



HeatWave

**Three-dimensional steady-state and
dynamic thermal transport with
temperature-dependent materials**

Field Precision

Copyright 2001

Internet: www.fieldp.com

E Mail: techninfo@fieldp.com

PO Box 13595, Albuquerque, New Mexico 87192 U.S.A.

Telephone: 505-220-3975, FAX: 505-294-0222

HeatWave

Three-dimensional steady-state and dynamic thermal transport with temperature-dependent materials

Table of contents

Chapter 1. Introduction to HeatWave

- 1.1. Function of the program
- 1.2. Walkthrough example

Chapter 2. Organizing HeatWave simulations

- 2.1. HeatWave input and output files
- 2.2. **AMaze** script conventions

Chapter 3. Run control commands

- 3.1. General control commands
- 3.2. Static mode control commands
- 3.2. Dynamic mode commands

Chapter 4. Choosing a time step

Chapter 5. Defining material properties

- 5.1. Materials and regions
- 5.2. Material properties in steady-state calculations
- 5.3. Material properties in dynamic calculations

Chapter 6. Materials with temperature-dependent properties

Chapter 7. Radiation boundaries

Chapter 8. Diagnostic commands in dynamic solutions

Chapter 9. Running HeatWave in a window

Chapter 10. Automatic runs under batch file control

- 10.1. Command line operation

- 10.2. Batch files
- 10.3. Introduction to **GCon**
- 10.4. Running **GCon**

Chapter 11. HWV - file menu

Chapter 12. HWV - plane plots

Chapter 13. HWV - slice plots

- 13.1. Setting the slice view
- 13.2. Setting slice plot properties
- 13.3. Analyses in a slice

Chapter 14. HWV - surface plots

Chapter 15. HWV - analysis functions and script operation

- 15.1. Point calculation - interpolation method
- 15.2. Line scans and matrix files
- 15.3. Automatic analyses
- 15.4. Script operation

Chapter 16. Structure of the HeatWave output file

Chapter 1. Introduction to HeatWave

1.1 Program function

HeatWave performs accurate calculations of thermal transport in three-dimensional systems with arbitrary geometries. The program employs finite-element methods to generate solutions to the steady-state or dynamic thermal equations:

$$\nabla \cdot k \nabla T + Q = 0, \quad (1.1)$$

$$\rho C_p \frac{\partial T}{\partial t} = \nabla \cdot k \nabla T + Q. \quad (1.2)$$

In the equations T is the temperature (in °C), k is the thermal conductivity (J/s-m-°C), Q represents volumetric heat sources (W/m³), ρ is the material density (kg/m³) and C_p is the specific heat at constant pressure (J/kg-°C). Both modes support radiation boundaries and materials with temperature-dependent thermal conductivity. Temperature variations are defined through a system of user-generated tables. As a result, the program has the flexibility to handle any type of material. The dynamic mode supports temperature-dependent specific heat and arbitrary time variations of thermal sources and fixed-temperature boundaries. In dynamic mode solutions the program can create multiple data files consisting of full spatial information at specified times. In the steady-state mode the data file contain the spatial variation of thermal quantities for the boundary value solution. Three programs are supplied with the package:

- **HEATWAVE . EXE** creates one or more thermal solutions in a window or under batch file control,
- **HWV . EXE** generates plots and analyses from **HeatWave** data files.
- **PROBE . EXE** is a utility for analyzing and plotting probe files created by **HeatWave** in the dynamic mode. These files consist of complete temporal information on the solution at a specific location in space

MetaMesh (the **AMaze** conformal mesh generator) is required to create solutions. **HeatWave** features fast and accurate calculations in random-access memory. The programs use dynamic memory allocation; therefore, the size of the solution is limited only by the installed RAM. A computer with 1 GB of memory can handle over 10 million elements.

Input and output quantities are in the following SI units.

- Time, t : seconds
- Temperature, T : °C
- Material density: kg/m³
- Specific heat: J/(°K·kg)
- Thermal conductivity: J/(m·s·°K)
- Thermal sources: W/m³

Coordinates may be defined in **MetaMesh** in alternative units. They are converted to meters for internal use by the program by the parameter *DUnit* described in Chap. 3. Table 1.1 lists useful thermal conversion factors.

Table 1.1. Thermal conversion factors			
Quantity	From	To	Multiply by
Specific heat	cal/(°C·gm)	J/(°C·kg)	4180.0
Density	gm/cm ³	kg/m ³	1000.0
Thermal conductivity	cal/(cm·s·°C)	J/(m·s·°C)	418.0

The remainder of this chapter describes a step-by-step run that will test your setup and familiarize you with the programs. Chapter 2 reviews the programs used for a complete solution and the organization of input and output files. Operation of **HeatWave** is controlled by a script that sets parameters for program operation and defines the material properties of regions. Chapter 3 through 8 discuss the creation of scripts. Chapter 3 covers the structure of the script and commands for general run control. Chapter 4 covers the choice of time step parameters in dynamic solutions. Chapter 5 discusses script commands to set material properties while Chapter 6 gives detailed information on the preparation of tables to describe time- and temperature-dependent quantities. Chapter 7 covers code capabilities to represent thermal radiation from structures at high temperature. Chapter 8 summarizes diagnostics commands to set times for the creation of data files and location of probes. Chapter 9 covers features of the program when run as an interactive Windows utility. Chapter 10 describes how to run complex simulations automatically under batch file control. The chapter reviews features of the Field Precision utility program **GCon**. Chapters 11 through 15 describe the analysis and plotting capabilities of the **HWV** postprocessor. Finally Chapter 16 summarizes

the format of the **HeatWave** output file. This information is useful if you want to write your own analysis programs.

1.2. Test run

A quick way to understand the solution procedure is to step through an example. We shall determine dynamic temperature profiles for an array of four thermal probes inserted into tissue. In addition to **HeatWave** procedures, the example illustrates the effective use of symmetry boundaries and a variable-resolution mesh to find accurate thermal flux values on small objects. Figure 1.1 shows the geometry of a single probe. The cylindrical metal probe at a fixed temperature of 100 °C has a diameter of 1.0 mm. The exposed region of the probe extends from spherical tip at $z = 20.5$ mm to $z = 0.0$ mm. A 1.5 mm diameter sheath with poor thermal conductivity covers the probe from the solution boundary at $z = -10.0$ mm to $z = 0.0$ mm. The four probes in the array have centers at $(x = 5.0$ mm, $y = 5.0$ mm), $(x = -5.0$ mm, $y = 5.0$ mm), $(x = -5.0$ mm, $y = -5.0$ mm) and $(x = 5.0$ mm, $y = -5.0$ mm). By symmetry, we need model only the probe in the first quadrant and apply the boundary condition $\partial T / \partial \mathbf{n} = 0.0$ along the lines $x = 0.0$ and $y = 0.0$. The condition means that there is no thermal flux across the boundary. We can apply the same condition at $z = -10.0$ mm to approximate the effect of an infinite extension of the sheathed probe in the $-z$ direction. The solution volume has relatively large dimensions in x and y : $x_{\max} = y_{\max} = 25.0$ mm. The condition $T = 0.0$ °C is applied on these boundaries to approximate the effect of a large mass of exterior tissue at ambient temperature.

To begin, transfer the example files `WALKTHROUGH.MIN`, `WALKTHROUGH.WIN` and `WALKTHROUGH.SCR` to a working directory like `\AMAZE\BUFFER`. Run the program launcher `AMAZE.EXE` and make sure the program and data directories are set correctly (*i.e.*, to `\AMAZE` and `\AMAZE\BUFFER`). Run **MetaMesh** and click the command *LOAD MIN FILE* in the *FILE* menu. Pick `WALKTHROUGH.MIN` in the dialog and click *OK*. You can view or edit the script with the *EDIT MIN FILE* command in the *FILE* menu. Table 1.1 shows the contents of the file. Note that variable element resolution has been applied along all directions to ensure accurate representations of the cylindrical and spherical surfaces of the probe. Click on the *PROCESS MESH* command. **MetaMesh** sets up a foundation mesh, divides the volume into regions that represent physical entities, and then fits

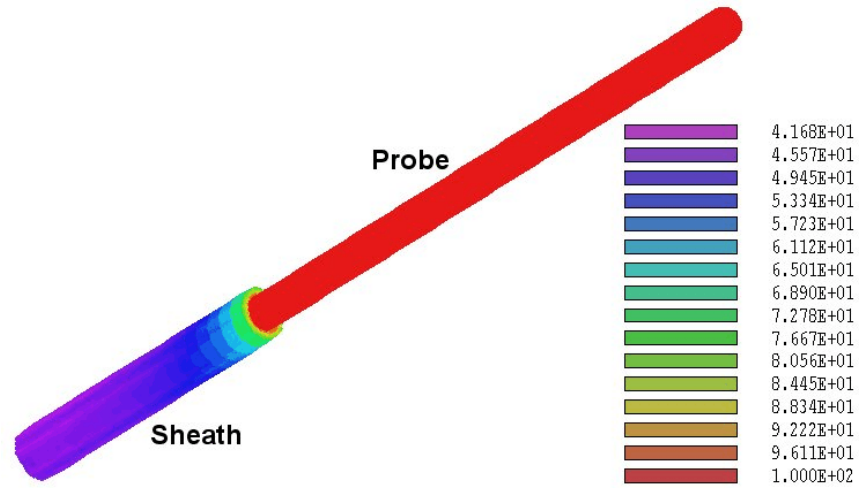


Figure 1.1. Probe geometry for the *WALKTHROUGH* example, surfaces color-coded by temperature.

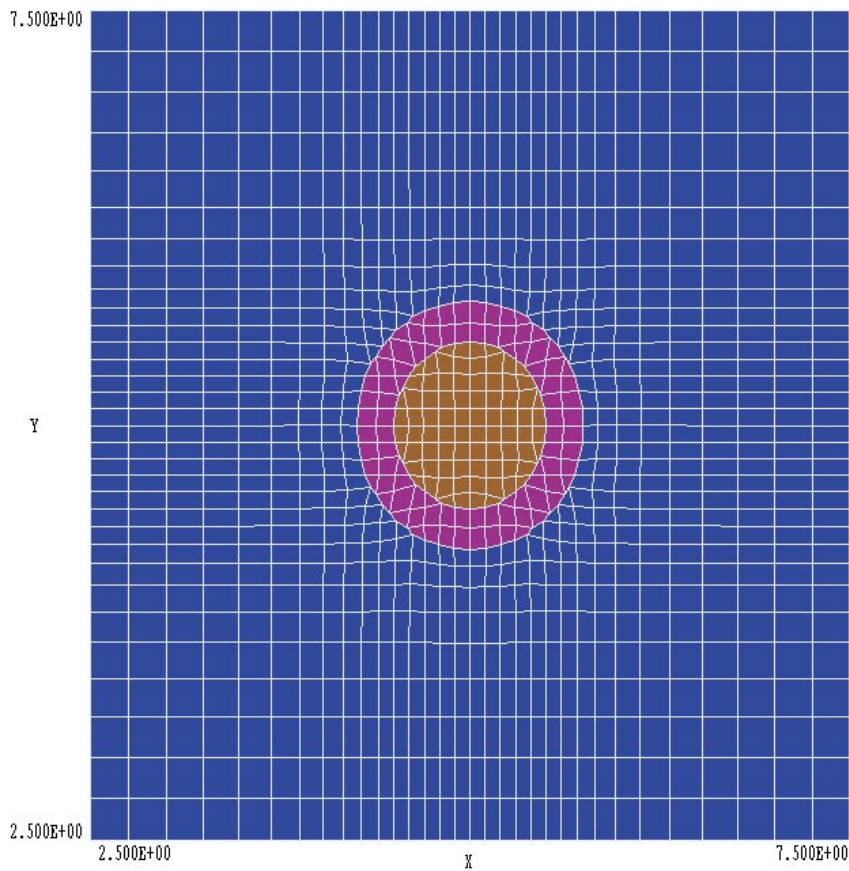


Figure 1.2. Detail of mesh for the *WALKTHROUGH* example in the plane $z = -5.0$ mm near $x = 5.0$, $y = 5.0$

designated surfaces. The program requires about 10 seconds to create a mesh with 890,847 nodes. Click on the *SAVE MESH* command in the *FILE* menu. The resulting file *WALKTHROUGH.MDF* will be used as input by **HeatWave**. At this point you may want to experiment with some of the plotting capabilities of **MetaMesh** before proceeding to the next step. Figure 1.2 shows a detailed cross section of the mesh near the probe normal to the *z* axis at *z* = -5.0 mm.

Table 1.1. Contents of the file WALKTHROUGH.MIN

<pre>* File WALKTHROUGH.MIN * Dimensions in mm Global XMesh 0.00 4.00 0.25 4.00 6.00 0.10 6.00 15.00 0.25 15.00 25.00 0.50 End YMesh 0.00 4.00 0.25 4.00 6.00 0.10 6.00 15.00 0.25 15.00 25.00 0.50 End ZMesh -10.00 18.00 0.50 18.00 22.00 0.20 22.00 35.00 0.50 End RegName(1) = Tissue RegName(2) = Sleeve RegName(3) = Probe RegName(4) = Boundary End Part Region Tissue Type Box Fab 50.00 50.00 70.00 End</pre>	<pre>PPart Region Sleeve Type Cylinder Fab 0.75 10.00 Shift 5.00 5.00 -5.00 Surface Region Tissue End Part Region Probe Name ProbeBody Type Turning L -20.0 -0.5 20.0 -0.5 S A 20.0 -0.5 20.5 0.0 20.0 0.0 S A 20.5 0.0 20.0 0.5 20.0 0.0 S L 20.0 0.5 -20.0 0.5 S L -20.0 0.5 -20.0 -0.5 End Shift 5.00 5.00 0.00 Surface Region Tissue Surface Region Sleeve End Part Region Boundary Type BoundXUp End Part Region Boundary Type BoundYUp End Part Region Boundary Type BoundZUp End ENDFILE</pre>
--	--

Table 1.2. Contents of the file WALKTHROUGH.WIN

```
* File WALKTHROUGH.WIN
* ----- Run parameters -----
DUNIT 1000.0
MODE TVAR
TMAX 252.0
DTMIN 0.001
DTMAX 1.000

* ----- Material properties -----
* Tissue
COND(1) = 0.5
DENS(1) = 1000.0
CP(1) = 4800.00
* Sleeve
COND(2) = 0.05
DENS(2) = 1000.0
CP(2) = 4800.00
* Probe
FIXED(3) = 100.0
* Boundary
FIXED(4) = 0.0
* ----- Region assignments -----
REGION(1) = 1
REGION(2) = 2
REGION(3) = 3
REGION(4) = 4
* ----- Diagnostic commands -----
SETTIME 20.0
SETTIME 50.0
SETTIME 100.0
SETTIME 250.0
HISTORY 0.00 0.00 25.00
ENDFILE
```

The next step is to run **HeatWave** from the **AMaze** program launcher. The program requires two files: the MDF file generated by **MetaMesh** and a script to control the solution. Table 1.2 shows the contents of the file `WALKTHROUGH.WIN` (the suffix stands for **HeatWave INput**). You can also view the file using the *EDIT SCRIPT FILE* command in the *FILE* menu. The first group of commands sets general run parameters. For

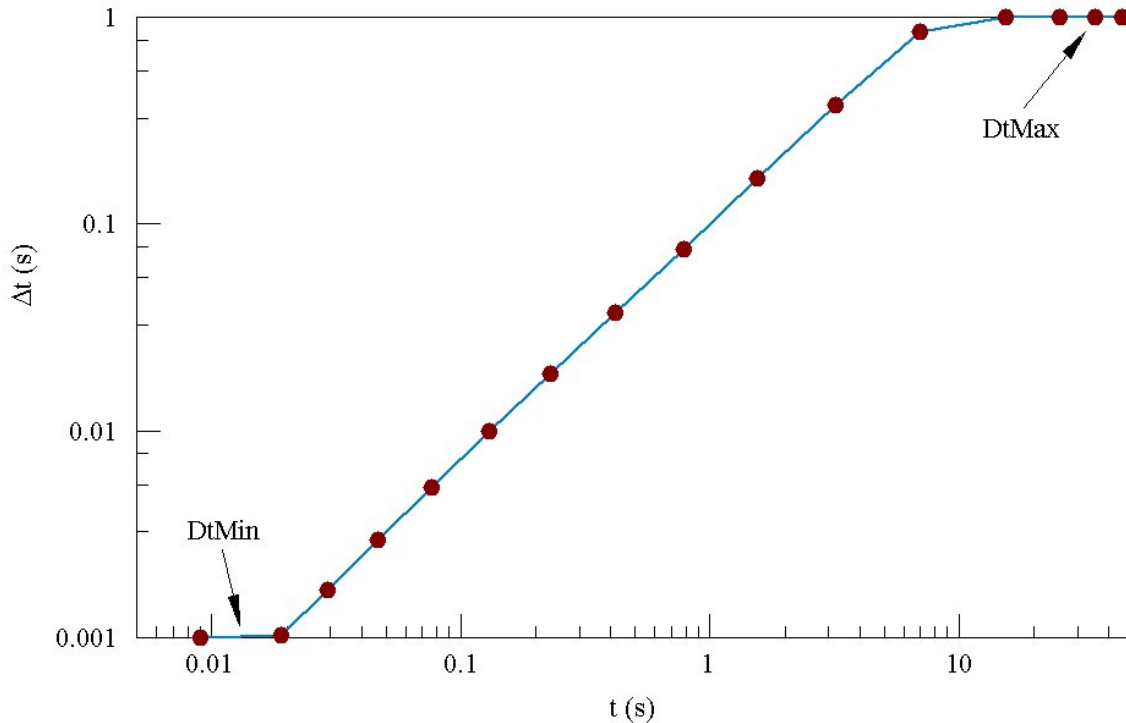


Figure 1.3. Automatic adjustment of time step in the *WALKTHROUGH* example.

example *DUNIT 1000.0* indicates that coordinates in the **MetaMesh** file are in millimeters and **MODE DYNAMIC** specifies a dynamic solution. The command *TMAX 252.0* gives the total simulation time. The *DTMIN* and *DTMAX* commands control the automatic time-step feature for the initial-value solution. The next command group assigns physical properties to materials. The tissue has $k = 0.5 \text{ W/m}^\circ\text{C}$, $\rho = 1000 \text{ kg/m}^3$ and $C_p = 4800 \text{ J/kg}^\circ\text{C}$. The conductivity of the sheath is 10 per cent that of the tissue. The probe and outer boundaries are fixed-temperature materials with values 100°C and 0°C . The *REGION* commands associate material properties with the regions defined in **MetaMesh**. The final set of commands instructs the program to create data files at times 20, 50, 100 and 250 seconds and to place a probe on-axis at position $z = 25.0 \text{ mm}$.

Choose the command *START RUN* in the *RUN* menu. In the dialog pick the file *WALKTHROUGH.WIN* and click *OK*. The program reads the mesh file, analyzes parameters in the script, calculates coupling coefficients for the finite-element calculation, advances the diffusion equation and creates

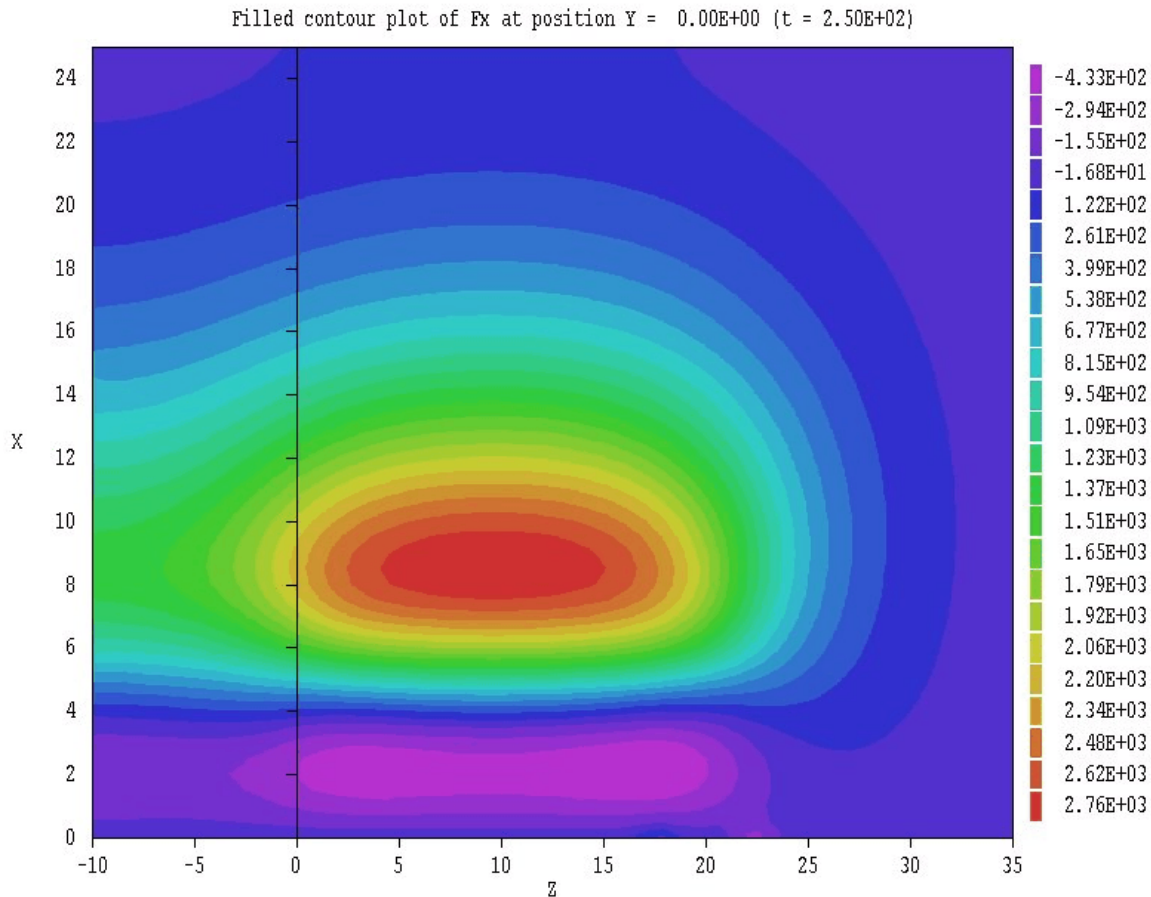


Figure 1.4. Example of a plane plot – thermal flux f_x in the plane $y = 0.0$.

the files WALKTHROUGH.001, WALKTHROUGH.002, WALKTHROUGH.003, WALKTHROUGH.004 and WALKTHROUGH.P01. The first four files are data records at the specified times and the final file is the probe record. The on-screen display shows the progress of the run. Initially the time step is quite small in order to resolve the strong temperature gradient near the probe at 100 °C in the tissue at 0°C. **HeatWave** automatically increases the time step while maintaining accuracy as the temperature profile spreads (Fig. 1.3). The entire solution process takes about 238 seconds on a 2.6 GHz computer.

Run the **HWV** program to analyze the data files. Click the *SET SOLUTION SERIES* command in the *FILE OPERATIONS* menu. Pick WALKTHROUGH.001 and click *OK*. The *LOAD NUMBERED SOLUTION IN SERIES* command is now active. Pick the command and enter the number 4 in the dialog. The program loads data from the file

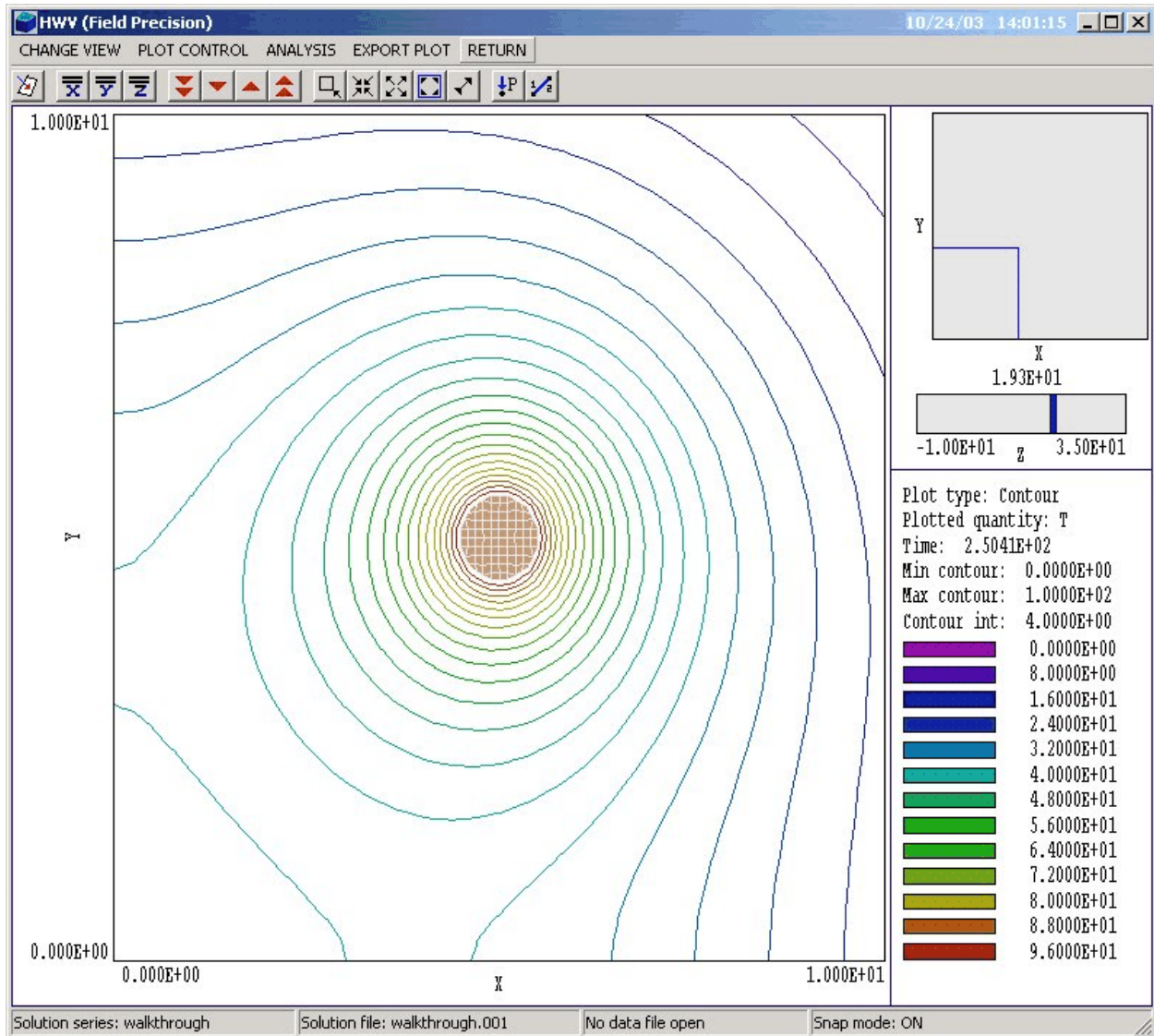


Figure 1.5. HWV working environment in the slice plot menu.

WALKTHROUGH.004 created near the end of the run at $t = 250$ seconds. After the data is loaded the *PLANE PLOTS*, *SLICE PLOTS* and *SURFACE PLOTS* menus become active. Click on the *PLANE PLOTS* menu. Plane plots show the variation of quantities in a specified plane normal to one of the axes. The plot is constructed by projecting interpolated quantities on a regular rectangular mesh. Plane plots achieve dramatic effects but may not show small details of the conformal mesh solution. Take the following actions to construct the thermal flux plot of Fig. 1.4:

- Click the *SET PLANE* command. In the dialog, choose *Y* as the normal axis and move the slider to $Y = 0.0$.
- Click the *PLOT STYLE* command in the *PLOT CONTROL* menu and choose *FILLED CONTOURS 2D*.
- Click the *PLOT QUANTITY* command in the *PLOT CONTROL* menu and choose *FX*.

The plot shows the value of the x -component of thermal flux (f_x in W/m^2) in the plane $y = 0.0$. To export a copy to the default Windows printer, click the command *DEFAULT PRINTER* in the *EXPORT PLOT* menu.

Click *RETURN* to return to the main menu and then choose *SLICE PLOTS*. Like plane plots, slice plots show the variation of quantities in a plane normal to one of the axes. The difference is that slice plots compute plot quantities directly from the conformal mesh. The procedures are more difficult but can show small details in the solution. Figure 1.5 shows the working environment for slice plots. The main plot area shows a zoomed view of isotherms in a plane normal to z near the probe. The upper area on the right shows an outline of the view region in the normal plane and the location of the slice along the normal axis. The lower area gives information on the plot.

Because slice plots are referenced to the conformal mesh, you can control visual operations (such as zooming and panning) with the mouse. It is also possible to find information on thermal quantities at a point or along a scan line by pointing and clicking. As a final activity, we shall determine the variation of temperature along a radial line from $(x = 0.0, y = 0.0)$ to $(x = 10.0, y = 10.0)$ in the plane $z = 20.09$. Take the following actions:

- If you changed the default slice view, click on *SLICE NORMAL TO Z* in the *CHANGE VIEW* menu. Use the red arrows in the toolbar to move to the plane $z = 20.9$. Finally, click on *GLOBAL VIEW* in the *CHANGE VIEW* menu.
- Click the *OPEN HISTORY* command in the *ANALYSIS* menu. Click *OK* to accept the default file prefix of *WALKTHROUGH*. **HWV** will write a record of the scan calculation to the file *WALKTHROUGH.DAT*.

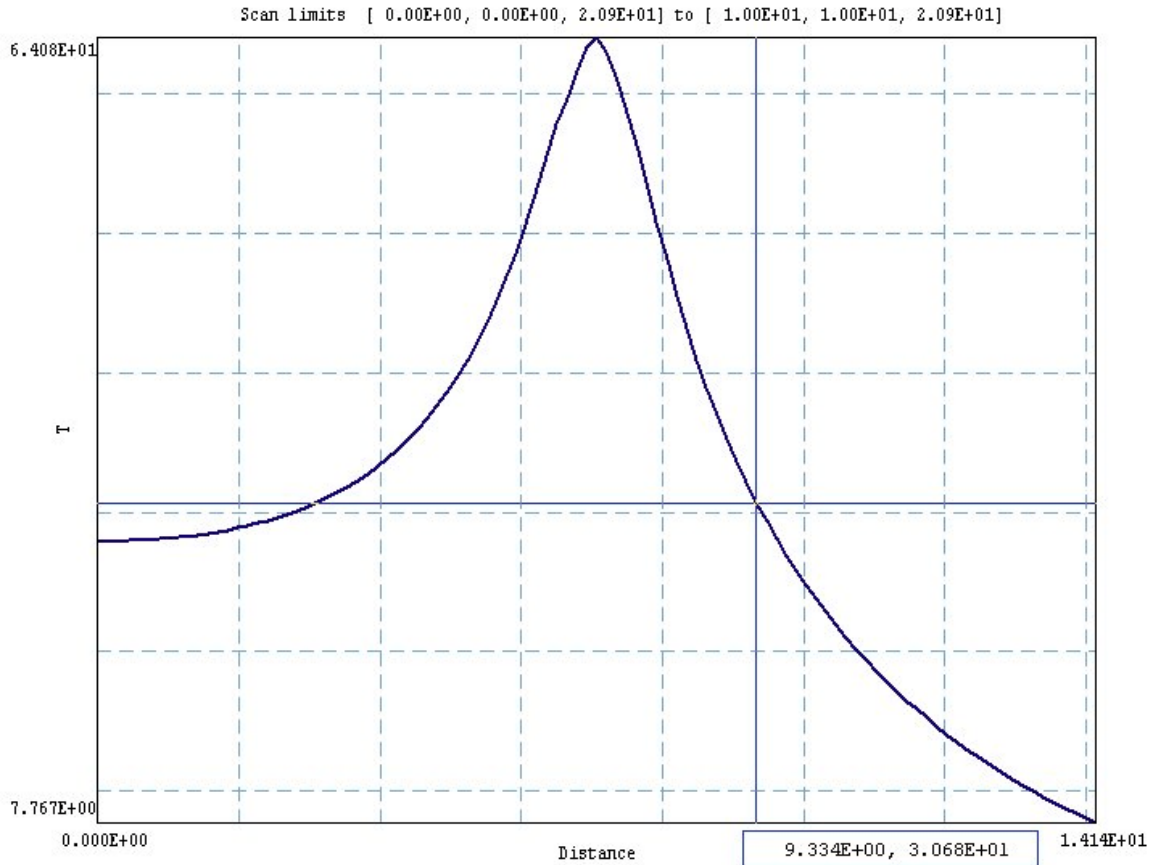


Figure 1.6. Scan plot in the oscilloscope mode.

- Click the command *LINE SCAN* in the *ANALYSIS* menu and move the mouse into the plot area. The cursor changes to a cross-hair pattern to show that pointing is active. Move the mouse toward the lower-left corner of the plot until the entries in the status bar at the bottom of the screen read $x = 0.0$ and $y = 0.0$. (Note that the values change discontinuously – mouse pointing is set to the snap mode by default.) When the mouse is positioned, click the left button.
- Next move the mouse to the position ($x = 10.0$, $y = 10.0$) and click the left button. The program computes 100 values along the scan line, enters the scan menu and creates the plot shown in Fig. 1.6.
- Click on *the OSCILLOSCOPE MODE* command. In this mode you can drag cross-hairs with the mouse to query specific points on the plot. Values are listed at the bottom of the plot area.

- Click the right mouse button to exit the oscilloscope mode and click *RETURN* to exit the scan menu. Click *RETURN* again to return to the main menu.

Because a data file (*WALKTHROUGH.DAT*) has been opened, the *ANALYSIS* submenu in the main menu is active. Choose the command *FULL ANALYSIS* in the *ANALYSIS* menu. There will be a significant delay while **HWV** performs a complex set of volume and surface integrations to find average temperature, thermal fluxes and other quantities. When the program has completed the task, click on the *CLOSE DATA FILE* command in the *FILE* menu and then choose the command *EDIT DATA FILE*. Choose *WALKTHROUGH.DAT* in the dialog and click *OK*. The data file contains quantitative information on the line scan and the global analysis. Exit the editor when you are finished. This discussion covered only a fraction of the capabilities of **HWV**. Chapters 11 through 15 cover operation of the program in detail.

Chapter 2. Organizing HeatWave simulations

2.1. HeatWave input and output files

The **HeatWave** package contains three components:

- The program **HeatWave** computes the physical solution and may generate several data and history files.
- **HWV** is a dedicated post-processor for plotting and analyses of data-file results.
- **Probe** is a standard Field Precision utility for plotting and analysis of history-file results.

HeatWave and **HWV** can run in two modes: interactively in a window or autonomously in the background under batch file control. The autonomous mode allows automatic processing of large or repetitive data sets.

Two input files are required for a **Magnum** solution:

- A script that controls the run and defines the physical properties of materials.
- A **MetaMesh** output file that describes the conformal hexahedron mesh. The file contains node coordinates and the region numbers of elements and nodes.

Additional files may be necessary to define time variations of sources and fixed temperatures and temperature variations of conductive materials.

The **MetaMesh** output file always has a name of the form `RUNNAME.MDF`. The **HeatWave** script must have a name of the form `RUNNAME.WIN`, where `RUNNAME` is the same prefix as that of the mesh file and `WIN` stands for **HeatWave INput**. **HeatWave** issues an error message if any input files are not available in the current working

directory. To organize data, the resulting output files are assigned names of the form `RUNNAME.001`, `RUNNAME.002`, . . . , `RUNNAME.P01`, `RUNNAME.P02`,

In summary, a simulation generally consists of the following steps:

- Prepare a **MetaMesh** script (`RUNNAME.MIN`) that defines the solution space.

- Run **MetaMesh** to create a file `RUNNAME.MDF` of standard mesh information. This file could be used as input for multiple **HeatWave** simulations or for other **AMaze** solution programs.

- Prepare a **HeatWave** script (`RUNNAME.WIN`) that sets control parameters and defines the material properties of regions.

- Run **HeatWave** to create data and history files. Data files are in binary format and contain the following node and element quantities: spatial coordinates, temperature, thermal conductivity, specific heat and thermal sources. History files are in text format and contain a listing of the same quantities as a function of time at a specific location.

- Optionally, prepare a standard **AMaze** analysis script (`ANYNAME.SCR`) to control an **HWV** analysis session.

- Run **HWV** to create plots or to generate numerical data using the information in any of the data files. The analysis can be performed interactively or automatically under the control of a script.

- Run **Probe** to plot and to analyze probe files.

You may feel that the procedure involves a large number of steps and a considerable amount of preparation. In the long run, the method will save you time and aggravation. Breaking complex solutions into several small steps is always helpful. The scripts form permanent records of the setup

and can often be used in different solutions with small modifications. With a good editor, script preparation is usually faster than repetitive menu operations. Note that we include two full-featured Windows editors (**Programmer's File Editor** and **SynEdit**) on the distribution disk.

Table 2.1 summarizes the input and output files used in **HeatWave**.

Table 2.1. HeatWave files		
Name form	Function	Status
RUNNAME . MIN	Description of simulation geometry, input to MetaMesh	<i>Required</i>
RUNNAME . MDF	Output from MetaMesh , input to HeatWave	<i>Required</i>
RUNNAME . WIN	Run control and description of material properties, input to HeatWave	<i>Required</i>
FNAME . TMP	Time-dependent temperature, input to HeatWave	<i>Optional</i>
FNAME . SRC	Time-dependent thermal source, input to HeatWave	<i>Optional</i>
FNAME . CND	Temperature-dependent conductivity, input to HeatWave	<i>Optional</i>
FNAME . CP	Temperature-dependent specific heat, input to HeatWave	<i>Optional</i>
RUNNAME . 001	Output data files from HeatWave , input to HWV	<i>Optional</i>
RUNNAME . P01	Output history files from HeatWave , input to Probe	<i>Optional</i>
ANYPREFIX . SCR	Analysis control, input to HWV	<i>Optional</i>
ANYPREFIX . DAT	Analysis data output from HWV	<i>Optional</i>

2.2. AMaze script conventions

The input script for all **AMaze** solution programs is a text file (ASCII format) with data lines containing commands and parameters. The script must end with the *EndFile* command. The programs make no distinction between upper and lower case. Entries on a line can be separated by the following delimiters

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

You can use any number of delimiters in a line. This means that you can add indentations and customize the appearance of the script. For example, the two lines

```
Epsi 2 5.56
Epsi(2) = 5.56
```

have the same effect.

HeatWave ignores blank lines and comment lines. Comment lines begin with the symbol [*] (asterisk). Most parameters are real numbers. The following formats are valid.

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

The final number is interpreted as 5.0.

HeatWave accepts commands in any order. The program reads and analyzes all commands before starting the solution. Generally, it is good practice to put control commands at the beginning and to group commands that set material properties by region. The following example illustrates a complete script for **HeatWave** for a static solution

```
* File RADBOUNDARYDEMO.WIN
* ----- Run parameters -----
DUNIT 100.0
MODE STATIC
NCHECK = 10
MAXCYCLE = 1500
OMEGA = 1.95
RESTARGET 1.0E-7
* --- Material properties -----
VOID(1)
```

```
COND(2) = 200.0
EMISS(2) = 1.0
INIT(2) = 800.0
COND(3) = 200.0
SOURCE(3) = 1.0E8
* -
--- Region assignments -----
REGION(1) = 1
REGION(2) = 2
REGION(3) = 3
INIT(3) = 800.0
ENDFILE
```

You can place any amount of text in any format after the *EndFile* command. Therefore you can add documentation that can be helpful when you return to a simulation after a long period of time.

Chapter 3. Run control commands

3.1. General control commands

This section reviews script that control program operation. Each command is shown in symbolic form that it might assume and also as it might appear in an application. Some commands may apply to all simulations and other commands are specific to the solution mode. This section reviews general commands

MODE [STATIC, DYNAMIC]

MODE STATIC

HeatWave can find two types of thermal solutions: static and dynamic. The term static denotes a boundary-value solution of Eq. (1.1). In this mode, the program always creates a single data file (RUNNAME.001) at the end of the run and does not create any probe files. A dynamic run is an initial-value solution of Eq. (1.2). In this mode the code may create several data and probe files in response to diagnostic commands.

It is important to note that **HeatWave** uses different routines for the two types of calculations. Static simulations involve a standard finite-element representation of the Poisson equation and a matrix solution for the temperature. Dynamic simulations utilize reduced forms for spatial derivatives to ensure numerical stability for all choices of element geometry and time step. The static method has higher inherent accuracy for a given element size. As a result, there may be small differences between a static solution and a dynamic solution that approaches a steady state. Always use the static mode for steady-state solutions.

DUNIT DUnit

DUNIT 100.0

Internally **HeatWave** uses SI units, with lengths in meters (m). Use the *DUnit* command to work in alternate length units. The real-number quantity is the number of working units per meter. For example, if you used dimensions of centimeters in the **MetaMesh** script, set *DUnit* = 100.0. Alternatively, if the mesh dimensions are in inches, use *DUnit* = 39.37. The quantity *DUnit* is recorded in the output file and is used in **HWV** for the input and output of positions. Default: *DUnit* = 1.0.

FORMAT [BINARY, ASCII]

FORMAT ASCII

By default, **HeatWave** creates data files `RUNNAME.001, . . .` in binary format. This format occupies less disk space. Binary output files are required if you want to use **HWV** to analyze the solution. We have included an option to create output files in ASCII format to make it easier to port results to your own analysis programs. The string parameter can assume the values *BINARY* or *ASCII*. Chapter 16 describes the file structure for both formats.

NGAUSS NGauss

NGAUSS 3

The parameter *NGauss* controls the number of intervals in Gaussian-quadrature integrations to determine element matrices. It should not be necessary to make any change from the default value of *NGauss* = 4. Lower values decrease the program setup time but may reduce accuracy.

INTERPOLATION [SPLINE, LINEAR]

INTERPOLATION = LINEAR

This command controls the type of interpolation performed on the tables used to define temporal and temperature variations of material properties. Cubic spline interpolations [*SPLINE*] are more accurate for smooth data. Because they preserve continuous values of the first and second derivatives, cubic splines may give anomalous results for poor data. Use the *LINEAR* option for noisy or discontinuous table values. Default: *SPLINE*.

3.2. Static mode control commands

The following commands control operation of **HeatWave** in the static mode. The program issues an error message if any of the commands are detected in a dynamic mode simulation.

RESTART ResTarget

RESTART 5.0E-8

The numerical solution of the Poisson equation involves a large set of coupled linear equations, one for each active node in the solution

volume. **HeatWave** uses an iterative technique based on corrections that reduce the error in T at a point compared to averages of values at neighboring nodes. The *residual* is an average of the relative errors over all nodes in the solution space during an iteration. This command sets a target value for the residual. The program stops if the error drops below the value. For good accuracy, the relative residual should be less than 10^{-6} . If the value of *ResTarget* is too low, the program may not converge because of roundoff errors. In this case, you can terminate a solution manually if you running **HeatWave** in the interactive mode by using the *Stop and save run* or *Abandon run* menu options. Default value: $ResTarget = 5.0 \times 10^{-7}$.

MAXCYCLE MaxCycle
MAXCYCLE 2500

The integer parameter *MaxCycle* is the maximum number of iteration cycles. **HeatWave** saves the solution and stops when it reaches *MaxCycle* even if *ResTarget* has not been attained. Default value: $MaxCycle = 2500$.

NCHECK NCheck
NCHECK= 20

The operations to calculate *ResTarget* take time; therefore, **HeatWave** usually does not make a check on every iteration cycle. The integer parameter is the number of relaxation cycles between convergence checks. Default value: $NCheck = 50$.

OMEGA Omega
OMEGA = 1.92

The real-number parameter *Omega* is the over-relaxation factor used to correct potential errors during the solution. This quantity may have a significant effect on the solution time. If the *Set omega* command does not appear, **HeatWave** automatically picks values that vary with the iteration cycle following the Chebyshev acceleration prescription. With this command, you can set a value manually. For numerical stability the value must be between 0.0 and 2.0. Higher values generally give faster convergence.

3.3. Dynamic mode control commands

This section lists commands that control operation of **HeatWave** in the dynamic mode. Chapter 4 gives a more detailed discussion of consideration for choices of time step in the initial-value solution. The program issues an error message if any of the commands are detected in a static mode simulation.

TMAX TMax

TMAX = 251.2

A required command that defines the stop time *TMax* for the run. Enter the value in seconds. All runs begin at $t = 0.0$. The quantity *TMax* should be at least one time step larger than the latest time for recording a data file.

DTMIN DtMin

DTMIN = 0.001

Sets a minimum value for the time step, over-riding the automatic step selection routines. Enter the value in seconds. Default: 1.0×10^{-9} s.

DTMAX DtMax

DTMAX = 0.05

Sets a maximum value for the time step, over-riding the automatic step selection routines. Enter the value in seconds. Default: ∞ .

NSTEP NStep

NSTEP = 550

Set a maximum allowed number of time steps in the solution. The program stops when it reaches either *TMax* or *NStep*. This command can be used to ensure that solutions controlled by a batch file do not run out of hand. Default: 100,000.

SAFETY Safety

SAFETY = 10.0

Sets a safety factor for automatic time-step adjustment. Typical values are in the range 5.0 to 30.0. Higher values may improve accuracy at the expense of longer computation times. Default: 25.

NCHECK NCheck
NCHECK = 20

The parameter *NCheck* is the interval (in time steps) between automatic time step adjustments. The calculation of *Dt* requires some work; therefore, the choice *NCheck* = 1 will probably extend the run time. Similarly, a large value of *NCheck* may result in a longer run because *Dt* is not optimized. The default value is *NCheck* = 10.

Chapter 4. Choosing a time step for a dynamic solution

HeatWave follows dynamic thermal transport by advancing in small time increments Dt . 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 element. If the solution space is about 100 elements on a side, the initial time step must be about 10,000 times shorter than the time step appropriate to model thermal diffusion in the full solution space. To perform this large adjustment, **HeatWave** has a routine that estimates a good time step (Fig. 1.3). 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 **HeatWave** handles. Chapter 3 introduced three commands that allow you to modify the time step: *SAFETY*, *DTMIN* and *DTMAX*.

SAFETY = 2.0

The automatic time-step routine estimates a maximum allowed time increment by averaging over the full solution volume and then divides by the factor *Safety*. Generally, *Safety* should exceed 5.0. It may be necessary to use higher values if there are large differences in element size on the mesh or local concentrations of temperature gradients.

DTMIN = 5.0E-7

The automatic time step routines of **HeatWave** can cope with difficult solutions where objects have initial temperatures that differ widely from their surroundings. Depending on the geometry, **HeatWave** may have to use extremely short steps to achieve accuracy early in time. Sometimes, details of the initial local temperature relaxation are not important. To speed such calculations, you can over-ride automatic selection of Dt by setting a minimum time step with this command. The automatic routines take over when the estimate of Dt rises above $DtMin$. The *DTMIN* command is also important for solutions driven by thermal sources where initially all nodes are at ambient temperature.

Because the second derivative of temperature is zero everywhere, **HeatWave** cannot estimate Dt . In this case, the program would assign the small default value $Dt = 10^{-9}$ s resulting in a long run time.

DTMAX 1.0E-3

Occasionally, the automatic routines in **HeatWave** may pick a time step that is too large 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 use this command to constrain the maximum allowed value of time increment. In picking $DtMax$ note that a good solution typically runs 500 to 2000 steps.

HeatWave uses the Dufort-Frankel method to advance node temperatures in time. The method has two advantages: 1) it is fully implicit and therefore involves a minimum number of numerical operations and 2) it is stable for any choice of Dt . The first advantage accounts for the high speed of **HeatWave**, even on large meshes. Although numerical stability is essential for a practical code, you should exercise caution in interpreting solutions. **HeatWave** will always give an answer, even if quite inaccurate. You can use the *DTMIN*, *DTMAX* and *SAFETY* commands to check the fidelity of initial solutions. Run a solution with a coarse time step and then change parameters to cut Dt approximately in half. Check for significant changes in the predicted temperature. Note that because of potential round-off errors, there is usually no advantage to using very short time steps.

Chapter 5. Defining material properties

5.1. Regions and materials

Material properties are handled differently in **HeatWave** than they are in the other **AMaze** solution programs. Physical properties are not assigned directly to the regions defined in **MetaMesh**. Instead, materials are defined independently and one or more mesh regions are associated with each material. The motivation is that a material may involve multiple tables to define time and/or temperature dependencies. Several regions can access the material information, avoiding redundancy.

Section 5.2 and 5.3 describe commands to define materials for static and dynamic calculations. Chapter 6 gives a detailed discussion of how to specify arbitrary time and temperature variations. The following command must appear once for each region in both calculation modes.

REGION *RegNo* *MatNo*
REGION(3) = 1

Assign the properties of material number *MatNo* to the region *RegNo* defined in **MetaMesh**. The default assignment is Material 1 → Region 1, Material 2 → Region 2, ...

The **HeatWave** script must contain one *REGION* command for each region in the mesh. Note that **MetaMesh** records a list of regions at the beginning of the *MLS* file in the form of comment lines. As a reminder, you can paste the text into the **HeatWave** script.

Several regions may be associated with a single material. The program issues an error message if a *REGION* command references an undefined material (*i.e.*, one for which no commands of the type described in Sects. 5.2 and 5.3 appear). **HeatWave** also issues an error messages if a region is undefined. The program makes formatted listings in the *WLS* file so you can check the validity of your material definitions and region assignments.

5.2. Material properties in static calculations

The following commands may be used in static mode calculations. In all commands the parameter *MatNo* is an integer number in the range $1 \leq MatNo \leq 32$ that references the material. Multiple commands may be necessary to define a single material. Material commands may appear in any order in the script, although it is good practice to group commands relating to a single material.

FIXED MatNo TFix**FIXED(3) = 110.5**

Assign a fixed temperature $TFix$ to all nodes whose region number is associated with material $MatNo$. Enter the value of $TFix$ in °C. Nodes with a fixed temperature are not affected by the relaxation process and will have temperature $TFix$ in the final solution. As an example, a fixed temperature could be assigned to the surface of a metal pipe that contains a constant-temperature cooling fluid

INIT MatNo TInit**INIT(5) = -20.0**

Set the temperature of nodes associated with material $MatNo$ to the value $TInit$ at the beginning of the relaxation solution. Enter $TInit$ in °C. This command will not affect the final state of a static calculation, but it may affect the convergence rate. Note that the $INIT$ command has no effect on fixed-temperature nodes. (Default: $TInit = 0.0$ °C for all materials.)

SOURCE MatNo QSource**SOURCE(2) = 5.0E6**

A source power density is assigned to elements with a region number associated with material $MatNo$. Enter the value of $QSource$ in W/m^3 . The power-density is uniform over the region. You can split a volume into several regions to approximate a spatial variation of power. If you want to set a specific total power, you can find calculated region volumes near the end of the **MetaMesh** MLS file. In this case, remember to divide by $DUnit^3$ to determine the volume in m^3 . A $SOURCE$ command should be combined with a $COND$ command for a complete definition of a conductive material. The $SOURCE$ command should not be used with a fixed-temperature material.

COND MatNo Cond**COND(5) = 0.567**

A uniform thermal conductivity $Cond$ is assigned to elements with region number associated with $MatNo$. Enter the conductivity in units of $W/m\text{-}^\circ C$.

COND MatNo TABLE TableName
COND(5, TABLE) = BRAIN.CND

Elements with region numbers associated with material *MatNo* have a temperature-dependent thermal conductivity defined by values in the tabular file *TableName*. The file must be available in the working directory. Although the tabular file may have any name, we suggest using a suffix CND. Each data line in the conductivity table contains a value of the temperature T (in °C) and the associated conductivity k (in W/m-°C). Chapter 6 discusses the preparation of tabular files. Although all elements associated with *MatNo* reference the same table, they will not have the same conductivity if they have different temperature.

EMISS MatNo Epsi
EMISS(5) = 0.82

Elements associated with material *MatNo* have an emissivity equal to *Epsi* (a number in the range $0.0 < \epsilon \leq 1.0$). Elements with non-zero emissivity emit thermal radiation through facets that border on a void material. The radiation power density is given by $p = \epsilon\sigma(T + 273.15)^4$, where p is the power density in W/m², ϵ is the emissivity, σ is the Stefan-Boltzmann constant and T is the temperature in °C. Thermal radiation is discussed in Chap. 7.

VOID MatNo
VOID 6

Elements with region number associated with material *MatNo* have the *void* property. Void elements have zero thermal conductivity. They are used to mark which surfaces of emissive elements ($\epsilon > 0$) radiate into free space. Void elements are usually located on the outer surface of the solution volume. Thermal radiation is discussed in Chap. 7

5.3. Material properties in dynamic calculations

The following commands have the same function in the dynamic mode as in the static mode:

FIXED MatNo TFix
SOURCE MatNo QSource
COND MatNo Cond
COND MatNo TABLE TableName
EMISS MatNo Epsi
VOID MatNo

The following additional material commands are necessary calculations with temporal variations.

INIT MatNo Tinit
INIT(3) = 37.0

Set the initial temperature ($t = 0.0$) of nodes with region number associated with material *MatNo* to the value *Tinit* (°C). In contrast to the static mode, the initial condition may have a significant effect on the subsequent solution. This command has no effect on fixed-temperature materials.

FIXED(MatNo) TABLE TableName [TimeAdjust,TempAdjust]
FIXED(5, TABLE) = GRAPHITE.TMP (0.01, 1.00)

The definition of a fixed-temperature material in the dynamic mode is more flexible than in the static mode. You can define a material whose temperature follows a prescribed temporal variation. For example, you could check the dynamic response of an assembly to a change in cooling water temperature. The keyword *TABLE* designates a time-dependent temperature with waveform defined by the tabular file *TableName*. The file must be available in the working directory. Although the tabular file may have any name, we suggest using a suffix *TMP*. Each data line in the table contains a value of the time t (in seconds) and the associated temperature T (in °C). Note that simulations always begin at time $t = 0.0$. The optional parameters *TimeAdjust* and *TempAdjust* are scaling parameters. Values of time are multiplied by *TimeAdjust* and values of temperature by *TempAdjust* as they are entered in the program. In this way, you can construct a set of normalized functions that can be used in a variety of simulations.

**SOURCE MatNo TABLE TableName [TimeAdjust,SrcAdjust]
SOURCE(7, TABLE) = NUCWEAP.SRC**

Materials with specified time-dependent sources can be defined in the dynamic mode. The keyword *TABLE* instructs the program to seek the file *TableName*. The file must be available in the working directory. Although the tabular file may have any name, we suggest using a suffix *SRC*. Each data line in the table contains a value of the time *t* (in seconds) and the associated power density *Q* (in W/m³). Note that simulations always begin at time *t* = 0.0. The optional parameters *TimeAdjust* and *SrcAdjust* are scaling parameters. Values of time are multiplied by *TimeAdjust* and values of power density by *SrcAdjust* as they are entered in the program. To illustrate an application, a time-dependent source could be used to find the response temperature in a biological system to a change in metabolic rate.

**DENS MatNo Rho
DENS MatNo 968.0**

A dynamic thermal solution requires specification of the thermal inertia of materials, ρC_p (ρ is the density and C_p is the specific heat at constant pressure). Use this command to set a value of density for elements with region number associated with material *MatNo*. Enter the value in kg/m³. To convert density values expressed in gm/cm³, multiply by 1000.

**CP MatNo Cp
CP(5) = 4800.0**

Set a constant value of specific heat for elements with region number associated with material *MatNo*. Enter the value in J/kg-°C.

**CP(MatNo) TABLE TableName
CP(8, TABLE) = LIVER.CP**

This form of the *CP* command sets a temperature-dependent specific heat in elements with region number associated with material *MatNo*. The string *TableName* refers to a tabular file in the working directory. Each data line in the table contains a value of the temperature *T* (in °C) and the associated value of specific heat C_p (in J/kg-°C). Note that the density and specific heat always appear in the combination ρC_p . Variations within a material where both the density and specific heat change with temperature can be incorporated in the C_p file.

Chapter 6. Time and temperature dependencies

One of the most powerful features of **HeatWave** is the ability to handle time-dependent sources and temperature-dependent thermal properties. The code has the flexibility to model any material or transport problem. The key to this flexibility is the definition of material properties through *tabular functions*. A tabular function is a text file consisting of up to 128 data lines of values for an independent and dependent variable. For example, to define a temperature-dependent thermal conductivity, each line has a temperature followed by the corresponding value of conductivity. You can prepare tabular function files with a text editor or spreadsheet. You can also incorporate published data or digitized experimental traces. The example below shows a tabular function generated with a spreadsheet. The intention was to model a sharp transition in thermal conductivity between two constant values. Note that the file syntax conforms to the same rules as the **HeatWave** script. The free-form parser accepts real numbers in any format separated by the standard delimiters. You can add documenting comment lines starting with an asterisk. The *ENDFILE* command designates the end of data. You may add any descriptive text after *ENDFILE*.

Although the lines of the example are ordered by increasing value of the independent variable, this ordering is not required. **HeatWave** sorts the list before use and records the final order in the listing file `RUNNAME.WLS`. Furthermore, the intervals of the independent variable 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. For spline interpolations a table requires a minimum of 5 entries. You must ensure that the tabular functions extend over the full range of time or temperature that will be encountered in the solution. The interpolation routines return 0.0 for values of the independent variable off the table. Off-table values can often be a source of error and confusion. Therefore, **HeatWave** writes a message in the listing file if any interpolation errors occurred.

```

* TWAVE.CND
* Tabular function example
* Thermal conductivity with a sharp transition
* 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
  4000.0   200.0
ENDFILE

```

The interpolation method is controlled by the *INTERPOLATION* command. By default **HeatWave** uses cubic splines. This method 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. Noisy data may produce large oscillations in a spline interpolation. The dependent quantity and its first derivative must vary smoothly over the table range. The example above illustrates the approach. Note how the transition between the values 10.0 and 200.0 is represented by a continuous variation. You can check the fidelity of spline interpolations

by inspecting the listing file RUNNAME .WLS. After listing the sorted table, **HeatWave** records a sample set of values calculated with spline interpolations. Use the linear interpolation method for noisy or discontinuous data.

Table 6.1 summarizes the HeatWave commands that employ tabular input.

Table 6.1. Commands to define temporal or temperature variations				
Command	Independent quantity	Dependent quantity	Scaling	Mode
COND	T (°C)	k (W/m-°C)	No	Both
FIXED	t (s)	T (°C)	Yes	Dynamic
SOURCE	t (s)	Q (W/m3)	Yes	Dynamic
CP	T (°C)	C_p (J/kg-°C)	No	Dynamic

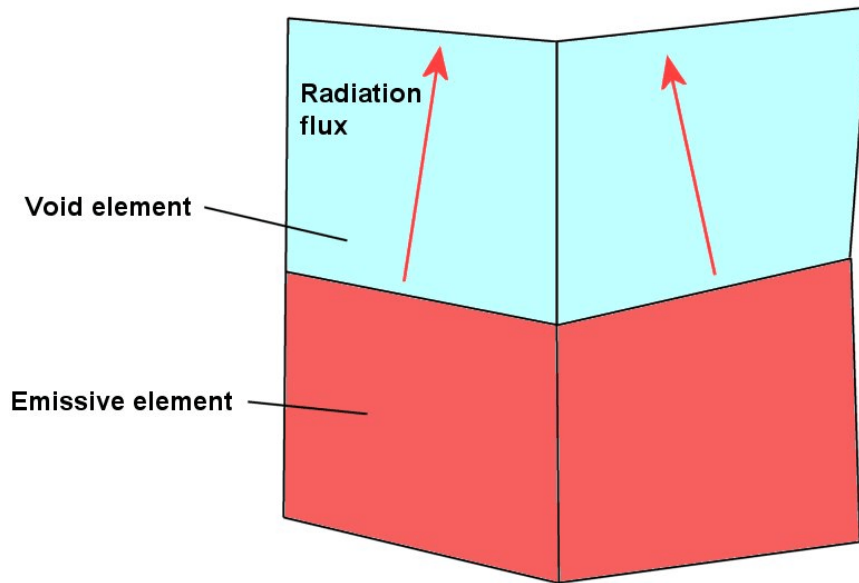


Figure 7.1. Definition of a radiation boundary in **HeatWave**.

Chapter 7. Radiation boundaries

HeatWave 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 to understand the capabilities and limitations of the model. Radiating surfaces of arbitrary shape can be defined by creating regions in **MetaMesh** and then assigning radiation properties in the **HeatWave** script using the *EMISS* and *VOID* commands. As shown in Fig. 7.1, thermal radiation leaves elements with non-zero emissivity ($\epsilon \neq 0.0$) through facets that border on void elements. The power loss through a facet is given by

$$dP = \iint_S dS \epsilon \sigma (T + 273.15)^4 \quad (7.1)$$

The surface integral in Eq. (7.1) is taken over the facet. The temperature T is computed from linear basis functions and temperatures at the four connected nodes. The quantity dP is the facet power flux, ϵ is the emissivity of the radiating element and σ is the Stefan-Boltzmann constant. The foundation of the **HeatWave** model is that thermal radiation

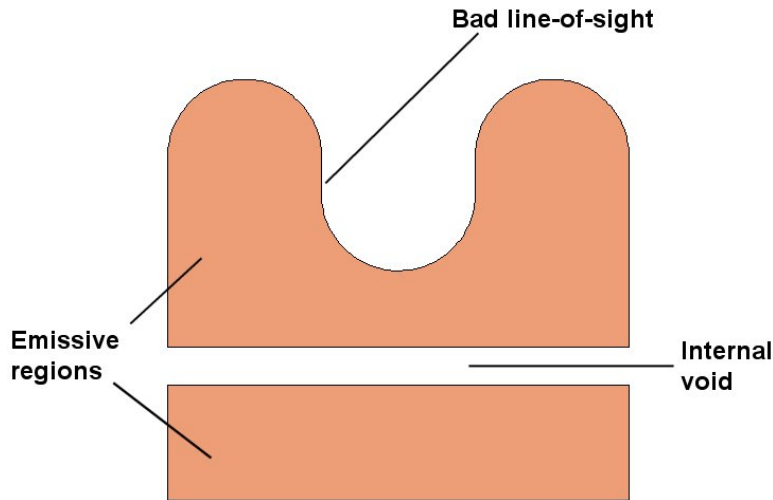


Figure 7.2. System that violates validity conditions for a HeatWave simulation with thermal radiation.

always leaves emissive elements – there is no incoming radiation. A calculation with several bodies at different temperatures that exchange radiation is extremely complex and is not handled by HeatWave. The following conditions must be satisfied for a valid thermal-radiation simulation:

- 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 effective temperature of absolute zero. By *effective*, we mean that the absolute temperature of the surroundings T_s is much less than that of the radiating elements T_e , so that $T_s^4 \ll T_e^4$. This condition ensures that there is negligible incoming radiation from the surroundings.
- All radiating facets have a clear line-of-sight to infinity.

Figure 7.2 illustrates a system where the conditions are violated.

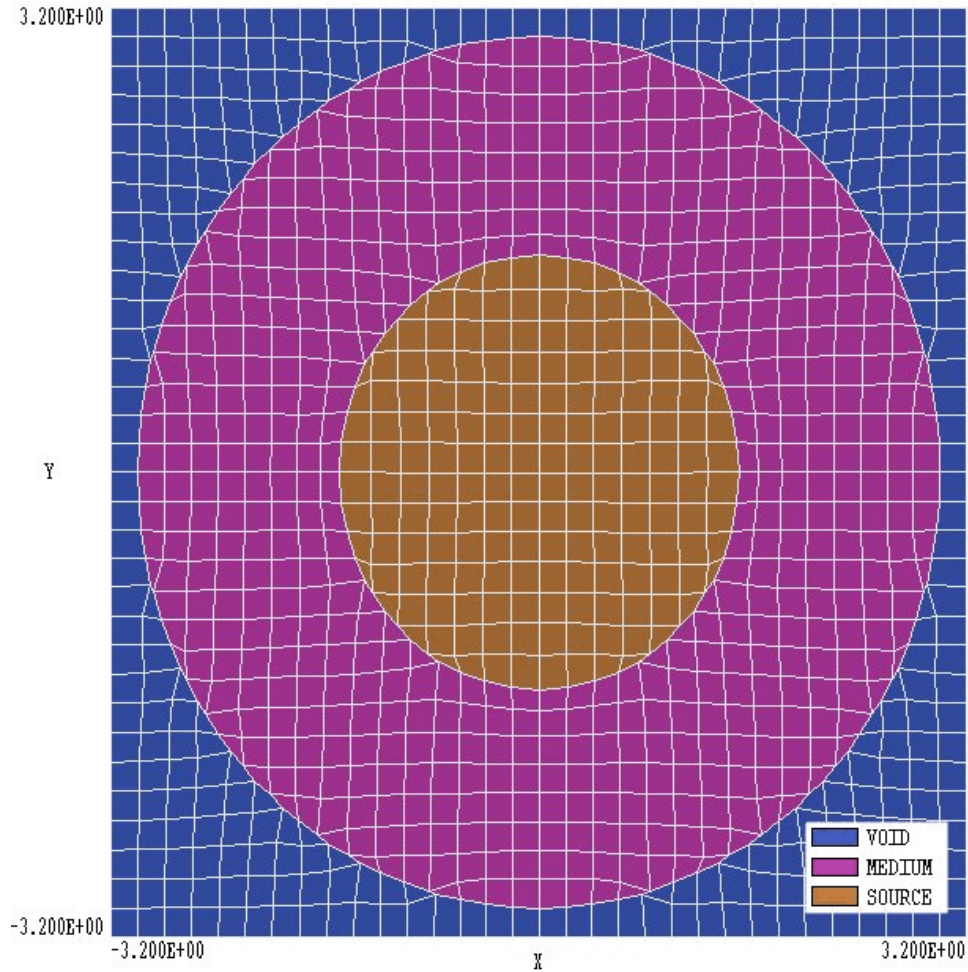


Figure 7.3. Mesh geometry for the RADBOUNDARYDEMO example – slice in the plane $z = 0.0$.

Simulation of heat transfer in a high-temperature structure involves two tasks:

- In **MetaMesh** set the solution volume boundary so that it is at least one element larger than the structure in all directions. Fill the solution volume with a region that will be defined as a void in **HeatWave**. Finally, carve out the parts that define the physical structure.
- In **HeatWave** assign the void condition to the material associated with the peripheral region. Assign appropriate values of emissivity to materials associated with radiating regions of the structure that border on the void elements.

In the subsequent calculation **HeatWave** will include thermal radiation as a negative source term in radiating elements. In dynamic calculations, sources are calculated using values of the temperature that are time-centered in the difference scheme, giving second-order accuracy. In static calculations, **HeatWave** simultaneously surface temperatures 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 *TINIT* command.

The example `RADBOUNDARYDEMO` illustrates the accuracy of the method. Figure 7.3 shows the geometry for the calculation. A sphere of radius 3.0 cm has thermal conductivity 200 W/m-°C and emissivity $\epsilon = 1.0$. The inner portion of the sphere of radius 1.5 cm is a thermal source with a total volume-integrated power of 1.414×10^3 W. The outer surface of the sphere has a surface area of 1.131×10^2 m². In equilibrium, the radiation flux is 1.25×10^5 W/m². The corresponding temperature predicted from Eq. 7.1 is 946.3 °C.

Table 7.1 shows the **HeatWave** control script. Region 1 (blue elements in Fig. 7.3) is defined as a void and Region 2 (violet elements) has $\epsilon = 1.0$.

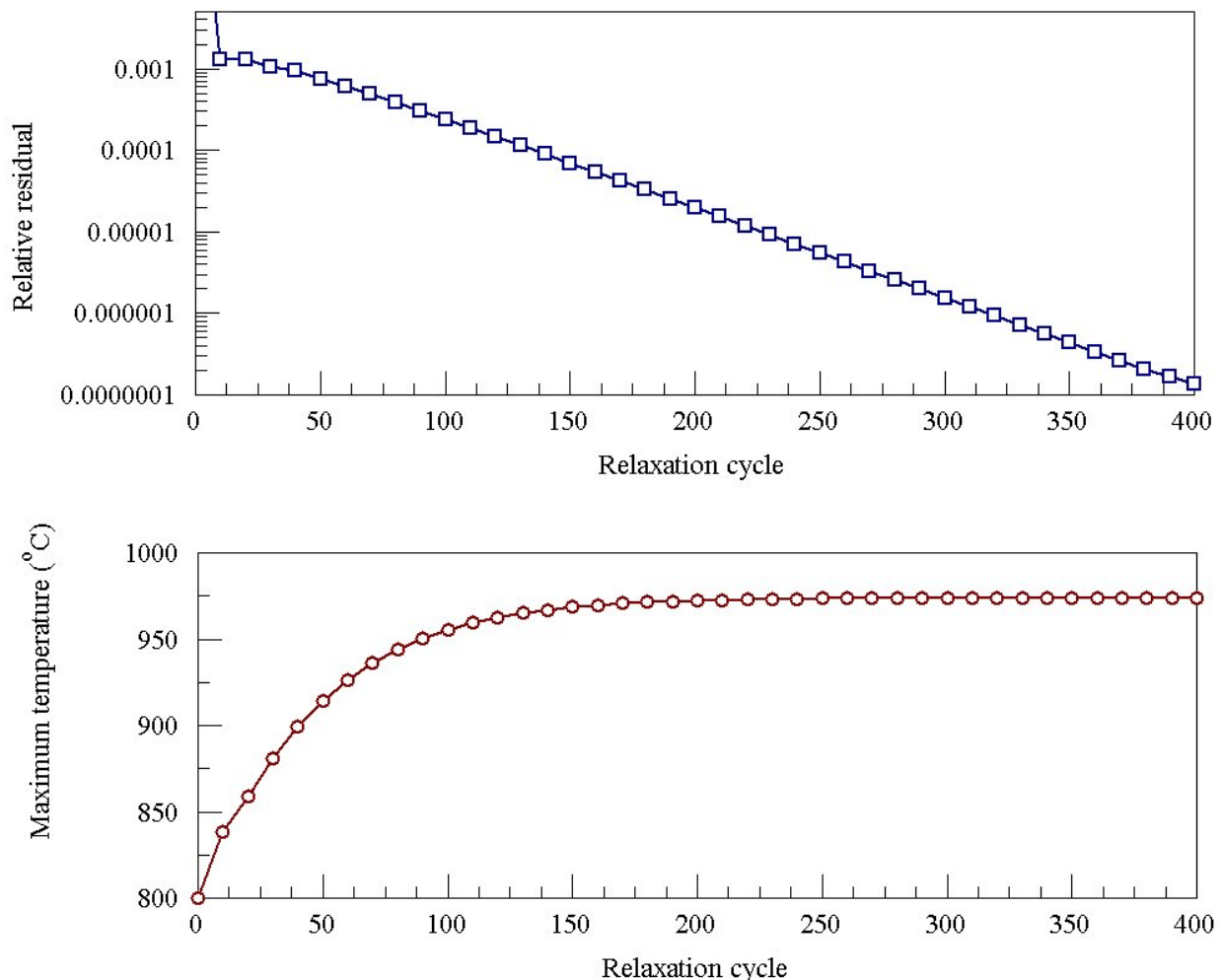


Figure 7.4. Variation of the relative residual and peak temperature during the relaxation solution for the example `RADBOUNDARYDEMO`.

Therefore, radiation emanates from the outer surface of the sphere. Note that both Region 2 and Region 3 (the section of the sphere with a source) are assigned an initial temperature of 800 °C to speed the calculation. Figure 7.4 shows the evolution of the residual and maximum temperature in the solution volume as a function of relaxation step. The surface temperature determined by **HeatWave** is 943.3 °C, a difference of only 0.3% from the theoretical prediction.

Table 7.1. HeatWave script RADBOUNDARYDEMO.WIN

```
* File RADBOUNDARYDEMO.WIN
* ----- Run parameters -----
DUNIT 100.0
MODE STAT
NCHECK = 10
MAXCYCLE = 1500
OMEGA = 1.95
RESTARGET 1.0E-7
* ----- Material properties -----
VOID(1)
COND(2) = 200.0
EMISS(2) = 1.0
INIT(2) = 800.0
COND(3) = 200.0
SOURCE(3) = 1.0E8
* ----- Region assignments -----
REGION(1) = 1
REGION(2) = 2
REGION(3) = 3
INIT(3) = 800.0
ENDFILE
```


Chapter 8. Diagnostic commands in dynamic solutions

Diagnostic commands control the creation of data and probe files in dynamic solutions. The first three commands set times to record data files. The last command sets probe positions for history files.

DTIME DTime

DTIME = 0.2

Use this command to create data files at uniform time intervals. The quantity *DTime* is the approximate interval (in seconds) between dumps. Because **HeatWave** 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, **HeatWave** will write dump 010 when $t \geq 10 \times DTime$. Default: $DTime = \infty$.

SETTIME RecordTime

SETTIME 3.45

Sometimes you may want to inspect data at certain critical times or at the end of a run. In this case you can set up to 100 time markers using multiple *SETTIME* commands. **HeatWave** will write data as soon as possible after passing each marked time. To ensure that the data is recorded, set TMax slightly larger than the last recording time. Note that multiple *SETTIME* commands must appear in the script in the order of increasing time. Default: All recording time set to ∞ .

NSTEP NStep

NSTEP = 100

In response to the *NSTEP* command, **HeatWave** creates data files at a uniform interval of times steps.

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

HISTORY XPos YPos ZPos

HISTORY = (5.00, 9.85, 10.00)

HeatWave opens a new history file `RUNNAME.P01`, `RUNNAME.P02`, . . . for each occurrence of the **HISTORY** command. The real-number parameters give the spatial location of a probe to record temperature and other thermal quantities as a function of time. Enter the coordinates in the distance units set by *DUnit*. You can define up to 12 probe positions with multiple *HISTORY* commands. **HeatWave** issues an error message if the specified position is outside the solution volume.

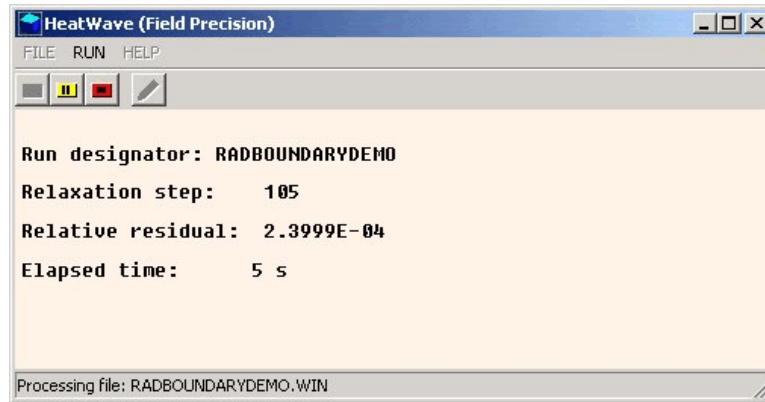


Figure 9.1. HeatWave display during a solution.

Chapter 9. Running HeatWave interactively

The program HEATWAVE . EXE can be run interactively in a window. In this mode you can perform several solutions in a session and temporarily leave the program to work on other tasks. You can launch the program from **AMaze** or create your own shortcuts. Figure 9.1 shows the program window.

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

Edit script file

Edit listing file

Edit file

The commands call up the internal editor to inspect or to modify ASCII input and output files for the solution program. With the *Edit script file* command you can work on files with names of the form FPREFIX . WIN. With the *Edit listing file* command you can pick files with names of the form FPREFIX . WLS. The *Edit file* command shows all available files. Choosing a file from an alternate directory does not change the working directory of the program.

The *Run* menu has three commands.

Start run

Pick an input file with a name of the form `RUNNAME .WIN` to start a solution. The working directory changes if you pick a file from an alternate directory. The run begins if the file `RUNNAME .MDF` is available in the working directory. **HeatWave** displays information to the progress of initial operations: loading a mesh file, calculating element matrices and generating coupling coefficients of the linear equation set. When these tasks are completed, the program begins the relaxation or initial-value solution and reports the status in the text window (Fig. 9.1).

Pause run

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

Stop run

This command terminates the program. In a static calculation, **HeatWave** creates an output file of quantities at the current state of the relaxation process. In a dynamic calculation, the program terminates history files and stops. Data files recorded at earlier times will be available.

The *Help* menu has a single command.

Instructions

Displays the HTML help file using your default browser. In order for this command to operate correctly, the file `HEATWAVE .HTML` must be in the same directory as `HEATWAVE .EXE`.

Chapter 10. Automatic runs under batch file control

10.1 Command line operation

Batch file control is a useful option for running large technical programs like **HeatWave**. You can prepare scripts to organize complex operations. The sequenced programs run automatically in the background or overnight.

To make a single **HeatWave** simulation in the background, go to the Command Prompt from Windows and log to the data directory that contains the required MDF and WIN files.. For example, suppose the data files SWITCH.MDF and SWITCH.WIN are stored in \AMAZE\BUFFER and that the program HEATWAVE.EXE is in the directory \AMAZE. From \AMAZE\BUFFER type

```
..\HeatWave SWITCH <Enter>
```

The program runs silently, writing detailed information in the listing file SWITCH.WLS. If the solution is successful, the program writes the data and history files in the data directory. During lengthy runs you can perform other tasks in Windows. Note that the response of the computer will be considerably slower because **AMaze** programs seek to use the full power of the CPU.

10.2. Batch files

The main function of the command mode is autonomous operation under batch file control. As an example, assume you have prepared the input files SWITCH01.MIN, . . . , SWITCH08.MIN and SWITCH01.WIN, . . . , SWITCH08.WIN in the directory \AMAZE\BUFFER.. Next you create the following batch file SW_RUN.BAT in the data directory using an ASCII text editor.

```
@ECHO OFF
ECHO Main switch data run
START ..\METAMESH.EXE SWITCH01
START ..\HEATWAVE.EXE SWITCH01
START ..\METAMESH.EXE SWITCH02
START ..\HEATWAVE.EXE SWITCH02
. . .
```

```
START ..\METAMESH.EXE SWITCH08
START ..\HEATWAVE.EXE SWITCH08
```

Type

```
SW_RUN <Enter>
```

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

10.3. Introduction to GCon

Microsoft has released over thirty versions of its 32-bit operating system since Windows 95. There is considerable inconsistency in DOS emulation between versions. For example, Windows 2000 runs a set of programs in the proper sequential order for any valid batch format. Windows 95 starts all programs simultaneously unless you use the command form:

```
START /WAIT ProgName
```

Windows ME does not recognize the `START` command at all. And so on.... To ensure consistent batch file operation we supply the utility program **GCon** with our software. To avoid problems we advise running batch scripts from **GCon** rather than from the Command Prompt.

GCon handles the following subset of DOS commands. Additional capabilities are described in the file `GCON.HTML`.

CD

Function: Displays the name of or changes the current directory.

Example: `CD C:\AMaze\Buffer`

COPY ... TO ...

Function: Copies a file to another location.

Example: `COPY Diode.min TO C:\AMaze\Diode.min`

ERASE

Function: Deletes a file.

Example: `ERASE diode.mls`

REM

Function: Displays the line, no action taken

MOVE ... TO ...

Function: Moves a file from one directory to another directory.

Example: MOVE Diode.mou TO C:\AMaze\Diode.mou

RENAME

Function: Renames a file or files.

Example: RENAME diode.mou diode_compare.mou

RUN

Function: Runs a specified program or command.

Example: RUN C:\tricom\mesh diode

As an example, the following script creates a mesh and an electrostatic solution using the input files DIODE .MIN and DIODE .WIN in the directory C:\EStudies\Inputs and stores the results in the directory C:\EStudies\Outputs

```
REM Sample GCon script
CD C:\EStudies\Inputs
RUN C:\AMAZE\METAMESH Diode
RUN C:\AMAZE\HEATWAVE Diode
MOVE Diode.MDF TO
C:\EStudies\Outputs\Diode.MDF
```

10.4. Running GCon

GCon is simple to run. Start the program and click on the command *Run GCon script*. The dialog shows all files with the suffix GCN. The working directory is changed if you change directories in the dialog. Pick a batch file and click *OK*. The program executes the commands in the file in sequence and displays the current status.

The *Stop on error* command toggles a switch that determines whether the program will stop the batch process if there is an error in one of the batch commands. The command *Instructions* in the *Help* menu displays a help file in your default browser. The function requires that the file GCON .HTML is in the same directory as GCON .EXE.

Chapter 11. HWV - file operations menu

The **HWV** post-processor is designed for the analyses of data files created by **HeatWave**. These files contain complete spatial information at specified times. The program has the following popup menus: *File operations*, *Analysis*, *Plane plots*, *Slice plots*, *Surface plots* and *Help*. When the program starts, only the *File* and *Help* menus are active. You must load a data file in order to create plots or to perform analyses. This section reviews options in the *File operations* menu.

Set solution series

A dynamic run may create several data files with the same prefix, `FPREFIX`. Use this command to specify a file prefix for subsequent load operations. Moving to a new directory in the dialog changes the program working directory. The command must be the first activity in an analysis session. Note that one of three following load operations must be performed to activate the plot and analysis functions.

Load first solution in series

This command (which follows *SET SOLUTION SERIES* command) loads the file `FPREFIX.001`. This file is either the first in a series for dynamic runs or the sole output from a static run. **HWV** loads the solution and updates the status bar. If data retrieval is successful, the analysis and plot menus become active.

Load next solution in series

If you are currently working with data from `FPREFIX.003`, this command loads the file `FPREFIX.004`. If no further files are available, **HWV** issues an error message and does not replace the current data. This command is useful only for multiple-file output from dynamic solutions.

Load numbered solution in series

Pick a specific data file from the output of a dynamic run. For example, to load `FPREFIX.004` supply the number 4 in the dialog. If the file is not available, **HWV** issues an error message and does not replace the current data. This command is useful only for multiple-file output from dynamic solutions.

Solution file information

The command displays information on the currently-loaded data file.

Run script

Sometimes you may want to perform complex or repetitive analyses on a set of similar solutions. Script file operation is a powerful feature of **HWV**. This command displays a dialog with a list of analysis files that you have created with the suffix `SCR`. Pick a file and click *OK*. The script file can load data files, open and close history files, and perform any of the quantitative analysis functions described in this manual. Chapter 15 describes the standard script command language used in all **AMaze** postprocessors. Note that the script file must be in the same directory as the data files.

Create script

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

Edit script

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

Open data file

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

Close data file

Use this command if you want to start a new file to record data. The data file is automatically closed when you exit the postprocessor. Otherwise, you must close the file before using the *Edit data file* command or loading the file into another program. Failure to close the file may result in a Windows Resource Sharing Error.

Edit data file

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

Edit file

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

The *Help* menu shows program information and contains the following command:

Instructions

Displays the HTML help file using your default browser. In order for this command to operate correctly, the file `HEATWAVE.HTML` must be in the same directory as `HWV.EXE`.

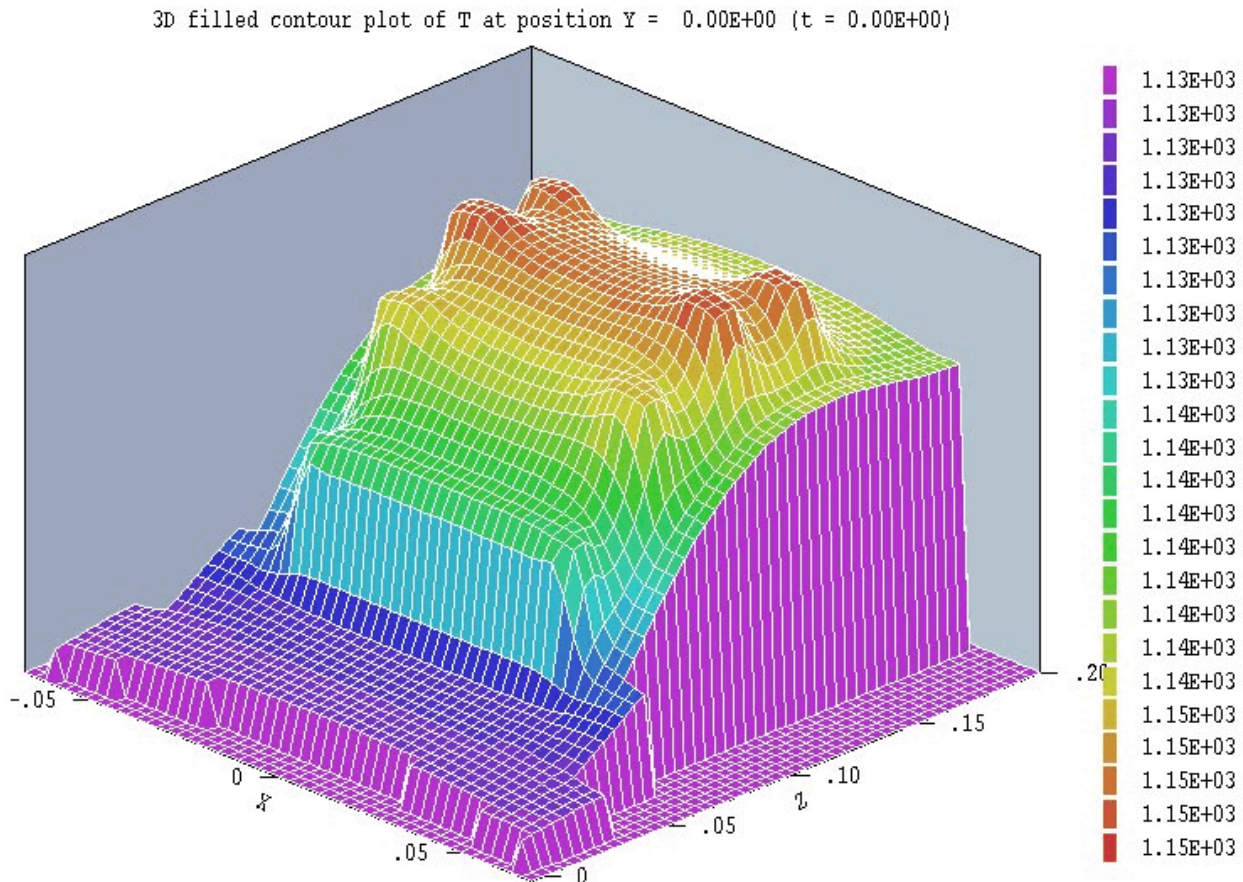


Figure 12.1. Illustration of a plane plot - 3D filled contour type.

Chapter 12. RFE3View - plane plots

Plane plots are two-dimensional representations that show the variation of quantities over a plane normal to one of the Cartesian axes. Plane plots provide a simple and quick views of the solution space. The plots utilize a rectangular mesh of values interpolated in a plane. The rectangular mesh is not keyed to the conformal mesh of the simulation. Slice plots, (discussed in the next chapter) are also two-dimensional in a plane normal to a Cartesian axis. The difference is that slice plots are built from the computational mesh. They show more precise details but are more limited in the choice of plot styles.

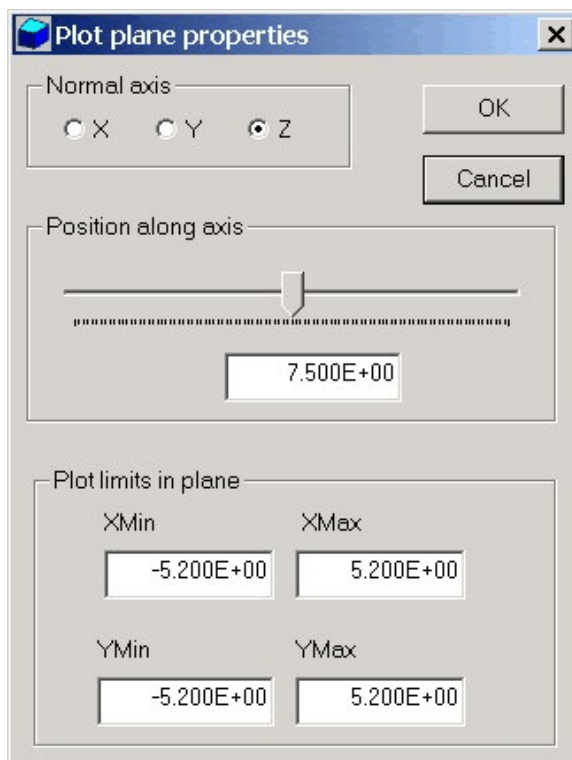


Figure 12.2. Set plane dialog.

Set plane

This command brings up the dialog of Fig. 12.2 to set the plane for the plot. Set the normal axis with the radio buttons at the top. For example, for a choice of z the plot will be created in the x - y plane. You can use the slider bar to set the position along the normal axis or type a value in the box. The range of the slider bar is automatically set to the limits of the solution volume along the normal axis. The boxes at the bottom determine the plot range in the normal plane. The default settings are the limits of the solution volume. Note that plane plots are constructed to fill the maximum area. They do not preserve scaling in the normal plane.

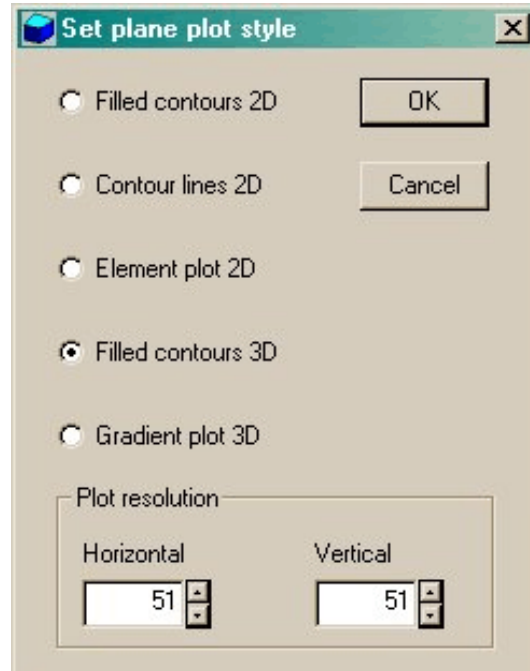


Figure 12.3. Plot style dialog.

The following commands appear in the *PLOT CONTROL* popup menu:

Plot style

This command brings up the dialog of Fig. 12.3 to set the plot style. As an example, Fig. 12.1 shows the *Filled contours 3D* plot style. The numbers at the bottom of the dialog give the resolution of the mesh used to create the plot. Higher values give more detail but require longer regenerate times. The default is a 51×51 mesh.

Plot quantity

Set the quantity to be plotted. The following options are available:

- T (temperature) in $^{\circ}\text{C}$,
- $|F|$, F_x , F_y , F_z amplitude and components of the thermal flux in W/m^2 .
- k (thermal conductivity) in $\text{W}/\text{m}\text{-}^{\circ}\text{C}$. This plot is meaningful only for simulations that include materials with temperature-dependent

conductivity,

- C_p (specific heat) in J/kg-°C. This plot is meaningful only for dynamic imulations that include materials with temperature-dependent specific heat.
- Q (source) in W/m³. This plot is generally not meaningful in **HeatWave** where sources are uniform over regions. It is included for compatibility with special **AMaze** programs that perform coupled solutions.

Plot limits

Set limits for the plotted quantity. When *Autoscale* is activated, **HWV** automatically sets limits based on the range of values in the current plot plane.

Rotate plot

This command is active only for the *Filled contours 3D* and *Gradient plot 3D* plot styles. You can rotate the plot in 90° increments for the best view.

The commands of the *Export plot* menu are used to send plots to printers or to create plot files.

Default printer

With this command, a plot can be ported to any installed Windows printer (including network printers, postscript drivers, PDF drivers...). You can generate colored plots if you have a color printer. Note that the current screen plot is sent to the **default** Windows printer. If necessary, change the default using the Windows *Settings* command **before** issuing the command.

Plot file (EPS)

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

Plot file (BMP)

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

Plot file (PNG)

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

Copy to clipboard

The current plot is copied to the clipboard in Windows Metafile format. You can then paste the image into compatible graphics software.

The screen capture capability of **MWSnap** (supplied with the program) is a useful alternative to transfer screen plots to documents. The method was used for the illustrations of this manual.

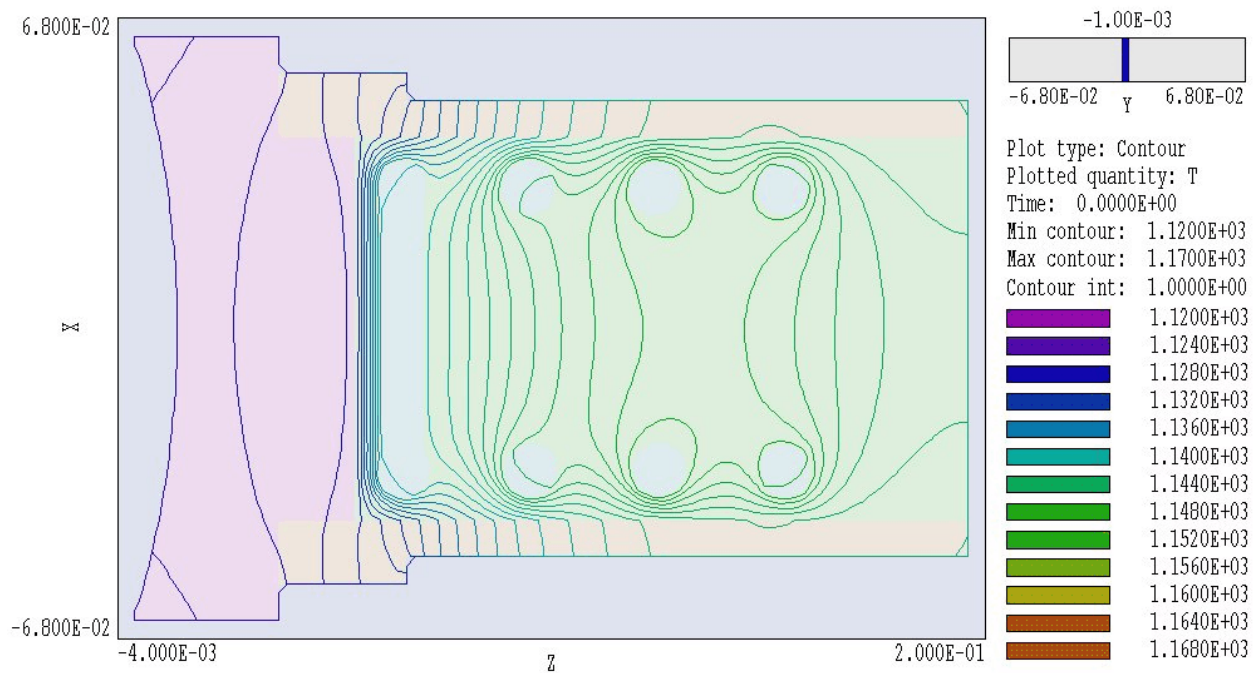


Figure 13.1. Slice plot showing isotherm lines and region boundaries.

Chapter 13. HWV - slice plots

13. 1. Setting the slice view

Slice plots are two-dimensional plots that show the variation of quantities over a plane normal to one of the Cartesian axes. In contrast to plane plots, slice plots are based on the structure of the mesh projected to a slice plane. This structure may be quite complex for a conformal mesh; therefore, slice plots require more computational effort. To facilitate the process, slices are constructed at discrete locations along the normal axis corresponding roughly to element planes in the foundation mesh. The precise rendering of spatial information makes it possible to apply mouse point-and-click analysis operations (point calculation, line scan,...) in the slice.

The *Change view* popup menu contains commands to set the slice plane and to adjust the dimensions of the plot.

Set slice plane properties

This command calls up the same dialog as the *Set plane* command in the plane plot menu (Fig. 12.2). You can change the normal axis, change the position along the normal axis, and set plot limits in the normal plane.

Slice normal to X

Slice normal to Y

Slice normal to Z

Quick commands to change the normal axis.

Jump forward

Step forward

Step backward

Jump backward

Move along the slice axis by small or large steps. The small step is approximately one layer of the foundation mesh and the large step is 5 layers. The term *forward* implies motion toward higher indices of the normal axis. The slider bar in the orientation area to the right of the plot (Fig. 13.1) shows the current location.

Zoom window

As an alternative to the entries in the *Set slice plane* dialog, you can interactively change plot limits in the normal plane using the mouse. Choose the command and move the mouse pointer into the plot area. The status bar enters coordinate mode. It shows the current mouse position in the plot and the snap mode. Use the left button to pick one corner and then move the mouse to create a view box. Click the left button again, and the plot regenerates. On any coordinate operation, press the *FI* key if want to enter values from the keyboard. The normal plane box in the orientation area to the right of the plot shows the dimensions of the slice plane and the outline of the current zoomed view.

Zoom in

Enlarge the plot about the current view center.

Expand view

Expand the plot about the current view center.

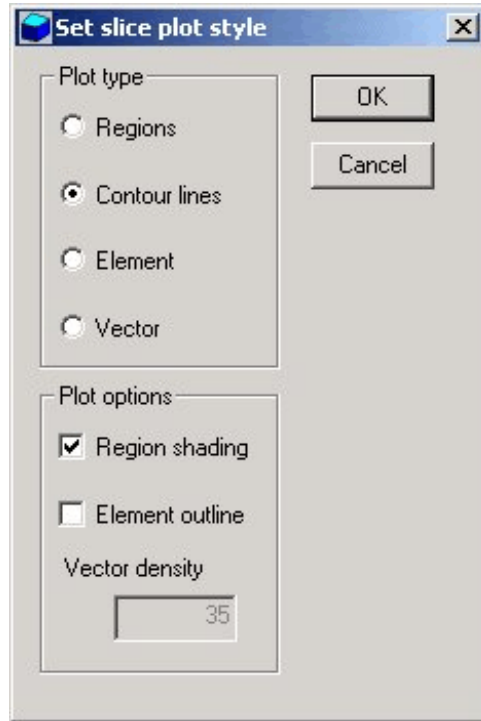


Figure 13.2. Slice plot style dialog.

Global view

Enlarge the plot boundaries to show the entire normal plane.

Pan

When the plot is zoomed, you can use this operation to shift the current view center. Use the mouse to define a relative starting and ending point for the shift.

13.2. Setting slice plot properties

The commands in the *Plot control* popup menu are used to set the plot style and mouse options.

Set slice plot style

This command brings up the dialog of Fig. 13.2. The *Region plot* style is a cross-section view of the mesh element divisions color-coded by region. In contrast to the logical plane plot of **MetaMesh**, **HWV** attempts to resolve the exact mesh structure in the plane. The *Contour plot* style shows lines of constant temperature. The vector component

of thermal flux in the normal plane is perpendicular to the isotherms with magnitude inversely proportional to the distance between lines. The *Element plot* style shows elements in the slice color-coded according to temperature. A *Vector plot* superimposes lines of thermal flux on an element plot. The check box *Region shading* is active for contour plots. When the box is checked HWV adds light background colors to show the boundaries between different conductive regions. Only fixed-temperature regions are colored when region shading is inactive. When the *Element outline* box is checked the program shows the boundaries of elements in element and vector plots. The quantity *Vector density* determines the density of lines in vector plots. Increase the number for a higher density of shorter lines.

Plot limits

In the Auto-scale mode, HWV chooses defaults for the minimum and maximum temperature values in the plots. Deactivate Auto-scale to set the values manually.

Number of contour lines

Change the number of lines for contour plots.

Toggle snap mode

Mouse coordinates for commands such as *Zoom window*, *Pan*, and *Scan in slice* can be entered in two modes. In the normal mode, the returned position corresponds to the mouse position on the screen. In the snap mode, the program picks a point at an even interval close to the mouse position. The returned coordinates depend on the value of the parameter *DSnap*. For example if $DSnap = 0.1$ and the mouse is at position (6.2345,-5.6113), the returned position is (6.2000,-5.6000). The status bar displays the actual or snapped position of the mouse.

Set snap distance

Change the value of *DSnap* from the default value determined by the program.

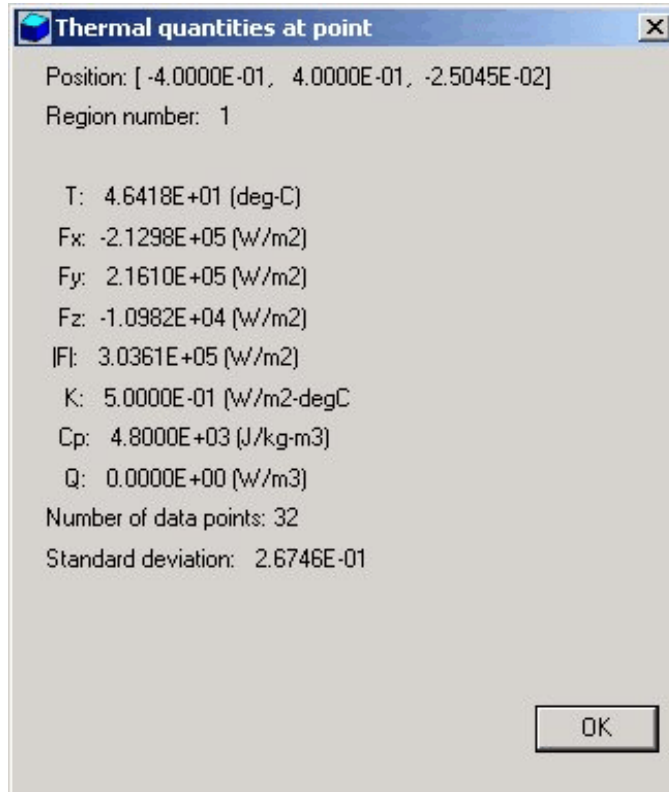


Figure 13.3. Information display – point calculation

Toggle grid

A set of dashed grid lines can be superimposed on slice plots. **HWV** automatically chooses intervals and positions so that the lines occur at convenient values along the horizontal and vertical directions (for example, 0.01 rather than 0.01153). Listings of the grid intervals are included in the axis labels. Grids corresponding to the normal plane axes are plotted as solid lines.

13.3. Analyses in a slice

You can determine field values at points and along scan lines with the commands of the *Analysis* popup menu.

Point calculation

This command is useful to make quick checks of thermal quantities in the solution volume. After you click the *Point calculation* command, move the mouse into the plot area. The mouse pointer changes to a cross-hair pattern and the status bar enters coordinate mode. Click the

left button to specify a point or press the *F1* key to enter the coordinates from the keyboard. Note that mouse coordinates will shift between discrete values if snap mode is active. **HWV** calculates the temperature, thermal flux and other quantities at the chosen location in the normal plane. The following chapter describes the interpolation method. Figure 13.3 shows the information display. Results of the point calculations are recorded if a data file is open.

Line scan

Line scans are one of the most useful **HWV** capabilities. After clicking on the command, supply two points with the mouse to define a scan line (or press the *F1* key to enter coordinates manually). The snap mode is useful in this application (for example, you may want the scan to extend from 0.000 to 5.000 rather than 0.067 to 4.985.) The program computes a series of values of thermal quantities in the normal plane at equal intervals along the line. Complete information is recorded if a data file is open. The program also makes a screen plot of the currently-selected quantity versus distance along the scan and activates the *Scan plot* menu (Fig. 1.6). **HWV** adds fiducial lines to the plot using intelligent grid selection. This means that the plot is adjusted to fill the screen and grids are drawn at useful intervals (*i.e.*, 0.05 or 2.00).

Set scan quantity

With this command you can pick the quantity that will be displayed in screen and exported plots of line scans. Pick the quantity from the list box and click *OK*. The following quantities are available in **HWV**:

- T (temperature, °C),
- $F_x, F_y, F_z, |F|$ (thermal flux components and amplitude, W/m²)
- k (thermal conductivity, W/m-°C).
- C_p (specific heat, J/kg-°C)
- Q (source power density, W/m³)

Plots of k and C_p are of interest on in simulations with temperature-dependent materials. Q is included for forward compatibility with advanced versions of **HeatWave**.

Set number of scan points

This command sets the number of line scan points in the screen plot and data file listing. The default value is 50 and the maximum number is 500.

In addition to the standard *Export plot* options, the *Scan plot* menu contains the following command:

Oscilloscope mode

In the oscilloscope mode, a scan plot assumes characteristics of a digital oscilloscope. **HWV** 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 records information when a data file is open. Press the right mouse button to exit the oscilloscope mode.

Click *Return* to exit the scan plot and return to the slice plot.

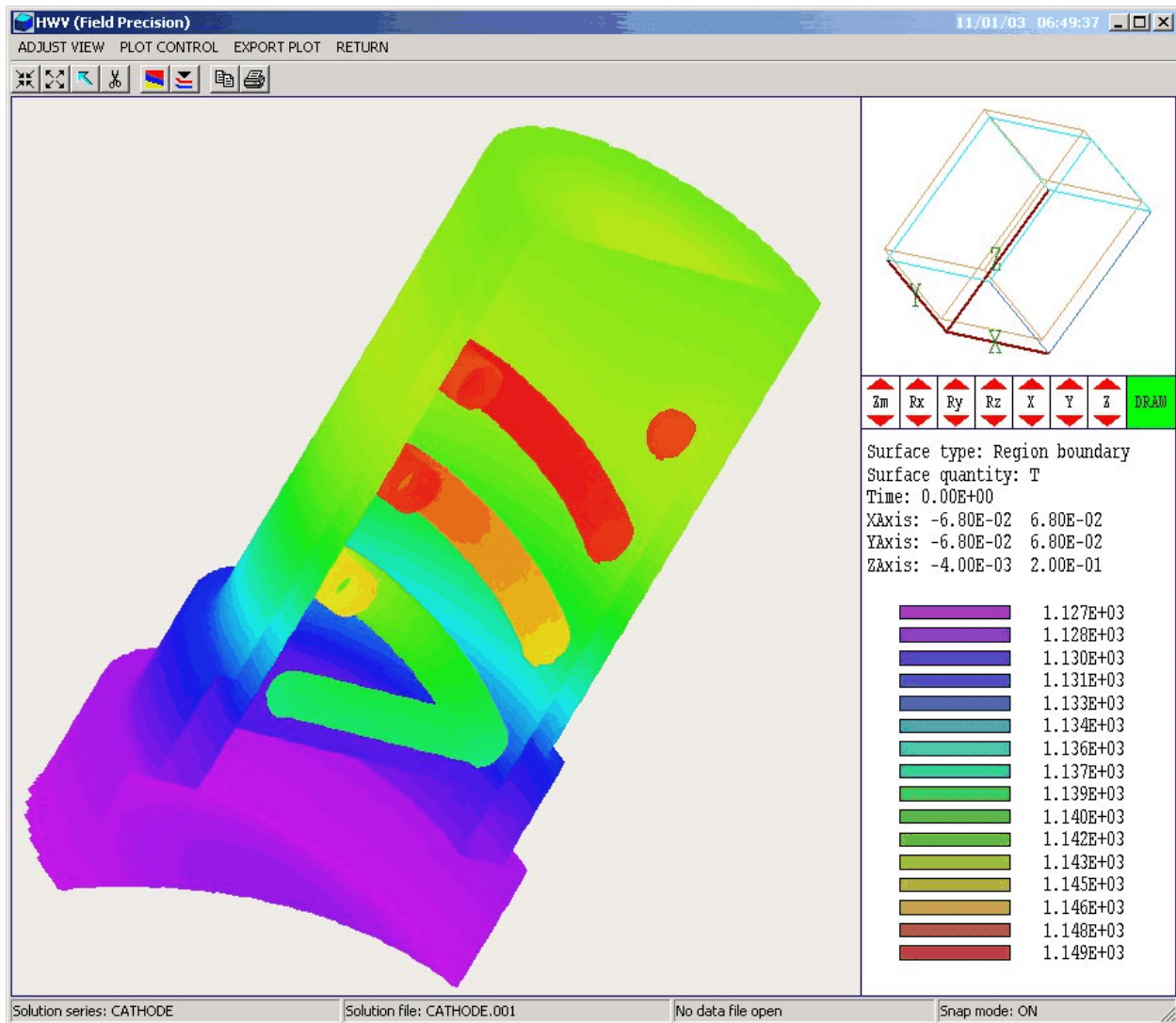


Figure 14.1. HWV working environment – surface plot menu.

Chapter 14. HWV - surface plots

Surface plots are three-dimensional views of the solution space. Four types of information can be displayed: 1) region boundaries color-coded by region number, 2) computed quantities in a slice plane normal to one of the Cartesian axes, 3) isothermal surfaces with color-coding by the amplitude of thermal flux $|F|$ and 4) boundaries of regions color-coded by temperature (Fig. 14.1). Region boundaries can be superimposed on plots of types 2 and 3. Surface plots are created from the conformal mesh and preserve true spatial scaling.

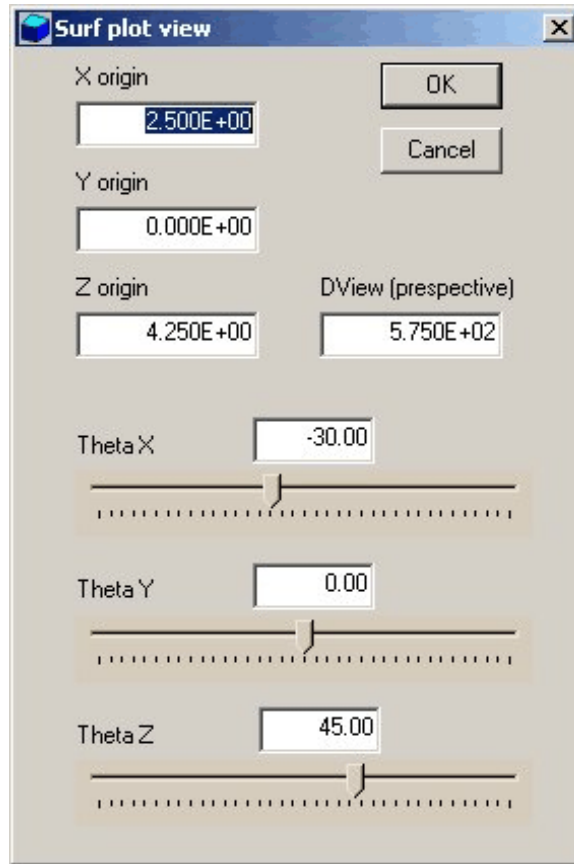


Figure 14.2. Set surface view dialog.

The three-dimensional display of the current plot style is controlled either with the onscreen controls or with the *Set surface view* command in the *Adjust view* popup menu (Fig. 14.2). The onscreen control bar is the section in the center-right portion of the of the screen(Fig. 14.1). The mouse cursor changes to a cross-hair pattern when you move it into the control region. The controls are identical to those used in **MetaMesh** for projection and surface plots. The commands have the following functions:

Zm

Zoom or expand the view. The approximate size the view volume is shown in the orientation section in the top-right portion of the screen.

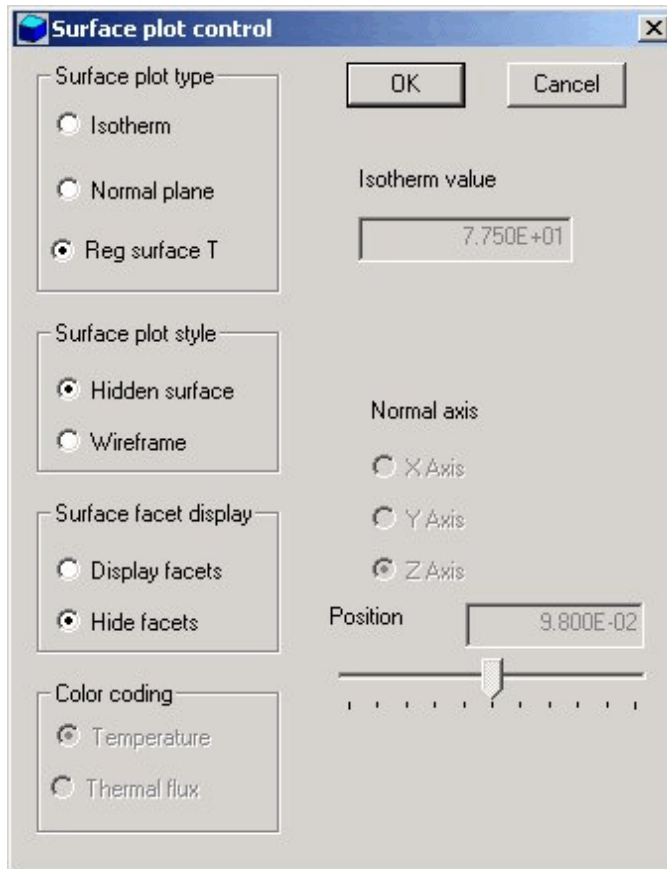


Figure 14.3. Surface plot style dialog.

Rx, Ry, Rz

Rotate about the x , y or z axes. The plot orientation is shown in the top-right portion of the screen.

X, Y, Z

Shift the view point along the x , y or z axes.

Draw

Update the main plot to reflect the rotation and shift commands. You can also click the right mouse button inside the control region to update the plot.

The commands of the *Plot control* popup menu control the appearance of the plot.

Surface plot style

This command brings up the dialog of Fig. 14.3. The *Surface plot type* radio buttons determine the type of data plot. The three options are *Isotherm*, *Normal plane* and *Region surface temperature*. In the isotherm plot **HWV** constructs a 3D surface of a surface of constant temperature within the solution volume. This is a complex process – if some of the facets are missing, rotate the plot (**Rx**, **Ry** or **Rz**) back-and-forth to correct the surface. Enter the temperature value for the isotherm in the box on the right. The surface can be plotted in wireframe or hidden surface formats (*Surface plot style* group). A wireframe plots regenerates quickly but may be confusing to view. It is often useful to set up the plot in wireframe mode and then to switch to hidden surfaces for the final rendition. Facets of the surface are color-coded according to the magnitude of thermal flux $|\mathbf{F}|$. The information window on the lower-right side of the main program screen contains a legend for the color values. The radio buttons in the *Surface facet display* group activate or deactivate plotting of the boundaries of surface facets. Turn off the facet display for a dense mesh. The *Normal plane* plot shows color-coded information on a temperature T or thermal flux magnitude $|\mathbf{F}|$ in a plane normal to one of the Cartesian axes (Fig. 14.4). The plot is similar to the plane plot discussed in Chap. 12 except that it preserves true spatial scaling. For *Normal plane* plots the commands to choose the normal plane on the right-hand side of the dialog become active. You can set the normal axis and pick the plane position along the axis. Choose the plot quantity using the radio buttons in the *Color coding* group. Normal planes can be plotted in either wireframe or hidden surface mode. The *Region surface temperature* plot shows the boundaries of selected regions in the solution volume color coded by temperature (Fig. 14.1). Note that a plot will not be created until you select one or more regions in the *Displayed regions* dialog. By default, all regions are deactivated. You can superimpose region boundaries on both Isotherm and Normal plane plots using the *Displayed regions* dialog. In this case, region facts are color-coded by region number (Fig. 14.4).

Displayed regions

The command brings up a dialog where you can pick region boundaries to display in a Region surface temperature plot or to superimpose on Isotherm and Normal plane plots. The numbered regions correspond to those defined in MetaMesh. The default is that no regions are displayed.

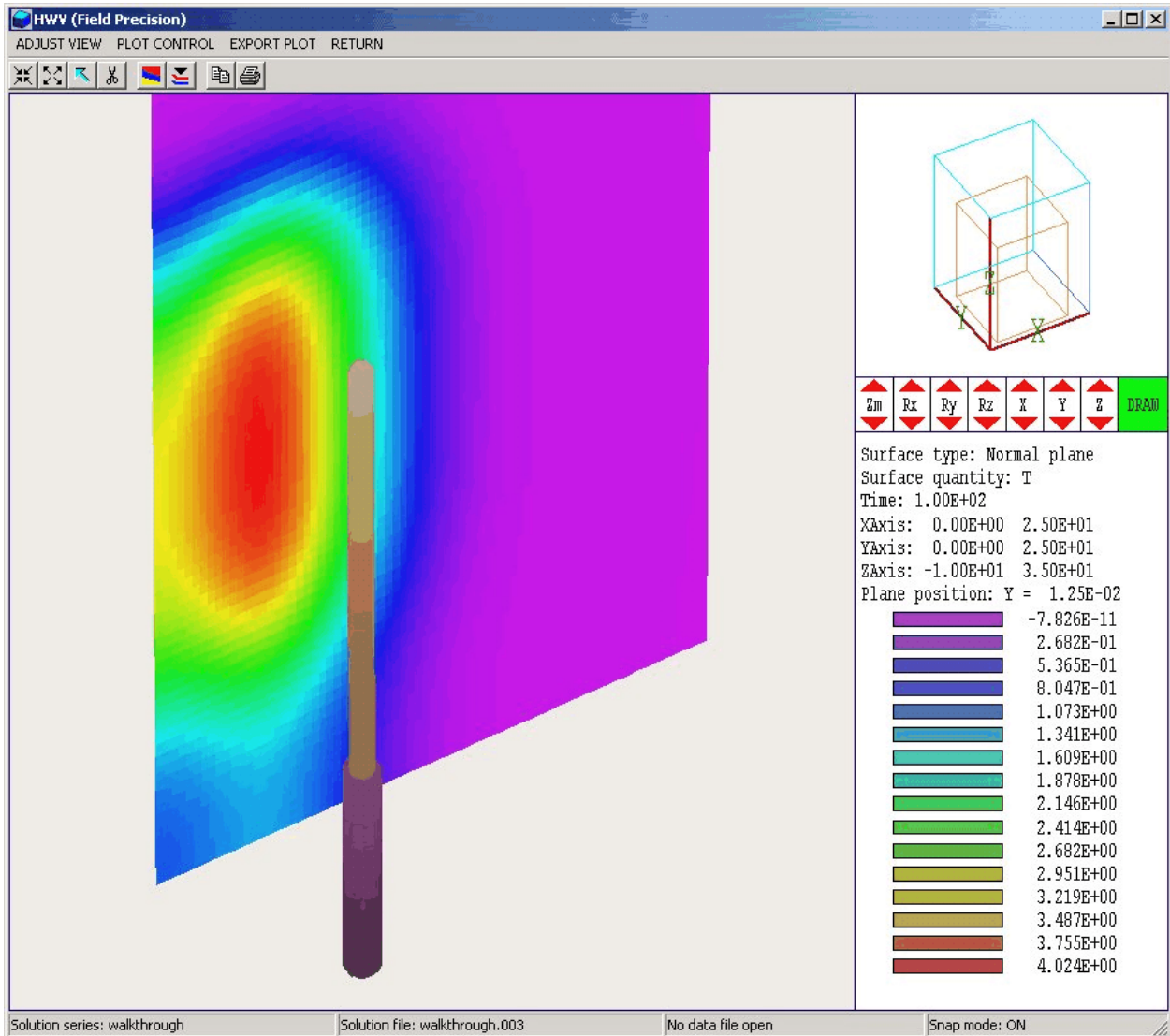


Figure 14.4. Normal plane plot with superimposed region boundaries of the probe and sheath in the WALKTHROUGH example.

Set cut planes

In a hidden surface plot, internal details may be obscured by surrounding parts or regions. This command brings up a dialog that allows you to adjust the display areas along the x , y and z axes. **HWV** does not display facets that lie outside the limits. With this feature you can create cutaway views. The default is that cut limits are set equal to the dimensions of the solution volume so that all facets are included.

Chapter 15. Structure of the HeatWave output files

HeatWave creates data and history files. Data files may be generated in binary or ASCII format. The mode is controlled by the *FORMAT* command (Sect. 3.1). The *BINARY* mode must be used to communicate results to **HWV**. The ASCII mode is useful to transfer results to your own analysis programs.

As in **MetaMesh**, nodes are referenced with the indices $[I, J, K]$ where I (the index along the x axis) extends from 0 to I_{\max} , J (y axis) from 0 to J_{\max} , and K (z axis) from 0 to K_{\max} . The number of elements is approximately equal to the number of nodes. A single element (in the direction of positive x , y and z) is associated with each node for storage. Nodes and elements have integer *region numbers* to associate them with structures in the solution space.

The format of the **HeatWave** binary output file is simple and compact, making it easy to transfer information to other programs. The code extract shown in Table 15.1. comprises the entire output algorithm. In the header, the quantities I_{\max} , J_{\max} and K_{\max} are 4-byte integers. $DUnit$ is a 4-byte real, t is a 4-byte real, CalcMode is a 1-byte integer and $NCQuant$ is a 1-byte integer. Here $DUnit$ gives the dimension units used in the MetaMesh file, t is the simulation time of the data file, CalcMode gives the type of calculation (1 for static, 2 for dynamic) and $NCQuant$ is the number of stored quantities for the solution program. In **HeatWave** $NCQuant = 4$. The recorded quantities are the node temperature and the conductivity, specific heat and source power density of the associated element. **HeatWave** then records the name of the quantity as a string of length 12. This information is used to create labels in **HWV**.

The next step is to record information for each node. The quantities $RegNo$ (the region number of the node) and $RegUp$ (the region number of the adjacent element in the direction of increasing I , J and K) are 1-byte integers. The coordinates (X , Y and Z) and the stored quantities T , KUp , $CpUp$ and QUp are 4-byte real numbers. The final step is to record region information. The quantity $NRQuant$, the number of physical quantities stored per region, is equal to 6 in **HeatWave**. The quantities are thermal conductivity k , material density ρ , specific heat Cp , initial or fixed temperature T , thermal power density Q and emissivity ϵ . The region quantity names ($RQuantName$) are strings of length 12. The variable $NFix$ is a 1-byte integer and the region quantities are 4-byte real numbers.

Table 15.1. Code to create the HeatWave binary output file

```
! Mesh size parameters
WRITE (OutData) IMax, JMax, KMax, DUnit, t, PStat.CalcMode, NCQuant
! Names of calculated quantities
DO N=1, NCQuant
  WRITE (OutField) CQuantName(N)
END DO
! Node and element information
DO K=0, KMax
  DO J=0, JMax
    DO I=0, IMax
      MC = M(I, J, K)
      WRITE (OutData) &
        C(MC).RegNo, C(MC).RegUp, C(MC).x, C(MC).y, C(MC).z, &
        C(MC).T, C(MC).KUp, C(MC).CpUp, C(MC).SUp
    END DO
  END DO
END DO
! Number of region quantities and names of quantities
WRITE (OutField) NRQuant
DO N=1, NRQuant
  WRITE (OutField) RQuantName(N)
END DO
! Number of regions
WRITE (OutField) NRegMax
! Region quantities
DO N=1, NRegMax
  MatNo = Reg(NReg).Mat
  IF (IAND(Mat(MatNo).STAT, FixedMat).EQ.FixedMat) THEN
    N = 1
  ELSE
    N = 0
  ENDIF
  WRITE (OutData) N, Mat(MatNo).Cond, Mat(MatNo).Dens, &
    Mat(MatNo).Cp, Mat(MatNo).TInit, Mat(MatNo).Source, &
    Mat(MatNo).EMiss
  ENDIF
END DO
```

Table 15.2. Code to create the HeatWave ASCII file

```
! --- Mesh size parameters
WRITE (OutData,5000)
WRITE (OutData,5100) OutDataName,IMax,JMax,KMax,DUnit, &
  t,PStat.CalcMode,NCQuant
WRITE (OutData,5200) (CQuantName(N),N=1,NCQuant)
! --- Write node and element properties
DO K=0,KMax
  DO J=0,JMax
    DO I=0,IMax
      MC = M(I,J,K)
      WRITE (OutData,5300) &
        C(MC).RegNo,C(MC).RegUp,C(MC).x,C(MC).y,C(MC).z, &
        C(MC).T,C(MC).KUp,C(MC).CpUp,C(MC).SUp
    END DO
  END DO
END DO
! --- Write region properties
WRITE (OutData,5400) (RQuantName(N),N=1,NRQuant)
DO NReg=1,NRegMax
  MatNo = Reg(NReg).Mat
  IF (IAND(Mat(MatNo).STAT,FixedMat).EQ.FixedMat) THEN
    N = 1
  ELSE
    N = 0
  ENDIF
  WRITE (OutData,5500)
  NReg,N,Mat(MatNo).Cond,Mat(MatNo).Dens,&
    Mat(MatNo).Cp,Mat(MatNo).TInit,Mat(MatNo).Source, &
    Mat(MatNo).EMiss
END DO
!
5000 FORMAT ('AMaze series, Program: 'A)
5100 FORMAT (/ &
  ' File: ',A/ &
  ' IMax: ',I6/ &
  ' JMax: ',I6/ &
  ' KMax: ',I6/ &
  ' DUnit: ',1P,E12.4/ &
  ' Time: ',E12.4/ &
  ' CalcMode: ',I2/ &
  ' NCQuant: ',I3)
5200 FORMAT (/ &
  ' RegNo ', &
  ' RegUp ', &
  ' X ', &
```

Table 15.3 shows an example of the standard probe file format. The first section is a header that contains the following information: 1) generating program name, 2) dimensionality of the generating program (1, 2 or 3), 3) the spatial position of the probe (from 1 to 3 quantities), 4) the index of the element sampled by the probe, 5) the region number of the element, 6) conversion factors for the probe position and the recorded quantities, and 7) labels for the recorded quantities. In HeatWave, each data line contains the following quantities determined by interpolation at the probe position: temperature, thermal conductivity, specific heat and source power density.

Table 15.3. Example of probe file output

Field Precision probe file				
Program: HeatWave				
NDimen: 3				
XPosition: 0.000E+00				
YPosition: 0.000E+00				
ZPosition: 0.000E+00				
ElementNo: 0				
RegionNo: 0				
NQuant: 4				
DConv: 1.000E+02				
QConv1: 1.0				
QConv2: 1.0				
QConv3: 1.0				
QConv4: 1.0				
QLabel1: Temp (deg)				
QLabel2: Cond (W/deg-m)				
QLabel3: Cp (J/kg-deg)				
QLabel4: Source (W/m3)				
Time (s)	Temperature (deg-C)	Conductivity (W/deg-m)	Spec heat (J/kg-deg)	Source (W/m3)
0.000000E+00	0.000000E+00	5.000000E-01	4.800000E+03	1.910000E+06
5.000000E-03	3.930107E-03	5.000000E-01	4.800000E+03	1.910000E+06
1.000000E-02	4.027017E-03	5.000000E-01	4.800000E+03	1.910000E+06
1.500000E-02	7.862603E-03	5.000000E-01	4.800000E+03	1.910000E+06
2.000000E-02	8.051704E-03	5.000000E-01	4.800000E+03	1.910000E+06
2.500000E-02	1.179737E-02	5.000000E-01	4.800000E+03	1.910000E+06
...				