TDiff 8.0
Thermal transport with temperature-dependent materials and radiation boundaries

PO Box 13595, Albuquerque, NM 87192 U.S.A.
Telephone: +1-505-220-3975
EMail: techinfo@fieldp.com URL: http://www.fieldp.com
## Contents

1 **Introduction**  
   1.1 Program functions ............................................. 4  
   1.2 Learning TDiff .................................................. 4  
   1.3 Finite-element solution procedure ............................. 5  
   1.4 Scripts and data files ......................................... 7  
   1.5 Basic equations and definitions ................................. 7  
   1.6 Numerical stability in dynamic solutions ....................... 9  
   1.7 Boundary conditions in thermal solutions ..................... 11

2 **Creating and analyzing a solution with TDiff**  
   2.1 Mesh generation ............................................... 12  
   2.2 Creating the TDiff script ...................................... 13  
   2.3 Finite-element solution ........................................ 15  
   2.4 Solution analysis ............................................... 18

3 **TDiff solution reference**  
   3.1 TDiff script format ............................................ 21  
   3.2 General commands .............................................. 23  
   3.3 Steady-state solutions ......................................... 24  
   3.4 Temperature-dependent thermal conductivity .................. 25  
   3.5 Control of dynamic solutions .................................. 28  
   3.6 Choosing a time step ........................................... 29  
   3.7 Material commands in dynamic solutions ....................... 31  
   3.8 Diagnostic commands for dynamic solutions ................... 33  
   3.9 Running TDiff .................................................. 34  
   3.10 Format of the TDiff output files ............................... 37

4 **TriComp analysis functions**  
   4.1 File menu commands ............................................ 40  
   4.2 Plot menu commands ............................................ 42  
   4.3 User specified contours ........................................ 45  
   4.4 Saving and loading views ...................................... 46  
   4.5 Analysis menu commands ....................................... 48  
   4.6 Scan plot menu ................................................ 51  
   4.7 Vector tools .................................................... 52  
   4.8 Analysis script commands ..................................... 53

5 **Standard TDiff calculated quantities**  
   5.1 TDiff variables ................................................ 57  
   5.2 Special commands .............................................. 59

6 **Probe – history file plot utility**  
   6.1 Introduction .................................................... 60  
   6.2 Loading data files .............................................. 60  
   6.3 Plot settings .................................................. 62
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4</td>
<td>Plot functions</td>
<td>63</td>
</tr>
<tr>
<td>6.5</td>
<td>Information</td>
<td>64</td>
</tr>
<tr>
<td>7</td>
<td><strong>Thermal source import from RFE2, Nelson, WaveSim and GamBet</strong></td>
<td>66</td>
</tr>
<tr>
<td>7.1</td>
<td>Function and commands</td>
<td>66</td>
</tr>
<tr>
<td>7.2</td>
<td>Application example</td>
<td>68</td>
</tr>
<tr>
<td>8</td>
<td><strong>Radiation boundaries</strong></td>
<td>72</td>
</tr>
<tr>
<td>9</td>
<td><strong>Building custom analysis configurations</strong></td>
<td>76</td>
</tr>
<tr>
<td>9.1</td>
<td>Configuration file structure</td>
<td>76</td>
</tr>
<tr>
<td>9.2</td>
<td>Introduction to RPN</td>
<td>78</td>
</tr>
<tr>
<td>9.3</td>
<td>Operators, parameters and variables</td>
<td>79</td>
</tr>
<tr>
<td>9.4</td>
<td>Building expressions</td>
<td>81</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Program functions

TDiff handles a wide variety of thermal transport problems in solids and gels. The program solves the thermal diffusion equation with the option to include perfusion contributions for biological applications (bioheat equation). TDiff employs finite-element methods on variable-resolution conformal triangular meshes for high accuracy and speed. The mesh size is limited only by the installed memory on your computer. The program handles three-dimensional cylindrical problems (symmetry in $\theta$) and two-dimensional rectangular problems (arbitrary variations in $x$ and $y$ with infinite extent in $z$). TDiff can find both steady-state and dynamic solutions. In time-dependent solutions, you can define multiple sources with arbitrary temperature histories using a flexible system of tabular function input. You also have the option to define temperature-dependent thermal conductivity and specific heat. The program can load power deposition data calculated by the Field Precision programs RFE2 (heating by RF electric fields), Nelson (inductive heating), WaveSim (microwave heating) and GamBet (deposition by energetic electrons and photons). In dynamic runs, the program produces multiple data files at specified times and history files of temperature and other quantities at given locations. The program incorporates an interactive graphical environment for making plots and calculating numerical quantities from data files. The package also contains the Probe utility for inspecting history files.

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

1.2 Learning TDiff

The size of this manual reflects the extensive capabilities of the TDiff package. It is not necessary to read the entire document to address most applications. We have organized the chapters to help you get started quickly. You can investigate advanced topics as you gain more experience. The first step is to follow the walkthrough example in Chapter 2. This will give you an overview of the package organization and the solution process. Then, run some of the prepared examples supplied with the package. The examples give insights into solution options and may serve as templates for your own work. Here are suggestions for additional activities:

- Browse the following sections in this chapter to review some basics of thermal transport. You will probably return to the material to resolve issues (such as boundary conditions) as you get more involved in the programs.

- Scan the section on FPController (the component program launcher) in the Mesh manual.
- 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 introduces fundamental tools you will need for your own simulations. Chapter 5 describes how to convert a script into a conformal mesh.

- As you gain experience you will want to take advantage of the full range of Mesh/TDiff 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. Chapters 4 and 5 in this manual summarize the solution and analysis functions of TDiff. Chapter 6 describes the capabilities of the Probe program. Chapter 7 summarizes how to incorporate information from RFE2, Nelson, WaveSim or GamBet calculations. Chapter 8 covers radiation boundaries. Finally, Chap. 9 describes how you can customize TDiff analysis function for your application.

- 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 directly from photographic and data images. This feature is useful to model complex or irregular systems that are difficult to describe with simple geometric specifications.

1.3 Finite-element solution procedure

An in-depth understanding of finite-element numerical methods is not necessary to use TDiff. 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 TDiff 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 thermal diffusion equation (Sect. 1.5). These equations can be solved directly by analytic methods if the system geometry and material properties are simple (for example, layers with uniform thermal conductivity 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 TDiff. The figure defines three terms that will be used throughout this manual:

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

The elements in Fig. 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 TDiff, solution of the set representing the thermal diffusion equation gives the temperature at each node. We can then perform two-dimensional interpolations to find...
temperature at intervening points, or take numerical spatial derivatives to find the components of the thermal flux.

With this background, we can understand the steps in a TDiff solution:

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

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

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

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

5. TDiff solves the coupled equations to find the node temperatures and self-consistent element properties. The numerical method depends on whether the calculation is an initial-value or boundary-value problem. The program records all information 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 TDiff to explore the solution. The program creates a wide variety of plots and performs quantitative calculations of thermal quantities. You can also write scripts for automatic control of complex analyses.

1.4 Scripts and data files

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

1.5 Basic equations and definitions

The reference S. Humphries, Field Solutions on Computers (CRC Press, Boca Raton, 1997) gives detailed information on the theory of thermal transport and the application of finite-element methods in TDiff. This section summarizes some basic concepts that may be helpful in preparing solutions.

The dynamic thermal transport equation has the following differential form:
Table 1: TDiff files

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>MName.MIN</td>
<td>Mesh input script (definition of foundation mesh and region outlines)</td>
</tr>
<tr>
<td>MName.MLS</td>
<td>Mesh diagnostic listing</td>
</tr>
<tr>
<td>MName.MOU</td>
<td>Mesh output (node locations and element identifies)</td>
</tr>
<tr>
<td>TName.TIN</td>
<td>TDiff input script (run control and material properties)</td>
</tr>
<tr>
<td>TName.TLS</td>
<td>TDiff diagnostic listing</td>
</tr>
<tr>
<td>TName001.TOU</td>
<td>One or more TDiff data dumps (node locations and temperature at specific times)</td>
</tr>
<tr>
<td>TName.P01</td>
<td>From 1 to 20 TDiff history files (time-dependent temperature at specific positions)</td>
</tr>
<tr>
<td>AName.SCR</td>
<td>TDiff script for automatic data analysis</td>
</tr>
</tbody>
</table>

Table 2: TDiff standard units

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial dimensions</td>
<td>meters or units set by <em>DUnit</em></td>
</tr>
<tr>
<td>Time</td>
<td>seconds</td>
</tr>
<tr>
<td>Temperature</td>
<td>degrees Centigrade, °C</td>
</tr>
<tr>
<td>Material density</td>
<td>kg/m³</td>
</tr>
<tr>
<td>Specific heat (constant pressure)</td>
<td>J/kg-°C</td>
</tr>
<tr>
<td>Thermal conductivity</td>
<td>W/m-°C</td>
</tr>
<tr>
<td>Thermal sources or sinks</td>
<td>W/m³</td>
</tr>
</tbody>
</table>

\[ \rho C_p \frac{\partial T}{\partial t} = \nabla (k \nabla T) + Q_{reg}(t) + Q_{file}(t) + Q_{perf}(t) + Q_{rad}(t). \]  

where \( \rho \) is the local mass density in kg/m³, \( C_p \) is the specific heat in J/kg-°C, \( T \) is the temperature in °C \( k \) is the heat conductivity in J/m-s-°C and \( Q \) represents a source or sink of thermal energy in W/m³. In steady-state, the temperature is determined by the Poisson equation:

\[ \nabla (k \nabla T) = -Q_{reg}(t) - Q_{file}(t) - Q_{perf}(t) - Q_{rad}(t), \]  

Note that the static equation does not involve the material specific heat or mass density. The standard SI units used for thermal quantities are listed in Table 2 and Table 3 gives some useful conversion factors.

Four sources or sinks of thermal energy may be included in a calculation:

- **Region sources**, \( Q_{reg} \). These sources are spatially uniform throughout a region of the solution volume. Values are specified in the TDiff script. Time variations may be assigned to region sources in dynamic solutions.
Table 3: Thermal conversion factors

<table>
<thead>
<tr>
<th>Quantity</th>
<th>From</th>
<th>To</th>
<th>Multiply by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific heat, ( C_p )</td>
<td>cal/(gm-oC)</td>
<td>J/(kg-oC)</td>
<td>4180.0</td>
</tr>
<tr>
<td>Density, ( \rho )</td>
<td>gm/cm(^3)</td>
<td>kg/m(^3)</td>
<td>1000.0</td>
</tr>
<tr>
<td>Conductivity, ( k )</td>
<td>cal/(cm-s-oC)</td>
<td>J/(m-s-oC)</td>
<td>418.0</td>
</tr>
</tbody>
</table>

- **File sources**, \( Q_{file} \). These sources have spatial variations over the solution volume. Values are loaded from files generated by **RFE2**, **Nelson** or **WaveSim**. In a dynamic solution, you can specify a modulation function to introduce a global time variation.

- **Thermal radiation**, \( Q_{rad} \). These sources represent losses through thermal radiation from the boundaries of extremely hot objects (e.g., thermionic cathodes).

- **Perfusion**, \( Q_{perf} \). These sources represent heat transfer by fluid perfusion. They are usually used to represent effects of blood flow through a microscopic network of capillaries in biomedical solutions.

Procedures to load file sources and to apply modulations are discussed in Chap. 7 while Chap. 8 covers thermal radiation from boundaries. The process of perfusion is the permeation of animal tissue by a flow of blood through a multiplicity of capillaries. The scale size for the effect is usually too small to treat directly. Instead, perfusion is modeled by a source or sink term that represents an average mass flow of blood that enters the tissue at an ambient temperature \( T_b \) and exits at the local tissue temperature. The expression for the power exchange per unit volume is

\[
Q_{perf} = W_b C_b (T_b - T),
\]

where \( T \) (in °C) is the local tissue temperature, \( T_b \) (in °C) is the ambient entering blood temperature, \( W_b \) (in kg/m\(^3\)-s) is the average blood mass flow per unit volume and \( C_b \) (in J/kg-°C) is the specific heat of blood.

### 1.6 Numerical stability in dynamic solutions

In dynamic solutions **TDiff** uses the time-centered Dufort-Frankiel method to advance the diffusion equation. The method 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 triangular meshes where a contiguous set of elements have an internal angle that exceeds 90.0 °. This condition may occur when elements have small height-to-width ratio (\( \Delta y \ll \Delta x \)). **Mesh** includes routines to correct the shapes of elements to avoid the instability condition. For meshes that will be used in dynamic calculations, you must include the command **DCORRECT** in the **Global** section of the **Mesh** script (**FPrefix.MIN**). Figure 2 shows the effect of the correction on a mesh region where the height to width ratio is about 0.4. The following global section was used to generate the mesh of the lower plot:
Figure 2: Effect of the $DCorrect$ command in Mesh. Top: before correction. Bottom: after correction.

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

Note that mesh correction is not necessary for static thermal solutions.
1.7 Boundary conditions in thermal solutions

The boundary is the outer edge of the solution volume. It may or may not correspond to the edges of the solution rectangle. For thermal problems, nodes on the boundary may assume one of two conditions.

- **Dirichlet.** Boundary nodes have a fixed temperature (either constant or with a specified time variation) that does not depend on the temperature at neighboring nodes. Thermal flux is normal to such a surface.

- **Neumann.** A Neumann boundary is one where the normal derivative of the temperature is specified. The boundaries in TDiff are limited to the special case \( \partial T / \partial n = 0 \). The special Neumann condition implies that the thermal flux is parallel to the surface. In this case, the boundary acts as an ideal thermal insulator. One advantage of the finite-element method is that all boundaries that are not fixed will automatically satisfy the special Neumann condition, even if they are slanted or curved. Neumann boundaries are often used to reduce computation time for symmetric systems.
Figure 3: Geometry for the WALKTHROUGH example, showing region numbers and the location of the probe (close-up view).

2 Creating and analyzing a solution with TDiff

2.1 Mesh generation

As a quick introduction to TDiff techniques and capabilities, we shall work through the example shown in Fig. 3. The simulation illustrates a dynamic calculation with a time-variation of applied temperature and a temperature-dependent thermal conductivity. A cylindrical metal probe penetrates a spherical volume of material with a relatively high value of non-linear thermal conductivity $k$. This material is surrounded by an infinite region of poor conductor at ambient temperature (37 °C). The procedure is to raise the temperature of the probe to 95 °C and to determine the range of heated material after a treatment time of 1000 seconds.

In preparation move the files WALKTHROUGH.MIN, WALKTHROUGH.TMP and WALKTHROUGH.CND to a working directory. Run Mesh and use the Edit file command to load WALKTHROUGH.MIN. Note the presence of the DCorrect command (diffusion correction) in the Global section. Following the discussion of Sect. 1.6, the command must appear in all Mesh scripts used for dynamic TDiff solutions to avoid a numerical instability. Enter the drawing editor with the Edit script/Graphics command and confirm that the solution volume contains the following regions:
1. **Tissue** (Filled). Main section of the solution volume with a fixed thermal conductivity.

2. **InfusedTissue** (Filled). Volume surrounding the probe with a temperature dependent conductivity.

3. **ProbeSheath** (Filled). Shield with poor thermal conductivity covering the inactive portion of the probe.

4. **Probe** (Filled). Metal structure with a specified, time-dependent temperature.

5. **Boundary** (Open). Outer and left boundaries, set to 37 °C to approximate a large surrounding medium.

Close the editor and return to the main menu. Click the Process command and then the Save mesh (MOU) command. Before leaving the program, check the mesh by clicking the Plot/Repair command. In the plot, note the use of variable resolution to achieve high accuracy. It is not possible to simulate an infinite system in a finite-element calculation, but we can make a good approximation. The active probe area is surrounded by an extended volume at low resolution and the boundary is set to ambient temperature.

### 2.2 Creating the TDiff script

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

1. Set up program controls and material properties.

2. Generate and solve the finite-element equations.

3. Analyze the solution.

To start, click the 1 tool or the Setup menu command. The initial dialog determines whether the solution is dynamic or static. Accept the default and proceed. In the next dialog, identify the **Mesh** output file that defines the system geometry. Choose the file WALKTHROUGH.MOU. TDiff loads the information and displays the dialog of Fig. 4. Note that the grid contains a row for each mesh region with a notation of the Fill status. In the dynamic mode, there are five columns in the grid where you can enter values: **Cond** (thermal conductivity in W/m·°C), **Cp** (specific heat in J/kg·°C), **Dens** (material density in kg/m³), **Source** (uniform thermal source or sink in W/m³) and **Temp** (fixed or initial temperature in °C).

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

- **GEOMETRY.** The symmetry of the solution: *planar* or *cylindrical*. A planar solution varies in *x*-y and has infinite length in *z*, while a cylindrical solution has symmetry in *θ*.

- **DUNIT.** Set a factor to convert the units used for coordinates in the **Mesh** file to meters. Use the standard options or define your own. The value is the number of mesh units per meter: 39.37 for inches, 100.0 for cm.
Figure 4: Dialog to set TDiff control parameters and region properties.
• TMAX. The duration of the solution in seconds.
• DTMIN. A limit on the minimum time step in seconds. Section 3.6 discusses criteria for choosing values.
• DTMAX. A limit on the maximum time step.

The column options in the large grid box specify the material properties to associate with regions. The values in Fig. 4 define the following characteristics:

- Region 1. Thermal conductor with \(k = 0.05 \text{ W/m}\cdot\text{oC}, C_p = 5000.0 \text{ J/kg}\cdot\text{oC} \) and \(\rho = 1100.0 \text{ kg/m}^3\) at an initial temperature 37.0 °C.
- Region 2. Thermal conductor with temperature dependent thermal conductivity defined by the file WALKTHROUGH.CND, \(C_p = 5000.0 \text{ J/kg}\cdot\text{oC} \) and \(\rho = 1100.0 \text{ kg/m}^3\) at an initial temperature 37.0 °C.
- Region 3. Thermal conductor with \(k = 0.01 \text{ W/m}\cdot\text{oC}, C_p = 2500.0 \text{ J/kg}\cdot\text{oC}, \) and \(\rho = 800.0 \text{ kg/m}^3\) at an initial temperature 0.0 °C.
- Region 4. Fixed-region with at time-dependent temperature defined by the file WALKTHROUGH.TMP.
- Region 5. Fixed-temperature boundary at 0.0 °C.

Finally, the values in the Diagnostics section control data dumps and history files. The entry DTime = 200.0 gives data dumps at 200, 400, 600, 800 and 1000 seconds. The entries in the Data record times list give additional data dumps at 100.0 and 500.0 seconds. Finally, the entries in the Probe positions grid set two probes at positions \((z, r) = (0.0, 0.0)\) and \((0.0, 1.5)\).

Enter the values shown in Fig. 4 and click OK. TDiff uses the information in the dialog to create the script WALKTHROUGH.TIN shown in Table 4. Chapter 3 reviews the script format and advanced program capabilities.

### 2.3 Finite-element solution

The next step is to generate the finite-element equations and to advance the solution in time. Click the tool marked 2 or the menu command Solve. Accept the default entry of WALKTHROUGH.TIN in the dialog by clicking OK. The screen lightens to indicate that the program is in solution mode and the status bar reports the progress of the operation. As the simulation advances, the program creates a listing file of information (WALKTHROUGH.TLS) and six data dumps: WALKTHROUGH001.TOU at \(t = 100\) s through WALKTHROUGH006.TOU at 1000 s. The file WALKTHROUGH.P01 contains temporal information at \((z = 0.0, r = 0.0)\) while WALKTHROUGH.P02 list variations at \((z = 0.0, r = 1.5)\). Section 3.10 describes the format of the output files.

The solution is complete. Before proceeding to analyses, we shall take a look at some components of the simulation. In the File menu pick Edit files and load WALKTHROUGH.CND. This file (shown in Table 5) defines a temperature-dependent conductivity. The value of \(k\) is 0.5 W/m\cdot°C at low temperatures and drops to 0.3 W/m\cdot°C in the range 60 °C to 90 °C. In a biological application the difference could represent chemical changes in tissue at increasing temperature. You can also check the file WALKTHROUGH.TMP. The values describe a temperature that rises smoothly from 0.0 °C to 95.0 °C in 15 s and then remains constant for the duration of the solution.
Table 4: **TDiff** script for the **WALKTHROUGH** example

* ---- CONTROL ----*
Mesh = WalkThrough
Mode = TVar
Geometry = Cylin
DUnit = 1.0000E+02
TMax = 1.0010E+03
DtMin = 1.0000E-02
DtMax = 5.0000E-01

* ---- REGIONS ----*
* Region 1: TISSUE
Cond(1) = 5.0000E-02
Cp(1) = 5.0000E+03
Dens(1) = 1.1000E+03
Init(1) = 3.7000E+01

* Region 2: INFUSEDTISSUE
Cond(2) = TABLE walkthrough.CND
Cp(2) = 5.0000E+03
Dens(2) = 1.1000E+03
Init(2) = 3.7000E+01

* Region 3: PROBESHEATH
Cond(3) = 1.0000E-02
Cp(3) = 2.5000E+03
Dens(3) = 8.0000E+02
Init(3) = 0.0000E+00

* Region 4: PROBE
Fixed(4) = TABLE walkthrough.TMP

* Region 5: BOUNDARY
Fixed(5) = 3.7000E+01

* ---- DIAGNOSTICS ----*
DTime = 2.0000E+02
SetTime = 1.0000E+02
SetTime = 5.0000E+02
History = 0.0000E+00 0.0000E+00
History = 0.0000E+00 1.5000E+00

EndFile
Table 5: Definition of temperature-dependent conductivity - contents of the file WALKTHROUGH.CND

* Walkthrough example, decreasing thermal conductivity

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Conductivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>0.5000</td>
</tr>
<tr>
<td>20.0000</td>
<td>0.5000</td>
</tr>
<tr>
<td>40.0000</td>
<td>0.5000</td>
</tr>
<tr>
<td>59.0000</td>
<td>0.5000</td>
</tr>
<tr>
<td>60.0000</td>
<td>0.5000</td>
</tr>
<tr>
<td>62.0000</td>
<td>0.4978</td>
</tr>
<tr>
<td>64.0000</td>
<td>0.4914</td>
</tr>
<tr>
<td>66.0000</td>
<td>0.4809</td>
</tr>
<tr>
<td>68.0000</td>
<td>0.4669</td>
</tr>
<tr>
<td>70.0000</td>
<td>0.4500</td>
</tr>
<tr>
<td>72.0000</td>
<td>0.4309</td>
</tr>
<tr>
<td>74.0000</td>
<td>0.4105</td>
</tr>
<tr>
<td>76.0000</td>
<td>0.3895</td>
</tr>
<tr>
<td>78.0000</td>
<td>0.3691</td>
</tr>
<tr>
<td>80.0000</td>
<td>0.3500</td>
</tr>
<tr>
<td>82.0000</td>
<td>0.3331</td>
</tr>
<tr>
<td>84.0000</td>
<td>0.3191</td>
</tr>
<tr>
<td>86.0000</td>
<td>0.3086</td>
</tr>
<tr>
<td>88.0000</td>
<td>0.3022</td>
</tr>
<tr>
<td>90.0000</td>
<td>0.3000</td>
</tr>
<tr>
<td>92.0000</td>
<td>0.3000</td>
</tr>
<tr>
<td>95.0000</td>
<td>0.3000</td>
</tr>
<tr>
<td>100.0000</td>
<td>0.3000</td>
</tr>
</tbody>
</table>

ENDFILE
2.4 Solution analysis

We can now use the information in the data files (WALKTHROUGH001.TOU...) to create plots and to perform quantitative analyses. Choose either the tool marked 3 or the Analyze menu command. TDiff shifts the analysis menu and toolbar. Choose the Load solution file menu command or tool. Pick WALKTHROUGH001.TOU and click OK. The program loads the data and creates a default plot of temperature contours at time $t = 100$ s. Use the menu command Load from series and pick the sixth data file ($t = 1000$ s). Figure 5 shows a zoomed view of the temperature distribution near the probe. Take a moment to inspect the organization of the working environment. The status bar at the bottom lists the following information: solution file name, data file name, plot quantity, plot type, interpolation method and mouse snap status. (The data file gives you the option to record the results of analysis operations in text format.) The plot window is divided into four areas. The main plot is at the upper left and the plot legend is at the upper right. The orientation area at the lower right is a miniature representation of the solution volume with an outline showing the current view. The information area at the bottom left appears in response to analysis commands such as Point calculation or Region properties.

Next, pick Plot type in the Plots/Plot settings menu and specify Element. In this plot the temperature is displayed by color-coding of the solution-volume elements. Pick Plot quantity in the settings menu and specify $k$ (thermal conductivity). The program creates the plot of Fig. 6. Note that a sheath of low conductivity has formed around the probe, inhibiting thermal conduction.

In addition to plots, TDiff can perform several quantitative calculations. In the File menu click on Open data record and accept the default file prefix WALKTHROUGH. The results of all subsequent operations will be recorded in the text file WALKTHROUGH.DAT. We want to find the
Figure 6: Color-coded element plot of thermal conductivity, WALKTHROUGH example.

Figure 7: Plot of the time-variation of temperature at the probe position created by the Probe program.
total power flux into the infused region from the probe (both directly and through the sheath). Choose the command Surface integral in the Analysis menu. In the dialog, set the probe and sheath regions to Internal and the infused region to External. The data are recorded when you click OK. After closing the data file with the Close data record command, you can inspect it using the Edit file command. The entry for the surface integral is:

<table>
<thead>
<tr>
<th>RegNo</th>
<th>Status</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>External</td>
<td>INFUSEDTISSUE</td>
</tr>
<tr>
<td>3</td>
<td>Internal</td>
<td>PROBESHEATH</td>
</tr>
<tr>
<td>4</td>
<td>Internal</td>
<td>PROBE</td>
</tr>
</tbody>
</table>

Area of internal/external boundary: 3.137468E-04 (m2)

Quantity: Power
Normal integral: 1.281702E+00

About 1.28 W enters the infused region. To find the power leaving the infused volume, set it as the Internal region and set the tissue as the external region. The results is 0.405 W. In the steady state solution, the power entering and leaving the infused region should be equal. Therefore, the infused region is still heating at $t = 1000$ seconds. To confirm this conclusion, we can look at the record of the probe at the position shown in Fig. 3 (colored triangle). From TC, run Probe, click on Load probe file and choose WALKTHROUGH.P02. You should see the plot of Fig. 7. The temperature in the outer portion of Region 2 is still rising and will take several thousand seconds to approach an equilibrium.
Table 6: Material models from entries in the TDiff dialog – steady-state calculations

<table>
<thead>
<tr>
<th>Material type</th>
<th>Cond</th>
<th>Source</th>
<th>Temp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conductor</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conductor/source</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Conductor/initial temp.</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Fixed temperature</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Table 7: Material models from entries in the TDiff dialogs – dynamic calculations

<table>
<thead>
<tr>
<th>Material type</th>
<th>Cond</th>
<th>Cp</th>
<th>Dens</th>
<th>Source</th>
<th>Temp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conductor</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conductor/source</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Conductor/initial temp.</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Fixed temperature</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

3 TDiff solution reference

3.1 TDiff script format

You can create a control script for TDiff interactively using the dialog described in Sect. 2.2 or by writing the commands directly with a text editor. You can easily alter any script with an editor. This chapter gives a detailed description of the syntax and functions of script commands. Section 3.2 covers commands that control general program operation in both the static and dynamic modes. Section 3.3 covers commands to control the program and to set simple material properties in static solutions. Table 6 summarize how entries in the setup dialog control the type of material associated with a region. Section 3.4 covers methods to define temperature-dependent material characteristics. The next four sections cover script commands to control dynamic solutions. Section 3.6 covers the important topic of setting the time step to advance the diffusion equation. Finally, Sect. 3.10 summarizes the format of the TDiff data dump file.

The TDiff script is a text file composed of data lines that contain commands and parameters. You can construct a script using the Setup dialogs in TDiff. You can also write and modify scripts with an editor. Direct script editing is required for some advanced TDiff 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 [',']
Any number of delimiters may be used in a line. Blank lines and comment lines are ignored. Comment lines begin with an asterisk (*). TDiff accepts commands in any order. The following example illustrates a complete script for a steady-state solution:

```
* File: VARCALC_BV.TIN
Mesh = VARCALC_BV
Mode = Stat
Geometry = Cylin
DUnit = 1.0000E+02
ResTarget = 5.0000E-07
MaxCycle = 600
* Region 1: SOLVOLUME
Cond(1) = TABLE varcon.CND
* Region 2: INNERSPHERE
Fixed(2) = 1.0000E+02
* Region 3: OUTERSPHERE
Fixed(3) = 0.0000E+00
EndFile
```

There are two classes of commands that appear in all solutions: program control and region properties. Dynamic solutions also contain diagnostic commands. 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

```
FIXED 2 -48.5
```

sets nodes with region number 2 to the fixed temperature 48.5 °C. (Note: For back-compatibility, TDiff 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 General commands

The commands listed in this section may appear in all TDiff scripts. Note that in this section
and the following ones, commands are written symbolically and as they might appear in the
script:

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

MODE [TVar, Stat]
MODE = Stat
A required command to set the solution type. The option TVar signifies a time-dependent,
initial-value solution while Stat implies a steady-state, boundary-value solution. The allowed
commands and material models depend on the solution type.

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 TDiff. 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 × 10^6. The default value is 1.00. (Note.
Spatial quantities recorded in the output files FPrefix.001 are always in meters). In an analysis
session with TDiff, 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
TDiff 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.

SOURCEFILE FileName [SFact]
SOURCEFILE = LiverProbe.ROU 2.5
Load a thermal source file from RFE2, Nelson, WaveSim or GamBet. Chapter 7 describe
this program function in detail.
INTERP [Linear, Spline]
INTERP = Linear
Set the interpolation mode for quantities where a time or temperature variations is described by a table of values (Sect. 3.4). The default cubic spline method gives the highest accuracy for smooth data (continuous value and slope). Use linear interpolation when data points are noisy or discontinuous.

3.3 Steady-state solutions

TDiff performs a steady-state calculation when the command Mode Stat appears in the script. In this case, the program solves the static thermal transport equation (Eq. 2). Static solutions are straightforward and the associated command set is compact. Specified temperatures and thermal sources have fixed values that do not vary in time. The only advanced technique is the assignment of temperature-dependent k discussed in the next section. The first set of commands control program operation.

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

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

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.

The next four commands set the material properties of regions. The commands have the form:

Keyword RegNo Value

The quantity RegNo (an integer) is the region number defined in the Mesh input file, while Keyword is one of the options listed below and Value is a real number.
FIXED RegNo TFix
FIX(5) = 37.2
Sets the region to a fixed temperature that does not depend on the temperature at neighboring nodes. Enter the value in °C.

The remaining commands control thermally-conductive materials. Multiple commands may be used to define a region.

INIT RegNo TInit
INIT(3) = 2815
Sets an initial temperature for a conductive region in °C. The settings has no effect on the final solution, but it may influence the convergence rate in solutions with radiative boundaries (Sect 8). The default starting temperature for variable points is 0.0 °C.

COND RegNo KReg
COND(8) = 3.45
Sets a fixed value of thermal conductivity in W/(m-°C).

SOURCE RegNo QReg
SOURCE(15) = -4.5E6
Creates a heat source or sink uniformly distributed over the region volume. The units are W/cm³.

3.4 Temperature-dependent thermal conductivity
One of the most powerful features of TDiff is the ability to handle time-dependent sources and temperature-dependent thermal properties. For example, in a simulation of a biological system the conductivity may decrease as tissue cooks or dessiclates. In this case, the Poisson equation assumes the non-linear form:

$$\nabla \cdot [k(T) \nabla T] = -Q.$$  (4)

It is important to realize that non-linear equations may not always have a unique solution. For example, consider a case where a region with a thermal source $Q$ has a conductivity $k$ that decreases with temperature. A solution may be possible at low values of $Q$, but at higher values an instability may occur. The temperature rises with decreasing conductivity while $k$ drops with increasing $T$. The end result is a divergent temperature. In the physical world, such a system disassembles. In the numerical world the program generates invalid real numbers and crashes. In the example, the allowed range of $Q$ depends on the functional dependence $k(T)$. There is no foolproof method to predict stability of non-linear solutions.

TDiff gives you considerable flexibility to model temperature-dependent conductivity. The key to this flexibility is program input through tabular functions. A tabular function is a text file consisting of up to 256 data lines containing values for an independent and dependent variable. For the case of temperature-dependent thermal conductivity, each line has a temperature (in °C) followed by the corresponding value of $k(T)$ (in W/m-°C). You can prepare tabular function...
files with a text editor or spreadsheet. You can also use published data or digitized experimental traces. Table 8 shows a tabular function generated with a spreadsheet. The intention was to model a sharp transition in thermal conductivity between two values. Note that the file syntax conforms to the general rules of TriComp scripts. The free-form parser accepts real numbers in any format with a choice of delimiters. You may add documenting comment lines starting with an asterisk (*). The end of the data is marked with the EndFile command. You can add text in any format after EndFile.

Although the lines in Table 8 are arranged by increasing value of the independent variable, this ordering is not required. TDiff sorts the list before use and records the final order in the file FPREFIX.TLS. Furthermore, the independent variable intervals need not be uniform. Note that the entries in the example are clustered near the transition temperature. 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 temperature that will be encountered in the solution. The interpolation routines for \( k \) return 0.0 for values of the independent variable that are out of the table range. Zero conductivity values can lead to non-physical results or a program crash. Be sure that the table extends below \( T_{Init} \) and well above the maximum temperature anticipated in the region.

In the default mode TDiff 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, as in Table 8. You can check the fidelity of the interpolations by inspecting the file FPREFIX.TLS. After listing the sorted table, TDiff records a sample set of interpolated values.

Temperature-dependent conductivity is defined by a modified form of the Cond command in both steady-state and time-dependent solutions.

**COND RegNo TABLE TabName**

**COND(2) = (TABLE, ALUM.CND)**

Define the thermal conductivity of the region as a function of temperature from a tabular function file. The command contains the keyword Table and the full name of a data file in the working directory. The file may contain up to 256 data lines. Each data line contains two real-number values: temperature (in °C) and the corresponding value of \( k \) (in W/m·°C).

**AVG Alpha NAAlpha**

**AVG = (0.35, 20)**

In a steady-state solution, TDiff adjusts values of \( k \) in variable elements during the relaxation process. For a stable system the default adjustment parameters in the program are usually sufficient to achieve convergence. For difficult cases where \( k \) may change by a large factor over a small temperature range, you can use this command to change the adjustment factors. The first real-number parameter \( \alpha \) is a smoothing factor to prevent oscillations during the relaxation. Enter a value from 0.0 to 1.0. The conductivity in an element is adjusted according to

\[
k_{new} = (1.0 - \alpha) k_{old} + \alpha k(T_{new}).
\]

Small values (\( \alpha \ll 1 \)), which give strong smoothing, may be required for difficult solutions. In this case, be sure to set MaxCycle high enough so that the material can reach equilibrium.
Table 8: Example of a data table to define a temperature-dependent thermal conductivity

* TWAVE.CND
* T(deg-C) K(J/(m-s-degC))
* ================
    0.0  10.00
    200.0  10.00
    290.0  10.00
  3.000E+02  1.000E+01
  3.025E+02  1.019E+01
  3.050E+02  1.076E+01
  3.075E+02  1.177E+01
  3.100E+02  1.331E+01
  3.125E+02  1.551E+01
  3.150E+02  1.854E+01
  3.175E+02  2.266E+01
  3.200E+02  2.815E+01
  3.225E+02  3.538E+01
  3.250E+02  4.472E+01
  3.275E+02  5.653E+01
  3.300E+02  7.105E+01
  3.325E+02  8.828E+01
  3.350E+02  1.079E+02
  3.375E+02  1.290E+02
  3.400E+02  1.502E+02
  3.425E+02  1.699E+02
  3.450E+02  1.859E+02
  3.475E+02  1.963E+02
  3.500E+02  2.000E+02
  360.0  200.0
  400.0  200.0
  500.0  200.0
  1000.0  200.0
  2000.0  200.0
ENDFILE
The default value is $\alpha = 0.25$. The second parameter $N_\alpha$ (an integer) equals the number of relaxation cycles per material adjustment. The default value is $N_\alpha = 10$. Lower the value for solutions that converge rapidly.

### 3.5 Control of dynamic solutions

**TDiff** performs a dynamic calculation when the command *Mode TVar* appears in the script. In this case, the program solves Eq. 1. There are more options in dynamic solutions, so there is a larger command set. This section reviews commands that control general program operation.

**TMAX Tmax**

$TMAX = 345.2$

This required command sets the duration of the run. Note that all runs begin at $t = 0.0$. Enter the value in seconds.

**SAFETY Safety**

$SAFETY = 35.0$

Set a safety factor for automatic time-step adjustment. Higher values improve accuracy at the expense of longer computation times. The default value is 20.0.

**DTMIN DtMin**

$DTMIN = 0.0001$

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

**DTMAX DtMax**

$DTMAX = 0.25$

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

**NCHECK NCheck**

$NCHECK = 10$

Set the number of time steps between recalculations by the automatic time step routine. The default value is 5.

**NSTEP NStep**

$NSTEP = 5000$

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 $NStep = 100,000$. 

28
SOURCEFILE FileNa ne [SFa ct]
SOURCEFILE = LiverProbe.ROU 2.5
Load a thermal source file from RFE2 or Nelson (Chapter 7).

SOURCEMOD TabFileName
SOURCEMOD = TenPulse.MOD
Use this command to assign a time-variation to spatially-distributed thermal power sources from Nelson, RFE2 or WaveSim loaded with the SourceFile command. See Chap. 7 for a complete discussion.

TSTEPMOD TabFileName
TSTEPMOD = PulseTrain.MOD
Use this command to define a time-dependent integration time step, over-riding automatic time step selection. This command (valid only in dynamic calculations) is particularly useful when a source file loaded from RFE2, Nelson or GamBet (Chap. 7) has a strong time modulation (i.e., a periodic train of short pulses). In this case, it is essential that TDiff uses a very short time step while the source is active and a long time step in the intervening periods. The string TabFileName is the full name of a file that defines the modulation waveform, f(t). The integration time step equals the product of the modulation function and the minimum time step, \( \Delta t = DtMin \times f(t) \). Therefore, you must define a nonzero value for DtMin. The modulation file has the form of a tabular function (Chap. 3.4). It may contain up to 256 data lines – each line contains two numbers: t and f(t). Enter the time t in seconds. (Note that all dynamic simulations start at t = 0.0.) Note that the linear interpolation mode is always used for the time-step modulation function. Therefore, the function may have discontinuities of value or slope. If the simulation time t exceeds the maximum time \( t_{max} \) of table entries, the interpolation time is taken as \( t = \text{mod}(t, t_{max}) \). This feature is useful for defining a long train of periodic time-step changes.

TSTEPMOD > Function
TSTEPMODE > 10.0 + cos(3.1416*$t/25.6)
Define a modulation function for the time step from a mathematical expression. The keyword TStepMod followed by the > symbol designates that a function string occupies the remainder of the line. The function may be up to 230 characters in length and follows the format described in Sect. 7.1. The function defines a variation in time, f(t). (Note that the parser uses the Perl standard for the time variable: $t stands for t. The integration time step equals the product of the function and the minimum time step, \( \Delta t = DtMin \times f(t) \).

3.6 Choosing a time step
In the dynamic mode, TDiff follows thermal transport 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 initial temperature on a body in a surrounding medium at ambient temperature. Initially, the full temperature drop 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 thermal diffusion in the full solution space. To perform this large adjustment, **TDiff** has a routine that estimates the best time step. The procedure relies on a calculation of the root-mean-squared value for the second derivative of temperature 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 **TDiff** handles. We shall review in detail the three commands that allow you to modify the time step: *Safety*, *DtMin* and *DtMax*:

**SAFETY = Safety**

**SAFETY = 15.0**

The automatic time-step routine calculates the maximum possible time increment and then divides by the factor *Safety*. A conservative value is *Safety* = 20. Higher values improve accuracy but lengthen the computation time.

**DTMIN DtMin**

**DTMIN = 5.0E-7**

The automatic time step feature of **TDiff** can cope with difficult solutions where objects have initial temperatures different from their immediate surroundings. Depending on the geometry, **TDiff** 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. To speed 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 thermal sources where all nodes start at the same temperature. Because the second derivative of temperature is zero everywhere, there is no way for **TDiff** to estimate a time step. In this case, the program assigns a small default increment. You can set a better value using *DtMin*. In picking a time step, note that a good run should extend over about 1000 to 3000 steps.

**DTMAX DtMax**

**DTMAX = 1.0E-3**

Occasionally, the automatic routine in **TDiff** may pick a time step that is too long for good accuracy. This may happen 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 1000 to 3000 steps.

Ironically, one disadvantage of the stable numerical procedure used in **TDiff** is that it always gives an answer, even if inaccurate. You can use the *DtMax* command to check the accuracy of solutions. First, run the problem using automatic time step selection. Note the final time step and use the analysis functions of **TDiff** 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 temperature.
3.7 Material commands in dynamic solutions

The following commands set the physical properties of materials in dynamic solutions. We shall first cover simple commands and then consider options for time and temperature variations.

**FIXED RegNo TFix**
**FIXED(4) = 37.0**
Sets the region to a fixed temperature that does not change during the solution. Enter the temperature in °C.

Other commands refer to thermally-conductive materials. Several commands may be necessary for a full region definition.

**DENS RegNo Dens**
**DENS(6) = 5.67**
Sets the mass density of the region in kg/m\(^3\).

**INIT RegNo TInit**
**INIT(3) = 1000.0**
Set an initial temperature (in °C) for a conductive region. The default starting temperature at all variable points is 0.0 °C.

**COND RegNo k**
**COND(2) = 3.45**
Set a fixed value of the region thermal conductivity (in W/m·°C).

**CP RegNo Co**
**CP(2) = 5000.0**
Set the region specific heat (in J/kg·°C).

**SOURCE RegNo Q**
**SOURCE(4) = -4.5E6**
Set a constant heat source or sink uniformly distributed over the region volume. Enter the value in W/m\(^3\).

The following commands apply time or temperature dependance to materials. All commands require a prepared tabular function file available in the working directory.

**FIXED RegNo TABLE TabName**
**FIXED(4) = (TABLE, PTEMP.TMP)**
Sets the temperature of a fixed region to a prescribed function of time rather than to a constant value. The numerical values are contained in the named file in the form
Time (sec) Temperature (degC)

If the simulation time $t$ exceeds the maximum time $t_{\text{max}}$ of table entries, the interpolation time is taken as $t = \text{mod}(t, t_{\text{max}})$. This feature is useful for defining periodic or pulse-train modulation functions.

The following commands apply to thermally conductive regions.

**COND RegNo TABLE TabName**

COND(2) = (TABLE, ALUM.CND)
Determines the thermal conductivity of the region as a function of temperature from the named tabular function. The file data lines have the form

Temperature (degC) Conductivity (J/m-s-degC)

Note the values of $k$ for the elements of a region are determined from the same table. This does not mean that all elements have the same conductivity, because the temperature may vary over the region.

**CP RegNo Table TabName**

CP(2) = (TABLE, M2.SPH)
Determines the specific heat of the region as a function of temperature from the named tabular function. The file data lines have the form

Temperature (degC) Specific heat (J/kg-s-degC)

**SOURCE RegNo TABLE TabName**

SOURCE(4) = (TABLE, EVAP.SRC)
Adds a time-dependent volumetric heat source or sink uniformly distributed over a region. The source temporal variation is defined by the named tabular function file. The data lines have the format

Time (sec) Source (W/m3)

If the simulation time $t$ exceeds the maximum time $t_{\text{max}}$ of table entries, the interpolation time is taken as $t = \text{mod}(t, t_{\text{max}})$. This feature is useful for defining periodic or pulse-train modulation functions.
3.8 Diagnostic commands for dynamic solutions

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

- From 1 to 999 data dump files with the names of the form FPREFIX.001, FPREFIX.002, .... Each file contains a complete record of spatial information (the mesh geometry and temperature distribution) at a specified time. These files may be used for plots and calculation in the analysis mode of TDiff.

- A listing file with the name FPREFIX.TLS that contains a variety of information about the run. This information is useful mainly to check the validity of input parameters and to diagnose problems.

- Optional history files with names FPREFIX.P01, FPREFIX.P02,.... Here, the temperature at one or more probe locations is recorded at each time step of the solution. The information can be inspected with a text editor or the utility program Probe (Chap. 6).

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

DTIME DTime
DTIME = 0.2
For many runs a convenient way to make spatial data dumps is at uniform time intervals. This command sets the approximate interval (in seconds) between dumps. Because TDiff 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, TDiff will write dump 010 when \( t \geq 10 \times DTime \). The default value is \( DTime = \infty \).

SETTIME STime
SETTIME = 3.45
Sometimes you may want to inspect data at certain critical times or at the end of a run. For this situation, you can set up to 100 time markers using multiple SetTime commands. TDiff will write data as soon as possible after passing each marked time. Note that multiple SetTime commands must be in order of increasing time.

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

HISTORY XPos YPos
HISTORY ZPos RPos
HISTORY = (5.00, 9.85)
TDiff creates probe files with names when it detects a History command. The parameters in the command give the spatial location of a probe where the time and temperature should be recorded. The location is given as \((x, y)\) or \((z, r)\). Enter the coordinates in the units set by DUnit. You can define up to 20 probes with multiple History commands.
NHSTEP Ns
NHSTEP = 10
In runs with a large number of time steps, the history files may be larger than necessary and difficult to export to other programs. Use this command to set the number of time steps per data record in the history files. The default value is \( N_s = 1 \).

MOVIE NFrame [Nx Ny] [QMin QMax]
MOVIE 100 800 600 0.0 110.0
Create a set of \( N_{frame} \) graphics files in BMP format separated by time interval \( t_{max}/N_{frame} \). The plots show isothermal lines in the full solution volume. The number of contour lines is determined by any previous setting in the Analysis menu. The integer parameters \( N_x \) and \( N_y \) give the horizontal and vertical image size in pixels. If they do not appear, the default is \( N_x = 1024 \) and \( N_y = 768 \). The two real-number parameters are the upper and lower temperature limits for isotherms. Autoscaling is applied to each frame if they are absent. You can use our utility program Cecil_B to stitch the files into a movie in AVI format. The program is available at

http://www.fieldp.com/cecil_b.html

3.9 Running TDiff

TDiff 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]\TDIFF [DataPath]\FPrefix.TIN <ENTER>

where the file FPREFIX.TIN and the appropriate Mesh output file are available in the data directory. With this capability, you can set up extended autonomous TDiff runs using a DOS batch file or a Python script.

The remainder of this section discusses commands in the main menu when TDiff 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 (TIN)
EDIT LISTING (TLS)
EDIT FILE
These commands call the internal editor to inspect or to modify TDiff input and output files. Choosing a file from an alternate directory does not change the working directory. The Edit script (TIN) command shows a list of all files with names of the form FPREFIX.TIN while Edit listing (TLS) displays files with names FPREFIX.TLS. Note that you must exit the editor to resume normal operation of the program.

RUN ANALYSIS SCRIPT
An analysis script allows you to perform complex or repetitive analyses on a set of similar solutions. This command displays a dialog listing files with the suffix SCR. Pick a file and
click OK. The script can load data files, open and close data records, and perform any of the quantitative analysis functions described in this chapter. The script command language is described in Sect. 4.8. Note that the analysis script must be in the same directory as the data files.

**SETUP**
The function of this command is to create a TDiff script to control a thermal 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. 4. 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. 2.2). You must edit the script directly to invoke advanced functions like temperature-dependent materials.

The Solve popup menu includes the following three commands.

**RUN**
Pick an input file (such as FPrefix.TIN) 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 lightens and progress is shown in the status bar.

**STOP**
This command terminates TDiff 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, TDiff plots equithermal lines. The lines show the convergence of steady-state solutions and the evolution of dynamic solutions. (Note that the plot operations increase the run time.)

**ANALYZE**
Pick a file of the type FPrefix.001 and call up the analysis menu for plotting and numerical analysis.

**TDIFF MANUAL**
Display this manual in your default PDF display software. The file tdiff.pdf must be available in the same directory as tdiff.exe.

**PHYSICAL CONSTANTS**
Display a list of physical constants using the default PDF viewer. The file PHYSCONS.PDF must be in the same directory as tdiff.exe.
THERMAL PROPERTIES
Display a table of thermal conductivity and specific heat for a variety of materials. list of physical constants using the default PDF viewer. The file THERMALPROPERTIES.PDF must be in the same directory as tdiff.exe.
3.10 Format of the TDiff output files

Data dump files created by TDiff are in text format. The file has three sections:

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

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

--- Run parameters ---
XMin: -6.00000000E-02
XMax: 6.00000000E-02
KMax: 141
YMin: 0.00000000E+00
YMax: 6.00000000E-02
LMax: 73
DUnit: 1.00000000E+02
NReg: 5
ICylin: 1
Time: 1.00500000E+02

Lines 2 and 3 list $x_{\text{min}}$ and $x_{\text{max}}$, the limits along horizontal axis ($x$ or $z$) of the solution volume. Dimensions are given in meters. The quantity $k_{\text{max}}$ in Line 4 is the number of nodes along the horizontal direction. Lines 5-7 describe the vertical axis ($y$ or $r$). Line 8 contains the quantity $DUnit$, the conversion factor from dimensions used in Mesh to meters. Line 9 gives the number of regions in the solution, while Line 10 specifies the symmetry (0: planar, 1: cylindrical). Line 11 lists the simulation time at which the data were recorded.

The node section consists of 4 title lines and $k_{\text{max}} \times l_{\text{max}}$ data lines, one for each node of the solution space. Each data line contains the following quantities:

- The indices of the node ($K,L$)
- The region number of the node ($RgNo$) and region numbers for two associated elements ($RgUp$ and $RgDn$). The upper element lies above the line between nodes ($K,L$) and ($K+1,L$) and the lower element lies below the line.
- The coordinates of the node in meters, ($x,y$) or ($z,r$).
- The temperature at the node in °C.
- The thermal conductivity and specific heat in the upper and lower elements. This information is useful if the solution includes temperature-dependent material characteristics.
- The sum of volumetric thermal sources (region, file and perfusion) in the upper and lower elements.

The node section has the following appearance:
--- Nodes ---

<table>
<thead>
<tr>
<th>k</th>
<th>l</th>
<th>RgNo</th>
<th>RgUp</th>
<th>RgDn</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>-6.00000000E-02</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>-5.90709655E-02</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>-5.8100568E-02</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>-5.70773543E-02</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>-5.6054516E-02</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>-5.5034571E-02</td>
<td>0.00000000E+00</td>
</tr>
</tbody>
</table>

The region section consists of four title lines following by $NReg$ data lines, one for each region. Entries of "1" designate fixed-temperature regions in column 2, a region with a radiation boundary in column 3 and a region with perfusion in column 4. The first four recorded region quantities are thermal conductivity $k$, material density $\rho$, specific heat $C_p$ and the thermal source density $q$. Note that the values are meaningful only for uniform regions. The next three quantities apply to regions with radiation boundaries: emissivity $\epsilon$, temperature of surrounding objects $T_s$ and the radiative area $R_a$. The last three quantities apply to regions with perfusion: the fluid mass flow rate $W_b$, the fluid specific heat and the fluid entrance temperature $T_b$. 

--- Region properties ---
====================================================================
<table>
<thead>
<tr>
<th>RegNo</th>
<th>Fix</th>
<th>Rad</th>
<th>Perf</th>
<th>Conductivity</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5.00000000E-02</td>
<td>1.1000000E+03</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00000000E+00</td>
<td>1.1000000E+03</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.00000000E-02</td>
<td>8.00000000E+02</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.00000000E+00</td>
<td>0.00000000E+00</td>
</tr>
</tbody>
</table>

SpecHeat Source Emissivity TSurround
5.00000000E+03 0.00000000E+00 0.00000000E+00 0.00000000E+00
5.00000000E+03 0.00000000E+00 0.00000000E+00 0.00000000E+00
2.50000000E+03 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

RadArea Wb Cb Tb
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

Finally, the TDiff file lists region name inherited from Mesh:

--- Region names ---
TISSUE
INFUSEDTISSUE
PROBESHEATH
PROBE
BOUNDARY
4 TriComp analysis functions

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

4.1 File menu commands

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

LOAD CONFIGURATION FILE
A configuration file is a text file of information that controls operation in the Analysis menu. The file defines quantities for plots, interpolations, volume integrals and surface integrals. The program loads a default configuration the first time you run it. This file contains a broad spectrum of useful quantities for the solution type. Some programs may automatically switch between two default configurations, depending on the solution type (e.g., static or dynamic solutions in TDiff). 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 9 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. 8 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.

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.
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. 4.8. Note that the analysis script must be in the same directory as the data files.

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

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

EXIT
Exit the program.
4.2 Plot menu commands

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

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

The Settings popup menu contains the following commands.

**PLOT TYPE**
Change the plot type.

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

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

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

**GRID CONTROL**
This command displays the dialog of Fig. 9 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 $y$. 

42
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
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. 10). 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 viewpoint.

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

### 4.3 User specified contours

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

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

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

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

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

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

Next, choose *Plot/Plot settings/Plot limits* (or click the *Plot limits* tool) to open the dialog of Fig. 11. Note the checkbox marked *From clipboard* at the bottom-right. When you check this box, **TDiff** performs the following actions:
• Reads and analyzes the contents of the clipboard, checking that the entries are valid numbers in ascending order.

• Sets the desired contour values.

• Exits the dialog and updates the plot.

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

Note that the user-specified values affect only Contour line and Filled contour type plots (not plots of type Element and Surface). Figure 12 illustrates how the values are interpreted.

In a Contour line plot (upper), each line is displayed with a corresponding listed value on the right if the number of lines is 15 or less. When the number is greater, the value list contains a partial set of selected values. In the Filled contour plot (lower), the list gives the minimum value for each colored interval. In other words, an interval covers the range from its listed value to the listed value of the next interval. For example, the dark violet band in the figure extends from $1.0 \times 10^{-7}$ to $2.0 \times 10^{-7}$. Values below the first are plotted in the next lowest color (in this case, light violet) and values above the top value are plotted in white.

4.4 Saving and loading views

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

SAVE NAMED VIEW
Save the view parameters for the current plot. Quantities such as the zoom limits, plotted quantity and plot type (spatial or scan) are saved. The information is stored in a text file in the current directory with a name of the form FPREFIX.FPV.

LOAD NAMED VIEW
Load a view file and refresh either a spatial or scan plot.

The file contains the complete set of plot parameters. This excerpt illustrates the format:
Figure 12: Display of user-specified contour intervals. Top: Contour line plot. Bottom: Filled contour plot.
Program: TriComp
PlotStyle: Spatial
Outline: ON
Grid: ON
Scientific: OFF
FixedPoint: OFF
Vectors: OFF
XGMin: -1.000000E+00
XGMax:  1.000000E+01
YGMin: 0.000000E+00
YGMax: 1.000000E+01
PlotType: Elem
NPQuant:  1
...

If a specific solution file is loaded, the plot will be restored exactly. The saved view feature has two useful features if a different mesh is loaded:

- Dynamic adaptation to different solutions.
- Option for user control of the view parameter set.

Regarding the first feature, there are situations where you want to create consistent views of a set of solutions with different geometries, maintaining a similar appearance. Some plot properties (like the plot type or quantity) are applicable to any solution, but others (like zoom limits) depend on the geometry. The program checks each plot parameter for validity. If a parameter is outside the allowed range for the currently-loaded solution, the program computes an alternative. The goal is to preserve as many features of the view as possible.

You can modify view files with an editor. The order of entries is not rigid. On input, the programs use a free-form parser. If a parameter is missing, the program simply makes no change from the value current in the program. The implication is that you can modify a saved view to include only elements essential to your application.

4.5 Analysis menu commands

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

POINT CALCULATION

Click on the command and move the mouse cursor to a position inside the solution volume. Current coordinates are displayed at the bottom of the screen. Press F2 if snap mode is active and you want to specify an intermediate point. Press F1 to type in the coordinates. The program writes a subset of quantities to the information area below the plot. If a data file is open, the program also records complete information on all quantities defined in the Interpolation section of the configuration file.
LINE SCAN
After clicking on the command, supply two points with the mouse or keyboard to define a scan line. The snap mode is useful in this application (for example, you may want a scan to extend from 0.000 to 5.000 rather than 0.067 to 4.985.) The program computes a series of values of field quantities at equal intervals along the line. The program makes a screen plot of the currently-selected quantity versus distance along the scan and activates the commands in the Scan plot menu (Sect. 4.6). The information is recorded if a data file is open. The default is to include all computed quantities. Use the Set recorded quantities command to limit the information.

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

SURFACE INTEGRALS
Calculate surface integrals of quantities defined in the SURFACE section of the configuration file over region boundaries. In response to the command, the program displays the dialog of Fig. 13. Pick one or more regions as internal regions and bounding regions as external. The program computes the integral over the surfaces of the internal region(s) in contact with the external regions.

LINE INTEGRALS
Take integrals of vector quantities specified in the SURFACE section of the current configuration file over a line in the solution volume. Specify the line with the mouse or keyboard as in the Line scan command. The program reports both parallel and normal integrals in the data file (i.e., the vector quantity is taken both parallel and normal to the differential line segments).
MATRIX FILE
The program can write matrix files of field values to help you create your own analysis routines. Although information is available in the output file of the solution program, it may be difficult to deal with the conformal triangular mesh. The Matrix file command uses the interpolation capabilities of the program to create a text data file of field quantities on a rectangular grid in x-y or z-r. The command displays the dialog of Fig. 14. Here you can set the matrix file name, the dimensions of the box and the number of intervals along x and y (or z and r). The program creates the file in the current directory. The default is to include all computed quantities. Use the Set recorded quantities command to limit the information.

The Analysis settings popup menu contains the following entries.

INTERPOLATION METHOD
The default interpolation method for plots and analysis commands such as Point calculation and Line scan is a second-order least-squares fit with a selective choice of data points. For example, only points on the side of a material boundary that contains the target point are included to give the correct field discontinuity. The least-squares fit may fail in very small regions or enclosed areas if the program cannot identify enough data points. In this case, toggle to the linear mode. Here, field values are determined by a first order fit in the element that contains the target point. The status bar reports the current interpolation type.

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

NUMBER OF SCAN POINTS
This command sets the number of line scan points in plots and data file records. The default is 50 and the maximum number is 500.
Figure 15: Dialog to set quantities recorded in response to the Line scan and Matrix commands.

SET RECORDED QUANTITIES
The default for data records of line scans and matrix files is to include all quantities defined in the INTERPOLATION section of the current configuration file. This state may result in large files with unnecessary information. Use this command to set the quantities that will be included.

In the dialog of Fig. 15, activate the check box to include a quantity. There are a few rules:

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

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

OSCILLOSCOPE MODE
In the oscilloscope mode, a scan plot assumes characteristics of a digital oscilloscope (Fig. 16). The program superimposes a cross-hair pattern on the graph. Plot values at the intersection are displayed in the information window. Move the marker along the plot by moving the mouse. If you click the left mouse button at a point, the program displays the plot values along with the numerical derivative and integral of the curve. The definite integral is taken from the left-hand side of the plot to the current point. Values are displayed on the screen and written to the data file if open. Press the right mouse button to exit the oscilloscope mode.
TOGGLE SCAN SYMBOLS
The setting determines whether plot symbols are added to the scan plot showing calculated points.

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

CLOSE SCAN PLOT
Close the scan plot plot and return to the Analysis menu.

4.7 Vector tools
The analysis menu includes useful tools to display directional quantities in the VECTOR section of the current configuration file. Figure 17 shows the corresponding entries on the toolbar.

PROBE
This feature was inspired by the familiar Magnaprobe illustrated in Fig. 17. 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.

### 4.8 Analysis script commands

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

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

The program can perform analyses autonomously under script file control from the command prompt. If the file `GTEST.SCR` is in the data directory, then use a command of the form:
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 = \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/m³, then the output quantities will have units of C for cylindrical solutions and C/m for planar calculations.

**SURFACEINT Reg1 Reg2 ... RegN**

**SURFACEINT 5 -9 -12**

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

**MATRIX FName Nx Ny Xs Ys Xe Ye**

**MATRIX FPrefix Nz Nr Zs Rs Ze Re**

**MATRIX = Switch1 (10, 20) (0.00, 0.00, 5.00, 10.00)**

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

**NSCAN NScan**

**NSCAN = 150**

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

**INTEPOLATION [LSQ, LINEAR]**

**INTERPOLATION = Linear**

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

**PLOT FSaveView FOutput Nx Ny**

**PLOT (XAxis3D VIEW001 800 600)**
Write a plot file based on data in the currently loaded solution. The string *FSaveView* is the prefix of the file *FSaveView.FPV*, a set of view parameters created with the *Save view* command (Sect. 4.4). The view file must be available in the working directory. The plot is saved in the working directory. It is in Windows Bitmap format and has the name *FOutput.BMP*. The integers $N_x$ and $N_y$ give the width and height of the image in pixels. To optimize compatibility with graphics format converters, pick values that are multiples of 16 (e.g., 1024 × 768).

**ENDFILE**

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

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

```
NSCAN 200
OUTPUT COMP.DAT
INPUT SWITCH01.EOU
SCAN 0.00 -50.00 0.00 50.00
INPUT SWITCH02.EOU
SCAN 0.00 -50.00 0.00 50.00
INPUT SWITCH03.EOU
SCAN 0.00 -50.00 0.00 50.00
INPUT SWITCH04.EOU
SCAN 0.00 -50.00 0.00 50.00
ENDFILE
```
5 Standard TDiff calculated quantities

The TDiff package includes two standard configuration files:

- `tdiff_static.cfg` for static solutions.
- `tdiff_dynamic.cfg` for dynamic solutions.

Table 9 lists the standard configuration file for dynamic solutions. The program automatically switches to match the loaded solution if one of the standard configurations is active. For example, if `tdiff_static.cfg` is the current configuration and a dynamic solution is loaded, TDiff automatically loads the dynamic configuration. Table 10 shows quantities defined in the standard configuration for plots and scan, vector tools, volume integrals and surface integrals. Note that the internal energy density in dynamic solutions is simply the product

\[ u = \rho C_p(T)T. \]  

Therefore, it should be considered only as a differential or relative quantity for materials with temperature-dependent specific heat.

5.1 TDiff variables

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

- \$T$: temperature in °C or K.
- \&Fxz$: x (planar) or z (cylindrical) component of thermal flux in W/m².
- \&Fyr$: y (planar) or r (cylindrical) component of thermal flux in W/m².
- \&FMag$: magnitude of the thermal flux in W/m².
- \&K$: thermal conductivity in W/°C-m.
- \&Cp$: specific heat in J/°C-kg.
- \&Qs$: heat source in W/m³.

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

- \$RegCond$: constant value of thermal conductivity for uniform regions (W/°C-m)
- \$RegDens$: region mass density in dynamic solutions (kg/m³)
- \$RegCp$: constant value of specific heat for uniform regions (J/°C-kg)
Table 9: Contents of the file tdiff_dynamic.cfg.

PROGPARAM
END
RUNPARAM
* Time of data dump for a dynamic solution
  $Time = 0.0
END
REALTIMEPARAM
END
INTERPOLATION
  T = &T
  Fx/Fz = &Fxz
  Fy/Fr = &Fyr
  |F| = &FMag
  Cond = &K
  PowerDens = &Qs
  SpecHeat = &Cp
  MassDens = $RegDens
  IntEng = &Cp $RegDens * &T *
END
VECTOR
  FVector = &Fxz;&Fyr
END
VOLUME
  Power = &Qs
  IntEnergy = &Cp $RegDens * &T *
END
SURFACE
  Power = &Fxz;&Fyr
END
ENDFILE
Table 10: Quantities for thermal solutions (those marked with + apply only to dynamic solutions).

<table>
<thead>
<tr>
<th>Section</th>
<th>Quantity</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interpolation</td>
<td>$T$ (temperature)</td>
<td>°C or °K</td>
</tr>
<tr>
<td></td>
<td>$f_x$ or $f_z$ (thermal flux)</td>
<td>W/m²</td>
</tr>
<tr>
<td></td>
<td>$f_y$ or $f_r$ (thermal flux)</td>
<td>W/m²</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(thermal flux magnitude)</td>
</tr>
<tr>
<td></td>
<td>$k$ (thermal conductivity)</td>
<td>W/°C-m</td>
</tr>
<tr>
<td></td>
<td>$q_s$ (source power density)</td>
<td>W/m³</td>
</tr>
<tr>
<td></td>
<td>$C_p$ (specific heat +)</td>
<td>J/°C·kg</td>
</tr>
<tr>
<td></td>
<td>$\rho$ (mass density +)</td>
<td>kg/m³</td>
</tr>
<tr>
<td></td>
<td>$u$ (internal energy density +)</td>
<td>J/m³</td>
</tr>
<tr>
<td>Vector</td>
<td>$F$ (thermal flux)</td>
<td>W/m²</td>
</tr>
<tr>
<td>Volume</td>
<td>$Q_s$ (source power)</td>
<td>W or W/m</td>
</tr>
<tr>
<td></td>
<td>$U$ (internal energy +)</td>
<td>J or J/m</td>
</tr>
<tr>
<td>Surface</td>
<td>$F$ (power flow)</td>
<td>W or W/m</td>
</tr>
</tbody>
</table>

- $\text{RegEmiss}$: constant value emissivity for radiative regions.
- $\text{RegTSurr}$: the temperature of materials surrounding a radiative region (°C)
- $\text{RegRadArea}$: free external surface area of a radiative region (m²).
- $\text{RegWb}$: fluid mass flow rate in perfused regions (kg/m³-s)
- $\text{RegCb}$: specific heat of the fluid in perfused regions (J/°C·kg)
- $\text{RegTb}$: fluid temperature in perfused regions (°C or °T)

$\text{TDiff}$ records one run parameter in solution files: $\text{Time}$. The quantity is the simulation time of the data file for dynamic solutions. It equals 0.0 for static solutions.

### 5.2 Special commands

The following command appears in the Analysis menu:

**AVERAGE TEMPERATURE**

Determine the temperature extremes and average temperature over the solution volume and list them on the screen. A detailed analysis by region is created if a data file is open.

$\text{TDiff}$ supports the following special script command in addition to the standard options listed in Sect. 4.8:

**TEMPANAL**

Loop over all elements in the solution volume and make a global and regional listing of the maximum temperature, minimum temperature and average temperature if a data file is open.
6  Probe – history file plot utility

6.1  Introduction

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

Table 11 shows the standard probe file format. The first section is a header that contains the following information:

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

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

6.2  Loading data files

When you start Probe the only active menu option is \textit{Load probe}. Plotting and analysis functions become active when a probe file has been opened. The program displays a dialog showing all files with suffixes of the form \texttt{P01,...,P12}. Pick a file to analyze and click \textit{OK}. Changing directories in the dialog will change the working directory of the program. If the load is successful, Probe creates a default plot of the data (Fig. 18).

The status bar at the bottom of the window contains the name of the probe file, the current plot quantity, and the temporal range of data. The default plot shows the first quantity recorded in the probe file over the full range of time. The horizontal and vertical scales are chosen so that the plot fits on the screen and the grid lines are automatically adjusted so that they lie on even values of the plotted quantity with easily recognized intervals (\textit{e.g.}, 0.02, 0.05, 0.10, ...).
Table 11: Example of the **Probe** file format

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

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Re (m)</th>
<th>Rho (kg/m3)</th>
<th>Press (Pa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.250000E-09</td>
<td>4.357041E-02</td>
<td>1.000000E-01</td>
<td>-1.238464E+02</td>
</tr>
<tr>
<td>1.250000E-08</td>
<td>4.357041E-02</td>
<td>1.000000E-01</td>
<td>-1.238464E+02</td>
</tr>
<tr>
<td>1.874999E-08</td>
<td>4.357041E-02</td>
<td>1.000000E-01</td>
<td>-1.238464E+02</td>
</tr>
<tr>
<td>2.499996E-08</td>
<td>4.357041E-02</td>
<td>1.000000E-01</td>
<td>-1.238464E+02</td>
</tr>
<tr>
<td>3.124995E-08</td>
<td>4.357041E-02</td>
<td>1.000000E-01</td>
<td>-1.238464E+02</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Temp (deg-K)</th>
<th>Epsi (J/kg)</th>
<th>V(Av) (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000000E-03</td>
<td>8.620001E+00</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>1.000000E-03</td>
<td>8.620001E+00</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>1.000000E-03</td>
<td>8.620001E+00</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>1.000000E-03</td>
<td>8.620001E+00</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The grid intervals are shown in parentheses next to the labels of the horizontal and vertical axes. The title line at the top of the plot shows the following information: generating program, probe file name, element number, region number and position. This information is recorded in hardcopy plots to help you archive your data.

6.3 Plot settings

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

PLOTTED QUANTITY

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

TIME LIMITS

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

VERTICAL LIMITS

In the default mode Probe picks a scale to display the full range of the plotted quantity. You
can narrow or expand the range by supplying minimum and maximum values. The program returns to full range if you check the Autoscale box or if you change quantities using Pick plotted quantity.

TOGGLE GRID
Switch between grid and fiducial lines in the plot.

TOGGLE PLOT SYMBOLS
Include or remove symbols to mark the recorded points.

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

OSCILLOSCOPE MODE
When you issue this command, Probe simulates a digital oscilloscope. As shown in Fig. 19, the mouse cursor changes to a cross-hair pattern when it is inside the plot window. The program adds movable fiducial lines to mark the current point. You can drag the fiducials along the time axis by moving the mouse. A box at the bottom of the plot shows values of the time and plotted quantity at the current position. If you press the left mouse button, the program displays a box with the following information about the current point:

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

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

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

DEFAULT PRINTER
Probe can port copies of the plot to any installed Windows printer. The program sends output
Figure 19: **Probe** in the *Oscilloscope mode*.

to the default printer, so be sure to select the correct device using the *Settings/Printer* function of Windows before making the plot.

**SAVE PLOT FILE**
Use this command to create a graphics file of the current plot in either Windows Bitmap (BMP) or Portable Network Graphics (PNG) formats. In the dialog, specify the format, the size in pixels and the file prefix. The graphics file is created in the current directory.

**COPY TO CLIPBOARD**
Copy the plot to the clipboard in Windows MetaFile format.

### 6.5 Information

**PROBE FILE INFORMATION**
Display information on the probe file in a message box (Fig 20). The quantity *NSkip* in line 7 is used for long files. There is no reason to store more than 1000 points for plots on typical screens and hardcopy devices. When there are less than 1000 data lines, **Probe** loads all points (*NSkip = 1*). When the file contains 1000 to 2000 data lines, the program loads every second
point ($NSkip = 2$), and so forth. In this way the **Probe** can handle probe files of any length without exceeding memory limits.

**VIEW PROBE FILES**

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

**PROBE MANUAL**

Show this document in your default PDF viewer.
7 Thermal source import from RFE2, Nelson, WaveSim and GamBet

7.1 Function and commands

TDiff can include spatially-distributed thermal sources calculated by the codes RFE2, Nelson, WaveSim and GamBet. The programs determine power deposition by electromagnetic fields or high-energy particles in the following regimes:

- **RFE2** calculates resistive power deposition by low- to mid-frequency electric fields created by electrodes. The field distribution includes contributions from both resistive and displacement current. The code has application to biothermal processes and RF heating.

- **Nelson** calculates eddy-current losses by low- to mid-frequency magnetic fields created by drive coils. The code has application to inductive heating treatments and high-frequency magnetic circuits.

- **WaveSim** solves the Maxwell equations to find high-frequency solutions for electromagnetic fields. Power dissipation is determined from the imaginary parts of the dielectric constant and magnetic permeability and may include contributions from both resistive current flow and material imperfections. The code has application to microwave heating.

- GamBet is a Monte Carlo code that tracks the distribution of power and dose deposited by high-energy X-rays, electrons and positrons. The run must be performed in the continuous-beam mode, where a steady-state current or flux is assigned to primary particles.

The following commands are used to load and to modify thermal-source files.

**SOURCEFILE FileName [SFact]**

`SOURCEFILE = LiverProbe.ROU 2.5`

Load an output file from RFE2, Nelson, WaveSim or GamBet to define thermal sources. The command is valid in both the static and dynamic modes of TDiff. The same mesh must be used for both the field calculation and the thermal calculation. In the case of GamBet, regions must have the same material density values for both simulations. The input file contains values for the thermal source (in W/m³) in each element of the mesh. Values are multiplied by the optional parameter `SFact` when loaded. In a dynamic solution, the source values may also be multiplied by a modulation function defined with the `SourceMod` command (Sect. 3.5). Only a single `SourceFile` command may appear in a script. File sources may be combined with other types of thermal sources (region, perfusion or radiation).
SOURCEMOD TabFileName
SOURCEMODE = TenPulse.MOD

Use this command to assign a time-variation to a spatially-distributed thermal power source loaded with the SourceFile command. This command is valid only in dynamic calculations. The string TabFileName is the full name of a file that defines a modulation waveform. The file has the form of a tabular function (Chap. 3.4). It may contain up to 256 data lines – each line contains two numbers: \( t \) and \( f(t) \). Enter the time \( t \) in seconds. (Note that all dynamic simulations start at \( t = 0.0 \).) The modulation function \( f(t) \) is dimensionless and usually has values on the order of unity. The stored file source values for elements in the solution volume are multiplied by \( f(t) \) to find sources at the current time. Be sure to include the command Interp Linear if the modulation function has discontinuities of value or slope. The time variation applies to file sources, and does not affect region, perfusion or radiation sources. If the simulation time \( t \) exceeds the maximum time \( t_{\text{max}} \) of table entries, the interpolation time is taken as \( t = \text{mod}(t, t_{\text{max}}) \). This feature is useful for defining periodic or pulse-train modulation functions.

SOURCEMOD > Function
SOURCEMODE > 10.0 + \( \cos(3.1416*\$t/25.6) \)

Define a modulation function for power density from a mathematical expression. The keyword SourceMod followed by the > symbol designates that a function string occupies the remainder of the line. The function may be up to 230 characters in length and follows the format described below. The function defines a variation in time, \( f(t) \). The parser uses the Perl standard for the time variable: \( \$t \) stands for \( t \).

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

- The time variable \( \$t \).
- Real and/or integer numbers in any valid format (e.g., 3.1415, 476, 1.367E23, 6.25E-02, 8.92E+04,...). Integers are converted to real numbers for evaluation.
- Binary operations: + (addition), - (subtraction), * (multiplication), / (division) and \( \wedge \) (exponentiation).
- Functions: abs (absolute value), \( \sin \) (sine), \( \cos \) (cosine), \( \tan \) (tangent), \( \ln \) (normal logarithm), \( \log \) (base 10 logarithm), \( \exp \) (normal exponent) and \( \sqrt{t} \) (square root).
- Up to 20 sets of parentheses to any depth.
- Any number of space delimiters.

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

\[
1 - \exp(-1.0*((\$t^2)/24))
\]
corresponds to

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

(7)

7.2 Application example

The example FSourceDemo illustrates several techniques for building coupled solutions with TDiff using calculated data from the RFE2 code. Figure 21 shows the geometry. Rod electrodes in air contact a layer of tissue resting on an insulating substrate. The applied bipolar voltage has amplitude ±1.0 V and frequency \( f = 10.0 \) MHz. The same mesh is used for both the RFE2 and TDiff solutions. Table 12 shows region definitions and physical properties assigned for the two solutions. Regions 1, 2 and 3 represent the insulating base, the tissue layer and air respectively. In the RF solution, the insulator and air are ideal dielectrics while the tissue has a non-zero conductivity. The bipolar electrodes (Regions 4 and 5) are assigned a fixed amplitude of oscillating voltage. Region 6 is a line of nodes with fixed voltage 0.0 V. It approximates the effect of an infinite surrounding medium at ground potential. In the dynamic thermal solution Regions 1, 2 and 3 are thermal conductors with assigned values of \( k \), \( C_p \) and \( \rho \) and initial temperature \( T_i = 25.0^\circ \). The water-cooled stainless-steel probes have the fixed temperature \( T_f = 10.0^\circ \). The boundary (Region 6) is modeled as a fixed temperature surface at \( T_f = 25.0^\circ \) to approximate the effect of an infinite surrounding medium at ambient temperature.

Table 13 shows the complete TDiff script. Most of the script was constructed using the dialog of Fig. 4. The advanced commands SourceFile and SourceMod were added with a text editor. The adjustment factor of 1200 in the SourceFile command gives a power deposition of 963.9 W/m. The file FSourceDemo.MOD defines a double-pulse function that equals 1.0 from \( t = 0.0 \) s to 70.0 s, 0.0 from 70.0 s to 130.0 s, and 1.0 for the remainder of the 201 s run. Figure 22 shows the distribution of electric field magnitude near the probes in the RFE2 solution, while
Figure 22: Distribution of $|\mathbf{E}|$ in the normalized RFE2 solution.

Fig. 23 shows the temperature distribution at times $t = 40$ and $t = 200$ s.

Table 12: Region properties – source file application example

<table>
<thead>
<tr>
<th>Reg</th>
<th>Property</th>
<th>RF</th>
<th>Thermal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Insulator</td>
<td>$\epsilon_r = 5.3$, $\sigma = 0.0$</td>
<td>$k = 0.24$, $C_p = 1000$, $\rho = 1370$, $T_i = 25.0^\circ$</td>
</tr>
<tr>
<td>2</td>
<td>Tissue</td>
<td>$\epsilon_r = 600.0$, $\sigma = 0.5$</td>
<td>$k = 0.45$, $C_p = 5000$, $\rho = 1000$, $T_i = 25.0^\circ$</td>
</tr>
<tr>
<td>3</td>
<td>Air</td>
<td>$\epsilon_r = 1.0$, $\sigma = 0.0$</td>
<td>$k = 0.0024$, $C_p = 1000$, $\rho = 1.25$, $T_i = 25.0^\circ$</td>
</tr>
<tr>
<td>4</td>
<td>Left electrode</td>
<td>$</td>
<td>V</td>
</tr>
<tr>
<td>5</td>
<td>Right electrode</td>
<td>$</td>
<td>V</td>
</tr>
<tr>
<td>6</td>
<td>Boundary</td>
<td>$</td>
<td>V</td>
</tr>
</tbody>
</table>
Table 13: TDiff script FSourceDemo.TIN

Mesh = SFileDemo
Mode = TVar
Geometry = Rect
DUnit = 1.0000E+02
TMax = 2.0100E+02
DtMin = 1.0000E-04
DtMax = 5.0000E-02
SourceFile = FSourceDemo.ROU 1200.0
SourceMod = FSourceDemo.MOD
Interp = Linear
Cond(1) = 0.24
Cp(1) = 1000.0
Dens(1) = 1370.0
Init(1) = 2.5000E+01
Cond(2) = 0.45
Cp(2) = 5000.0
Dens(2) = 1000.0
Init(2) = 2.5000E+01
Cond(3) = 2.4000E-03
Cp(3) = 1.0000E+03
Dens(3) = 1.2500E+00
Init(3) = 2.5000E+01
Fixed(4) = 1.0000E+01
Fixed(5) = 1.0000E+01
Fixed(6) = 2.5000E+01
DTime = 4.0000E+01
History = 0.0, 1.0
EndFile
Figure 23: Temperature distributions at $t = 40.0$ s and $t = 200.0$ s. The maximum temperature (red line) is $80.0^\circ$. 
8 Radiation boundaries

TDiff can include thermal radiation losses in models of high-temperature systems like thermionic cathodes. To begin, we shall review the procedure used in the code in order to understand the capabilities and limitations of the model. Radiating surfaces of arbitrary shape can be defined by creating regions in Mesh and then assigning radiation properties in the TDiff script using the Radiation command. As shown in Fig. 24, thermal radiation leaves elements with non-zero emissivity ($\epsilon > 0.0$) through facets that border on void elements. Void elements (with $RegNo = 0$) include any elements within the solution volume that have not been assigned to a material region and a virtual layer of elements around the boundaries of the solution volume. The power loss through a facet is given by

$$dP = \int \int_S dS \epsilon \sigma \left[ (T + 273.15)^4 - (T_s + 273.15)^4 \right],$$

where $\sigma = 5.6705 \times 10^{-8}$ W/m$^2$K$^4$ is the Stefan-Boltzmann constant. The surface integral in Eq. 8 is taken over the facet. The temperature $T$ is the average temperature (in °C) inside the emissive element while $T_s$ is the temperature of the surroundings (objects outside the solution volume). The quantity $dP$ is the facet power flux, $\epsilon$ is the emissivity of the radiating element and $\sigma$ is the Stefan-Boltzmann constant.

The fundamental assumption of the TDiff model is that thermal radiation that leaves emissive elements goes to the surroundings and that incoming thermal radiation is produced by the surroundings. In other words, there is no exchange of thermal radiation between emissive elements in different parts of the solution. A calculation with several bodies at different temperatures that exchange radiation is extremely complex and is beyond the capability of the TDiff model. The following conditions must be satisfied for a valid thermal-radiation simulation:
Figure 25: System that violates the assumptions of the TDiff radiation model.

- The physical system consists of contiguous regions with no internal voids. In other words, regions of the system connect directly to one another, and internal heat exchange is solely through conduction.

- Elements on the periphery of the system radiate to surroundings that have an average absolute temperature \((T_s + 273.15)\) K. In practical problems the absolute temperature of the surroundings is much less than that of the radiating elements, so that \((T_s + 273.15)^4 \ll (T + 273.15)^4\). Therefore, it is not necessary to know the precise distribution of surrounding temperature.

- All radiating facets have a clear line-of-sight to infinity. In other words, the projection of a normal vector from any point on an emissive boundary intersects the solution volume boundary. Figure 25 illustrates a system where the conditions are violated.

Simulation of radiative losses in a high-temperature structure involves two tasks:

1. Using Mesh, set up emissive regions with boundaries that contact either the boundaries of the solution volume or unassigned elements \((\text{Reg}No = 0)\).

2. In the TDiff script, assign radiation properties to the emissive regions. The program automatically determines the emissive facets and sets up source terms to include energy loss to the boundary elements.

In dynamic solutions, TDiff determines radiation loss using current values of element temperature. In static calculations, TDiff simultaneously adjusts surface sources while it performs the relaxation solution of the boundary-value problem. Therefore radiation problems may converge more slowly. You can reduce the run time significantly by starting nodes of the system close to the expected final temperature using the Init command. Thermal radiation is implemented by the following script command in either mode:
Figure 26: Geometry and temperature distribution, RADBENCHMARK example.

Figure 27: Temperature distribution with thermal radiation from two hemispheres.

RADIATION RegNo Emiss Ts
RADIATION(5) = (0.5, 25.0)

Specify an emissive region. Thermal radiation is lost through the region boundary in contact with void elements. The quantity $Emiss$ is the emissivity of the material and $Ts$ is the average temperature (in °C) of objects at infinity.

The example RADBENCHMARK illustrates the accuracy of the method. In the boundary-value problem, we want to set conditions to create a surface temperature of $T = 1000.0$ °C on a uniform sphere of material 2.5 cm in radius. We assume that the material has emissivity $\epsilon = 0.5$ and is surrounded by material at $Ts = 0.0$ °C. Figure 26 shows the mesh geometry for the system with cylindrical symmetry. Note that elements inside the solution volume that are not assigned to the sphere automatically assume the void condition. The calculated surface area of the sphere is $A = 7.854 \times 10^{-3}$ m$^2$ and the predicted thermal flux density at 1000.0 °C is $7.4334 \times 10^4$ W/m$^2$. The total expected radiative loss is 583.68 W. For the spherical volume of $6.5448 \times 10^{-5}$ m$^3$, we can generate the required power with a volumetric source $q = 8.918 \times 10^6$ W/m$^3$.

The code-generated surface temperature in Figure 26 is close to the target value. In a radiative calculation TDiff records the calculated radiative area and total radiative heat loss of regions in the listing files. The values generated by the code are $A = 7.852 \times 10^{-3}$ m$^2$ and
Finally, Fig. 27 shows a more interesting example where two hemispheres of different materials are attached. The material on the left has $k = 5.0 \text{ W/m}^{-\circ\text{C}}$ and $\epsilon = 0.5$ while the material on the right has $k = 10.0 \text{ W/m}^{-\circ\text{C}}$ and $\epsilon = 1.0$. The TLS file contains the following listing:

<table>
<thead>
<tr>
<th>NReg</th>
<th>Radiative flux (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-2.21855E+02</td>
</tr>
<tr>
<td>2</td>
<td>-3.61737E+02</td>
</tr>
</tbody>
</table>

In an initial-value problem with radiation, $\text{TDiff}$ includes such a listing each time a data dump is created.
9 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.

9.1 Configuration file structure

An analysis configuration is a text file with a name of the form FPREFIX.CFG. Table 9.1 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 9.3 covers parameters and variables that may appear in expressions.
Table 14: Configuration file estat_dielectric.cfg.

PROGPARAM
    $\varepsilon_0 = 8.854187 \times 10^{-12}$
END

RUNPARAM
    $\text{CondFlag} = 0.0$
END

REALTIMEPARAM
END

INTERPOLATION
    $\Phi = \&\Phi$
    $E_x/E_z = \&E_{xz}$
    $E_y/E_r = \&E_{yr}$
    $|E| = \&E_{Mag}$
    $D_x/D_z = \varepsilon_0 E_{xz} \times$
    $D_y/D_r = \varepsilon_0 E_{yr} \times$
    $|D| = \varepsilon_0 E_{Mag} \times$
    $\text{EngDens} = E_{xz} E_{xz0} \times E_{yr} E_{yr0} \times + 0.5 \times \varepsilon_0 \times$
END

VECTOR
    $\text{Evect} = \&E_{xz};\&E_{yr}$
    $\text{Dvect} = E_{xz0} \varepsilon_0 \times;E_{yr0} \varepsilon_0 \times$
END

VOLUME
    $\text{Energy} = E_{xz} E_{xz0} \times E_{yr} E_{yr0} \times + 0.5 \times \varepsilon_0 \times$
    $\text{Charge} = \&\rho$
END

SURFACE
    $\text{Charge} = E_{xz0} \varepsilon_0 \times;E_{yr0} \varepsilon_0 \times$
END
ENDFILE
9.2 Introduction to RPN

Consider the expression

\[5.0 \times 4.5^3 + \left[9.2 + 0.6 \times 0.9^{(2 + 0.67)}\right].\]  \hfill (9)

Equation 9 could be represented in algebraic notation by the string:

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

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. 28. 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 15: Unary operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Operation</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>@SIN</td>
<td>sin(X)</td>
<td>Angle in radians</td>
</tr>
<tr>
<td>@COS</td>
<td>cos(X)</td>
<td>Angle in radians</td>
</tr>
<tr>
<td>@TAN</td>
<td>tan(X)</td>
<td>Angle in radians</td>
</tr>
<tr>
<td>@ASIN</td>
<td>sin⁻¹(X)</td>
<td>Returns angle in radians</td>
</tr>
<tr>
<td>@ACOS</td>
<td>cos⁻¹(X)</td>
<td>Returns angle in radians</td>
</tr>
<tr>
<td>@ATAN</td>
<td>tan⁻¹(X)</td>
<td>Returns angle in radians</td>
</tr>
<tr>
<td>@EXP</td>
<td>e^X</td>
<td></td>
</tr>
<tr>
<td>@LN</td>
<td>ln(X)</td>
<td></td>
</tr>
<tr>
<td>@LOG</td>
<td>log₁₀(X)</td>
<td></td>
</tr>
<tr>
<td>@SQRT</td>
<td>√X</td>
<td></td>
</tr>
<tr>
<td>@ABS</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>@EXCH</td>
<td>X ċ Y</td>
<td></td>
</tr>
<tr>
<td>@OVERX</td>
<td>1/X</td>
<td></td>
</tr>
<tr>
<td>@XSQ</td>
<td>X^2</td>
<td></td>
</tr>
<tr>
<td>@ENTER</td>
<td>X ⇒ X,Y</td>
<td>Copy X and push on stack</td>
</tr>
<tr>
<td>@CHS</td>
<td>X = −X</td>
<td></td>
</tr>
</tbody>
</table>

• Subtraction (-): Y − X
• Multiplication (*): Y × 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.

### 9.3 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 15 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 -$\varepsilon_{0}$ and -&grady[3] that mix a minus sign with a variable are invalid. Instead, use forms like $\varepsilon_{0}$ @CHS.

You may define up to 10 constants in the PROGPARAM section that can be used in your expressions. For example:
A data line contains a parameter name, an equal sign and a value in any valid real-number form. 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 \texttt{RUNPARAM} section:

\begin{verbatim}
RUNPARAM
  $CondFlag = 0.0
END
\end{verbatim}

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 \texttt{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

\begin{verbatim}
$X, $Y, $Z, $NREG, $DUNIT
\end{verbatim}

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:

\begin{verbatim}
RDIST = $X^2 + $Y^2 + $Z^2 + SQRT
\end{verbatim}

The standard variable \(\$DUNIT\) is a dimension conversion factor passed from the solution program. For example, suppose dimensions in \texttt{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:

\begin{verbatim}
RDIST = $X^2 + $Y^2 + $Z^2 + SQRT $DUNIT *
\end{verbatim}

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

### 9.4 Building expressions

You can add your own expressions to the \textsc{Interpolation}, \textsc{Vector}, \textsc{Volume} and \textsc{Surface} sections of the configuration file, one expression per line. Lines for a scalar quantities in the \textsc{Interpolation} and \textsc{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:

\texttt{RectName/CylName}

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

\texttt{FVolX/FVolZ = $\textbf{RegCurrent}$ &Byr * $\textbf{RegArea}$ / @CHS}

Expression lines in the \textsc{Vector} and \textsc{Surface} sections have the following form:

\texttt{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 \textit{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 \textit{Load configuration} dialog so that you can pick an alternative. An error message is displayed if the alternate file is invalid.
Index

accuracy, numerical, 30

Analysis commands
  3D angle, 45
  3D grid, 45
  Close data record, 41
  Contour style, 43
  Create analysis script, 41
  Field lines at points, 52
  Grid control, 42
  Interpolation method, 50
  Line integrals, 49
  Line scan, 49
  Load configuration file, 40
  Load from series, 40
  Load solution file, 40
  Matrix file, 50
  Number display format, 44
  Number of contours, 44
  Number of scan points, 50
  Open data record, 40
  Oscilloscope mode, 51
  Pick vector quantity, 53
  Plot limits, 42, 45
  Plot quantity, 42
  Plot type, 42
  point calculation, 48
  Probe, 52
  Remove vectors, 53
  Rotate 3D, 44
  Run analysis script, 41
  Save named view, 46
  Scan plot quantity, 50
  Set recorded quantities, 51
  Snap distance, 43
  Surface integrals, 49
  Toggle fixed point, 43
  Toggle grid display, 42
  Toggle keyboard, 43
  Toggle outline, 43
  Toggle snap mode, 43
  Vector scatter plot, 53
  Volume integrals, 49

Analysis script
  Configuration, 54
  GenScan, 54
  Input, 54
  Interpolation, 55
  Matrix, 55
  NScan, 55
  Output, 54
  Plot, 56
  Scan, 54
  SurfaceInt, 55
  TempAnal, 59
  VolumeInt, 55

analysis script commands, 53
analysis script example, 56
Analysis script format, 53
analysis, batch file control, 53
analytic solutions, 5
autoscale, 42

boundary conditions, 11

calculated quantities
  standard variables, 80
comment lines, 22
computational mesh, 6
configuration file, 76
  calculated quantities, 76
  expressions, 81
  load errors, 81
  structure, 76
conformal mesh, 6
contour lines, user-specified, 45
conversion factors, 8
cubic spline, 26
data export, 50
delimiters, 21
Dirichlet boundary, 11
Dufort-Frankiel method, 9
element, 6
emissivity, 72

facet, 6
fiducials, 42
field
  definition, 5
  equations, 5
field line plots, 53
field probe, 52
finite-element method, 5
  basis, 6
graphical environment, 18
grids, 42
interpolation methods, 50
keyboard/mouse control, 43
line integral, 49
mathematical functions, 67
matrix file, 50
Mesh script
  DCorrect, 9, 12
mouse controls, 48
Nelson, 66
Neumann boundary, 11
node, 6
node equations, 5, 6
nonlinear solutions, 25
number notation, 44
numerical solutions, 5
numerical stability, 9
perfusion, blood, 9
plot control, 42, 45
plot types, 42
plot view control, 44
Plot views, saving, 46
Poisson equation, 8
Probe, 4, 20, 33, 60
  file format, 60
  file information, 64
  graphical environment, 60
loading files, 60
oscilloscope mode, 63
plot export, 63
plot settings, 62
  smooth display, 63
recorded quantities, 49
recorded quantities, rules, 51
reverse Polish notation, 78
RFE2, 66
rpn
  binary operators, 78
  parsing rules, 78
scan plot menu, 51
script
  advantages, 4
  dialog, 13, 21
script control of analyses, 41
snap mode, 43
specific heat, 8
standard analysis functions, 40
Stefan-Boltzmann constant, 72
surface integral, 49
surface plot control, 44
symmetry
  cylindrical, 4, 6
  planar, 4, 6
tabular function, 15, 25
TDiff
  analysis screen, 18
  Avg, 26
  capabilities, 4
  command prompt, 34
  Cond, 25, 26, 31, 32
  Cp, 31, 32
  Dens, 31
  DTime, 33
  DtMax, 15, 28, 30
  DtMin, 15, 28, 30
  DUnit, 13, 23
file sources, 66
file types, 7, 15, 33
Fixed, 25, 31
Geometry, 13, 23
History, 33
Init, 25, 31
Instruction manual, 35
Interp, 24
Maxcycle, 24
Mesh command, 23
Mode, 23
Movie, 34
NCheck, 28
NStep, 28
Omega, 24
output file format, 37
Physical constants, 35
Radiation, 74
radiation model, 72
radiation sources, 72
Restart, 24
Run, 35
Run analysis script, 34
running in a window, 34
Safety, 28, 30
script, 15, 21
script dialog, 13, 35
SetTime, 33
Solution display, 35
solution procedure, 7
Source, 25, 31, 32
SourceFile, 23, 29, 66
SourceMod, 29, 67
steps in a solution, 13
Stop, 35
Thermal properties, 36
TMax, 15, 28
TStepMod, 29
TDiff analysis variables, 57
temperature variation, 15, 25
text editor
text editor internal, 34, 41
thermal conductivity, 8
thermal material properties, 13, 15, 21
thermal radiation, 9, 72
thermal sources, 8, 66
thermal transport equation, 7
time step, setting, 29
user-specified contour lines, 45
vector display, 53
vector tools, 52
volume integral, 49

WaveSim, 66