Pulse

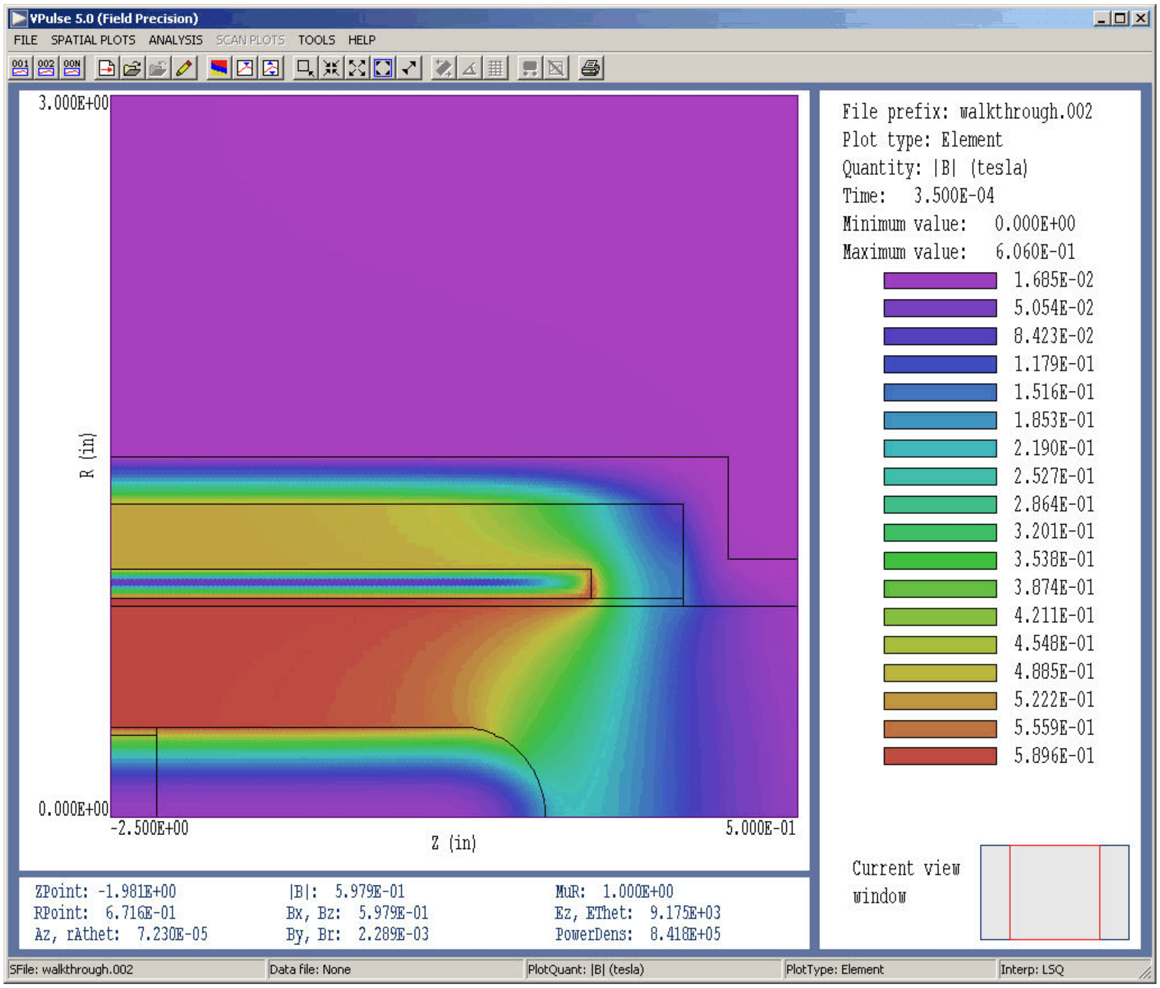
Time-dependent Magnetic Fields with Eddy Currents

Field Precision

Copyright 2004

PO Box 13595
Albuquerque, New Mexico 87192 U.S.A.
Telephone: 505-220-3975
FAX: 505-294-0222
E Mail: techinfo@fieldp.com
Internet: http://www.fieldp.com

Pulse 5.0 Manual - Contents



```
File prefix: walkthrough.002
Plot type: Element
Quantity: |B| (tesla)
Time:    3.500E-04
Minimum value:    0.000E+00
Maximum value:    6.060E-01
      1.685E-02
      5.054E-02
      8.423E-02
      1.179E-01
      1.516E-01
      1.853E-01
      2.190E-01
      2.527E-01
      2.864E-01
      3.201E-01
      3.538E-01
      3.874E-01
      4.211E-01
      4.548E-01
      4.885E-01
      5.222E-01
      5.559E-01
      5.896E-01

ZPoint: -1.981E+00          |B|:  5.979E-01         MuR:  1.000E+00
RPoint:  6.716E-01          Bx, Bz:  5.979E-01      Ez, EThet:  9.175E+03
Az, rAthet:  7.230E-05      By, Br:  2.289E-03      PowerDens:  8.418E+05

SFile: walkthrough.002 | Data file: None | PlotQuant: |B| (tesla) | PlotType: Element | Interp: LSQ
```

Figure 1. Screen shot – working environment of **VPulse**.

1. Program function

Pulse calculates the time-evolution of magnetic fields in the presence of eddy currents. The program solves the two-dimensional magnetic diffusion problem in cylindrical and rectangular geometries. **Pulse** utilizes finite-element methods on variable-resolution conformal triangular meshes. The mesh size is limited only by the installed memory of the computer. Using a simple command language, you can define up to 127 regions to represent materials or coils with different current densities. **Pulse** can be applied to time-dependent problems or as a relaxation program to derive steady-state boundary value solutions. You can define multiple coils with arbitrary current waveforms using a flexible system of tabular function input. The program also handles non-linear magnetic materials with user-defined μ(|**B**|) tables. In time-dependent runs, **Pulse** produces multiple data files at specified times and history files of magnetic and electric field at given locations. Data files can be analyzed with **VPulse** an interactive graphical post-processor. The program provides full information on the fields and magnetic permeability. Other features include automatic calculation of stored energy, peak field values, magnetic forces and torques. **VPulse** makes a wide variety of screen and hardcopy plots. History files can be analyzed and plotted with the utility program **Probe**. **Pulse** has applications to transformers, magnetic recording devices, high-field pulsed magnets, linear induction accelerators, high-frequency inductors and magnetic coupling in microelectronic devices.

2. Walkthough example

It is useful to review a complete solution to understand **Pulse** procedures. We shall consider a problem where the magnetic field generated by a pulsed coil penetrates surrounding metal structures. In preparation, move the files `WALKTHROUGH.MIN, WALKTHROUGH.PIN` and `CRITDAMP.DRV` from the example library to a convenient working directory like `\TRICOMP\BUFFER`. The file `WALKTHROUGH.MIN` (**M**esh **IN**put) contains geometry information for the mesh generator, while the file `WALKTHROUGH.PIN` (**P**ulse **IN**put) contains control parameters for **Pulse**. The file `CRITDAMP.DRV` contains a table that defines a curve for a normalized, critically-damped pulse. Start the **TriComp** program launcher (`TC.EXE`) and make sure that the program points to the working directory using the *SET DATA DIRECTORY* button. Note that you will need to scroll down the list of programs in `TC.EXE` to find buttons for **Pulse**, **VPulse**, **Probe.**

Figure 2 shows the geometry of the cylindrical system. Note that the figure is an *r*-*z* plot as opposed to the a longitudinal section. The physical system is a figure of revolution about the *z* axis and only values $r \geq 0.0$ are defined. The device is a miniature magnetically-insulated transmission line to carry pulsed voltages in the MV range to a high-current electron beam diode. The assembly is contained inside an aluminum vacuum chamber (indigo). The aluminum wall also confines the pulsed magnetic field, reducing inductance and minimizing external fringe fields. The pulse coil (cyan) has a uniform winding density that we approximate by a current density distributed uniformly over the cross-section. A thin layer of stainless steel (dark blue) acts as the outer conductor of the coaxial transmission line. The inner conductor is divided into three parts so we can study the effect of changing its material composition. In one configuration, *CCOND1* is vacuum, *CCONC02* is a thin stainless steel tube and *CCOND03* is a stainless steel cap. In the second configuration, all regions are assigned the properties of aluminum to represent a solid

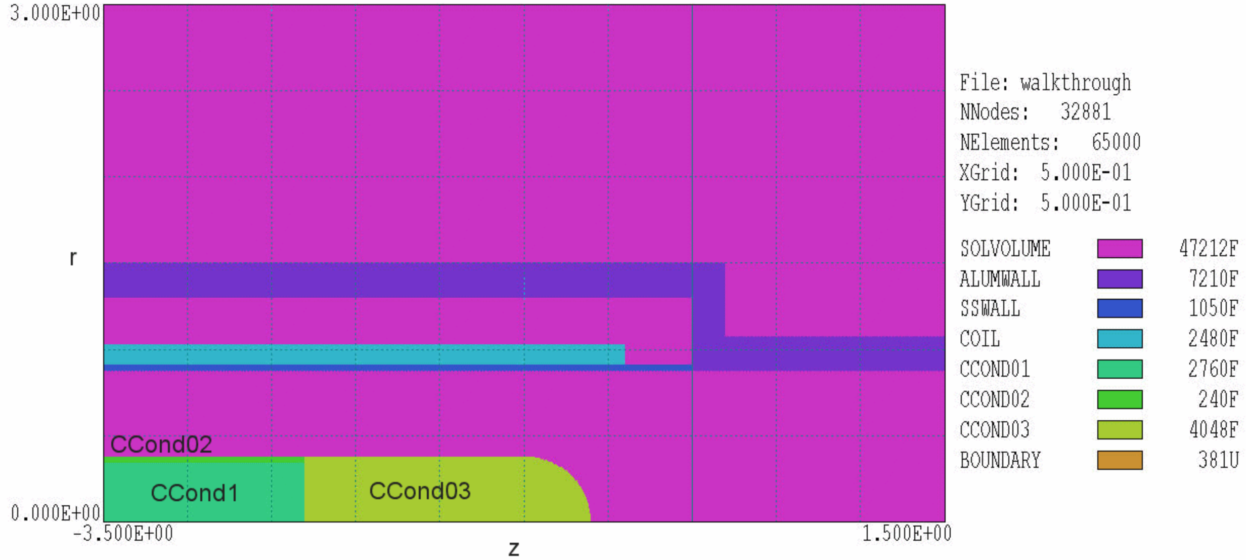


Figure 2. Region definitions for the `WALKTHROUGH` example.

assembly. The remaining elements of the solution volume (light violet) represent air or vacuum (μ_r = 1.0). There is an additional region (*BOUNDARY*) that is not displayed in Fig. 2. The region consists of the nodes on the bottom, right and top sides of the solution volume to which we shall apply a special condition (discussed below). To review, the term *nodes* applies to the vertices of triangles that constitute the mesh. The function of **Pulse** is to calculate time dependent values of the vector potential A_z (for planar solutions) or stream function rA_θ (for cylindrical solutions) at the node points. The **VPulse** program takes numerical derivatives of these quantities to find magnetic field components. Finally, the term *element* refers to the triangles. Physical properties like magnetic permeability and electrical resistivity apply to the triangle areas (which represent physical volumes).

To begin, launch **Mesh** from **TC**. (If the program does not start, check that the program is available in the *Program Directory* specified in `TC.EXE`). Click on *LOAD SCRIPT (MIN)* in the *FILE* menu. If the data directory is correctly set, you should the file `WALKTHROUGH.MIN` in the dialog. Choose it and click *OPENFILE*. Before creating a mesh, we will take a look at the contents of the file. In the *FILE* menu, click on *EDIT CURRENT SCRIPT (MIN)* to open an edit window. The script is divided into several sections. The *GLOBAL* section defines control parameters and the initial size of triangular elements. Each *REGION* section corresponds to one of the regions in Fig. 2. The outlines of the regions are defined by sets of line (*L*) or arc (*A*) vectors. Most of the *REGION* statements contain the *FILL* keyword, which means that the current region number is assigned to all elements and nodes inside the boundary. When the *FILL* statement is absent, **Mesh** only sets nodes on the boundary.

Exit the editor by clicking on *FILE/EXIT* or the close button and then choose the *PROCESS* command. In response, **Mesh** analyzes the contents of the script to create a conformal mesh. The program displays screen information on the progress of the run. The program also records the information in a listing `WALKTHROUGH.MLS`. Press any keyboard button or click the mouse inside the text area to continue. Click on *FILE/SAVE MESH (MOU)* to save the mesh configuration. The program creates the file `WALKTHROUGH.MOU` which contains information on node coordinates and the region numbers of nodes and elements. This file is one of the required inputs for the **Pulse** calculation. Before proceeding, you can check some of the **Mesh** plot capabilities by clicking on *PLOT/REPAIR*. The **Mesh** manual gives detailed descriptions. Note how the program has flexed triangles to create accurate and smooth surfaces. The right-hand side of Figure 2 shows some of the information displayed in **Mesh** plots. The mesh contains 32,881 nodes and 65,000 elements. The region representing vacuum/air is of the filled type and contains 47,212 elements, while the boundary region is unfilled and contains 381 nodes.

Next, launch **Pulse** from **TC**. Click on *FILE/EDIT INPUT FILES* and choose `WALKTHROUGH.PIN`. Table 1 shows a condensed version of the file content. There are three types of commands: *SET*, *REGION* and *DIAG*. *SET* commands define values for program control parameters. The first four commands in Table 1 control the automatic time-step calculation for the solution of the diffusion equation. The fifth and sixth commands specify that the geometry has cylindrical symmetry and that the coordinates from **Mesh** are in inches (39.37 inches/m). The *REGION* commands set the physical properties of regions. The vacuum/air region (number 1) has μ_r = 1.0 and ρ = 100.0 Ω-m. Note that the volume resistivity is much larger than the metal regions but not infinite. In a diffusion solution, information must propagate at a finite velocity. The speed of light is not inherent in the magnetic diffusion equations discussed in Sect. 4. Therefore, the choice $\rho = \infty$ would give an infinite mesh velocity leading to numerical instability. The choice ρ = 100.0 ensures that changes of magnetic field move rapidly through the air/vacuum regions compared to the rise-time of the drive current pulse.

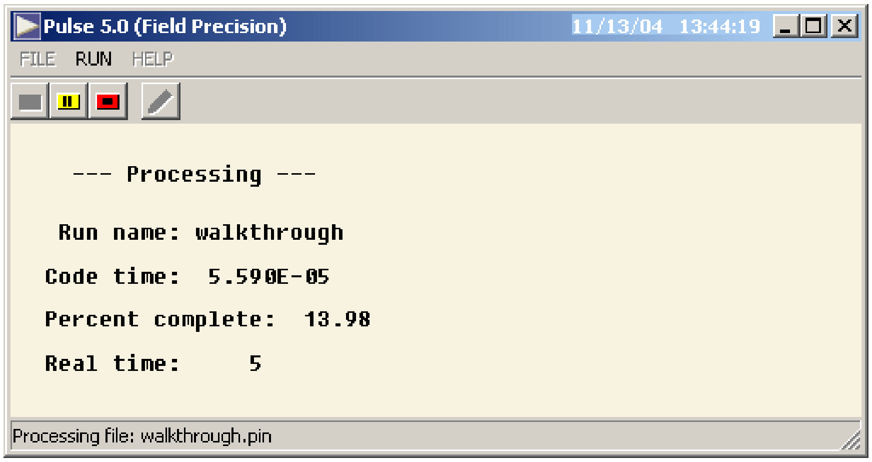
Table 1. Condensed content of the file WALKTHOUGH.PIN

```
SET Safety = 5.0
SET DTMin = 1.0E-7
SET DTMax = 1.00E-6
SET TMax = 400.0E-6
SET Geometry = Cylin
SET DUnit = 39.37
REGION(1) Mu = 1.0
REGION(1) Rho = 1.0E2
REGION(2) Rho = 2.7E-8
REGION(3) Rho = 81.0E-8
REGION(4) Rho = 1.0E2
REGION(4) Current(Table) = CRitDamp.DRV (350.0E-6, 6.967E4)
REGION(5) Rho = 2.7E-8
REGION(6) Rho = 2.7E-8
REGION(7) Rho = 2.7E-8
REGION(8) Fixed = 0.0
DIAG DTime = 175.0E-6
DIAG History = (-2.00, 1.55)
ENDFILE
```

Commands of the type *`REGION(RegNo) MU`* have been omitted from the other regions because the code default value is $\mu_r = 1.0$. The *RHO* commands set the volume resistivity for the stainless steel outer conductor and set all sections of the inner conductor to the volume resistivity of aluminum. The command

```
REGION(4) Current(Table) = CritDamp.DRV (350.0E-6, 6.967E4)
```

defines the total current in the coil region (number 4). The keyword *TABLE* and string *CRITDAMP.DRV* designate that the current waveform is defined by the table in the file. The file consists of 150 data lines – each line contains a value of time *t* and a functional value *f(t)*. In this case, the function is a normalized critically-damped waveform $f(t) = 2.7183\ t\ \exp(-t)$. The table was created from a spreadsheet. The two real-number parameters adjust the normalized function so that: 1) the peak current occurs at 350.0 μs and 2) the magnetic field in the transmission line gap has a magnitude of 0.6 tesla for an aluminum center-conductor.



```
   --- Processing ---

  Run name: walkthrough

 Code time:   5.590E-05

 Percent complete:   13.98

 Real time:     5
```

Processing file: walkthrough.pin

Figure 3. Pulse screen display during a calculation.

The final region command,

```
REGION(8) Fixed = 0.0
```

sets rA_θ = 0.0 for all nodes on the bottom, right and top boundaries. The condition constrains the magnetic field to be parallel to surfaces. It is clear that the condition holds on the bottom boundary (axis of rotation), but some explanation is needed for the other two. The field will not be completely contained by the aluminum vacuum chamber. Some flux will leak out, and one of our goals is to find the magnetic field amplitude outside the tube. In a finite-element calculation of magnetic-field diffusion, it is not possible to define an ideal free-space boundary. In **Pulse**, there are two choices for the boundary condition: 1) parallel **B** and 2) normal **B**. To approximate the free-space condition, we include an isolating air region around the system and apply the parallel-**B** condition. With no specification, the left boundary automatically assume the natural boundary conditions for a finite-element magnetic-field solution (normal **B**). Because there is a long, uniform section of transmission line to the left of the solution volume, the condition that B_r = 0.0 is approximately correct. The two *DIAG* commands control data output. The first initiates generation of field snapshots at 175 and 350 seconds. The second places a probe just outside the aluminum vacuum chamber in the transmission line region to measure leakage field.

Exit the editor and click on *RUN/START RUN*. **Pulse** reads the files `WALKTHROUGH.MOU` and `WALKTHROUGH.PIN` and starts the calculation. After reading data from `CRITDAMP.DRV`, the program initiates the magnetic diffusion calculation. The screen display (Figure 3)

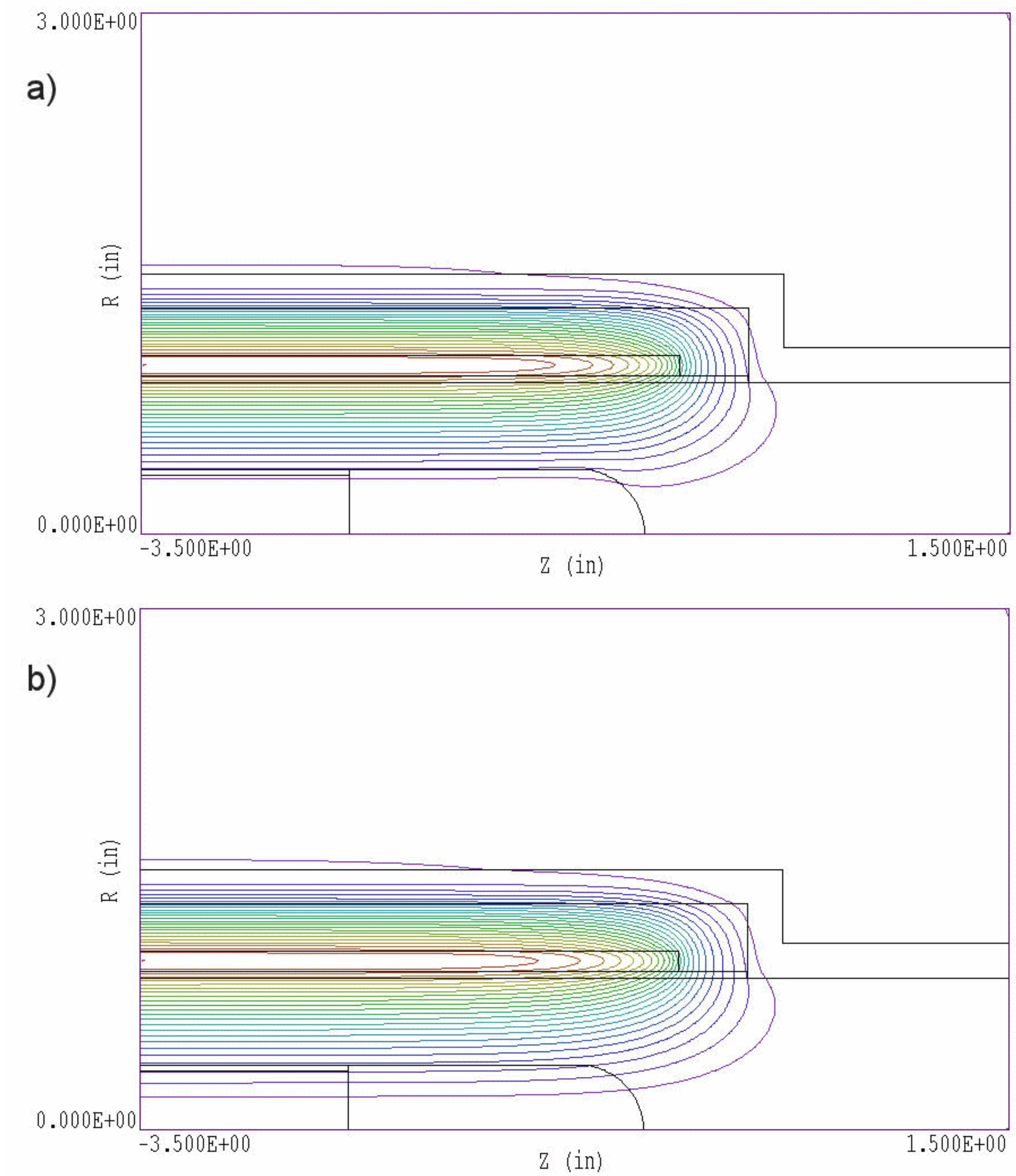


Figure 4. Magnetic field lines for the `WALKTHROUGH` example at t = 350 seconds. Note that lines are separated by equal intervals of flux so that the spacing appears larger near the axis. *a*) Solid aluminum center-conductor. *b*) Center-conductor comprised of stainless steel tube and cap.

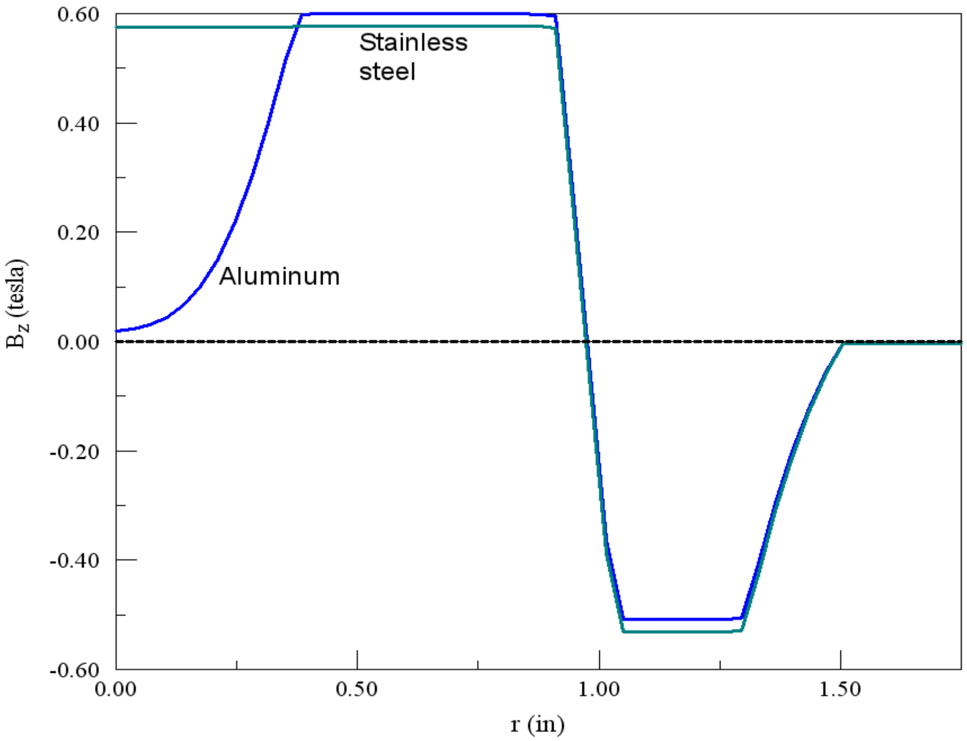


Figure 5. Variation of B_z along the left boundaries of Figs. 4*a* and 4*b*.

shows the progress of the run. The process takes about 40 seconds on a 3 GHz computer. Pulse adds the following files to the working directory: WALKTRHOUGH.001, WALKTHROUGH.002 and WALKTHROUGH.P01. The first two files are the field snapshots at 175 and 350 seconds, while the third is the probe record.

Launch **VPulse**, click on *FILE/LOAD FIRST SOLUTION FILE* and choose *WALKTHROUGH*. The program loads `WALKTHROUGH.001` and creates a default magnetic field-line plot (Figure 4). Click on *FILE/LOAD NEXT SOLUTION FILE* to advance to the solution at the peak of the drive current (350 seconds). At this point, you can experiment with the extensive plot and analysis functions of **VPulse**. Figure 4 shows field-line plots for two choices of center-conductor materials. Figure 5 plots the variations of B_z along the left boundary of Fig. 4. Values for the plot were calculated using the *LINE SCAN* command in **VPulse**. Figure 1 is an element-type plot of |**B**| for an aluminum center-conductor which clearly illustrates diffusive field penetration. Sections ## give detailed descriptions of **VPulse** capabilities.

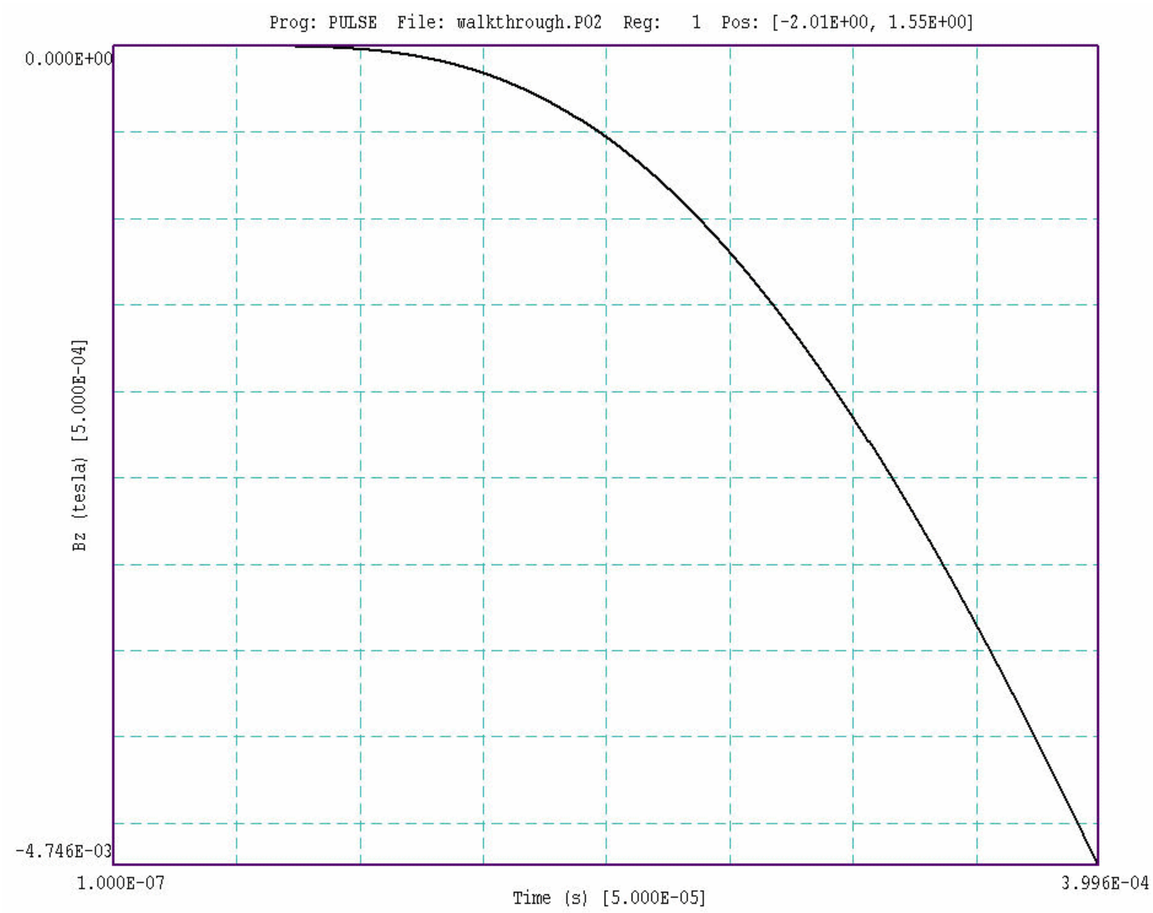


Figure 6. Probe plotting capabilities. *WALKTHROUGH* example – magnetic field $B_z(t)$ just outside the aluminum vacuum tube at z = -2.00 inches.

To conclude, launch **Probe** from **TC**. Click on *LOAD PROBE FILE* and choose `WALKTHROUGH.P01`. The program displays a summary of file information – click *OK* to proceed. **Probe** creates the default plot shown in Fig. 6. As expected, the external field rises exponentially with time, reaching a peak value of about 33 G at t = 350 seconds. Within the program you can changes quantities, modify the plot limits or use the digital oscilloscope features to find specific values. The **Probe** manual describes program capabilities in detail. Note that probe files are in text format so it is easy to port information to your own analysis programs.

3. Physical units

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

Length: meters
Time: seconds
Vector potential: tesla-m
Magnetic fields: tesla
Current: amperes
Current density: amperes/m^2
Volume resistivity: Ω-m

4. Theoretical background

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

The reduced set of the Maxwell equations is

$$\nabla \times \left(\frac{\boldsymbol{B}}{\mu} \right) = \boldsymbol{j_o} + \boldsymbol{j_r}. \qquad \textbf{(2)}$$

$$\nabla \times \boldsymbol{E} = - \frac{\partial \boldsymbol{B}}{\partial t}. \qquad \textbf{(3)}$$

Equation 2 is Ampere’s law. The quantity μ is the local value of magnetic field permeability. Including it in the equation automatically introduces the effects of atomic currents in materials. In the following discussions, we use the *relative magnetic permeability,* defined by

$$\mu_r = \frac{\mu}{\mu_o}. \tag{4}$$

In **Pulse** calculations, we assume that all magnetic materials are *soft* and *isotropic*, so that the magnetic permeability is a single-valued function of the magnitude of the local magnetic field, $\mu_r(|\mathbf{B}|)$. The non-atomic current density on the right-hand side of Eq. 2 is divided into two parts. The quantity j_o represents the current density in coils. This current density is specified by the user and does not depend on the field solution. The quantity j_r is the *eddy current* density induced in surrounding conductors by changing magnetic fields. If the conductors have an isotropic volume resistivity ρ, the eddy current density is related to the local value of electric field by

$$\boldsymbol{j}_{\boldsymbol{r}} = \frac{\boldsymbol{E}}{\rho}. \tag{5}$$

The third equation is Faraday's law, giving the electric field in terms of the changing magnetic field. We assume that there is no space charge so that only inductive electric fields are present. Equation 1 implies that the magnetic field can be written as the curl of a vector quantity, the vector potential,

$$\boldsymbol{B} = \nabla \times \boldsymbol{A}. \tag{6}$$

Combining Eqs. 2 and 6, we can show that the inductive electric equals field is given by the time derivative of the vector potential

$$\boldsymbol{E} = -\frac{\partial \boldsymbol{A}}{\partial t}, \tag{7}$$

or

$$\boldsymbol{j}_{\boldsymbol{r}} = -\frac{1}{\rho}\frac{\partial \boldsymbol{A}}{\partial t}. \tag{8}$$

Substituting Eqs. 6 and 8 into Eq. 3 gives the magnetic diffusion equation

$$\frac{\mu_o}{\rho}\,\frac{\partial \boldsymbol{A}}{\partial t} = \nabla \cdot \left(\frac{1}{\mu_r}\,\nabla \boldsymbol{A}\right) + \mu_o\,\boldsymbol{j_o}. \qquad \textbf{(9)}$$

The strategy is to solve Eq. 9 for the vector potential using appropriate boundary conditions and then to derive the magnetic fields from Eq. 6. The procedure is straightforward in two dimensional geometries where there is only one component of vector potential. Rectangular geometries have variations in *x* and *y* and are uniform in *z*. The conditions of symmetry imply that the applied currents, the vector potential, and the inductive electric field point in the z direction and that the magnetic field components B_x and B_y are non-zero. Cylindrical systems are symmetric in θ and have field components B_r and B_z. The current density, vector potential and induced electric field are directed in θ. **Pulse** does not directly solve the differential form of Eq. 9. Instead, the finite-element code solves the integral form of Ampere's law applied to the triangular elements of the mesh. In this approximation, values of A_z and $\partial A_z/\partial t$ are defined at the vertices of the mesh triangles. The material properties μ and ρ as well as the field components B_x and B_y are associated with the triangle areas.

To apply the finite element method to cylindrical problems, **Pulse** uses the *stream function* rather than the vector potential as the main solution variation. The stream function, denoted by ψ, is related to the azimuthal component of vector potential by $\psi = rA_\theta$. A complete review of the physical basis of Pulse is given in S. Humphries, **Field Solutions on Computers** (CRC Press, Boca Raton, 1997), Sect. 11.5.

5. Organizing runs

The **Pulse** package consists of a program that computes the physical solution (PULSE.EXE), a dedicated post-processor for the analyses of magnetic and inductive electric fields (VPULSE.EXE) and a utility program to plot history files from any Field Precision program that generates initial value solutions (PROBE.EXE). The solution program can run in two modes: interactively in a window or autonomously in the background with the option for batch file control. Similarly **VPulse** can run as an interactive application under user control or autonomously under script file control. The autonomous modes allow automatic processing of large or repetitive data sets.

Pulse requires two input files.

- A **Mesh** output file that describes the conformal triangular mesh. The file contains node coordinates and the region numbers of elements and vertices.

- A command script file that sets control parameters for the solution program and describes the physical properties associated with region numbers.

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

The mesh output file always has a name of the form FPREFIX.MOU, where FPREFIX is a valid file prefix (1 to 20 characters). The command script must have a name of the form FPREFIX.PIN where FPREFIX is the same prefix as that of the mesh file. **Pulse** issues an error message if both input files are not available in the working directory. To organize data, the resulting output files have names that begin with FPREFIX.

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

- Prepare a **Mesh** input script with a name of the form FPREFIX.MIN following the instructions in the **Mesh** manual. You can create the file directly with a text editor or graphically using the Drawing Editor of **Mesh**.

■ Run **Mesh** either interactively from the **TC** program launcher or from the Windows Command Prompt to create the file `FPREFIX.MOU`.

■ Prepare a command script (`FPREFIX.PIN`) using the *Create command file* menu entry in **Pulse** or your own text editor. The allowed file commands are described in this manual.

■ Optionally, prepare data files to define time-dependent current or the B-μ characteristics of a material. The text files may consist of experimental data and may be created with a text editor or spreadsheet.

■ Run **Pulse** to create one or more output files. Files with names of the form `FPREFIX.001` are data dumps at specific times that can be used as input to the **VPulse** program. The files are in text format and contain information on the mesh geometry, the physical properties of regions, and values of computed quantities at the nodes. In magnetic diffusion solutions the recorded region properties are relative magnetic permeability (for linear materials) and the volume resistivity. The code stores node values of the vector potential (or stream function) and its time derivative. The latter quantity is used to display inductive electric field in **VPulse**. Values of relative magnetic permeability are also stored to plot variations in non-linear materials. Files with names of the form `FPREFIX.P01` are probe records of temperature at specific locations. The contents of these files can be plotted with the **Probe** program.

■ Analyze the solution using **VPulse**. or **Probe**. You can also transfer the information in data dumps or history files to your own analysis programs.

6. Structure of the command file

The input script for all **TriComp** solution programs is a text file with data lines that contain commands and parameters. The file must end with the *ENDFILE* command. The entries on a line can be separated by the following delimiters

```
Space, Comma, Forward slash,
Colon, Tab, Equal sign
```

Any number of delimiters can be used in a line. Blank lines and comment lines are ignored. Comment lines begin with an asterisk (*). Most parameters are real numbers. The following formats are valid.

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

The bottom number is interpreted as 5.0.

The program accepts commands in any order. The following example illustrates a complete control file for **Pulse**:

```
* Run HMAG_T
SET SAFETY 2.5
SET NCHECK 10
SET TMAX 150.0
SET DTMIN 0.05
SET DTMAX 0.50
SET DUNIT 100.0
SET GEOMETRY Rect
SET INTERPOLATION Spline
* Region 1: Air
REGION 1 MU 1.0
REGION 1 RHO 1.0E-4
* Region 2: Fixed boundary
REGION 2 MU TABLE SOFTIRON.DAT
* REGION 2 MU 3000.0
REGION 2 RHO 10.0E-8
* Region 3: Coil
REGION 3 MU 1.0
REGION 3 RHO 1.0E-4
REGION 3 CURRENT TABLE SMOOTH.DRV 100.0 10000.0
* Region 4: Boundary
REGION 4 FIXED 0.0
* Diagnostics
DIAG DTIME 25.0
DIAG HISTORY 0.0 0.0
DIAG HISTORY 0.0 3.0
ENDFILE
```

There are three types of commands that begin with different keywords:

```
SET
REGION
DIAG
```

Commands that begin with the word SET control program settings. The keyword that follows determines the function of the command. One or more parameters follow the keyword. For example, the command

```
SET GEOMETRY CYLIN
```

instructs the program to use element weighting for cylindrical coordinates. The available SET commands depend on the solution type.

REGION commands set physical properties associated with elements and vertices. They all start with the word REGION and have the same format.

```
Region   RegNo    Keyword    Value
```

Here, RegNo (an integer) is the region number defined in the **Mesh** input file. The string Keyword specifies the physical property. The keyword is followed by one or more numbers. As an example, the command

```
REGION 2 Fixed 37.0
```

sets nodes with region number 2 to a fixed temperature of 37.0 °C. In **Pulse** several REGION commands may refer to the same mesh volume and set different properties. Commands that begin with the keyword DIAG are used to specify times for data dumps and locations for probes.

7. Set commands for general program control

Pulse command files have the suffix PIN. Commands divide into three groups: program control, region properties and diagnostics. We begin with the control commands. They begin with the keyword *SET*. Each command is written as it might appear in the RUNNAME.PIN file. The following section gives a more detailed explanation of the important *Set* commands that control the integration time step.

SET SAFETY 35.0

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

SET DTMIN 0.0001

Sets the minimum time interval, over-riding the automatic step selection. Enter the value in seconds. The default value is 1.0 x 10^{-9} s.

SET DTMAX 0.25

Sets the maximum time interval, over-riding the automatic step selection. Enter the value in seconds. The default is infinity.

SET NCHECK 10

Sets the number of time increments between a recalculation of the automatic time step routine. The default value is 5.

SET TMAX 100.0

Sets the stop time for the run. Enter the value in seconds. All runs begin at $t = 0.0$. The default is 0.0.

SET NSTEP 5000

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

SET GEOMETRY Cylin

Sets the problem geometry. The parameter options are *Cylin* and *Rect*. **Pulse** handles either rectangular or cylindrical problems. In the former case, the *x*-axis in Mesh corresponds to the *z*-direction and *y* corresponds to *r*. The program generates an error message if *YMin* < 0.0. In rectangular geometry, the system varies in *x* and *y* and extends an infinite distance in *z*.

SET DUNIT 100.0

You can use any convenient distance units for the **Mesh** input file. The quantity *DUnit* gives the conversion from **Mesh** units to the standard distance units of meters in **Pulse**. The quantity DUnit is the number of **Mesh** units per m. For example, if the **Mesh** dimensions are entered in microns, set DUnit = 1.0 x 10^6. The default value is 1.00.

8. Picking a time step

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

SET SAFETY

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

SET DTMIN 5.0E-7

The automatic time step feature of **Pulse** can cope with difficult solutions where there are initial discontinuities in vector potential. Depending on the geometry, **Pulse** may have to use extremely short time steps to achieve accuracy early in time. Sometimes, details of the initial local temperature relaxation are not important. One example is finding a steady-state field distribution in a medium with several fixed currents. Here, the final state does not depend on the accuracy of the initial relaxation. To speed these calculations, you can over-ride automatic time selection by setting a minimum time step with the *DtMin* command. Once the estimated time step rises above *DtMin*, the automatic routine takes over.

The *DtMin* command is also important for solutions with time-dependent currents that start from zero. In this case, the vector potential is initially zero everywhere. Because the second derivative is zero, there is no way for **Pulse** to calculate a time step. In this case, the program assigns a very small default increment. You can set a better value using the *DtMin* command. In picking a time step, note that a good solution usually require about 1000 or more steps.

SET DTMAX 1.0E-3

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

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

Finally, although **Pulse** is a time-dependent code, it is easy to derive steady state solutions for problems with fixed vector potential boundaries and constants. Use current waveforms that rise smoothly and then maintain a constant value. Run the program past the longest magnetic diffusion time and make a single data dump at the end. In this case, the calculation is identical to a relaxation solution for a boundary value problems. You can use a history probe to check the convergence. The process can be accelerated by lowering the value of *Safety* and raising *DtMin* or *DtMax*.

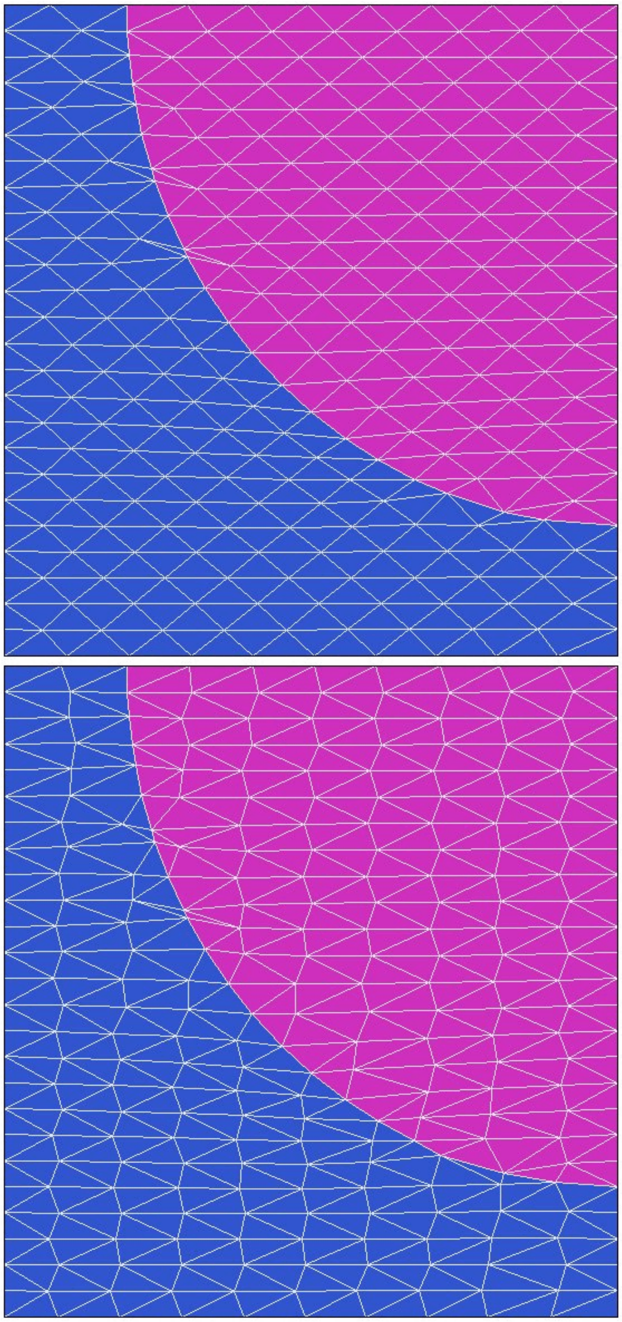


Figure 7. Effect of the *DCORRECT* command.

9. Avoiding numerical instabilities

Pulse uses the time-centered Dufort-Frankiel method to advance the diffusion equation in time. The explicit method gives fast calculations on large meshes and is stable for all choices of time step on a simple rectangular finite-difference mesh. A numerical instability may occur if the method is applied on conformal meshes that have triangular elements that are short in the *y* or *r* directions. For a discussion of the theory and conditions of the instability, see S. Humphries, **Field Solutions on Computers** (CRC Press, Boca Raton, 1997), Sect. 12.4. We have added an automatic correction feature in Version 5.0 of **Mesh** to eliminate the instability in diffusive programs like **TDiff** and **Pulse**. To employ the correction, you must include the command

DCORRECT

(for *diffusive correction*) in the **GLOBAL** section of the **Mesh** input (`MIN`) file. Figure 7 shows the effect for a mesh with elements where Δ*y* = 0.4Δ*x*. In the top plot with no correction most of the triangles have an internal angle greater than 90°, the condition for an instability. With the diffusive correction (bottom plot), **Mesh** automatically adjust the shapes of mesh elements so that there are no contiguous blocks of elements with obtuse internal angles. The following global section was used for the bottom plot:

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

As a rule, always include *DCORRECT* in **Mesh** input files with variable resolution that will be used for **Pulse** solutions.

10. Commands to set region properties

The **Pulse** command set for defining the material properties of regions is simple. The three commands to set the physical properties of materials all start with the word **Region** and have the same format.

Region   RegNo   Keyword   Value

Here, *RegNo* (an integer) is the Region Number defined in the **Mesh** input file. *Keyword* is one of the words listed below and *Value* is a real number. In this section, we limit attention to commands to set values that remain constant through the solution. The following section describes alternative forms to define variable quantities.

REGION 4 FIXED 0.0

Sets the region to a fixed value of vector potential that does not change during the solution. The parameters following the Region keyword are 1) the region number, 2) the keyword *Fixed*, and 3) the vector potential in tesla-m. In cylindrical problems, enter the value of the stream function, ψ, in tesla-m^2. Because the stream function always equals zero at r = 0, Pulse automatically creates a fixed potential region with ψ = 0 on the axis.

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

REGION 6 RHO 81.0E-8

Sets the volume resistivity of the region in Ω-m. The default is 1.0×10^{38} Ω-m.

REGION 3 MU 1000.0

Sets a constant value for the relative magnetic permeability. The default is 1.0. To model a floating ideal conductor with a constant value of vector potential that is not known in advance, set μ_r equal to a value much smaller than unity.

REGION 2 CURRENT -10000.0

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

In a magnetic diffusion problem, all areas in the solution that do not have a fixed value of vector potential are treated as electrical conductors. You must take care in solutions with vacuum or air regions where the resistivity is effectively infinite. Diffusion through such a medium is instantaneous, so that numerical solutions may exhibit inaccuracies or oscillations. A conservative approach is to set the resistivity in insulators to a high (but finite) value compared to that of the conductive media. For example, in a problem with field penetration into a stainless steel structure (ρ = 81.0 x 10^{-8} Ω-m), a good solution results by setting μ = 1.0 and ρ = 10^{-2} Ω-m in coil and air regions. In this case, the rapid field diffusion through the insulators is distributed over several time steps.

11. Variable quantities

A powerful feature of **Pulse** is the ability to handle time-dependent currents and field-dependent magnetic permeability. You have complete flexibility to model any material or coil variation. The key to this flexibility is input through tabular functions. A tabular function is a text file consisting of up to 256 entry lines of values for an independent and dependent variable. For example, to define a field-dependent permeability, each line has a value of field magnitude |**B**| (in tesla) followed by the corresponding relative permeability. You can prepare tabular function files with a text editor or spreadsheet. You can also use published data or digitized experimental traces. The example below shows the tabular function supplied for soft iron. Note that the file syntax conforms to the same rules as the **Pulse** command file. The free-form parser accepts real numbers in any format with a choice of delimiters. You can add documenting comment lines starting with an asterisk (*). The end of the data is marked with the *ENDFILE* command. You can also add any text after *ENDFILE*.

Although the data lines of the example are ordered by increasing value of the independent variable, this ordering is not required. **Pulse** sorts the list before use and records the final order in the `FName.PLS` file. Furthermore, the independent variable intervals need not be uniform. The maximum number of tables for all purposes is 32. A table requires a minimum of 5 entries. You must ensure that the tabular functions extend over the full range of time or field amplitude that will be encountered in the solution. The interpolation routines return 0.0 for values of the independent variable out of the table range.

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

Many of the commands of the preceding section will accept tabular function input for the quantities. The indicator is the keyword *TABLE* in the command line followed by the name of a text file that contains the numerical information. Note that the files must be in the current directory.

REGION 4 FIXED TABLE FLUX.DAT

Sets the vector potential of a fixed region to a prescribed function of time rather than a constant value. For rectangular problems, the numerical values are contained in the named file in the form

```
Time (seconds)     Vector potential (tesla-m)
```

In cylindrical problems, enter the values as

Time (seconds)   Stream function (tesla-m^2)

Note that the values of the independent variable must extend from t = 0.0 to $t > TMax$. This command is useful in problems where you want to specify the time variation of flux in a problem. In rectangular problems, the difference in A_z between two surfaces equals the integral of normal magnetic field along a line connecting the surfaces. In other words, the difference in vector potential equals the magnetic flux per length (in z) between the surfaces. In cylindrical problems, the quantity $\psi/2\pi$ equals the enclosed axial magnetic flux. The time-dependent fixed boundary condition is useful for inductor problems. Setting dA_z/dt = constant or $d\psi/dt$ = constant is equivalent to applying a constant voltage to the structure.

The following commands apply to electrically conductive regions.

REGION 2 CURRENT TABLE ALPHA.CUR

Determines the total current in the specified region as a function of time from the named tabular function. The file data lines have the form

```
Time (seconds)      Current (A)
```

REGION 2 CURRENT TABLE SMOOTH 5.0E-3 100.0

This is a useful alternative command to define the current as a function of time. The two additional real numbers give scaling factors. The values of the independent variable are multiplied by the first factor as they are loaded, while the values of dependent variable are multiplied by the second factor. With this feature, you can keep a library of standard waveforms, adjusting them to different problems. Note that scaling factors can also be added to the commands to set vector potential as a function of time and relative magnetic permeability as a function of field magnitude. Four normalized tables are supplied with **Pulse**.

`Linear.DRV`. Independent and dependent variables rise linearly from 0.0 to 1.0.

`Sine.DRV`. Independent variable x in the range 0.0 to 1.0 with dependent variable given by $y = \sin(\pi x)$.

`Smooth.DRV`. The dependent variable rises smoothly from 0.0 to 1.0 over the range $0.0 \leq x \leq 0.1$ and remains constant thereafter.

`CritDamp.DRV`. The waveform for a critically-damped current with a peak value of 1.0 at t = 1.0.

REGION 4 MU TABLE SOFTIRON.DAT

Specifies the magnetic permeability in a region as a function of the magnetic field amplitude. The values are contained in the named data file. The data lines have the format

```
B (tesla)      μr
```

Pulse uses cubic splines for interpolation of the tables. This method minimizes recalculation time and gives smooth interpolations that aid convergence. It is important to note that the quality of the interpolation depends on the nature of the numerical data. The dependent quantity and its first derivative should vary smoothly over the table range. You can check the fidelity of the interpolations by inspecting the file `FName.PLS`. After listing the sorted table, **Pulse** records a sample set of interpolated values.

12. Diagnostic commands

Diagnostic commands that begin with the keyword *DIAG* control data output from the program. **Pulse** produces three types of data, all in text format.

- From 1 to 999 data dump files with the names `RunName.001, RunName.002, ....` Each file contains a complete record of spatial information (the mesh geometry and distributions of vector potential, magnetic permeability and induced electric field) at a specified time. These files are usually analyzed with the **VPulse** program.

- A listing file with the name `RunName.PLS` that can be inspected with a text editor. This file contains a variety of information about the **Pulse** run used mainly to check the validity of input parameters and to diagnose problems.

- Optional history files with the names `RunName.P01, RunName.P02, ... ,RunName.P10`. Here, the magnetic and electric field components at one or more probe locations are recorded at each time step of the solution. The information can be inspected with a text editor or the utility program **Probe** (described in a separate manual).

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

DIAG DTIME 0.2

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

DIAG SETTIME 3.45

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

DIAG NSTEP 100

The *NStep* option sets the number of time cycles per data dump.

Note that the three commands for data dumps can work concurrently. Be careful setting the parameters to avoid generating excessively large amounts of data.

DIAG HISTORY 5.00 9.85

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

13. Vector potential and boundary conditions

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

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

$$B_y = -\frac{\partial A_z}{\partial x}.$$

and

$$B_x = \frac{\partial A_z}{\partial y}.$$

For problems in cylindrical coordinates, **Pulse** calculates the quantity rA_θ, where A_θ is the vector potential created by azimuthal currents. The quantity rA_θ is usually called the stream function and denoted by ψ. The magnetic fields are given by

$$B_r = -\frac{\partial A_\theta}{\partial z},$$

and

$$B_z = \frac{1}{r}\frac{\partial}{\partial r}\ rA_\theta\ .$$

The above equations show that changes in values of the vector potential are related to the flux of magnetic field lines (the surface integral of magnetic field). For example, in a rectangular problem with boundaries at y_1 and y_2, the difference in the boundary values of the vector potential at position x_o is given by

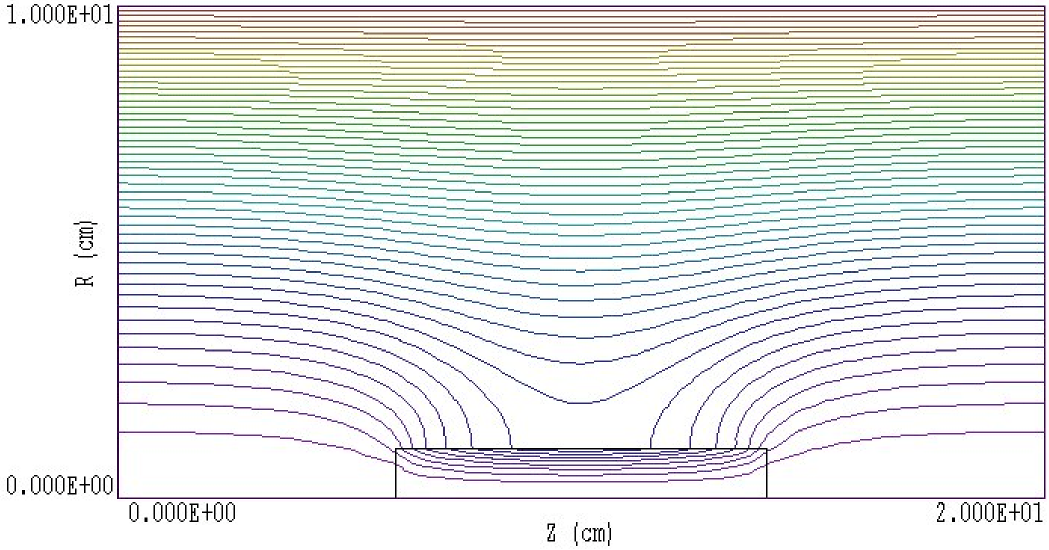


Figure 8. Setting boundary values of the vector potential to create a region of uniform flux. Cylindrical geometry. Vertical: *r* from 0.0 to 0.1 m. Horizontal: *z* from 0.0 to 0.2 m. Lower boundary: rA_θ = 0.0 tesla-m^2. Upper boundary: : rA_θ = 5.0 × 10^{-4} tesla-m^2.

$$A_z(x_o,y_2)-A_z(x_o,y_1) \;=\; \int_{y_1}^{y_2} B_x(y',x_o)\;dy'.$$

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

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

$$(rA_\theta)\Big]_{r_o} = \left(\frac{1}{2\pi}\right)\int_0^{r_o} 2\pi r' dr' B_z(r,z) = \frac{Flux}{2\pi} .$$

As an example, Fig. 8 shows the magnetic field lines around an iron cylinder immersed in a uniform solenoidal field of 0.1 tesla. The uniform field was created by setting $\psi = 5.0 \text{ x } 10^{-4}$ tesla-m^2 on the outer boundary at $r_o = 0.10$ m.

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

Dirichlet

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

Neumann

A Neumann boundary is one where the normal derivative of the vector potential is specified. The boundaries in **Pulse** are limited to the special case $\partial A/\partial n = 0$. The special Neumann condition implies that the magnetic field is normal to the boundary. One of the advantages of the finite-element method is that all boundaries that are not fixed automatically satisfy the special Neumann condition, even if they are slanted or curved. Neumann boundaries are often used to reduce computation time by modeling portions of symmetric systems.

14. Running Pulse interactively

The program `Pulse.EXE` can run interactively in a Window (Figure 3). In this mode you can carry out several solutions in a session or leave the program while you work in other Windows applications.

The most convenient way to run solution programs and post-processors is from the **TC** program launcher. Operation and setup of the launcher is described in the *TC Manual*. You can also run the executable programs individually from Windows Explorer or add shortcuts to the desktop or the Windows Start Menu.

The program has three popup menus: *File*, *Run* and *Help*. The following commands appear in the *File* menu.

Create input file

This command calls up a full-featured internal Windows editor to create an input script. Supply a run prefix (1-20 characters). The allowed commands for the specific solution program appear as a set of comment lines. Save or abandon the file to return to the solution program.

Edit input file
Edit listing file
Edit file

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

The *Run* menu has four commands.

Start run

Pick an input file (such as `RUNNAME.PIN`) to start a solution. The working directory is changed if you pick a file from an alternate directory. The run begins if the file `RUNNAME.MOU` and any data files specified in the command script are present. The small text window displays information on the run status.

Pause run

The intensive calculations of the solution program demand the full resources of your computer, causing other tasks to run slowly. If you need to perform critical work, you can pause the solution program and reactivate it later without loss of data.

Stop run

This command terminates the program 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.

15. Running Pulse from the command prompt

To run **Pulse** in the background, go to the Command Prompt from Windows and log to the data directory that contains the required `MOU` and `PIN` files.. For example, suppose the magnetostatic input data files `SOLENOID.MOU` and `SOLENOID.PIN` are stored in `\TRICOMP\BUFFER` and that the program `PULSE.EXE` is in the directory `\TRICOMP`. From `\TRICOMP\BUFFER` type

```
..\PULSE SOLENOID <Enter>
```

The program runs silently, writing detailed information in the listing file `SOLENOID.PLS`. If the solution is successful, the program writes the output files `SOLENOID.001, SOLENOID.002,`... to the data directory. During lengthy runs you can perform other tasks in Windows. Note that the response of the computer may be slowed considerably.

The main function of the command mode is autonomous operation under batch file control. This feature is useful if you want to make an extended series of solutions overnight or in the background. As an example, suppose you have prepared the input files `SWT01.MIN,...,SWT08.MIN` and `SWT01.PIN,...,SWT08.PIN` in the directory `\TRICOMP\BUFFER`. Furthermore, you have written the following batch file `SWRUN.BAT` in the data directory using a text editor.

```
@ECHO OFF
ECHO Main switch data run
START ..\MESH.EXE SWT01
START ..\PULSE.EXE SWT01
START ..\MESH.EXE SWT02
START ..\PULSE.EXE SWT02
 ...
START ..\MESH.EXE SWT08
START ..\PULSE.EXE SWT08
```

Type

```
SWRUN <Enter>
```

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

Microsoft has released over thirty versions of it's 32-bit operating system since Windows 95. There is considerable inconsistency in DOS emulation between versions. To ensure consistent batch file operation we supply the utility **GCon** with all our software. The program emulates many DOS commands and has advanced features to organize and to analyze large data runs. To avoid programs, we advise running batch scripts from **GCon** rather than from the Command Prompt.

16. VPulse file menu

VPulse is the interactive graphical analysis program for output data dumps from **Pulse**. **VPulse** has the following popup menus: *File, Spatial plots, Analysis, Scan plots* and *Help*. When the program starts only the *File* and *Help* menus are active. You must load a data file in order to create plots or to perform analyses. This section reviews options in the *File* menu.

Program units

Spatial dimensions: meters or units set by *DUnit*.
Magnetic fields: tesla
Vector potential: tesla-m
Current: amperes
Forces: newtons
Torques: newton-m
Electric field: V/m
Power density: J/m^3

Load first solution file

The *Load first solution file* command displays a dialog with a list of all data dump files with the suffix `001`. Changing the directory in the dialog changes the program working directory. Pick an available file and click *OK*. The program loads the first file in the series and updates the status bar. If data retrieval is successful, the *Spatial plots* and *Analysis* menus become active.

Load next solution file

Load the next solution in the current series. For example, if you are currently working with `RUNNAME.003`, **VPulse** loads `RUNNAME004`.

Load solution file number

Load a specific file in the current series. For example, if you are currently working with `RUNNAME.002` and specify the number 5, **VPulse** loads `RUNNAME.005` if it is available.

Open data listing file

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

Close data listing file

Use this command if you want to start a new file to record data. You must close the file before attempting to view or edit it using the built-in editor of the post-processor.

Run script

Sometimes you may want to perform complex or repetitive analyses on a set of similar solutions. Script file operation is a powerful feature of the **TriComp** postprocessing programs. This command displays a dialog with a list of analysis files that you have created with the suffix

SCR. Pick a file and click *OK*. The script file can load data files, open and close history files, and perform any of the quantitative analysis functions described in this manual. The script command language is described in Sect. 20. Note that the script file must be in the same directory as the data files.

Create script

This command allows you to create script files using the internal editor. A dialog requests a file prefix. The resulting script file will be saved as FPREFIX.SCR. Next, the program opens the file in the editor and writes a list of allowed commands for reference. This list follows the *EndFile* command and will be ignored. Enter analysis commands above the *EndFile* command. After saving the file, you can run it using the *Run script* command.

Edit script

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

Edit data file

Use this command to view or to modify files with names of the form FPREFIX.DAT.

Edit file

Use this command to view or to modify any text file.

17. VPulse spatial plot menu

Spatial plots show variations of quantities over the two-dimensional space of the simulation. **VPulse** makes a default plot when a data file is loaded.

Plot type

VPulse can create a variety of colored screen and hardcopy plots.

Mesh. Outline of the elements in the computational mesh.

Region. Computational mesh with elements color-coded by region number.

Contour. Lines that follow constant values of a calculated quantity.

Element. Elements of the solution space color-coded according to a computed quantity (such as magnetic-field magnitude). You can add element boundaries with the *Toggle element outline* command.

Vector. An element plot with orientation lines included in each element to show the local direction of a vector quantity.

Surface. A three-dimensional plot where a computed quantity is displayed along *z* 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*, *Contour*, *Element* or *Vector* plots.

A plot type may not support some plotted quantities. For example, a vector plot of electric field is undefined. If you receive a message when you switch plot types that the current quantity is not allowed, use the *Plotted quantity* command to pick a valid option.

Plotted quantity

A dialog shows a list of available quantities consistent with the current plot type. Note that the list will be empty for *Mesh* and *Region* plots.

VPulse supports the following plot quantities.

Countour plots

Vector potential (A_z) or stream function (rA_θ)
Magnetic field amplitude, $|\mathbf{B}|$
Relative magnetic permeability, μ_r
Induced electric field, E_z or E_θ

Note: In a rectangular solution, contours of A_z lie along magnetic field lines separated by constant increments of magnetic flux. In cylindrical coordinates, contours of rA_θ show magnetic field lines. Because of the equal flux condition, the spacing between field lines appears larger near the axis. Contours of μ are useful only if the solution contains non-linear magnetic materials.

Element plots

Vector potential (A_z) or stream function (rA_θ) Magnetic field amplitude, $|\mathbf{B}|$
Magnetic permeability, μ
Induced electric field, E_z or E_θ
Resistive power density, σE^2.

Vector plots

Magnetic field, **B**

Surface plots

Magnetic field amplitude, $|\mathbf{B}|$
Horizontal field, B_x or B_z
Vertical field, B_y or B_r
Magnetic permeability, μ
Vector potential, A_z or A_θ
Induced electric field, E_z or E_θ
Resistive power density, σE^2.

Plot limits

In the default *autoscale* mode **VPulse** picks limits in *Contour*, *Element*, *Vector* and *Surface* plots that span the full range of the current quantity. With this command you can set manual limits. In the dialog uncheck the *Autoscale* box and fill in minimum and maximum values. Note that the program does not check that the values you supply are physically reasonable. This operation will not affect scaling of other plot quantities. Check the box to return to *autoscale* mode.

Toggle grid

A set of dashed grid lines in *x*-*y* or *z*-*r* can be superimposed on *Mesh*, *Region*, *Contour*, *Element* and *Vector* plots. 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). Grids corresponding to the axes (x = 0.0 or y = 0.0) are plotted as solid lines.

Set contour plot style

This command is active only when the current plot type is *Contour*. There are four 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. Overlapping labels on closely-spaced lines may look better in a zoomed view.

Toggle element outline

This command determines whether the element boundaries are included in *Element* and *Vector* plots. It may be necessary to deactivate outlines for large meshes.

Toggle mouse/keyboard

By default the program uses interactive mouse entry of coordinates for commands like *Line scan* and *Zoom*. This command switches between mouse and keyboard input. Enter keyboard coordinates in the distance units used in **Mesh**. In other words, if the solution program has *DUnit* = 1.0×10^6, then enter dimensions in microns.

Toggle snap mode

The mouse snap mode is a convenient feature of **VPulse**. 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 [5.5,-2.5]. By default, snap mode is *ON*. Snap mode is automatically turned off for coordinate input to the commands *Point calculation* and *Element properties*. Otherwise, the program could pick a location closest to the snap point rather than the point at the tip of the mouse arrow, giving misleading information.

Set DSnap

Set the distance scale for the mouse snap mode.

The following commands change the view limits in *Mesh*, *Region*, *Contour*, *Element* and *Vector* plots. The current view limits of the two-dimensional plots are used when creating three-dimensional *Surface* plots.

Zoom window

You can zoom in on any area of the plot by specifying the two corners of a view box with the mouse. The coordinates are displayed in the window below the plot. The returned coordinate values depend on whether snap mode is active. You can also enter view coordinates from the keyboard by issuing the *Toggle mouse/keyboard* command.

Zoom in

Magnifies the view about the center of the current plot.

Expand view

Expands the view area about the center of the current plot.

Global view

Returns the plot boundaries to the full solution area.

Pan

Moves the center of the plot. Enter two points to define a displacement vector.

The following commands control the appearance of *Surface* plots. The commands are active only when a *Surface* plot is displayed. You may notice that there is a delay if you choose the *Surface* option in the *Plot type* command or if you change the plotted quantity when a *Surface* plot is active. To create the plot, the post-processor must map the current quantity to a rectangular grid, performing a large number of interpolations.

Rotate 3D image

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

View angle 3D

Change the angle of view for the *Surface* plot.

Set grid 3D

To create the plot a quantity is mapped to a rectangular grid with dimensions $NX \times NY$. These numbers also determine the total number of grid lines in the *Surface* plot. The default values are $NX = NY = 40$. You can change values with this command. Although higher numbers give plots with more detail, the regeneration time is longer and the screen display may be unattractive. The command causes a program delay because values must be recalculated.

Default printer

Spatial and scan plots can be ported to any installed Windows printer (including network printers, postscript drivers,...). You can generate colored plots if you have a color printer. When you issue the *Default printer* command, the current screen plot is sent to the default Windows printer. If necessary, change the default using the *Settings* command of Windows.

Plot file (PostScript)

Use this command to make a plot file of the current screen plot in Encapsulated PostScript format. Supply a file prefix in the dialog box. The plot file will be created in the current directory with the name `FPREFIX.EPS`.

Plot file (BMP)

Use this command to make a plot file of the current screen plot in Windows Bitmap format. Supply a file prefix in the dialog box. The plot file will be created in the current directory with the name `FPREFIX.BMP`.

Plot file (PNG)

Use this command to make a plot file of the current screen plot in Portable Network Graphics format. Supply a file prefix in the dialog box. The plot file will be created in the current directory with the name `FPREFIX.PNG`.

Copy to clipboard

Copy the current plot to the clipboard in Windows MetaFile format. The plot can then be pasted into a compatible graphics program.

18. VPulse analysis menu

The commands in the Analysis 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* plots is displayed.

Point calculation

VPulse employs a sophisticated interpolation technique that preserve discontinuities at material boundaries. The program gives correct values for thermal flux on both sides of material boundary. Click on the command and then point to any position in a *Mesh, Region, Contour*, *Element* or *Vector* plot. Note that snap mode is turned *OFF* for coordinate input. The program writes a subset of interpolated quantities to the window below the plot and also records complete information if a data file is open. To enter point coordinates by keyboard, use the *Toggle mouse/keyboard* command.

Toggle interpolation

The default interpolation method for the *Point calculation* and *Line scan* commands is a second-order least-squares fit with intelligent collection of data points. For example, only points on the side of a dielectric boundary that contains the target point are included to give the correct field discontinuity at the boundary. 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.

Line scan

The line scan is one of the most useful functions of **TView**. After clicking on the command, supply two points with the mouse in a view of a *Mesh, Region, Contour, Element* or *Vector* plot 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 thermal quantities at equal intervals along the line. The information is recorded if a history file is open. The program also makes a screen plot of the currently selected quantity versus distance along the scan and activates the commands in the *Scan plot* menu (see Section 19). The program adds fiducial lines to the plot using intelligent grid selection. This means that the plot is adjusted to fill the screen and grids are drawn at useful intervals (like 0.05 or 2.00).

Scan plot quantity

With this command you can choose the quantity to display in screen and hardcopy plots of line scans. Pick the quantity from the list box and click *OK*. This setting has no effect on the history file listing which includes all field quantities. VPulse supports the following line scan quantities: vector potential (A_z or A_θ), horizontal magnetic field component (B_x or B_z), vertical magnetic field component (B_y or B_r) magnetic field amplitude ($|\mathbf{B}|$), relative magnetic permeability (μ_r), induced electric field (E_z or E_θ) and resistive power density, σE^2.

Set number of scan points

This command sets the number of line scan points used for the screen plot and recorded in the history file. The default value is 50 and the maximum number is 500.

Element properties

Pick a triangular element with the mouse (or keyboard) and the post-processor writes material and field properties of the element to the screen. The information is also recorded if a history file is open. It is usually best to start from a *Mesh*, *Region*, *Element* or *Vector* plot to identify the element.

Region properties

To see the physical properties associated with a region of the solution space, click the mouse close to any arc or line vector of the region. Partial results are shown on the screen and a complete analysis is included in the data file. **VPulse** calculates volume integrals of field energy density and power density over the region. Several quantities are calculated from line integrals around the region boundary.

1) Region current. Automatic identification of region surfaces and calculation of surface integrals of Ampere’s law over each region to determine enclosed current. The total current is the integral $\int \boldsymbol{B}\cdot\boldsymbol{ds}/\mu_o$, while the free current is the integral $\int \boldsymbol{B}\cdot\boldsymbol{ds}/\mu$. The free current should equal the current assigned to the region. The quantity is included to check the accuracy of the surface integrals. Note that the free current

integral is invalid when two ferromagnetic materials ($\mu \neq \mu_o$) a common boundary. The surface integral includes the total atomic surface current along the shared boundary and therefore may give non-zero free current for a ferromagnetic region.

2) Region forces, given by a surface integral of magnetic stress over a filled region. For coil regions, this result can be compared to that of the volume integral method. Generally, the surface integral is less accurate for coils, particularly on coarse meshes with discontinuous region boundaries (*i.e.*, square cross section). The forces computed by the surface integral method are generally accurate for magnetic materials with surface currents. Check the accuracy by running two or more solutions with different mesh resolution. Because of ambiguities in resolving atomic surface current layers, the results are invalid when two ferromagnetic materials ($\mu \neq \mu_o$) share a common boundary.

3) Region torques. This computation applies only to filled regions in rectangular problems and uses the surface integral method. The torque is calculated with respect to an axis at the point (0,0). Pick the values of x_{min}, x_{max}, y_{min} and y_{max} in **Mesh** so that this point corresponds to the correct physical axis. The results are given in newton-m/m. Again, the results are invalid when ferromagnetic materials share a boundary.

4) Power density, volume integral of σE^2.

Line integrals

Input for this command is similar to that for the *Line scan* command. Enter two points to define a line.**VPulse** calculates line integrals that are useful for calculations of enclosed current and force. The quantity listed as *Total Current* is the Ampere's law integral ,

$$\int \boldsymbol{B}\cdot\boldsymbol{dl}/\mu_o \ .$$

If you take line scans around a closed path in the solution space, the sum of the integrals gives the enclosed total current (free current plus the contribution from magnetic materials). The quantity *Free Current* is the integral

$$\int \boldsymbol{B}\cdot\boldsymbol{dl}/\mu_r\mu_o.$$

The integral of the magnetic stress tensor (see, for instance, S. Humphries, **Field Solutions on Computers** (CRC Press, Boca Raton, 1997), Sect. 10.5) is also recorded. Multiple scans defining a closed path can be used to find the force on a region. The results are invalid if the path passes through a region with $\mu \neq \mu_o$.

Volume integrals

No input is needed for this command. **VPulse** automatically computes integrals of quantities over the full solution volume and over individual regions. Information is recorded on the screen or in a history file.The following analyses are performed in response to the *Volume integrals* command.

1) Magnetostatic field energy density, $u = B^2/2\mu_r\mu_o$ over the solution space and region volumes. The output units are J/m in rectangular solutions and J in cylindrical solutions.

2) Coil forces, the volume integrals of $\boldsymbol{j}\times\boldsymbol{B}$ for all regions with non-zero current. The units are newtons/m for rectangular problems and newtons for cylindrical problems.

3) Coil torques, the volume integrals of $\boldsymbol{x} \times \boldsymbol{j} \times \boldsymbol{B}$ for all regions with non-zero current. The vector **x** is taken relative to the origin, (x = 0.0, y = 0.0). This calculation is valid only for planar simulations. The units are N-m/m.

4) The location and value of the maximum magnetic field in the solution space and in each region.

5) Flux integrals to find the self-and mutual inductances of coils. In a planar simulation, the inductance factor for a region is given by

$$IndFact = \frac{\int dx \int dy \; A_z(x,y)}{\int dx \int dy} \; .$$

The integral is taken over the region area. In a cylindrical simulation, the inductance factor is

$$IndFact = \frac{\int dz \int dr\ 2\pi r A_\theta(z,r)}{\int dz \int dr} .$$

A discussion on the use of the vector potential integrals to find self-inductance and mutual inductance is given in S. Humphries, **Field Solutions on Computers** (CRC Press, Boca Raton, 1997), Sect. 10.4.

6) Power density, volume integral of σE^2.

Matrix file

VPulse can make matrix files of values to help you create your own analysis routines. Although information is available in the output file of the solution program, it is often 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 a dialog box where you set the matrix file prefix, the dimensions of the box and the number of intervals along *x* and *y* (or *z* and *r*). The program creates the file `FPREFIX.MTX` in the current directory.

19. VPulse scan plot menu

The commands of the *Scan plot* menu become active when a plot is created following the line scan command. The commands to export plots were described in Section 17. This section describes unique commands of the Scan plot menu.

Oscilloscope mode

In the oscilloscope mode, a scan plot assumes many of the characteristics of a digital oscilloscope. 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

The scan plot must be closed before you can use the *File* and *Analysis* functions of **VPulse**. This command closes the scan plot and returns the program to the previous spatial plot.

20. VPulse script file commands

Script files to control **VPulse** 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 '* ' (asterisk). Processing ends when the *EndFile* command is encountered.

To run a script, click on the *Run script* command in the *File* menu. The program shows a list of available script files. 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 can also sequentially open one or more data files.

VPulse can run autonomously under script file control from the Windows Command Prompt. Suppose you have a file `GTEST.SCR` in the directory `\TRICOMP\BUFFER` and that `VPULSE.EXE` is in the directory `\TRICOMP.` From `\TRICOMP\BUFFER`, type

```
..\VPULSE GTEST <Enter>
```

The main application of the Command Prompt mode is to generate data files and to perform extensive analyses under batch file control.

The following commands can appear in a script:

Input Switch1.003

Close the current data file and load a file for analysis. The parameter is the full name of the data file. For the command illustrated, the post-processor would load the file `SWITCH1.003`. You can load several files for sequential analysis.

Output SW02

Close the current history file and open an output file `SW02.DAT`.

Point 5.65 10.68

Perform interpolations at the specified point and write the results to the data file. The two real number parameters are the coordinates of the point (*x-y* or *z-r*) in **Mesh** units.

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: [x_{start} y_{start} x_{end} y_{end}] or [z_{start} r_{start} z_{end} r_{end}].

Interpolation LSQ

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

Element 5.65 10.68

Write the properties of the element at the specified point to the data file. The two real number parameters are the coordinates of the point (*x-y* or *z-r*) in **Mesh** units.

NScan 150

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

Region 5

Write volume and surface integrals for a region to the data file. The integer parameter is the region number.

LineInt 0.00 0.00 10.00 0.00

Write line integrals along a scan line to the data file. The four real number parameters are the starting and end coordinates in **Mesh** units: [x_{start} y_{start} x_{end} y_{end}] or [z_{start} r_{start} z_{end} r_{end}].

VolumeInt

Write volume integrals for the full solution and regions to the data file.

Matrix Switch1 10 20   0.00 0.00 5.00 10.00

Create a matrix file and record values. This command must have the following seven parameters.
Param 1: The prefix of the matrix file `FPREFIX.MTX` (string).
Param 2: Number of intervals along the *x* (or *z*) direction (integer).
Param 3: Number of intervals along the *y* (or *r*) direction (integer).
Param 4-7: The coordinates of the corners of a box in the solution volume, [x1,y1,x2,y2] or [z1,r1,z2,r2] (real).

Endfile

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

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

```
NSCAN 200
OPEN OUTPUT COMP
OPEN INPUT SWITCH01.001
SCAN 0.00 -50.00 0.00 50.00
OPEN INPUT SWITCH02.002
SCAN 0.00 -50.00 0.00 50.00
OPEN INPUT SWITCH03.003
SCAN 0.00 -50.00 0.00 50.00
OPEN INPUT SWITCH04.004
SCAN 0.00 -50.00 0.00 50.00
ENDFILE
```

21. Format of the Pulse output files

The **Pulse** output files `RUNNAME.001,...` are in text format. They can be inspected with a text editor. Each file has three sections:

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

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

```
--- Run parameters ---
XMin: -8.890018E-02
XMax:  3.810008E-02
KMax:  251
YMin:  0.000000E+00
YMax:  7.620016E-02
LMax:  131
DUnit:  3.937000E+01
NReg:    8
ICylin:  1
Time:  3.500025E-04
```

Lines 2 and 3 list x_{min} and x_{max}, the limits along the 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). Finally, Line 11 lists the code time for the data.

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             A
==================================================================
   1    1    9    5    0 -8.890018E-02  0.000000E+00  0.000000E+00
   2    1    9    5    0 -8.828106E-02  0.000000E+00  0.000000E+00
   3    1    9    5    0 -8.778497E-02  0.000000E+00  0.000000E+00
   4    1    9    5    0 -8.727399E-02  0.000000E+00  0.000000E+00
   5    1    9    5    0 -8.676673E-02  0.000000E+00  0.000000E+00
   6    1    9    5    0 -8.625852E-02  0.000000E+00  0.000000E+00
   ...
```

```
                  MuUp           MuDn           dAdt
           =========================================
             1.000000E+00  1.000000E+00  0.000000E+00
             1.000000E+00  1.000000E+00  0.000000E+00
             1.000000E+00  1.000000E+00  0.000000E+00
             1.000000E+00  1.000000E+00  0.000000E+00
             1.000000E+00  1.000000E+00  0.000000E+00
             1.000000E+00  1.000000E+00  0.000000E+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 about the line between nodes (*K*, *L*) and (*K*+1, *L*) and the lower element lies below the line.

- The coordinates of the element in meters (*x*, *y*)

- The vector potential A_z at the node in tesla-m or the stream function rA_θ in tesla-m^2.

- The values of relative magnetic permeability in the upper and lower associated elements.

- The time derivative of the vector potential or stream function to calculate induced electric field.

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

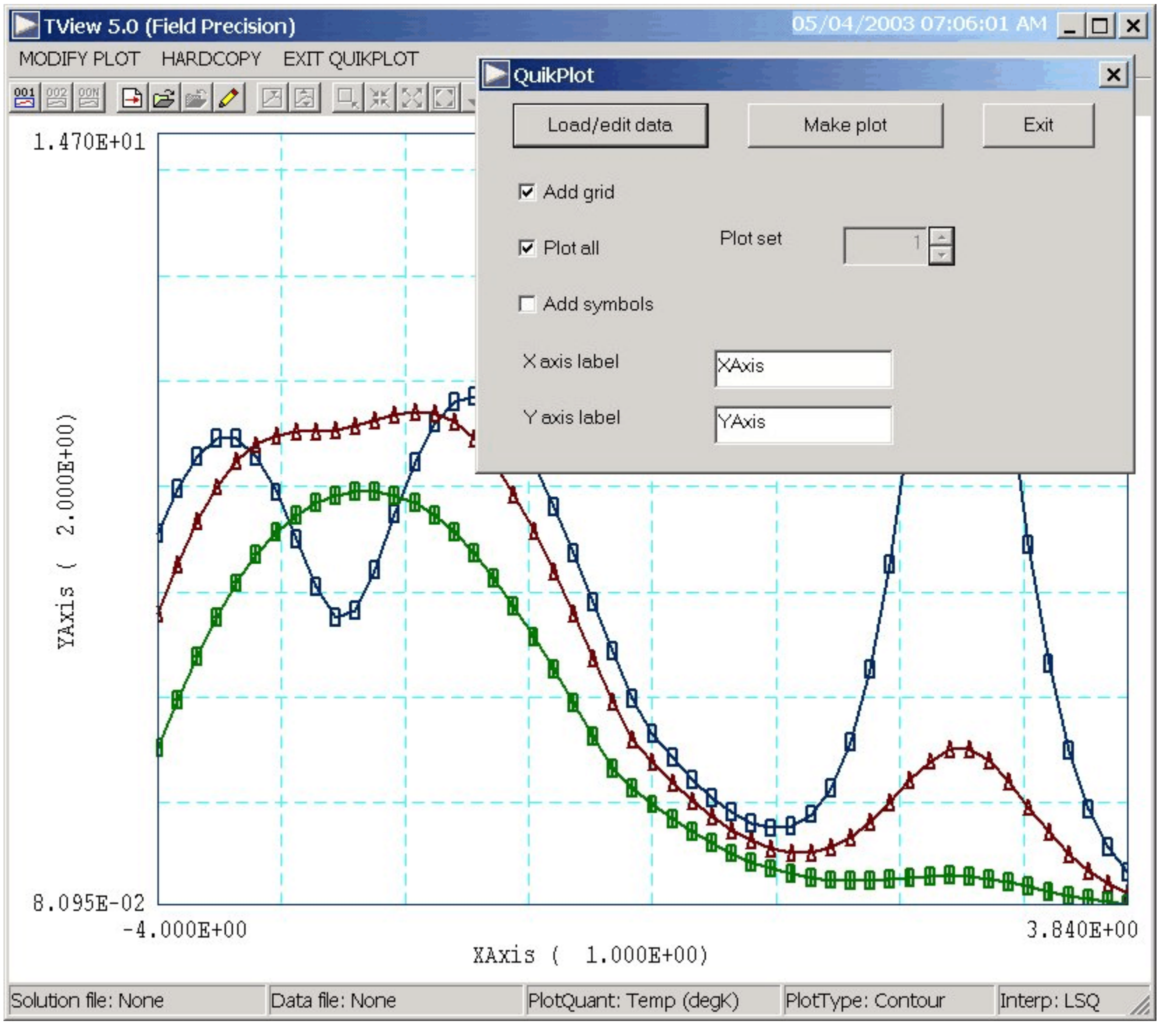
```
--- Regions ---
RegNo Fix                           Mu          Current

   1    0    0    0    0    0  1.000000E+00  0.000000E+00
   2    0    0    0    0    0  1.000000E+00  0.000000E+00
   3    0    0    0    0    0  1.000000E+00  0.000000E+00
   4    0    0    0    0    0  1.000000E+00  6.966977E+04
   5    0    0    0    0    0  1.000000E+00  0.000000E+00
   6    0    0    0    0    0  1.000000E+00  0.000000E+00
   7    0    0    0    0    0  1.000000E+00  0.000000E+00
   8    1    0    0    0    0  1.000000E+00  0.000000E+00


          Vector Pot        Area       Resistivity

        0.000000E+00  7.977806E-03  1.000000E+02
        0.000000E+00  7.000468E-04  2.700000E-08
        0.000000E+00  7.903292E-05  8.100000E-07
        0.000000E+00  2.399991E-04  1.000000E+02
        0.000000E+00  2.632221E-04  1.000000E+02
        0.000000E+00  2.709685E-05  8.100000E-07
        0.000000E+00  3.917838E-04  8.100000E-07
        0.000000E+00  0.000000E+00  1.000000E+37
```

A one in the second column designates a fixed-potential region such as a Dirichlet boundary. The quantity *MU* (μ_r) applies to linear materials and *Vector pot* applies to fixed-potential materials.

22. QuikPlot utility

Postprocessors for **TriComp** and **Amaze** programs contain a utility for making fast, simple plots of numerical data. The feature may be useful, for example, to check a table that defines non-linear magnetic permeability. Click on *QUIKPLOT* in the *TOOLS* menu to bring up a dialog (Fig. 10). Initially, all commands are deactivated except *LOAD/EDIT DATA*. Clicking this command brings up an editor window. You can paste numerical data that you have copied from text files or spreadsheets into the window. You can also use the editor function to make changes or to save the data to a file.

The data should consist of columns of numbers separated by any of the standard delimiters (space, tab, comma, equal sign, parentheses). The first column contains *x*-values. Additional columns contain from 1 to 5 sets of *y*-values. All columns must have the same number of rows. Although there is no limit to the number of rows, the program will pick a maximum of 250 evenly-spaced values to make the plots. Click *SAVE DATA* when you are through.



☑ Add grid

☑ Plot all

Plot set 1

□ Add symbols

X axis label XAxis

Y axis label YAxis

Solution file: None | Data file: None | PlotQuant: Temp (degK) | PlotType: Contour | Interp: LSQ

Figure 10. Control dialog superimposed on **QuikPlot** screen.

When you return to the main dialog all commands should be active. Click *MAKE PLOT* to see a plot of the data. The program adjusts the ranges and grid intervals automatically. When the plot is active you can export it in various formats or copy it to the clipboard. Click *MODIFY PLOT* to return to the main dialog to make changes. Here you can change the display of grids and plot symbols or add text labels to the *x* and *y* axes. Note that the *x*-*y* labels also show the grid intervals. By default, **QuikPlot** shows all data sets (columns of *y*-values) with different line colors and plot symbols.

You can plot a particular set by unclicking the *PLOT ALL* box and specifying the set number (1-5). The plot range is automatically adjusted to the set. In the main dialog, click on *EXIT QUIKPLOT* to return to the normal functions of **VPulse**.