil losses, [25](#)

comment lines, [17](#)

complex number

display quantities, [48](#)

magnitude and amplitude, [46](#)

computational mesh, [4](#)

configuration file, [51](#)

calculated quantities, [51](#)

expressions, [56](#)

load errors, [56](#)

standard, [45](#)

structure, [51](#)

conformal mesh, [5](#)

data export, [37](#)

delimiters, [17](#)

eddy current, [7](#), [25](#)

element, [4](#)

facet, [5](#)

fiducials, [33](#)

field

definition, [4](#)

equations, [4](#)

field line plots, [41](#)

field probe, [41](#)

- finite-element method, 4
  - basis, 5
- frequency run parameter, 46
- grids, 33
- interpolation methods, 37
- keyboard/mouse control, 34
- line integral, 37
- matrix file, 37
- mouse controls, 36
- Nelson
  - analysis screen, 13
  - analysis script, 13
  - command prompt, 20
  - DUnit, 12, 19
  - file types, 6
  - finte-element solution, 13
  - Geometry, 12, 19
  - instructions, 22
  - MaxCycle, 12
  - Maxcycle, 19
  - Mesh command, 19
  - Mu, 20
  - Omega, 12, 19
  - output file format, 22
  - Potential, 20
  - program function, 3
  - Restarget, 12, 20
  - Run, 21
  - Run analysis script, 21
  - running in a window, 20
  - script, 13, 17
  - script dialog, 11, 21
  - self-consistent drive current, 26
  - skin depth calculator, 22
  - Solution display, 21
  - solution procedure, 5
  - steps in a solution, 11
  - Stop, 21
  - units, 45
- Nelson analysis variables, 45
- node, 5
- node equations, 4, 5, 13
- number notation, 35
- numerical solutions, 4
- plot control, 33
- plot types, 33
- plot view control, 35
- Poisson equation, 5
- recorded quantities, 36
- recorded quantities, rules, 39
- reference phase, 48
- reverse Polish notation, 53
- rpn
  - binary operators, 53
  - parsing rules, 53
- scan plot menu, 39
- script control of analyses, 32
- skin depth, 26, 27
- snap mode, 34
- standard analysis functions, 31
- surface integral, 37
- surface plot control, 35
- symmetry
  - cylindrical, 5
  - planar, 5
- TDiff, export to, 3
- text editor
  - internal, 20, 32
- vector display, 41
- vector tools, 39
- virtual drive current, 25
- volume integral, 36
- winding losses, 25