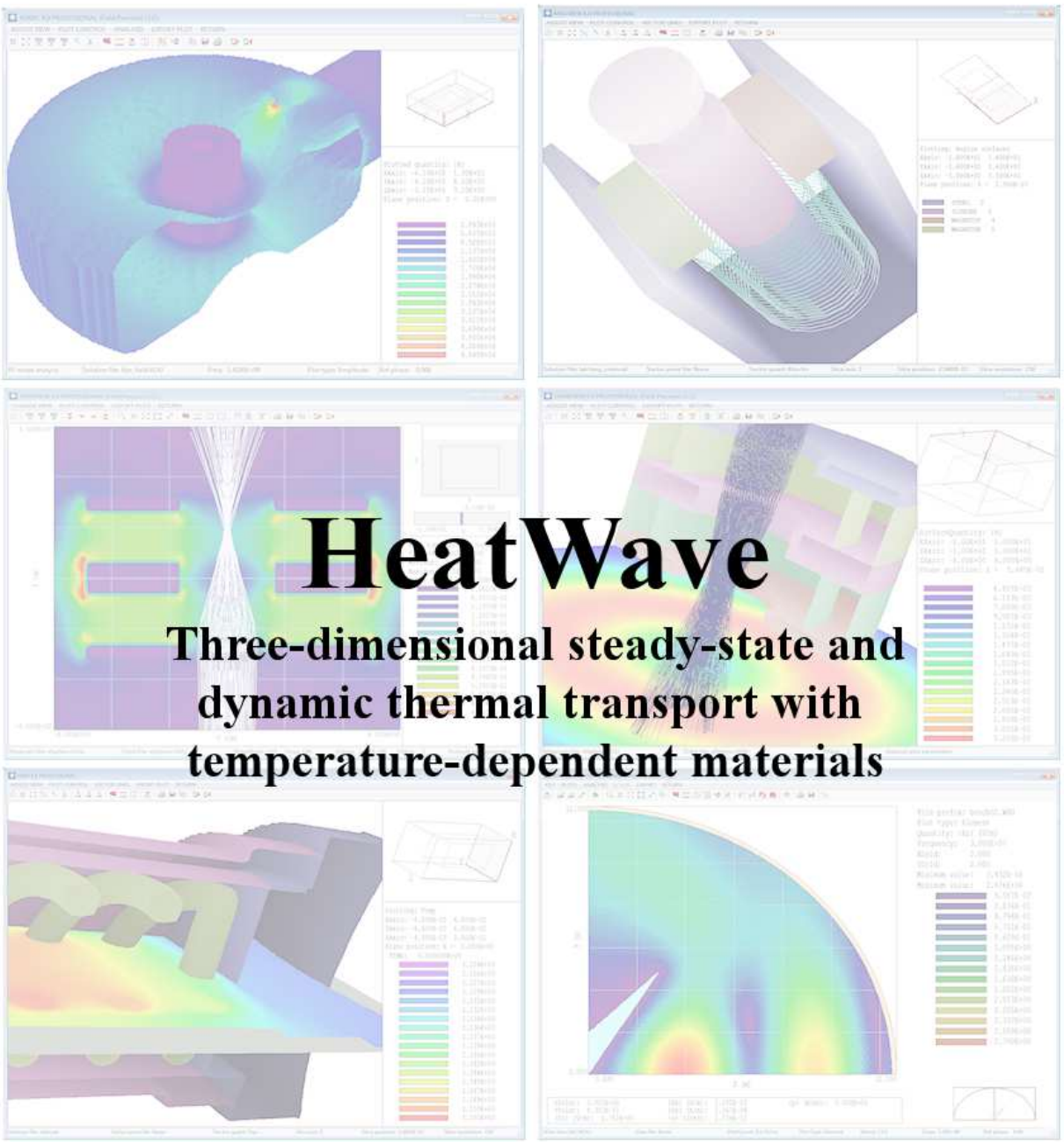
Field Precision LLC



HeatWave

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

EMail: techinfo@fieldp.com

URL: http://www.fieldp.com

Contents

1 Introduction

1.1 Program function

HeatWave handles the full range of three-dimensional thermal calculations in solids or gels. The program solves the thermal diffusion equation with the option for perfusion contributions for biological applications (*bioheat equation*). **HeatWave** employs finite-element methods on variable-resolution conformal hexahedron meshes for high accuracy and speed. **HeatWave** can determine both steady-state and dynamic solutions. In time-dependent solutions, you can define multiple sources with arbitrary temperature histories using a flexible system of tabular function input. You also have the option to define temperature-dependent thermal conductivity and specific heat. The program can load power deposition data calculated by the Field Precision programs **RFE3** (heating by RF electric fields) and **GamBet** (Monte Carlo transport of X-rays and electrons). In dynamic runs, **HeatWave** produces multiple data files at specified times and history files of temperature and other quantities at given locations. In the steady-state mode the data file contains the spatial variation of thermal quantities for the boundary value solution. **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.

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.

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

The next section briefly reviews the thermal transport equation. Section 1.3 gives a step-by-step description of a calculation to test your setup and to familiarize you with the programs. Chapter 2 reviews the functions of programs in the solution process 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. Chapters 3 through 7 discuss the creation of scripts. Chapter 3 covers commands for general run control and Chap. 4 reviews commands to set basic material properties. Advanced material properties

Table 1: **HeatWave** standard units

Quantity	**Unit**
Spatial dimensions	meters or units set by *DUnit*
Time	seconds
Temperature	degrees Centigrade, °C
Material density	kg/m^3
Specific heat (constant pressure)	J/kg-°c
Thermal conductivity	W/m-°C
Thermal sources or sinks	W/m^3

like temperature-dependent conductivity and radiation boundaries are covered in Chap. 5. Chapter 6 describes methods to import thermal source information from **RFE3** and **GamBet** and how to apply modulation functions to model time variations of sources. Chapter 7 concludes material on scripts by summarizing diagnostic commands for dynamic runs. These commands set times for the creation of data files and location of probes. Chapter 8 covers features of the program when run as an interactive Windows utility and procedures to run **HeatWave** under batch file control. Chapters 9 through 10 describe the analysis and plotting capabilities of the **HWV** postprocessor. Chapter 15 summarizes the format of the **HeatWave** output file. This information is useful if you want to write your own analysis programs.

1.2 Theoretical background

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

The dynamic thermal transport equation has the following differential form:

$$\rho C_p \frac{\partial T}{\partial t} = \nabla(k\nabla T) + Q_{reg} + Q_{file} + Q_{perf} + Q_{rad}. \tag{1}$$

In Eq. 1, ρ is the local mass density in kg/m^3, C_p is the specific heat in J/kg-°C, T is the temperature in °C, k is the heat conductivity in J/m-s-°C and the Q quantities represents sources or sinks of thermal energy in W/m^3. In steady-state, the temperature is determined by the Poisson equation:

$$\nabla(k\nabla T) = -Q_{reg} - Q_{file} - Q_{perf} - Q_{rad}. \tag{2}$$

Note that the static equation does not involve the material specific heat or mass density. The standard SI units used for thermal quantities are listed in Table 1 and Table 2 gives some useful conversion factors.

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

Region sources, Q_{reg}. These sources are spatially uniform throughout a region of the solution volume. Values are specified in the **HeatWave** script. Time variations may be

Table 2: Thermal conversion factors

Quantity	**From**	**To**	**Multiply by**
Specific heat, C_p	cal/(gm-°C)	J/(kg-°C)	4180.0
Density, ρ	gm/cm^3	kg/m^3	1000.0
Conductivity, k	cal/(cm-s-°C)	J/(m-s-°C)	418.0

assigned to region sources in dynamic solutions.

File sources, Q_{file}. These sources have spatial variations over the solution volume. Values are loaded from files generated by **RFE3** or **GamBet**. In a dynamic solution, you can specify a modulation function to introduce a global time variation.

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

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

Procedures to load file sources and to apply modulations are discussed in Chap. 6 while Sect. 5.3 covers thermal radiation from boundaries. The process of *perfusion* is the permeation of animal tissue by a flow of blood through a multiplicity of capillaries. The scale size for the effect is too small to treat directly. Instead, perfusion is modeled by a source or sink term that represents an average mass flow of blood that enters the tissue at an ambient temperature T_b and exits at the local tissue temperature. The expression for the power exchange per unit volume is

$$Q_{perf} = W_b C_b (T_b - T), \tag{3}$$

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

1.3 Walkthrough example

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 a conductive gel. In additional 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 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 a 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

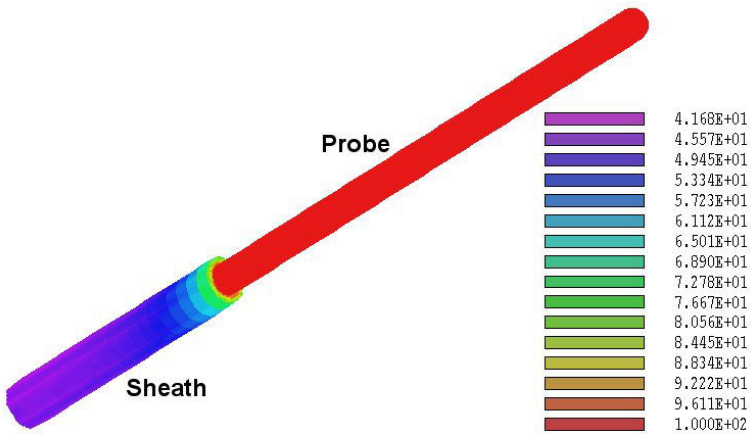


Figure 1: Probe geometry for the `WALKTHROUGH` example, surfaces color-coded by temperature.

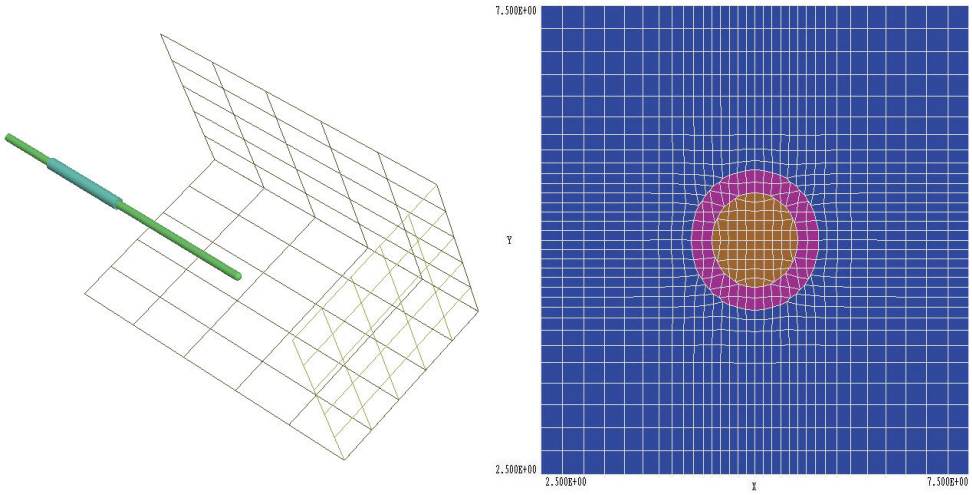


Figure 2: Mesh generation for `WALKTHROUGH` example. *a*) Perspective view in **Geometer**. *b*) Mesh detail in the plane $z = -5.0$ mm near $x = 5.0$, $y = 5.0$

condition $\partial T/\partial n = 0.0$ 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 exterior mass of material at ambient temperature.

To begin, transfer the files `WALKTHROUGH.MIN` and `WALKTHROUGH.SCR` to a working directory. Run the program launcher `FPController.EXE` and make sure the program and data directories are set correctly. Run **Geometer** and load the file `WALKTHROUGH.MIN`. You can experiment with different plotting capabilities of the program. Figure 2*a* shows a perspective view with the probe and sheath plotted as solid bodies and the external fixed-temperature boundary plotted in the wireframe mode. Note that the parts that define the probe and sheath extend out of the solution volume in the $-z$ direction. The extra length ensures that **MetaMesh** will not try to fit the ends of the cylinders. Exit or minimize **Geometer**, run **MetaMesh**

and click the command *File/Load MIN file*. Pick `WALKTHROUGH.MIN` in the dialog and click *OK*. You can view or edit the script with the *File/Edit MIN file* command. Table 3 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 designated surfaces. The program requires only a few seconds to create a mesh with 890,847 nodes. Click on the *File/Save mesh* command. 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 2*b* shows a detailed cross section of the mesh near the probe normal to the z axis at $z = -5.0$ mm.

The next step is to run **HeatWave** from the **AMaze** program launcher. The run requires two input files: the `MDF` file generated by **MetaMesh** and a script to control the solution. To generate the script, click on the *SetUp* command. In the initial dialog, choose the dynamic option and click *OK*. **HeatWave** then displays the dialog of Fig. 3. Note that the fields are divided into three groups:

Control parameters – quantities that control the solution process.

Region properties – physical characteristics of the regions of the solution volume.

Diagnostics – control data dumps (spatial information at specified times) and history files (temporal information at specified positions).

The field entries in Fig. 3 have the following meanings.

DUNIT
Dimensions in the `MDF` file are given in mm.

TMAX
The run extends from $t = 0.0$ s to $t = 252.0$ s.

DTMIN
Set a lower limit of 0.001 s for the integration time step determined by the automatic routines.

DTMAX
Set an upper limit on the integration time step of 1.0 s.

DTIME
This field can be used to set a fixed time interval between data dumps.

DATA RECORD TIMES
Create data dumps at 20.0, 50.0, 100.0 and 250.0 s.

Table 3: Contents of the 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) = Gel
  RegName(2) = Sleeve
  RegName(3) = Probe
  RegName(4) = Boundary
END
PART
  Region Gel
  Type Box
  Fab  50.00  50.00  70.00
END
PART
  Region Sleeve
  Type Cylinder
  Fab 0.75  12.00
  Shift 5.00  5.00  -6.00
  Surface Region Gel
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 Gel
  Surface Region Sleeve
END
PART
  Region Boundary
  Type BoundXUp
END
PART
  Region Boundary
  Type BoundYUp
END
PART
  Region Boundary
  Type BoundZUp
END
ENDFILE
```

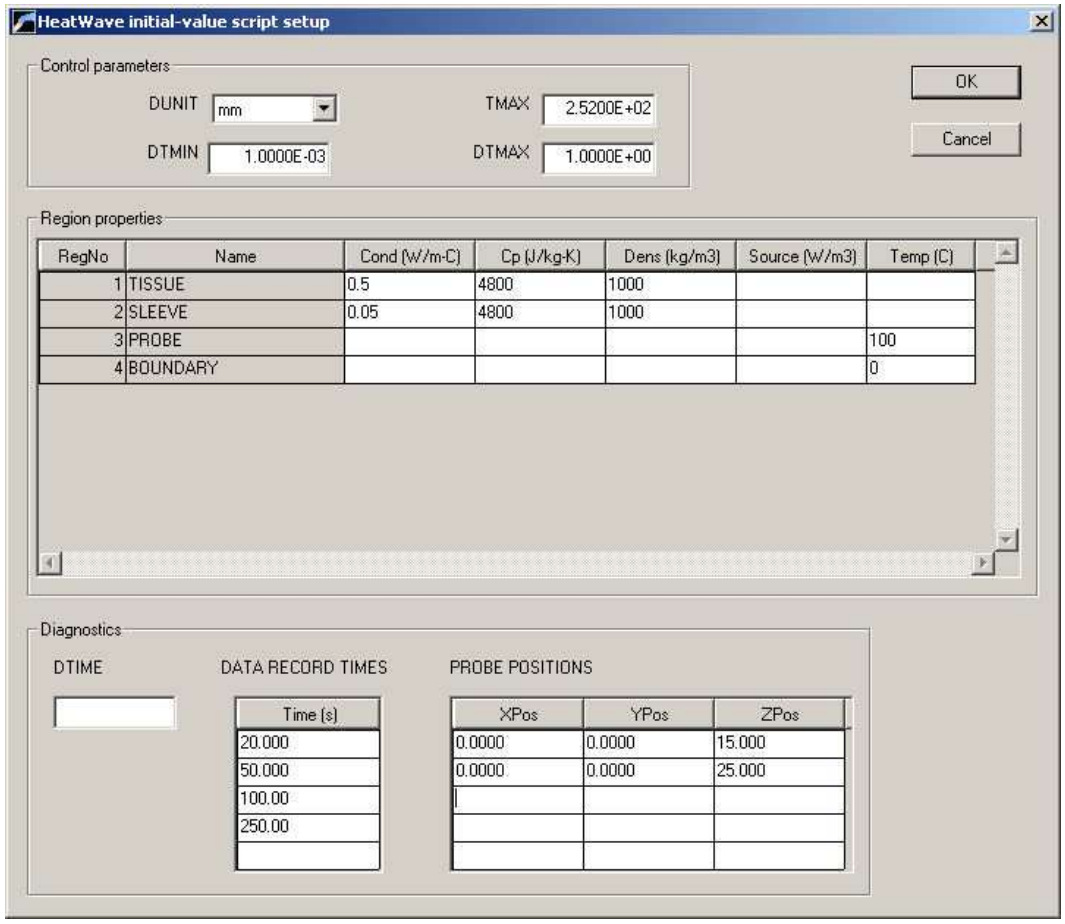


Figure 3: Dialog to generate the script `WALKTHROUGH.WIN`.

PROBE POSITIONS
Set probes at the following positions: [0.0, 0.0, 15.0] and [0.0, 0.0, 25.0].

The grid field in the middle contains material properties for the four regions of the solution volume. The bulk of the solution volume (gel) and the insulating sleeve are thermal conductors. The probe body is set to the fixed temperature 100.0 °C and the boundary is set to 0.0 °C. Enter the values shown, click *OK* and accept the default to save the script `WALKTHROUGH.WIN`. You can inspect the script and change values using the internal editor of **HeatWave** or any text editor. Table 4 shows the contents of the file. Note how the command groups in the script reflect the grouping of fields in the setup dialog.

Return to the main menu and choose the command *Run/Start run*. In the dialog pick the file `WALKTHOUGH.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 the files `WALKTHROUGH.001`, ..., `WALKTHROUGH.004`, `WALKTHROUGH.P01` and `WALKTHROUGH.P02`. The first four files are data records at the specified times and the final files are the probe records. 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. **HeatWave** automatically increases the time step as the temperature profile spreads (Fig. 4). The entire solution process takes only a few minutes.

Table 4: Contents of the file WALKTHROUGH.WIN

```
* HeatWave 2.0 Script (Field Precision)
* ---- CONTROL ----
Mesh = walkthrough
Mode = Dynamic
DUnit =   1.0000E+03
TMax =   2.5200E+02
DtMin =   1.0000E-03
DtMax =   1.0000E+00
* ---- MATERIAL PROPERTIES ----
* Material 1
Cond(1) =   5.0000E-01
Cp(1) =   4.8000E+03
Dens(1) =   1.0000E+03
* Material 2
Cond(2) =   5.0000E-02
Cp(2) =   4.8000E+03
Dens(2) =   1.0000E+03
* Material 3
Fixed(3) =   1.0000E+02
* Material 4
Fixed(4) =   0.0000E+00
* ---- REGION ASSIGNMENTS ----
* Region 1: GEL
Region(1) =   1
* Region 2: SLEEVE
Region(2) =   2
* Region 3: PROBE
Region(3) =   3
* Region 4: BOUNDARY
Region(4) =   4
* ---- DIAGNOSTICS ----
SetTime =   2.0000E+01
SetTime =   5.0000E+01
SetTime =   1.0000E+02
SetTime =   2.5000E+02
History =   0.0000E+00  0.0000E+00  1.5000E+01
History =   0.0000E+00  0.0000E+00  2.5000E+01

EndFile
```

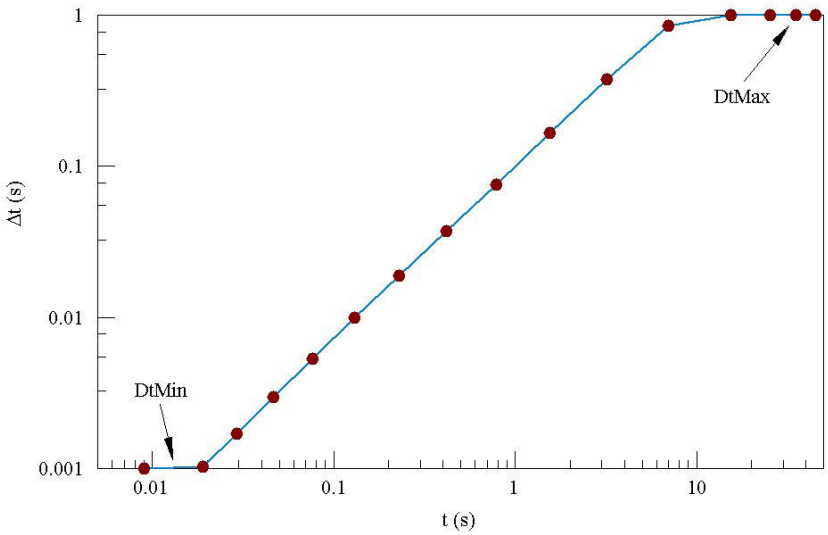


Figure 4: Automatic adjustment of time step in the `WALKTHROUGH.WIN` example.

Run **HWV** to analyze the data files. Click the command *Load solution file* in the *File operations* menu. Pick `WALKTHROUGH.004` and click *OK*. The program loads data from the file, created near the end of the run at $t = 250$ seconds. After the data are loaded the *Analysis*, *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 are attractive displays but may not show small details of the conformal mesh solution. Take the following actions to construct the thermal flux plot of Fig. 5:

> 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 *XFlux*.

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 go to the main menu and 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 display small details in the solution. Figure 6 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,

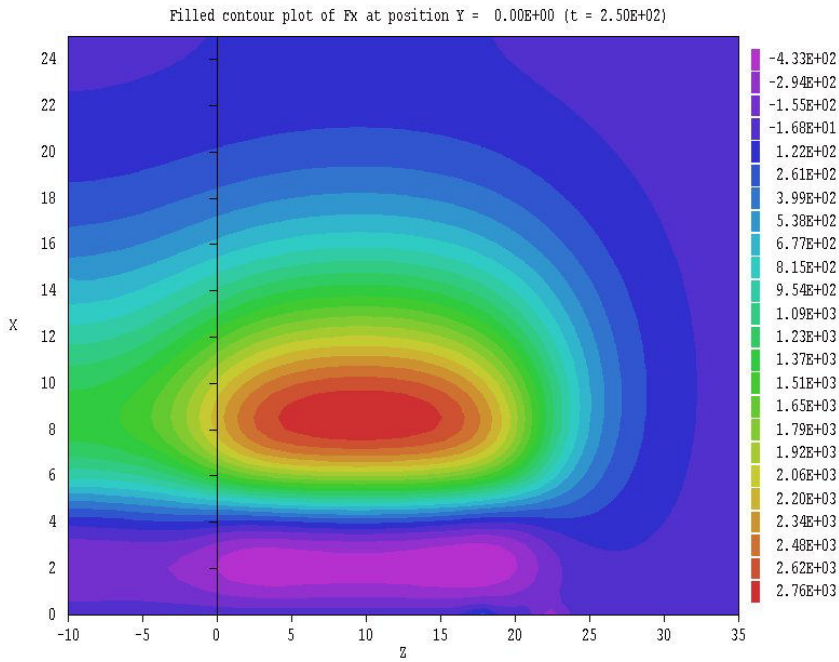


Figure 5: Example of a plane plot – thermal flux f_x in the plane $y = 0.0$.

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:

1. 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.

2. 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`.

3. Click the command *Analysis/Line scan* 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.

4. 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 of Fig. 7.

5. 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.

6. 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.

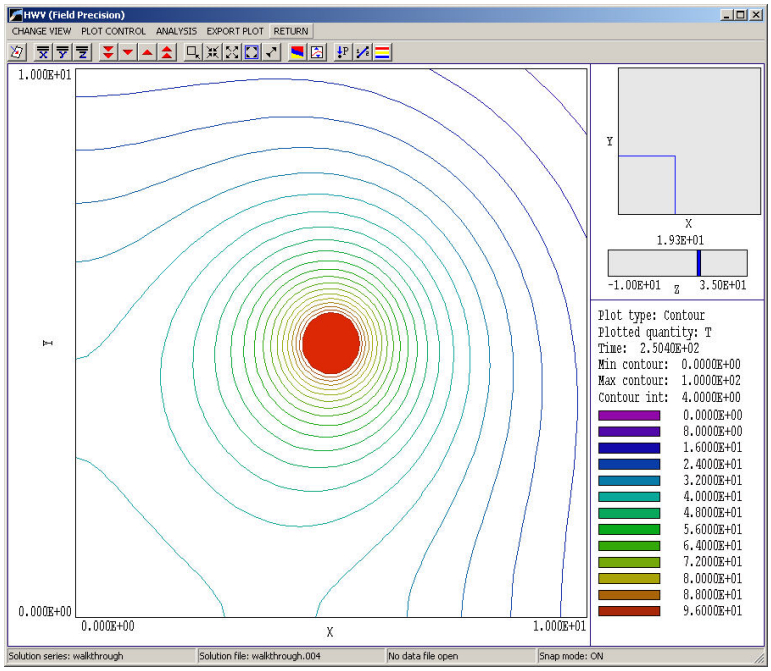


Figure 6: **HWV** working environment in the slice plot menu.

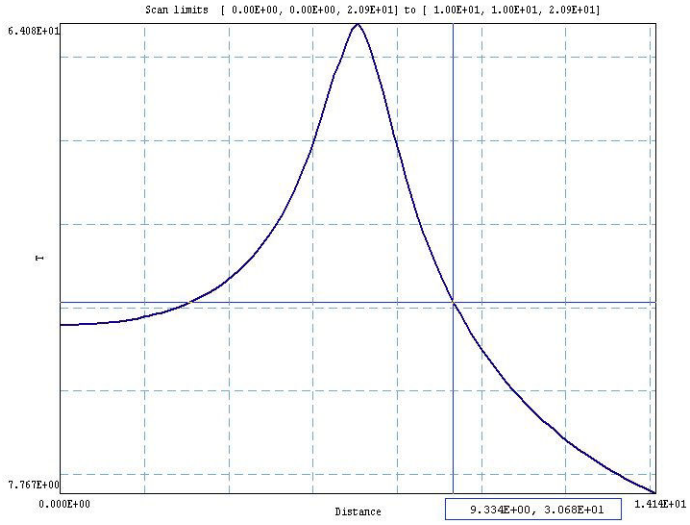


Figure 7: Scan plot in the oscilloscope mode.

Because a data file (`WALKTHROUGH.DAT`) has been opened, the *Analysis* submenu in the main menu is active. Choose the command *Analysis/Volume integral*. There will be a delay while **HWV** performs a complex set of volume integrations to find the average temperature. When the program has completed the task, click on the *File/Close data file* command and then choose the command *Edit data file*. Choose `WALKTHROUGH004.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 9 through 10 cover operation of the program in detail.

2 Organizing HeatWave calculations

2.1 Procedures and files

The **HeatWave** package contains three components:

> The program **HeatWave** computes the physical solution and generates data and history files.
>
> **HWV** is a dedicated post-processor for plotting and analyses of data-file results.
>
> **Probe** is a 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 **HeatWave** solution:

> A script that controls the run and defines the physical properties of materials.
>
> A **MetaMesh** output file (`MNAME.MDF`)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 **HeatWave** scripts must have names of the form `RUNNAME.WIN`. The program issues an error message if any input files are not available in the current working directory. To organize data, sequential data dumps are assigned names of the form `RUNNAME001.WOU`,`RUNNAME002.WOU`,... History files at multiple probe sites have names `RUNNAME.P01`, `RUNNAME.P02`,....

A calculation generally includes the following steps:

1. Using **Geometer** or a text editor, prepare a **MetaMesh** script with a name of the form `RUNNAME.MIN` that defines the solution space.

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

3. Using the *Setup* dialog or a text editor, prepare a **HeatWave** script (`RUNNAME.WIN`) that sets control parameters and defines the material properties of regions.

4. 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.

Table 5: **HeatWave** file types

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

5. Optionally, prepare a standard **AMaze** analysis script (`SNAME.SCR`) to control an analysis session in **HWV**.

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

7. Run **Probe** to plot and to analyze history files.

The procedure involves several steps and preparation effort. 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 may often be used in different solutions with small modifications. With a good editor, script preparation is usually faster than repetitive menu operations. Table 5 summarizes the input and output files used in **HeatWave**. Table 5 summarizes the input and output file used in **HeatWave**.

2.2 HeatWave script conventions

The **HeatWave** input script is a text file 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 may 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**:

```
* 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 may 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.

3 Script commands for run control

3.1 General control commands

This section reviews script commands 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

MESH MPrefix
MESH = XRayTarget
The prefix of the **MetaMesh** output file (`MPrefix.MDF`) to use for the definition of the solution geometry. If the command does not appear in a script `FPrefix.WIN`, **HeatWave** searches for the default file `FPrefix.MDF`.

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. 2. 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. In this mode the code may create several data and probe files in response to diagnostic commands.

PARALLEL [NProc]
PARALLEL = 4
This command invokes multithread support for the 64-bit program running on a multi-core machine. When the command appears with no value of $NProc$, **HeatWave** uses the full resources of the computer during the solution, giving a substantial reduction of the run time. Set a value of $NProc$ to assign a specific number of threads. If the command does not appear, the operating system assigns the program to a single thread, leaving the others free for other tasks. Do not employ parallel processing if you are running multiple instances of **HeatWave**. The command has no effect on the 32-bit program version.

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 Unit
DUNIT = MM
DUNIT = 100.0
Use this command to set the units used in **MetaMesh** to create the input mesh. The *Unit* argument may be one of the following strings for common units: `angstrom`, `nanometer`, `micrometer`, `mil`, `mm`, `cm`, `inch`, `foot`, `yard`, `meter`, `kilometer` or `mile`. For custom units, enter a real number equal to the number of mesh units per meter. For example, if you used dimensions of centimeters in the **MetaMesh** script, set $Unit = 100.0$. The quantity $DUnit$ is recorded in the output file and is used in **HWV** for the input and output of positions. Default: $Unit = 1.0$.

FORMAT [Text, Binary]
Format = Text
By default, **HeatWave** creates output files `FPREFIX.001` in binary format. This format can be loaded quickly and occupies minimal disk space. A binary output file is required if you want to use **HWV** to analyze the solution. Chapter 15 describes the binary file structure.We have included an option to create output files in text format to make it easier to port results to your own analysis programs. The string parameter may assume the values *Binary* or *Text*.

INTERP [SPLINE, LINEAR]
INTERP = Linear
This command controls the type of interpolation performed on the tables used to define temporal and temperature variations. Cubic spline interpolations 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*.

SOURCEFILE FileName [SFact]
SOURCEFILE = LiverProbe.ROU 2.5
Load a thermal source file from **RFE3** or **GamBet**. Chapter 6 describe this program function in detail.

SOURCEMOD TabFileName
SOURCEMOD > Function
Use this command to define a modulation function for thermal sources imported from **RFE3** or **GamBet**. The function determines time variations of the sources in dynamic runs. Details are covered in Chapter 6.

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.

RESTARGET ResTarg
RESTARGET = 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 $ResTarg$ is too low, the program may not converge because of roundoff errors. In this case, you can terminate a solution manually if you are running **HeatWave** in the interactive mode by using the *Stop* option. Default value: $ResTarg = 5.0 \times 10^{-7}$.

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

NCHECK NCheck
NCHECK = 20
The operations to calculate $ResTarg$ 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 *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.

AVG Alpha Nc [ATarg]
AVG = 1.00 8
The parameters α, N_c and A_{targ} are active in solutions where there is a temperature-dependent value of thermal conductivity k. In this case, **HeatWave** must perform two tasks: 1) solve the matrix inversion that represents the static thermal equation and 2) adjust conductivity values so that they are consistent with the computed temperatures. A dual iteration is necessary, and such solutions have longer run times. The code uses the following procedure:

1. Set values of k in nonlinear materials using the initial values of temperature in the elements. The temperature in a material is either 1) a value set with the *Init* command (Sect. 4.2) or 2) the default value of 0.0 °C.

2. Solve the thermal equation using the initial values of k. **HeatWave** stops when either 1) the number of cycles exceeds $MaxCycle$ or 2) the residual is less than $ResTarget$.

3. Use the new values of temperature to adjust k in variable materials using the formula $k' = \alpha k(T_{new}) + (1 - \alpha)k(T_{old})$.

4. Calculate the relative material adjustment factor A_r, the root-mean-squared value of $\Delta k/k$ and record the result in the listing file.

5. Terminate the calculation if $A_r \leq A_{targ}$ or the number of material adjustment cycles equals N_c. Otherwise, return to Step 2.

The adjustment parameter must be in the range $0.0 \leq \alpha \leq 1.0$. The choice $\alpha = 1.0$ is best if $k(T)$ varies smoothly. The number of cycles N_c determines the accuracy of the solution. The `WLS` file lists convergence information. Generally, A_r should be less than 10^{-3} for a good solution. If the parameter A_{targ} is omitted, the code runs for the full N_c cycles.

Usually, solutions will be successful if $k(T)$ has a smooth and simple variation with temperature. The are two circumstances where a solution may fail:

> The system as defined is physically unstable. For example, consider a case where a given thermal source Q heats a medium where k decreases with temperature. As Q increases, temperature in the medium rises non-linearly because of the decrease in conductivity. At some value, the temperature diverges to infinity. The interpretation is that runaway heating would lead to disassembly of a physical system.
>
> If k has a complex variation, there may be multiple equilibrium solutions. There is no guarantee that **HeatWave** will find a particular one. In the worst case, the code will be jump between solutions and the process will not converge.

In summary, you must ensure that material and system definitions are mathematically and physically valid when dealing with nonlinear materials.

3.3 Dynamic mode control commands

This section lists commands that control operation of HeatWave in the dynamic mode. Section 3.4 gives more details on setting the 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 stopping time t_{max} for the run. Enter the value in seconds. All runs begin at $t = 0.0$. The quantity t_{max} should be at least one time step larger than the latest time for recording a data file.

DTMIN DtMin
DTMIN = 0.001
Set a minimum for the time step, over-riding the automatic step selection routines. Enter the value in seconds. Default: $\Delta t_{min} = 1.0 \times 10^{-9}$ s.

DTMAX DtMax
DTMAX = 0.05
Set a maximum value for the time step, over-riding the automatic step selection routines. Enter the value in seconds. Default: $\Delta t_{min} = \infty$.

NSTEPMAX Ns
NSTEPMAX = 550
Set a maximum allowed number of time steps in the solution. The program stops when it reaches either t_{max} or N_s. This command may be used to ensure that solutions controlled by a batch file do not run out of hand. Default: $N_s = 100,000$.

SAFETY Fsafe
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: $F_{safe} = 25.0$.

NCHECK NCheck
NCHECK = 20
The parameter $NCheck$ is the interval (in time steps) between automatic time step adjustments. The parameter also equals the number of time steps between adjustment of materials with temperature-dependent thermal conductivity or specific heat. The choice of $NCheck$ strikes a balance between accuracy and run time. There is no simple prescription that covers all cases. In complex solutions, you must determine a good value by experimentation. The default is $NCheck = 10$.

3.4 Choosing a time step for a dynamic solution

HeatWave follows dynamic thermal transport by advancing in small time increments Δt. 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. 4). 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. Section 3.3 introduced three commands that allow you to modify the time step: *Safety*, *DtMin* and *DtMax*.

SAFETY FSafe

The automatic time-step routine estimates a maximum allowed time increment by averaging over the full solution volume and then divides by the factor F_{safe}. Generally, F_{safe} should exceed 5.0. In 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 DtMin

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 Δt by setting a minimum time step with this command. The automatic routines take over when the estimate of Δt rises above Δt_{min}. 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 Δt. In this case, the program would assign the small default value $\Delta t = 10^{-9}$ s resulting in a long run time.

DTMAX DtMax

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 Δt_{max}, 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 Δt. 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 *DtMax* command to check the fidelity of a solution. Run the solution with a coarse time step and then cut Δt 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.

4 Script commands for basic material properties

4.1 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.

Sections 4.2 and 4.3 describe commands to define simple materials for static and dynamic calculations. The following chapter covers 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 $Mat1 \Rightarrow Reg1$, $Mat2 \Rightarrow Reg2$,...

The **HeatWave** script should 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 the following sections 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.

4.2 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 retain the 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 T_{init} 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. The *Init* command has no effect on fixed-temperature nodes. Note that T_{init} is used to calculate initial values of k in temperature-dependent materials. (Default: $Tinit = 0.0$ °C for all materials.)

SOURCE MatNo Qs
SOURCE(2) = 5.0E6

A source power density is assigned to elements with a region number associated with material $MatNo$. Enter the value of Q_s in W/m^3. The power-density is uniform over the region. If you want to set a specific total power, you can find calculated region volumes near the end of the **MetaMesh** listing 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. Do not use a *Source* command 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-°C.

EMISS MatNo Epsi
EMISS(5) = 0.82

Elements associated with material $MatNo$ have an emissivity equal to ϵ. The number should be in the range $0.0 \leq \epsilon \leq 1.0$. Elements with non-zero emissivity emit thermal radiation through facets that border on a void material. The radiation power density is given by $p = \sigma(T + 273.15)^4$, where p is the power density in W/m^2, ϵ is the emissivity, σ is the Stefan-Boltzmann constant and T is the temperature in °C. Thermal radiation is discussed in Sect. 5.3.

VOID MatNo
VOID(6)

Elements with region number associated with material $MatNo$ have the *void* property. Void elements have zero thermal conductivity. Voids may be used to model internal insulators. They are also used to mark which surfaces of emissive elements ($\epsilon > 0.0$) radiate into free space. In this case, void elements must be located on the outer surface of the solution volume. Care should be exercised with internal voids. In **MetaMesh**, when the void is defined within a material region, nodes on the boundary are set to the void region number. Void nodes have undefined properties in the **HeatWave** calculation. Therefore, if the surrounding material has region number *MatRegNo*, include a command of the following form in the void part section:

```
COAT MatRegNo MatRegno
```

The command resets the region number of boundaries nodes to that of the encompassing material.

PERF MatNo Wb [Cpb Tb]
PERF(6) = 5.0 (3500.0, 38.0)
Set a fixed value for the perfusion fluid flow rate W_b within a material. Enter the value in kg/m^3-s. The optional parameters are the specific heat and enterting temperature of the perfusion fluid. If they are not specific, the defaults are the values for blood (C_{pb} = 3500.0 J/kg-°C, T_b = 37.0 °C.

4.3 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
EMISS MatNo Epsi
VOID MatNo

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

INIT MatNo Ti
INIT(3) = 37.0
Set the initial temperature of nodes with region numbers associated with material $MatNo$ to the value T_i (°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.

DENS MatNo Rho
DENS MatNo 968.0
A dynamic thermal solution requires specification of the thermal inertia of materials, ρC_p (where 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/m^3. To convert density values expressed in gm/cm^3, multiple 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.

PERF MatNo Wb [Cpb Tb]
PERF(6) = 5.0 (3500.0, 38.0)

Set a fixed value for the perfusion fluid flow rate W_b within a material. Enter the value in kg/m^3-s. The optional parameters are the specific heat and enterting temperature of the perfusion fluid. If they are not specific, the defaults are the values for blood (C_{pb} = 3500.0 J/kg-°C, T_b = 37.0 °C.

5 Advanced material properties

5.1 Temperature-dependent material properties

HeatWave can represent materials with reversible temperature-dependent properties. Any variation may be defined through flexible *tabular functions*. A tabular function is a text file consisting of up to 256 data lines – each line contains values for an independent and dependent variable. To define a temperature-dependent thermal conductivity, a data line contains a temperature value followed by the corresponding value of conductivity. You can prepare tabular function files with a text editor, spreadsheet or the *Table Generator* tool discussed in Sect. 8.2. You can also incorporate published data or digitized experimental traces. The example of Table 6 was created with a spreadsheet program. The intention was to model a sharp transition in 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 60. For spline interpolations a table requires a minimum of 5 entries. You must ensure that the tabular functions extend over the full range of temperature that will be encountered in the solution. The interpolation routines return 0.0 for values of the independent variable off the table. Off-table values may often be a source of error and confusion. Therefore, **HeatWave** writes a message in the listing file if any interpolation errors occur.

The interpolation method is controlled by the *Interp* command. By default **HeatWave** uses cubic splines. This method 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, as in the example of Table 6. 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 interpolated values. Use the linear interpolation method for data with a discontinuous slope.

Important note
Discontinuities of the dependent variable of tabular functions can result in non-convergent solutions. A smoothed curve gives a better numerical solution.

Table 6: Example of a tabular function file for temperature-dependent thermal conductivity

```
* TWAVE.CND
* 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
```

COND MatNo TABLE TabName
COND(5,TABLE) = BRAIN.CND
This command functions in either the static or dynamic modes. Elements with region numbers associated with material *MatNo* have a temperature-dependent thermal conductivity defined by values in the tabular file *TabName*. The file must be available in the working directory and have the 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). Note that although all elements associated with *MatNo* reference the same table, they will not have the same conductivity if they are at different temperatures.

CP(MatNo) TABLE TabName
CP(8,TABLE) = LIVER.CP
This command is valid only in dynamic solutions. It sets a temperature-dependent specific heat in elements with region number associated with material *MatNo*. The string *TabName* 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 may be incorporated in the C_p file.

5.2 Time variations of sources and fixed temperature

Tabular functions may also be used to define temporal variations for fixed-temperature conditions or thermal sources in dynamic solution. There are two difference from the tables discussed in the previous section: 1) the independent variable is time rather than temperature and 2) you can specify optional values to scale table values used in the program.

Important note
Temporal functions with discontinuities (*e.g.*, a square pulse) can result in invalid numerical solutions or computational artifacts. For the best numerical solution, use smooth transitions characteristic of physically realizable variations.

FIXED(MatNo) TABLE TabName [TimeAdj TempAdj]
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 *TabName*. The file must be available in the working directory and have the 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 calculations always begin at time $t = 0.0$. The optional parameters *TimeAdj* and *TempAdj* are scaling parameters. Values of time are multiplied by *TimeAdj* and values of temperature by *TempAdj* as they are entered in the program. In this way, you can construct a set of normalized functions that may be used

Table 7: Commands to define temporal or temperature variations

Command	IndVar	DepVar	Mode
COND	T (°C)	k (W/m-°C)	Both
CP	T (°C)	C_p (J/kg-°C)	Dynamic
FIXED	t (s)	T (°C)	Dynamic
SOURCE	t (s)	Q (W/m^3)	Dynamic

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

SOURCE MatNo TABLE TabName [TimeAdj SrcAdj]
SOURCE(7,TABLE) = NUCWEAP.SRC

Materials with specified time-dependent thermal sources may be defined in the dynamic mode. The keyword *Table* instructs the program to seek the file $TabName$. The file must be available in the working directory and have the 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/m^3). Note that simulations always begin at time $t = 0.0$. The optional parameters $TimeAdj$ and $SrcAdj$ are scaling parameters. Values of time are multiplied by $TimeAdj$ and values of power density by $SrcAdj$ 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. If the simulation time t exceeds the maximum time t_{max} of table entries, the interpolation time is taken as $t = \mathrm{mod}(t, t_{max})$. This feature is useful for defining periodic or pulse-train modulation functions.

Table 7 summarizes the **HeatWave** commands that employ tabular input.

5.3 Radiation boundaries

HeatWave can model heat loss by thermal radiation 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 may 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. 8, thermal radiation leaves elements with non-zero emissivity ($\epsilon > 0.0$) through facets that border on void elements. The power loss through a facet is given by

$$dP = \int_S \int \; dS \; \epsilon\sigma(T + 273.15)^4. \tag{4}$$

The surface integral in Eq. 4 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 always leaves

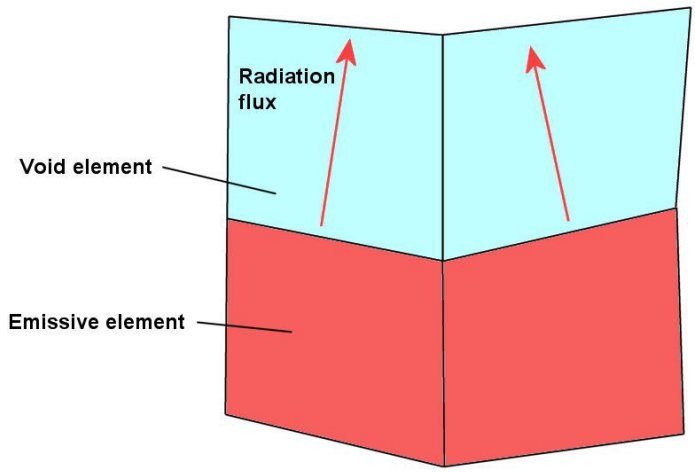


Figure 8: Definition of a radiation boundary in **HeatWave**.

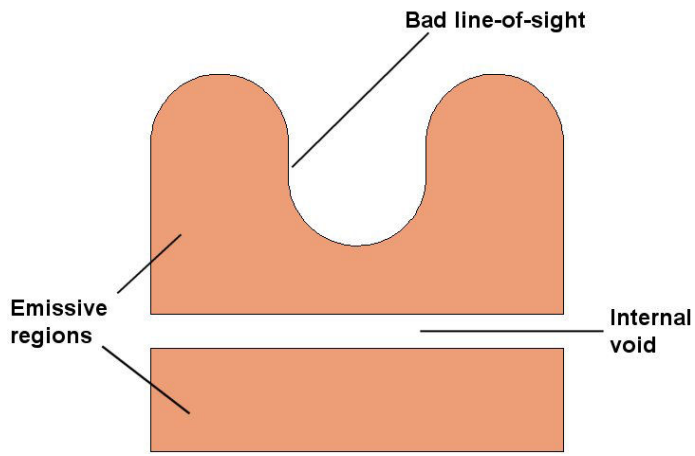


Figure 9: System that violates validity conditions for a **HeatWave** simulations with thermal radiation.

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 9 illustrates a system where the conditions are violated.

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

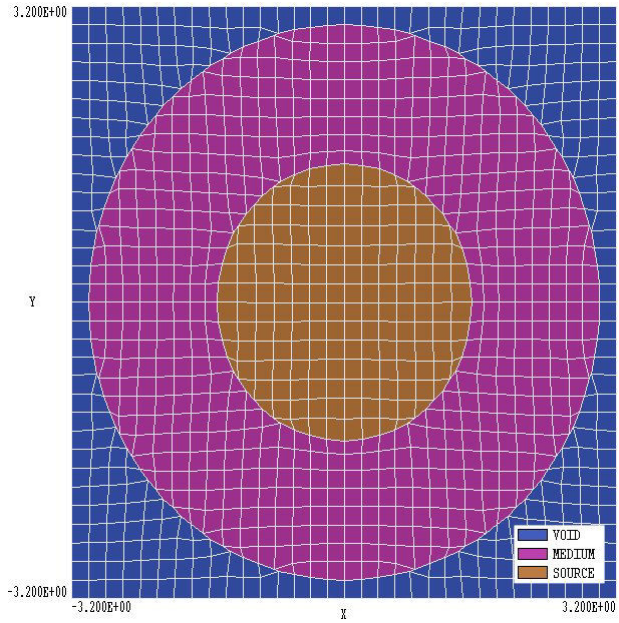


Figure 10: Mesh geometry for the RADBOUNDARYDEMO example – slice in the plane $z = 0.0$.

1. 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.

2. 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 updates 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 10 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} m^2. In equilibrium, the radiation flux is 1.25×10^5 W/m^2. The corresponding temperature predicted from Eq. 4 is 946.3 °C.

Table 8 shows the **HeatWave** control script. Region 1 (blue elements in Fig. 10) is defined as a void and Region 2 (violet elements) has $\epsilon = 1.0$. Therefore, radiation leaves 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 11 shows the evolution of the residual and maximum temperature in the solution volume as a

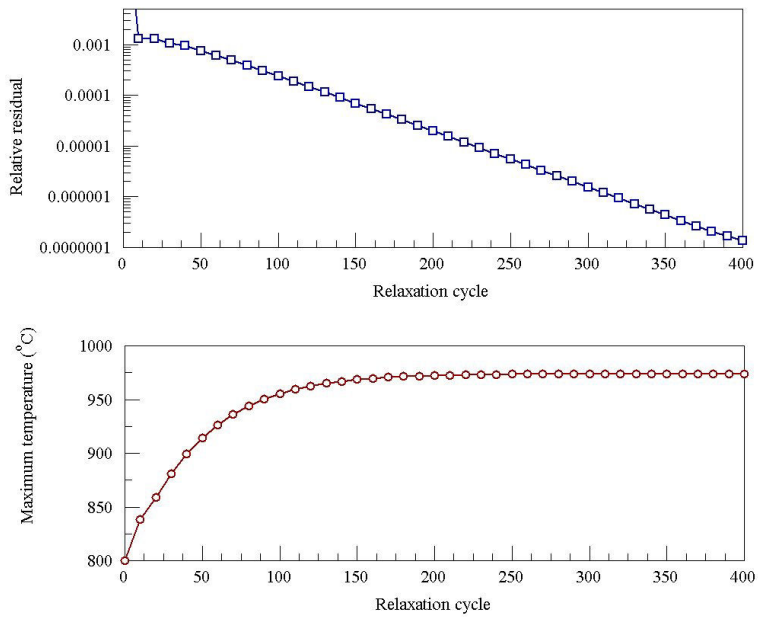


Figure 11: Variation of the relative residual and peak temperature during the relaxation solution for the example `RADBOUNDARYDEMO`.

Table 8: HeatWave script `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
```

function of relaxation step. The surface temperature determined by **HeatWave** is 943.3 /dg C, a difference of only 0.3% from the theoretical prediction.

6 Thermal source distributions from RFE3, GamBet and Aether

6.1 Function and commands

HeatWave can import spatially-distributed thermal sources generated by **RFE3**, **GamBet** and **Aether** (in the *RF* mode). **RFE3** calculates resistive power deposition by AC and RF electric fields. **GamBet** employs Monte Carlo methods to model transport of high-energy electron, photons and positrons in matter. The code creates profiles of dose and power deposition in materials. **Aether** calculates electromagnetic fields and the attendant power deposition in resistive materials.

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

SOURCEFILE FileName [SFact]
SOURCEFILE = LiverProbe.ROU 2.5

Load an output file from **RFE3**, **GamBet** or **Aether** to define thermal sources. The command is valid in both the static and dynamic modes of **HeatWave**. For **RFE3** and **Aether**, the same mesh must be used for both the field calculation and the thermal calculation. Similarly, the **GamBet** solution must employ the three-dimensional geometry mesh used for the **HeatWave** calculation. In either case, it may be necessary to include extra regions that assume different properties in the two calculations. The **RFE3** and **Aether** input files contain values for the thermal source (in W/m^3) in each element of the mesh. The **GamBet** file contains element dose rates that are converted to power density by multiplication by the material density ρ. In this case, **HeatWave** checks that the density values in the script are identical to the density values in the **GamBet** file. Power-density values are multiplied by the optional parameter *SFact* when loaded. In a dynamic solution, the source values may also be multiplied by a modulation function defined with the *SourceMod* command. Also, a time-dependent displacement of the source distribution in the solution volume may be specified with the *XDisp*, *YDisp* and *ZDisp* commands. Only a single *SourceFile* command may appear in a script. File sources may be combined with other types of thermal sources (region, perfusion or radiation).

SOURCEMOD TabFileName
SOURCEMODE = TenPulse.MOD

Use this command to assign a time-variation to a spatially-distributed thermal power source loaded with the *SourceFile* command. This command is valid only in dynamic calculations. The string *TabFileName* is the full name of a file that defines a modulation waveform. The file has the form of a tabular function (Chap. 5.2). It may contain up to 256 data lines – each line contains two numbers: t and $f(t)$. Enter the time t in seconds. (Note that all dynamic simulations start at $t = 0.0$.) The modulation function $f(t)$ is dimensionless and usually has values on the order of unity. The imported source values are multiplied by $f(t)$ to find the power-density distribution at the current time. Be sure to include the command *Interp Linear* if the modulation function has discontinuities of value or slope. The time variation does not affect independent region, perfusion or radiation sources. If the simulation time t exceeds the

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

SOURCEMOD > Function
SOURCEMODE > 10.0 + cos(3.1416*$t/25.6)
Define a modulation function for power density from a mathematical expression. The keyword *SourceMod* followed by the > symbol designates that a function string occupies the remainder of the line. The function may be up to 230 characters in length and follows the format described below. The function defines a variation in time, $f(t)$. The parser uses the Perl standard for the time variable: $\$t$ stands for t.

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

The time variable `$t`.

Real and/or integer numbers in any valid format (*e.g.*, `3.1415`, `476`, `1.367E23`, `6.25E-02`, `8.92E+04`,...). Integers are converted to real numbers for evaluation.

Binary operations: `+` (addition), `-` (subtraction), `*` (multiplication), `/` (division) and $\wedge$ (exponentiation).

Functions: `abs` (absolute value), `sin` (sine), `cos` (cosine), `tan` (tangent), `ln` (normal logarithm), `log` (base 10 logorithm), `exp` (normal exponent) and `sqt` (square root).

Up to 20 sets of parentheses to any depth.

Any number of space delimiters.

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

```
1 - exp(-1.0*(($t^2)/24))
```

corresponds to

$$1 - \exp\left[-\left(\frac{t^2}{24}\right)\right]. \tag{5}$$

XDISP TabFileName
XDISP = LinearScan.DAT
In some applications, it is useful to move a source distribution to different positions in the solution volume as time progresses. An example is the simulation of rotating X-ray target heating using input from **GamBet**. This command assigns a time-dependent source displacement from a tabular function file (Chap. 5.2). The file may contain up to 256 data lines – each line contains two numbers: t and $x(t)$. Enter the time t in seconds and the displacement

in units set by *DUnit*. In other words, $x(t)$ should be in mm if the command $DUnit = 1000$ appears in the input file. The time variation should start at $t = 0.0$ with $x(0) = 0.0$. If the motion is discontinuous, be sure to specify the option $Interp = Linear$ in the script. Note that the source translation operation is time consuming. Because thermal solutions tend to average variations, frequent small translations are unnecessary. Use the *NCheck* command (Sect. 3.3) to control the frequency of updates. Finally, the commands *YDisp* and *ZDisp* set motion in the y and z directions from independent tabular function files.

XDISP > Function
YDISP > Function
ZDISP > Function
XDISP > 25.0*$t

Source translations may also be controlled by algebraic functions using the conventions discussed for the *SourceMod* command. For the time in seconds, the function should give displacements in the units set by the *DUnit* command. Only one instance each of the *XDisp*, *YDisp* and *ZDisp* commands may appear in a HeatWave input file. Each instance may either use a tabular function file or algebraic function.

6.2 Application example

This section reviews a simple example that shows how to import power density files generated by **GamBet** into a **HeatWave** solution. The following input files are supplied in the example library:

`SOURCEGB.MIN`: specification of the mesh for the **GamBet** and **HeatWave** calculations representing a copper cube with sides of length 4.0 cm.

`SOURCEGB.PRT`: a set of primary electrons for the **GamBet** calculation. The file defines 611 electrons at the entrance face of the cube moving in the $+z$ direction. Electrons with kinetic energy 45 MeV are uniformly distributed over a beam of radius 0.5 cm. The beam current is 202.2 μA (total power, 10.0 kW). This file was created with the *Circular Beam Tool* of **OmniTrak**.

`SOURCEGB,GIN`: control file for **GamBet**. For good statistics, 10 showers are created for each primary electron.

`SOURCEGBBV.WIN`: control file for a static calculation in **HeatWave** to determine the steady-state temperature for a continuous beam.

`PULSE30.MOD`: modulation function to define a 30 second pulse.

`SOURCEGBIV.WIN`: control file for a dynamic calculation in **HeatWave**. The calculation gives time variations of temperature during and after the beam pulse.

The mesh volume (Region 1) consists of a single region with 64,000 uniform 1.0 mm cubic elements. Nodes on the solution volume are designated as Region 2. In the **GamBet** calculation, the boundary nodes have the same material identity as the volume. Figure 12 shows the

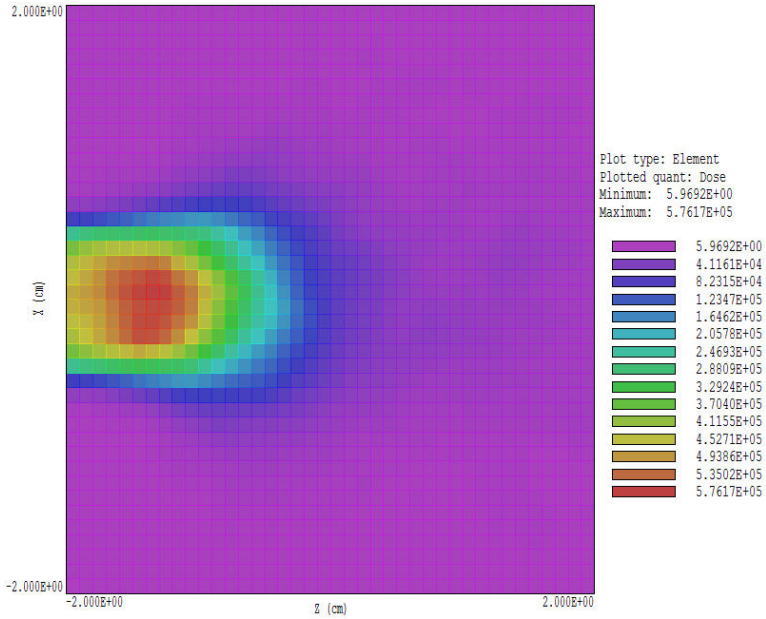


Figure 12: Dose distribution for the example `SOURCEGB` in the plane $y = 0.0$. Two cycles of smoothing have been applied.

dose distribution recorded in the **GamBet** output file `SOURCEGB.G3D`. A total of 7.781 kW is deposited in the medium. The remaining power is lost as escaping gamma rays and backscattered electrons. The peak dose (determined by electron scattering) occurs at a depth of about 0.6 cm. When the file is ported to **HeatWave**, the recorded dose in grays (J/kg-s) is multiplied by the material density to give power density in W/m^3.

Table 9 lists the **HeatWave** script for the static calculation. Region 1 has the thermal conductivity of copper, while Region 2 is set to the fixed condition $T = 0.0$ °C. The *SourceFile* command handles all details of loading and normalizing the **GamBet** output file. Figure 13 shows the resulting temperature distribution. Despite the large beam power, the high thermal conductivity of copper limits the peak temperature to about 162 °C. The peak occurs 0.8 cm from the entrance face, shifted from the position of peak dose because of the asymmetric thermal flux.

Table 10 shows the script for the dynamic calculation with $t_{max} = 51.0$ s. Values for the specific heat and density of copper must be supplied for the material of Region 1. The *SourceMod* command loads the file `PULSE30.MOD` to define time variation of all element power-density values derived from the **GamBet** solution. The entries in the modulation table equal 1.0 from $t = 0.0$ s to $t = 29.0$ s and 0.0 for $t > 31.0$ s. The linear interpolation option must be used because of the data discontinuity. The diagnostic commands set a probe at the position of peak temperature and record spatial distributions when the solution has approached the steady-state temperature distribution ($t = 25.0$ s) and during the cool-down period ($t = 35.0$ s). The probe output is illustrated in Fig. 14. The spatial data files show a shift in the temperature distribution after the end of the beam pulse. The initially skewed distribution of Fig. 13 becomes symmetric within the cube at long times.

Table 9: Contents of the file SOURCEGBBV.WIN

```
* ---- CONTROL ----
Mesh = SourceGB
Mode = Static
DUnit =   1.0000E+02
ResTarget =   1.0000E-07
MaxCycle =    5000
SourceFile = SourceGB.G3D
* ---- MATERIAL PROPERTIES ----
* Material 1
Cond(1) =   410.0
* Material 2
Fixed(2) =   0.0000E+00
* ---- REGION ASSIGNMENTS ----
* Region 1: COPPER
Region(1) =   1
* Region 2: BOUNDARY
Region(2) =   2

EndFile
```

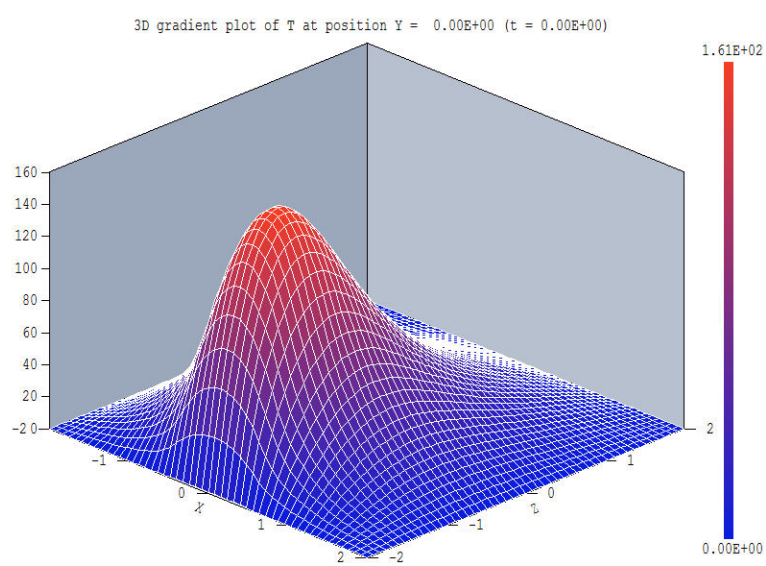


Figure 13: Steady-state temperature distribution in the plane $y = 0.0$ for the example SOURCEGBBV.

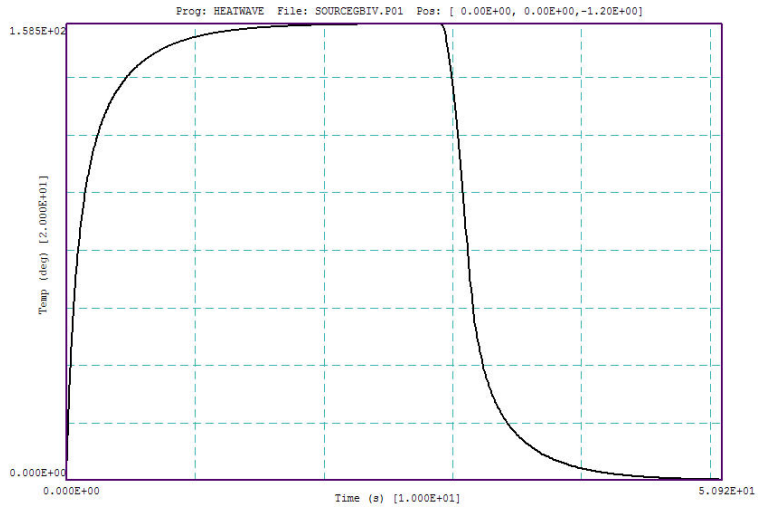


Figure 14: Time-dependent temperature recorded by a probe near the position of peak temperature [0.00, 0.00, -1.20], example `SOURCEGBIV`.

Table 10: Contents of the file `SOURCEGBIV.WIN`

```
* ---- CONTROL ----
Mesh = SourceGB
Mode = Dynamic
DUnit =   1.0000E+02
TMax = 51.0
DtMin = 0.001
DtMax = 0.1
Interp = Linear
SourceFile = SourceGB.G3D
SourceMod = Pulse30.MOD
* ---- MATERIAL PROPERTIES ----
* Material 1
Cond(1) =   410.0
Cp(1) = 3850.0
Dens(1) = 8960.0
* Material 2
Fixed(2) =   0.0000E+00
* ---- REGION ASSIGNMENTS ----
* Region 1: COPPER
Region(1) =   1
* Region 2: BOUNDARY
Region(2) =   2
* ---- DIAGNOSTICS ----
* Near maximum temperature point
History = (0.0, 0.0, -1.2)
SetTime = 25.0
SetTime = 35.0

EndFile
```

7 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 defines 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 immediately 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
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 T_{max} 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 times set to infinity.

NSTEP NStep
NSTEP = 100
In response to this command, **HeatWave** creates data files at a uniform interval of time 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.

NHSTEP Ns
NHSTEP = 10
In runs with a large number of time steps, the history files may be larger than necessary and difficult to export to other programs. Use this command to set the number of time steps per data record in the history files. The default value is $N_s = 1$.

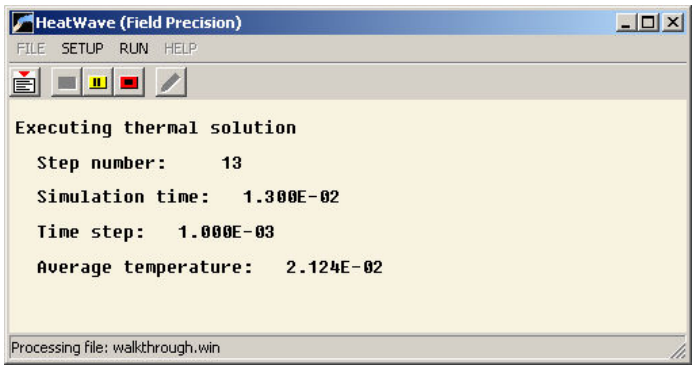


Figure 15: **HeatWave** display during a solution.

8 Running the HeatWave program

8.1 Interactive operation

The program `heatwave.exe` can 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 you own shortcuts. Figure 15 shows the program window.

The program menu has four main commands: *File*, *Setup*, *Run* and *Help*. The following commands appear in the *File* popup menu.

EDIT SCRIPT
EDIT LISTING FILE
EDIT FILE

The commands call up the internal editor to inspect or to modify text input and output files for the solution program. With the *Edit script* command you can work on files with names of the form `FPREFIX.HIN`. With the *Edit listing file* command you can pick files with names of the form `FPREFIX.HLS`. The *Edit file* command shows all available files. Choosing a file from an alternate directory does not change the working directory of the program. Note that the main program will be inactive until you exit the editor.

SETUP

For this command, **HeatWave** prompts for the name of a **MetaMesh** file and brings up the dialog of Fig. 3. Fill in values for control parameters and material properties to create a basic script. You can use an editor to modify the script or to add the advanced functions described in Chaps. 3, 4 and 5.

The *Run* menu has three commands.

START RUN

Pick an input file with a name of the form `FPREFIX.WIN` to start a solution. The working directory changes if you pick a file from an alternate directory. The run begins if the requested **MetaMesh** file `MPREFIX.MDF` is available in the working directory. **HeatWave** displays information in the window to show the progress of program operations (Fig. 15).

PAUSE RUN

The intensive calculations of **HeatWave** make demands on the resources of your computer, possibly causing other tasks to run slowly. If you need to perform critical work, you can pause the solution program during the integration or relaxation process and restart it later without loss of data. Note that if you have a dual-processor machine, the Windows operating system will shunt tasks so that you have approximately the power of one processor when **HeatWave** is running. To use the full power of the machine, you can launch two instances of **HeatWave** to perform two independent calculations.

STOP RUN

This command terminates the program during the relaxation process and saves available output data. For example, in a static run you may want to stop at a moderate value of convergence to check whether the problem has been correctly defined.

The *Help* menu has two commands.

HEATWAVE MANUAL

The command displays this manual using your default PDF viewer. The file `heatwave.pdf` must be in the same directory as `heatwave.exe`.

TABLE GENERATOR

Run the utility described in the next section to create a tabular function from a mathematical expression.

8.2 Table generation tool

This tool provides a quick way to create tabular function files to represent variations with temperature or time from mathematical functions. Click on the command *Help/Table generator* to bring up the dialog of Fig. 16. Supply information in the following four fields:

Table file name. Enter a name for the file, such as `LIVER.CND`. The string may contain up to 36 characters.

x(min) and x(max). The range of the independent variable (real numbers).

Function (f(x). An algebraic function of the variable `x` following the rules discussed below. The string may contain up to 255 characters.

NEntry. The number of data lines in the table. The maximum value is 256.

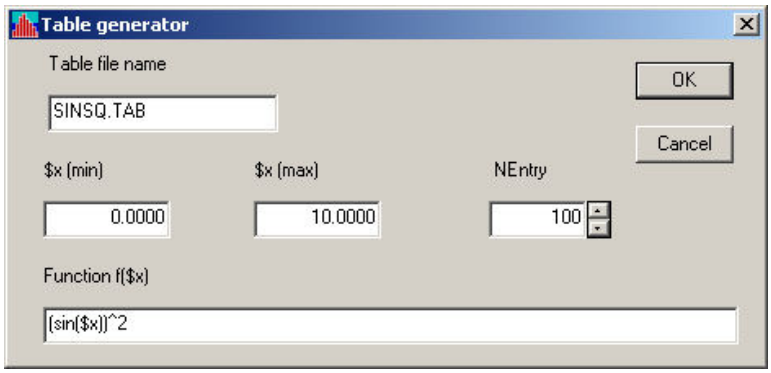


Figure 16: Table generation tool

When you click *OK*, the table generator opens a text file, writes a heading and adds *NEntry* data lines, [`x`, `f( x)`]. The values are evenly spaced along `x` from `x(min)` to `x(max)`. Finally, the table generator appends *ENDFILE*, closes the file and then closes the dialog.

A function is a string (up to 255 characters) that may include the following entities:

The independent variable `x`.

Real and/or integer numbers in any valid format (*e.g.*, `3.1415`, `476`, `1.367E23`, `6.25E-02`, `8.92E+04`,...). Integers are converted to real numbers for evaluation.

Binary operations: `+` (addition), `-` (subtraction), `*` (multiplication), `/` (division) and $\wedge$ (exponentiation).

Functions: `abs` (absolute value), `sin` (sine), `cos` (cosine), `tan` (tangent), `ln` (normal logarithm), `log` (base 10 logorithm), `exp` (normal exponent) and `sqt` (square root).

Up to 20 sets of parentheses to any depth.

Any number of space delimiters.

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

```
1 - exp(-1.0*(($z^2 + $r^2)/24))
```

corresponds to

$$1 - \exp\left[-\left(\frac{z^2 + r^2}{24}\right)\right]. \tag{6}$$

The entries in the dialog shown in Fig. 16 give the file shown in Table 11.

Table 11: Extract from the file SINSQ.DAT

```
* Field Precision Table Generation Tool
*   Number of entries:  100
*   XMin:   0.0000E+00
*   XMax:   1.0000E+01
*   Function: (sin($x))^2
     0.0000E+00  0.0000E+00
     1.0101E-01  1.0168E-02
     2.0202E-01  4.0260E-02
     3.0303E-01  8.9051E-02
     4.0404E-01  1.5456E-01
     5.0505E-01  2.3411E-01
     ...
     9.7980E+00  1.3293E-01
     9.8990E+00  2.0852E-01
     1.0000E+01  2.9596E-01
ENDFILE
```

8.3 Automatic runs under batch file control

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. This feature is particularly attractive on dual-processor machines.

To run a single **HeatWave** calculation 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 RUNAWAY.MDF and RUNAWAY.HIN are stored in \AMAZE\BUFFER and that the program heatwave.exe is in the directory \AMAZE. From \AMAZE\BUFFER type

```
..\HEATWAVE RUNAWAY <Enter>
```

The program runs silently, writing detailed information in the listing file SWITCH.WLS. If the solution is successful, the program creates output files in the data directory.

The main function of the command mode is autonomous operation under batch file control. As an example, suppose you have prepared the input files METAB01.MIN,...,METAB08.MIN and METAB01.HIN,...,METAB08.HIN in the directory \AMAZE\BUFFER. Next you create the following batch file MBRUN.BAT in the data directory using a text editor:

```
@ECHO OFF
ECHO Main switch data run
START /WAIT ..\METAMESH.EXE METAB01
START /WAIT ..\HEATWAVE.EXE METAB01
START /WAIT ..\METAMESH.EXE METAB02
START /WAIT ..\HEATWAVE.EXE METAB02
 ...
START /WAIT ..\METAMESH.EXE METAB08
START /WAIT ..\HEATWAVE.EXE METAB08
```

Type

```
SWRUN <Enter>
```

to generate all solutions without the need for further keyboard input. If there are data dependences, use the option `/WAIT` to ensure that each task completes before the next one starts.

9 HWV – file operations and plane plots

9.1 File operations

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. **HWV** counts the number of files in the series, records the times and displays the dialog of Fig. 17. Pick a solution file to load and click *OK*. Note that there will be only one entry in the list at $t = 0.0$ for a static solution.

LOAD SOLUTION

Load a different solution file in a series. **HWV** displays the dialog of Fig. 17. Pick a different solution and pick OK. This command is active only when a series has been specified.

SOLUTION FILE INFORMATION

The command shows a message box with 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. Analysis script operation is a powerful feature of **HWV**. This command displays a dialog with a list of analysis scripts (suffix `SCR`) that you have created. Pick a file and click *OK*. Changing directories in the dialog changes the working directory of the program. The analysis script can load data files, open and close history files, and perform any of the numerical functions described in this manual. Section 10.2 reviews the analysis script language.

CREATE SCRIPT

Use this command to create scripts using the internal editor. A 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 the reference list of allowed commands shown in Table 12. Enter commands in the space above *EndFile*. After saving the file, you can run it using the *Run script* command.

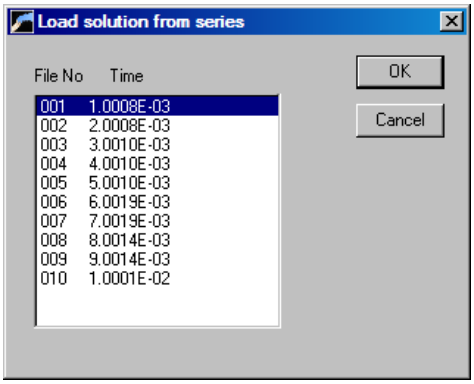


Figure 17: Dialog to pick a solution file from a series

Table 12: Create script - default file content

```
* AMaze script file
* Insert commands here...
ENDFILE

      --- Script command summary ----

INPUT FileName
 [Close current solution file and load FileName]
OUTPUT FPrefix
 [Close current data file and open FPrefix.DAT]
NSCAN  100
 [Set the number of points in a line scan]
POINT xp yp zp
 [Point field calculation at the given coordinates]
LINE xp1 yp1 zp1 xp2 yp2 zp2
 [Scan along a line between the given coordinates]
FULLANALYSIS
 [Write analyses for all region to the data file]
REGION RegNo
 [Write an analysis for region RegNo to the data file]
MATRIX FileName XMin XMax NX YMin YMax NY ZMin ZMax NZ
 [Write a matrix of field values to the file FileName]
ENDFILE
 [Terminate the analysis]
```

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 text data file has a name of the form `FPREFIX.DAT` and will be stored in the working directory. You can use an editor to view the file or to extract information for 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

View or modify files with names of the form `FPREFIX.DAT`.

EDIT FILE

Use the program editor to view or to modify any text file.

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

INSTRUCTION MANUAL

Displays this document in your default PDF viewer. The file `heatwave.pdf` must be in the same directory as `hwv.exe`.

9.2 Plane plots

Plane plots (Fig. 18) are two-dimensional plots that show the variation of quantities over a plane normal to one of the Cartesian axes. Plane plots provide simple and quick views of the solution space. The technique is to generate a rectangular mesh of values over a specified planar region and then to create plots in a variety of styles. No attempt is made to connect the plot mesh with the conformal mesh of the simulation. Slice plots, discussed in the next chapter, are also two-dimensional in a plane normal to an axis. The difference is that slice plots are built using the computational mesh. They show precise region boundaries and equipotential lines in true scale. Slice plots are more difficult to construct than plane plots.

SET PLANE

This command brings up the dialog of Fig. 19 to set the plane for the plot. Set the normal axis

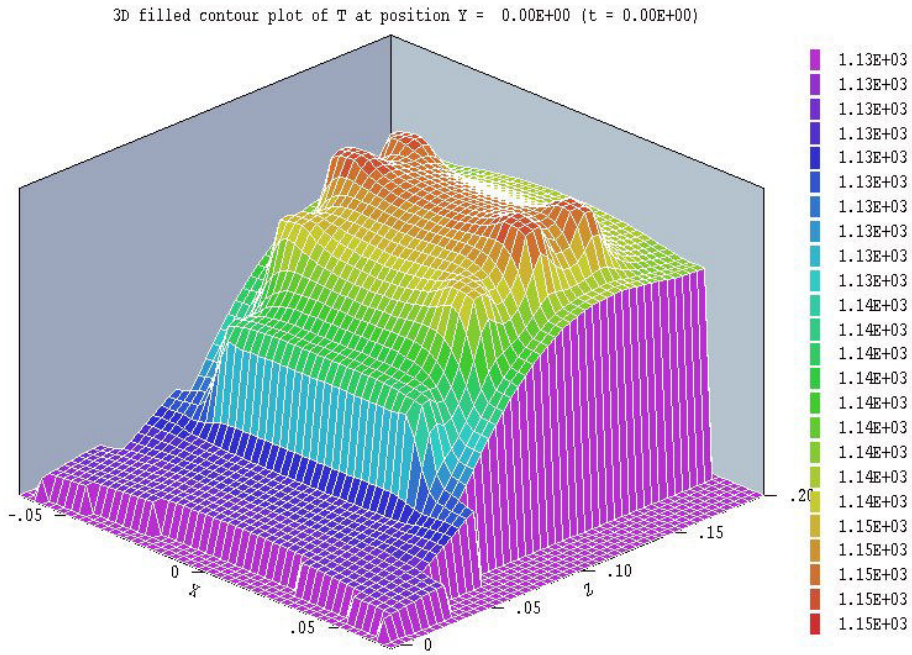


Figure 18: Illustration of a plane plot – *3D filled contour* type.

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.

The following commands are in the *Plot control* popup menu:

PLOT STYLE

This command brings up the dialog of Fig. 20 to set the plot style. As an example, Fig. 18 shows the *Filled contours 3D* plot style. The numbers at the bottom 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 ^{o}C.

$|\mathbf{F}|$, F_x, F_y and F_z: amplitude and components of the thermal flux in W/m^2.

Q: thermal source in W/m^3. This plot is useful mainly when spatially-dependent thermal information is imported from **RFE3** or **GamBet**.

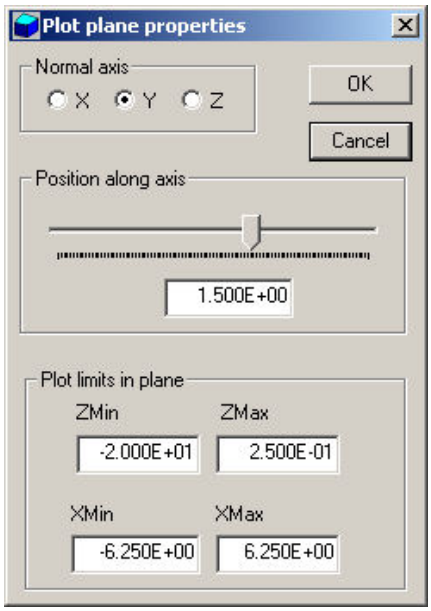


Figure 19: Set plane dialog.

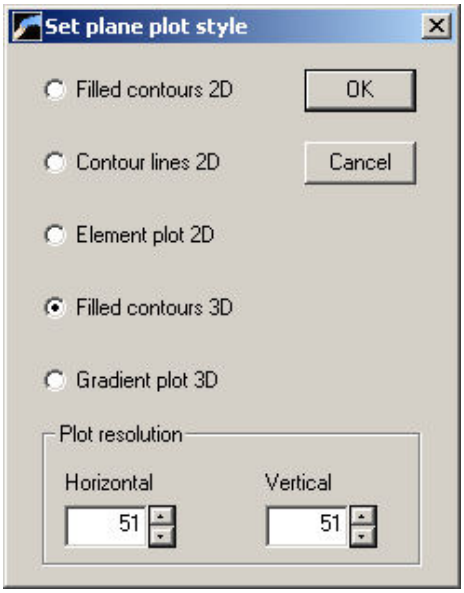


Figure 20: Plot style dialog.

k: thermal conductivity in W/m-°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 simulations that include materials with temperature-dependent specific heat.

PLOT LIMITS

Set limits for the plotted quantity. When *Autoscale* is active, **HWV** automatically sets limits based on the range of values in the solution file. To plot an absolute range, uncheck the box and enter numbers for the minimum and maximum values. In either case, the program shows lines in plots of type *Contour lines 2D* at a uniform interval. You can set values for contour lines manually. See Sect. 11.2 for information.

ROTATE PLOT

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

The commands of the *Export plot* menu are used to generate hardcopy or to create plot files.

DEFAULT PRINTER

With this command, a **HWV** 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 Settings command of Windows before issuing the command.

SAVE PLOT FILE

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

COPY TO CLIPBOARD

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

9.3 Saving and loading views

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

SAVE NAMED VIEW

Save the view parameters for the current plot. Quantities such as the slice axis, slice position

and zoom limits are saved for two dimensional plots. Parameters for three-dimensional plots include the viewpoint position, displayed regions and cut planes. The information is stored in a text file in the current directory with a name of the form `FPREFIX.FPV`.

LOAD NAMED VIEW

Load a view file and refresh the plot. Note that you must be in the appropriate plot menu to retrieve a view. Views of plane plots must be loaded to the Plane Plot Menu.

The file contains the complete set of plot parameters. This excerpt illustrates the format:

```
Program: AMaze
2D/3D: 2D
DisplayBy: Regions
Outline: On
NSlice:     40
SliceAxis: YAxis
PlotType: LogElemUp
XPMin: -1.500000E+00
XPMax:  4.250000E+00
...
```

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

Dynamic adaptation to different solutions.

Option for user control of the view parameter set.

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

You can modify view files with an editor. The order of entries is not rigid. On input, **HWV** uses a free-form parser. If a parameter is missing, the program simply makes no change from the value current in the program. The implication is that you can modify a saved view to include only elements essential to your application. For example, you could compare a series of assemblies with different sizes, maintaining an orthographic 3D view from the same point in Cartesian space. In this case, the view file would contain only the entries:

```
DView:  1.000000E+37
R11:  8.660253E-01
R12: -5.000002E-01
R13:  0.000000E+00
R21:  1.669031E-01
R22:  2.890846E-01
R23:  9.426408E-01
R31: -4.713208E-01
R32: -8.163510E-01
R33:  3.338061E-01
```

10 HWV – numerical analysis

The commands of the *Analysis* popup menu are used to determine numerical values for the solution. In the interactive mode, the menu is active only when a data file is open.

10.1 Automatic analysis operations

POINT CALCULATION

The *Point calculation* command brings up a dialog where you enter the x, y and z coordinates of any point in the solution space. In contrast to the *Point calculation* command of the *Slice plots* menu, the point is not limited to a slice plane. When you click *OK*, **HWV** displays an information box and also writes the results to the data file in the following format:

```
   --- Point Thermal Analysis ---
 Position: [  1.0000E+00,   1.0000E+00,   1.5000E+01]
 Region number:   1
    T:   4.6124E+01 (deg-C)
   Fx:  -3.9657E+02 (W/m2)
   Fy:  -3.9657E+02 (W/m2)
   Fz:   7.8387E+02 (W/m2)
  |F|:   9.6384E+02 (W/m2)
    K:   5.0000E-01 (W/m2-degC
   Cp:   4.8000E+03 (J/kg-m3)
    Q:   0.0000E+00 (W/m3)
 Number of data points: 32
 Standard deviation:   2.2214E-03
```

The set of values depends on the quantities defined in the `INTERPOLATION` section of the currently-loaded configuration file.

LINE SCAN

In the *Analysis* menu, scans may be performed along arbitrary straight lines between any two points in the solution volume. When you click the *Line scan* command a dialog appears where you can specify the start and end points. Enter values in units set by *DUnit*. In other words, if you used dimensions of cm in **MetaMesh** and the **HeatWave** script contains the command $DUnit = 100.0$, then enter the dimensions in cm. In the interactive mode, **HWV** creates a screen plot of a selected quantity determined by the *Set scan quantity* command. The command has no effect on the data file listing which contains all calculated quantities. Features of the scan plot environment are discussed in Sect. 11.3. Table 13 shows an example of a listing.

VOLUME INTEGRALS

In response to the command, **HWV** performs volume integrals of quantities defined in the

Table 13: Example of a line scan listing

```
------- Scan between points -------
   XStart:   1.0000E+00   YStart:   1.0000E+00   ZStart:   2.0000E+00
     XEnd:   1.0000E+00     YEnd:   1.0000E+00     ZEnd:   1.6000E+01

        X           Y           Z           T          Fx          Fy
==========================================================================
   1.0000E+00  1.0000E+00  2.0000E+00  4.2768E+01 -3.7302E+02 -3.7302E+02
   1.0000E+00  1.0000E+00  2.1400E+00  4.3052E+01 -3.7492E+02 -3.7492E+02
   1.0000E+00  1.0000E+00  2.2800E+00  4.3330E+01 -3.7681E+02 -3.7681E+02
   1.0000E+00  1.0000E+00  2.4200E+00  4.3602E+01 -3.7871E+02 -3.7871E+02
   1.0000E+00  1.0000E+00  2.5600E+00  4.3869E+01 -3.8106E+02 -3.8106E+02
   1.0000E+00  1.0000E+00  2.7000E+00  4.4128E+01 -3.8222E+02 -3.8223E+02
   1.0000E+00  1.0000E+00  2.8400E+00  4.4381E+01 -3.8339E+02 -3.8339E+02
   1.0000E+00  1.0000E+00  2.9800E+00  4.4628E+01 -3.8455E+02 -3.8455E+02
   1.0000E+00  1.0000E+00  3.1200E+00  4.4869E+01 -3.8553E+02 -3.8553E+02
   1.0000E+00  1.0000E+00  3.2600E+00  4.5103E+01 -3.8608E+02 -3.8608E+02
   1.0000E+00  1.0000E+00  3.4000E+00  4.5330E+01 -3.8663E+02 -3.8663E+02
   1.0000E+00  1.0000E+00  3.5400E+00  4.5551E+01 -3.8707E+02 -3.8706E+02
   1.0000E+00  1.0000E+00  3.6800E+00  4.5765E+01 -3.8714E+02 -3.8714E+02
   ...

        Fz          |F|          K           Cp           Q      NReg
 ==================================================================
  -1.0250E+03  1.1528E+03  5.0000E-01  4.8000E+03  0.0000E+00    1
  -1.0037E+03  1.1352E+03  5.0000E-01  4.8000E+03  0.0000E+00    1
  -9.8240E+02  1.1176E+03  5.0000E-01  4.8000E+03  0.0000E+00    1
  -9.6109E+02  1.1002E+03  5.0000E-01  4.8000E+03  0.0000E+00    1
  -9.3791E+02  1.0817E+03  5.0000E-01  4.8000E+03  0.0000E+00    1
  -9.1545E+02  1.0631E+03  5.0000E-01  4.8000E+03  0.0000E+00    1
  -8.9299E+02  1.0447E+03  5.0000E-01  4.8000E+03  0.0000E+00    1
  -8.7052E+02  1.0264E+03  5.0000E-01  4.8000E+03  0.0000E+00    1
  -8.4658E+02  1.0070E+03  5.0000E-01  4.8000E+03  0.0000E+00    1
  -8.2371E+02  9.8824E+02  5.0000E-01  4.8000E+03  0.0000E+00    1
  -8.0084E+02  9.6970E+02  5.0000E-01  4.8000E+03  0.0000E+00    1
  -7.7714E+02  9.5057E+02  5.0000E-01  4.8000E+03  0.0000E+00    1
  -7.5444E+02  9.3217E+02  5.0000E-01  4.8000E+03  0.0000E+00    1
  ...
```

VOLUME section of the configuration file over all elements of the mesh. The calculations are intensive, so there may be a delay for large meshes. **HWV** records results to the data file in the following form:

```
---------- Volume Integrals ----------
   Global volume:   2.812500E-05
   RegNo   Region volume
   ========================
       1  2.809154E-05
       2  9.707937E-09
       3  2.375377E-08

Quantity: Source
   Global value:   1.747416E_03
   RegNo   Region volume
   ========================
       1  0.000000E+00
       2  0.000000E+00
       3  1.747416E+03
```

The first table always shows the global and region volumes. Additional tables list results for quantities defined in the VOLUME section of the configuration file. In the standard file, the calculated quantity is the defined thermal sources in a region. These values are useful to check source input from **RFE3** and **GamBet**.

SURFACE INTEGRALS

In response to this command, **HWV** takes integrals of defined quantities over the selected region surfaces. The program displays the dialog of Fig. 21 which includes a row for each region along with the name assigned in **MetaMesh** and forwarded by **HiPhi**. Activate the check boxes in the *Internal* column to specify the regions to include in the set. The default is to include the full exterior surface of the region set (*i.e.*, all remaining regions are *External*). Use the check boxes to specify *External* regions. For example, to take a surface integral over the boundary between Regions 2 and 5, set Region 2 as *Internal* and Region 5 as *External*.

The calculated quantities are defined in the SURFACE section of the configuration file. In the standard configuration, the surface integral gives the total thermal flux out of the region set:

$$Q = \int\int_S k\nabla\mathbf{T}\cdot\mathbf{n}\, dA. \tag{7}$$

PEAK/AVERAGE TEMPERATURE

This command initiates volume integrals of temperature weighted by volume to find the global and regional temperature averages and peak values. Information is recorded to a data file in the form:

```
---------- Peak and average temperature analysis ----------
Volume (global):   2.812500E-05
Minimum temperature (global):   1.468349E-07
Maximum temperature (global):   1.000000E+02
```

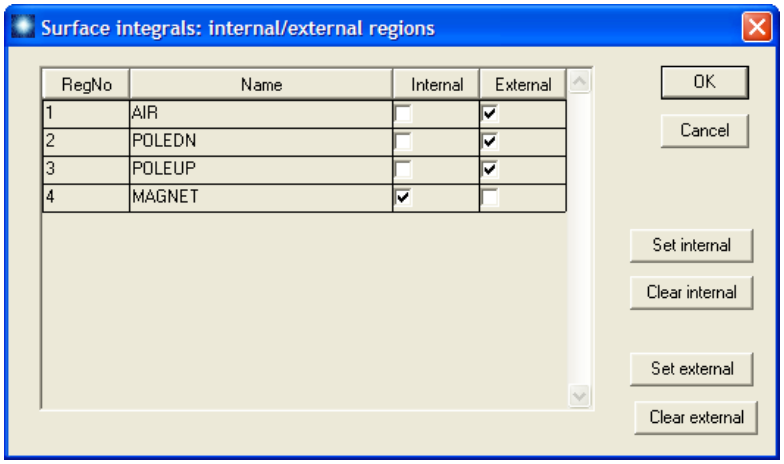


Figure 21: Dialog to pick the region set for surface integrals.

```
Average temperature (global):   6.324696E+00

 RegNo      Volume         Temp(Min)      Temp(max)      Temp(Avg)
==============================================================
     1  2.809154E-05  1.468349E-07  9.925679E+01  6.222838E+00
     2  9.707937E-09  4.680619E+01  9.757219E+01  7.185667E+01
     3  2.375377E-08  1.000000E+02  1.000000E+02  1.000000E+02
     4  0.000000E+00  1.000000+100 -1.000000+100  0.000000E+00
```

MATRIX FILE

The *Create matrix file* command controls a feature that is useful if you want to write your own analysis routines or port results to mathematical software. In response to the command, **HWV** performs interpolations over a specified box region on a regular grid of values. It is much easier to use results in this form than to deal with the conformal mesh directly. Clicking on *Create matrix file* calls up the dialog of Fig. 13.2. Specify the dimensions of the box along each axis (in units set by *DUnit*) and the number of calculation intervals. To illustrate, calculations are performed at positions with x coordinates given by

$$x = x_{min} + \frac{n\ (x_{max} - x_{min})}{n_x}, \tag{8}$$

where $n = 0, 1, 2, ..., n_x$. For example, if you set $x_{min} = 0.5$, $x_{max} = 1.5$ and $n_x = 10$, the calculations are performed at points with $x = 0.5$, 0.6, ..., 1.4, 1.5. You can also specify an output file name. Depending on the value of *NRecord* in the configuration file (Sect. 13), the listing may contain all quantities defined in the `INTERPOLATION` section. If you want a two-dimensional matrix of values in a single plane, set either N_x, N_y or N_z equal to zero. For example, if $N_x = 0$, then a set of calculations is performed over the specified range of y and z at the single position $x = (x_{min} + x_{max})/2.0$.

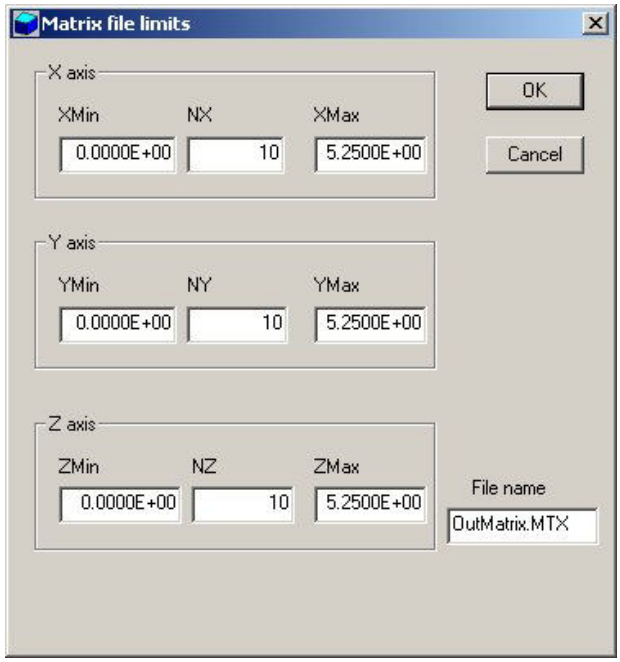


Figure 22: Create matrix file dialog.

10.2 HWV analysis scripts

HWV scripts are useful for automating complex or repetitive analysis procedures. For example, you may want to compare a line scan of values between two points in a set of similar solutions. In the interactive mode, it would be necessary to type in the same coordinates for each scan. An analysis script can load each solution file, perform the scan and record all information in a single data file. In the **HWV** main menu, the following commands in the *File* popup menu (Sect. 9.1) apply to analysis scripts.

RUN SCRIPT

Pick and run a script file with a name of the form `FPREFIX.SCR`. The script can operate on the currently-loaded solution file or load a different file. The last solution file loaded by the script becomes the current file for plots and interactive analysis.

CREATE SCRIPT

Use this command to create an analysis script using the internal program editor.

EDIT SCRIPT

Use this command to change an existing script with the internal program editor.

HWV may be called as a command line task. The calling syntax is

```
[progpath\]HWV [scriptpath\]SPREFIX
```

where the script has the file name `SPREFIX.SCR`. You should supply the full path `[scriptpath]` if the script is not in the current directory.

Command line operation opens several possibilities.

> **HWV** runs quickly and silently in the background when laucnhed from the *Command Prompt* in a terminal window. To minimize typing, open the terminal in the working directory. Use the DOS `CHDIR` command to change directories.
>
> The program may be called by DOS batch files with commands of the form
>
> ```
> START /wait [progpath]\HWV [scriptpath\]SPREFIX
> START [progpath]\HWV [scriptpath\]SPREFIX
> ```
>
> Use batch files to organize calculations that run autonomously. The option `/wait` in the first example signals that the batch file waits for completion of the program to continue. This form is useful if data from **HWV** is required for the next operation. Omit the option if you want to launch several instances of **HWV** to take advantage of a multiple-core machine. Use the `CHDIR` command to set the working directory.
>
> Run **HWV** as a subtask from your own compiled or interpreted programs. One application is to integrate finite element solutions and analysis into an optimization loop.

10.3 Script commands

Entries on a command line may be separated by the following set of delimiters: space, tab, equal sign [=], colon [:], left parenthesis [(] or right parenthesis [)]. Each command listed below is shown in symbolic form along with an example of how they might appear in a script.

INPUT [datapath\] FileName
INPUT = E:\FEData\AprilRuns\HVCross.WOU
Close the currently-opened solution file and load a solution file from the current directory. If `[datapath]` is not specified, the solution file must be in the working directory.

OUTPUT [datapath\] FileName
OUTPUT: WGUIDE1.DAT
Close the current data record file and open a new one. If a `[datapath]` is not specified, the file will be written in the working directory. A data file must be opened to use the following analysis commands.

CONFIGURATION [datapath\] FileName
CONFIGURATION = \FieldP\AMaze30\HWV_DYNAMIC.CFG
Load a new **HWV** configuration file to change the analysis characteristics. If a `[datapath]` is not specified, the configuration file must be available in the working directory. If this command does not appear, the program uses the current configuration or the one in force in the last run.

POINT xp yp zp
POINT = (0.00, 0.05, 4.67)
Perform a point calculation and write values of the quantities defined in the *Interpolation* section of the configuration file to the data record file. Enter coordinates in the units set by *DUnit*.

SCAN xp1 yp1 zp1 xp2 yp2 zp2
SCAN = (0.00, 0.00, 15.00) (12.00, 0.00, 15.00) Perform $(NScan + 1)$ calculations along a line in space. At each point, write the first *NRecord* quantities defined in the `INTERPOLATION` section of the configuration file to the data record file. Enter coordinates in units set by *DUnit*.

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

```
GENSCAN
  xp1 yp1 zp1
  xp2 yp2 zp2
  ...
  xpn ypn zpn
END
```

Each data line contains three real numbers separated by spaces to define a point in the solution volume. Enter coordinates in units set by *DUnit*.

VOLUMEINT [NReg]
VOLUMEINT
VOLUMEINT 8
Perform a volume integral of quantities defined in the `VOLUME` section of the configuration file and write the results to the data record file. If a region number does not appear, the integral is taken over the full solution volume. Otherwise, the integral extends over elements with region number *NReg*. Volume integrals may take a long time, so use the region-number option if there are several regions that are not of interest.

SURFACEINT Reg1 Reg2 ... RegN
SURFACEINT 5 7 9 12
Perform a surface integral of quantities defined in the `SURFACE` section of the configuration file and write the results to the data record file. Positive integer values correspond to *Internal* regions and negative to *External* regions. With no specification, all regions that are not *Internal* are taken as *External*. The integral is taken over the surface facets between any *Internal* and *External* elements. The command may include any number of regions. The regions in the *Internal* set may or may not be contiguous. Note that **HWV** does not include facets on the boundary of the solution volume in surface integrals.

TLIMITS
Make a listing of global and region values of the average temperature and temperature limits.

MATRIX FileName XMin XMax NX YMin YMax NY ZMin ZMax NZ
MATRIX WGUIDE.MAT 1.00 1.00 10 10.00 2.00 10 2.00 12.00 20

Write values calculated at an array of location to a data file in text format. The file is named $FileName$ and is created in the current directory. The real-number values x_{min} and x_{max} are the limits of the listing volume along the x direction. The parameter N_x is an integer. Listings are made at $(N_x + 1)$ evenly spaced positions along x. If you want a two-dimensional matrix of values in a single plane, set either N_x, N_y or N_z equal to zero. For example, if $N_x = 0$, then a set of calculations is performed over the specified range of y and z at the single position $x = (x_{min} + x_{max})/2.0$.

PLOT FSaveView FOutput Nx Ny
PLOT (XAxis3D VIEW001 800 600)

Write a plot file based on data in the currently loaded solution. The string *FSaveView* is the prefix of the file `FSaveView.FPV`, a set of view parameters created with the *Save view* command (Sect. 9.3). The view file must be available in the working directory. The plot is saved in the working directory. It is in Windows Bitmap format and has the name `FOutput.BMP`. The integers N_x and N_y give the width and height of the image in pixels. To optimize compatibility with graphics format converters, pick values that are multiples of 16 (*e.g.*, 1024×768).

NSCAN NScan
NSCAN = 100

Set the number of intervals for line scans. The default is $NScan = 150$ and the maximum value is $NScan = 500$.

RECORD [Field, Full]
RECORD = Full

Set the number of quantities to include in data files in response to the *Scan* and *Matrix* commands. The *Field* set (default) consists of E_x, E_y and E_z. Use this option to create data tables for **OmniTrak**. The *Full* set includes all quantities defined in the `INTERPOLATION` section of the configuration file.

INTERPOLATION [LSQ,Linear]
INTERPOLATION = Linear

Set the interpolation type to a multi-element least-squares fit or a single-element linear fit.

ENDFILE

Terminate the analysis

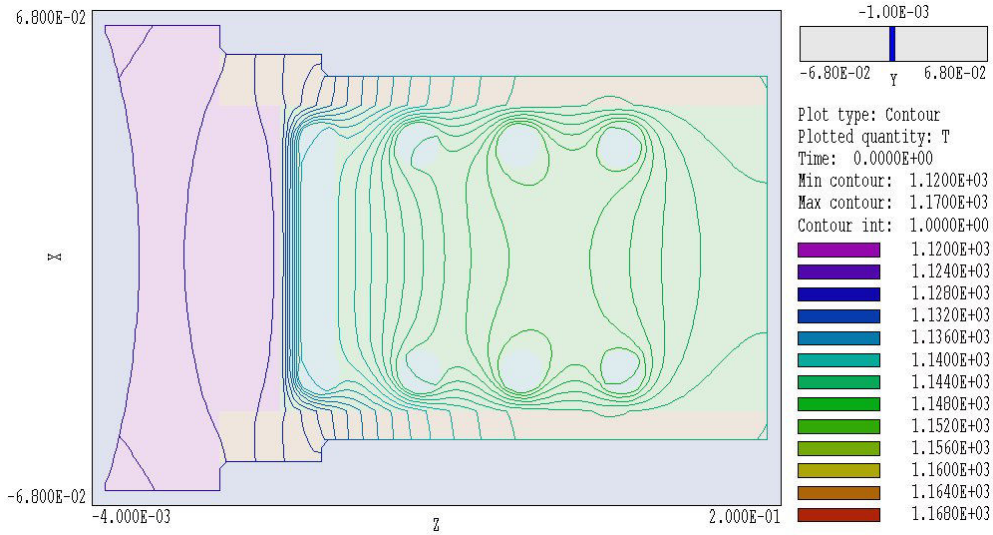


Figure 23: Slice plot showing isotherm lines and region boundaries.

11 HWV – slice plots

11.1 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 the planes of the foundation mesh. The precise rendering of mesh information enables 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. 19). 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. 23) 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. 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 *F1* key if want to enter values from the keyboard. Note that the normal plane box in the orientation area to the right of the plot (Fig. 23) 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.

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 relative start and end points for the shift.

11.2 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. 24. 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. In the *Filled contour* style, the program determines discrete bands of color coding according to values of the current plot quantity. The *Contour* style shows lines of constant values of the plot quantity.

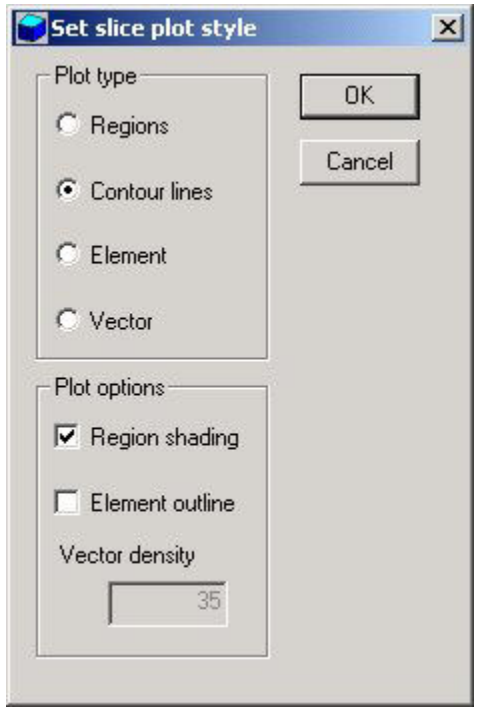


Figure 24: Slice plot style dialog.

Finally, an *Element* plot has color coding by the average value of the plot quantity in the element volume. When the *Element outline* box is checked, **HWV** includes facets in the *Region* and *Element* modes. In comparing relative advantages, the *Filled contour* plot provides the most attractive and accurate display in extended field regions. Use the *Element* plot for the best view of discrete field changes at the boundaries of materials. The *Contour* plot is useful for checking field uniformity over a volume.

PLOT QUANTITY

Use this command to set the quantity for color-coding in *Filled contour* and *Element* plots and line values in *Contour* plots. The following quantities are available in **HWV**:

T: temperature (°C).

F_x, F_y, F_z, $|\mathbf{F}|$: thermal flux components and amplitude (W/m^2).

Q: source power density (W/m^3).

k: thermal conductivity W/m-°C.

C_p: specific heat, J/kg-°C.

Plots of k and C_p are of interest when materials have temperature-dependent properties. The quantity Q is useful to check values loaded from **RFE3** or **GamBet**.

PLOT LIMITS

This command opens the dialog of Fig. 25. In the *Autoscale* mode, **HWV** chooses defaults for

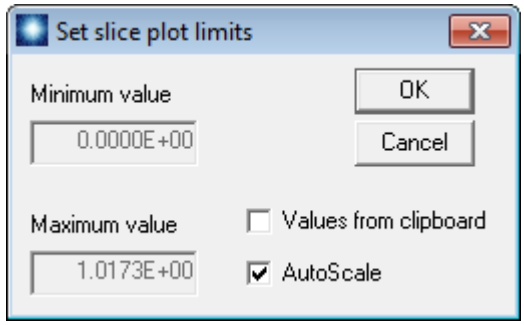


Figure 25: Dialog to set limits of the plotted quantity.

the minimum and maximum values for colors and contours based on the range of values of the current plot quantity in the current slice. Deactivate *Autoscale* to set the values manually. The manual mode is useful for comparing absolute field distributions in different slices. In either case, **HWV** uses a uniform interval between the limits based on the number of contour lines. For plots in the *Contour* style, you can set specific values for the contour lines. Applications include logarithmic intervals or the boundary of a region above a critical temperature. To use this feature, open a text document using the internal editor or an external program. Make a list of the values you want, using any valid numerical format. There should be one entry per line and the entries should appear in ascending order. Save the document if you want to compare a reference set. Highlight the values and copy them to the Windows clipboard. Then return to the plot limits dialog and check the *Values from clipboard box*. **HWV** displays an error message if the number set is invalid. Otherwise, the program reads the data, exits the dialog and updates the plot. Note that you will see a change only if the *Contour* style is displayed. Other plot styles are not affected.

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* may 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 point depends 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.

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

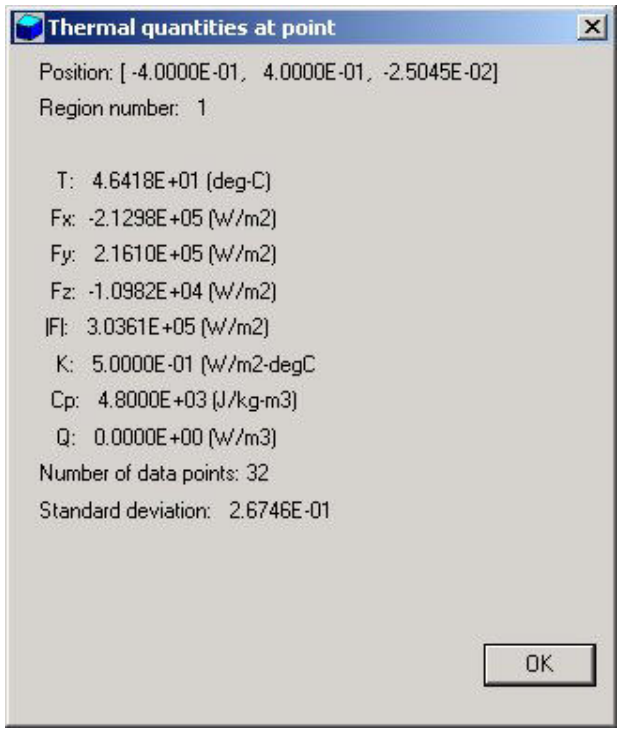


Figure 26: Information display – point calculation.

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.

11.3 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 26 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 field 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* display (Fig. 7). **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. The choices are the same as those in the element style slice plots.

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 250.

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

OSCILLOSCOPE MODE

In 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 mode.

11.4 Slice vector tools

HWV has useful tools to display the direction of the vector quantities **B** and **H** in slice plots. Figure 27 shows the corresponding entries on the toolbar.

FLUX PROBE

This feature was inspired by the familiar Magnaprobe illustrated in Fig. 27. When you click on the tool and move the cursor into the slice-plot area, it changes to a semi-transparent probe that rotates about a pivot point to show the direction thermal flux. The probe functions in all plot styles and may be combined with flux-line traces and scatter plots. The status bar shows the coordinates along with the temperature and thermal flux magnitude at the pivot point.

HEAT FLUX AT POINTS

Use this command to add projected lines of thermal flux. . The program enters coordinate entry mode when you click the command. Move the mouse to a point in the solution volume and click the left button. **HWV** calculates the three-dimensional heat-flow path that passes

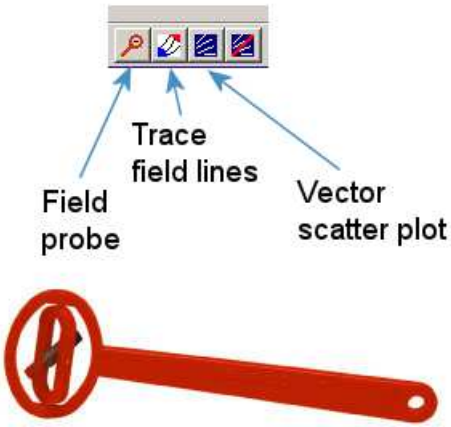


Figure 27: Tools to display vector information in the slice plot menu.

through the point and plots the projection in the slice plane. You can continue to add any number of lines. Click the right mouse button or press *ECS* to exit coordinate mode. The lines are not included in hardcopy or plot file exports and disappear if you change the slice view. Use a screen capture utility to record them. It is important to recognize the nature of the plots. The lines are three-dimensional curves projected to the slice plane. They may be difficult to interpret if the line does not lie close to the slice plane. For full three-dimensional field line plots, see the *Field line plot file* command in Sect. 12.

VECTOR SCATTER PLOT

Superimpose a uniform distribution of vector arrows pointing in the direction of heat flow in the slice plane. Arrows may be added to any of the plot types, including *Region*. They are included in hardcopy and plot file export. The arrows are preserved and adjusted if you change the slice view.

REMOVE VECTORS

Use this command to turn off the vector scatter plot mode and to remove vector tool displays from the current plot.

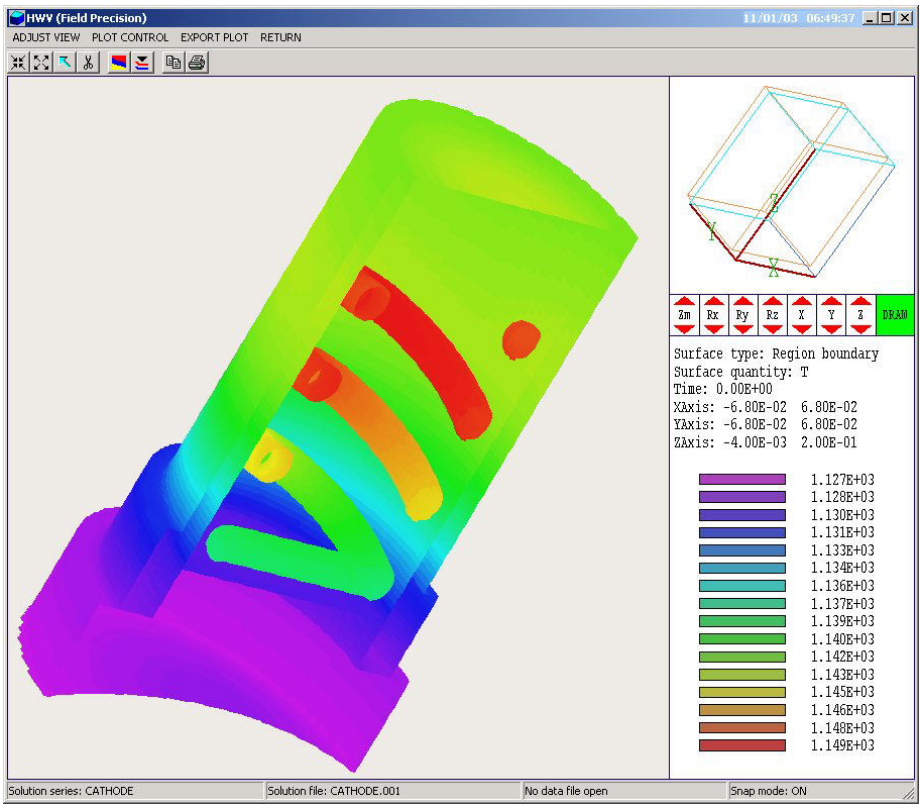


Figure 28: HWV working environment – surface plot menu.

12 HWV – surface plots

Surface plots are three-dimensional views of the solution space. Surface plots are created from the conformal mesh and preserve true spatial scaling. Four types of information may be displayed: 1) region boundaries color-coded by region number, 2) thermal quantities in a slice plane normal to one of the Cartesian axes, 3) isothermal surfaces with color-coding by the amplitude of thermal flux $|\mathbf{F}|$ and 4) boundaries of regions color-coded by local thermal quantities (Fig. 28). Region boundaries may be superimposed on both isothermal and normal plane plots.

The method to control the three-dimensional display with the mouse is identical to that used in **MetaMesh**. Figure 29 shows the active areas of the screen. The central zone (A) is used for zooming in (left button) and out (right button). Hold down the left mouse button in zones B, C, D and E to walk around the object. Hold down the right mouse button in zones B, C, D and E to move the viewpoint to the right, upward, to the left and downward. Note that changes are reflected in the orientation box in the upper-right portion of the screen. The plot is updated when you release the mouse button. You can control additional aspects of the three-dimensional view with the commands of the *Adjust view* popup menu.

You can control additional aspects of the three-dimensional view with commands of the *Adjust view* popup menu.

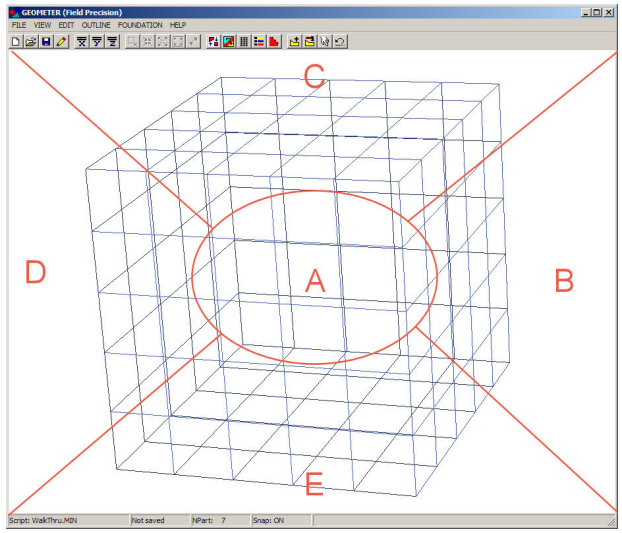


Figure 29: Active areas for mouse control of the 3D view.

RESTORE DEFAULT VIEW

This command is useful if you loose your orientation after several rotations and translations. The view is returned to the default: $\theta_x = -30^o$, $\theta_y = 0^o$ and $\theta_z = 45^o$ with the origin at the center.

+X VIEW
+Y VIEW
+Z VIEW

Rotate to views from the $+x$, $+y$ or $+z$ directions. Origin shifts are not affected.

CENTER VIEW

Remove shifts by setting the origin to the center of the solution volume.

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

SURFACE PLOT CONTROL

This command brings up the dialog of Fig. 30. The *Surface plot type* radio buttons determine the type of data plot. The three options are *Isotherm*, *Normal plane* and *Region only*. In the isotherm plot, **HWV** constructs a 3D 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 display. Enter the temperature value for the isotherm in the box on the right. The surface may be plotted in wireframe or hidden surface formats (*Surface plot style* group). A wireframe plot 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 local 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 *Show facets* check box in the *Surface facet*

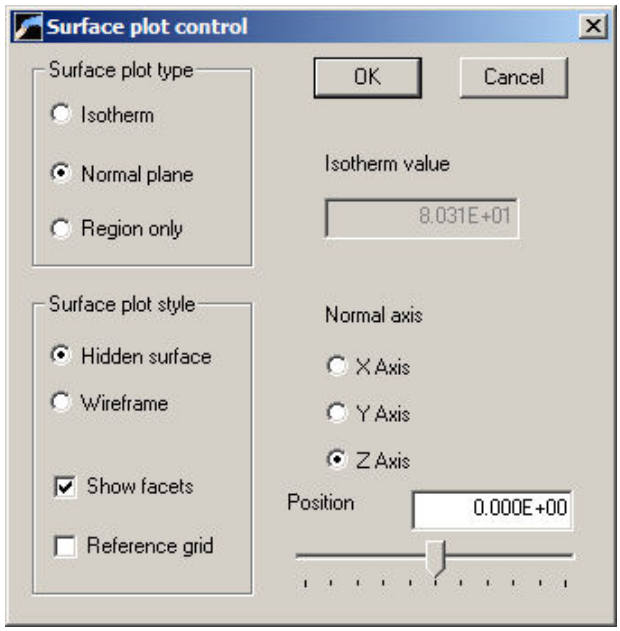


Figure 30: Dialog to control the surface plot style.

display group activates plotting of the boundaries of surface facets. Turn off the facet display for a dense mesh. When the *Reference grid* box is checked, lines are included to show the boundaries of the solution volume. The *Normal plane* plot shows color-coded information on thermal quantities in a plane normal to one of the Cartesian axes (Fig. 31). The plot is similar to the plane plot discussed in Sect. 9.2 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. Normal planes may be plotted in either wireframe or hidden surface mode. You can combine display of region boundaries with a normal plane. Finally, the *Region only* options deactivates the normal plane so that only region boundaries are displayed. In this case, a plot is not 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.

DISPLAYED REGIONS

The command brings up the dialog of Fig. 32 where you can pick region boundaries to include in the plot. Depress a button in the *Display* column to activate a region. The buttons in the column marked *Thermal* determine the presentation style for boundary facets. If the box is uncheck, **HWV** colors facets by region number. If one or more buttons are depressed in the *Thermal* column, color code is by the computed field quantity. The information window shows the correspondence between color and the quantity. Note that the field quantity is always calculated at a point near the facet **outside** the chosen region. If the region is a fixed-potential electrode, the color coding gives the surface field in the adjacent vacuum/dielectric region. If a boundary separates two conductive regions, the field values depend on which region is chosen for display.

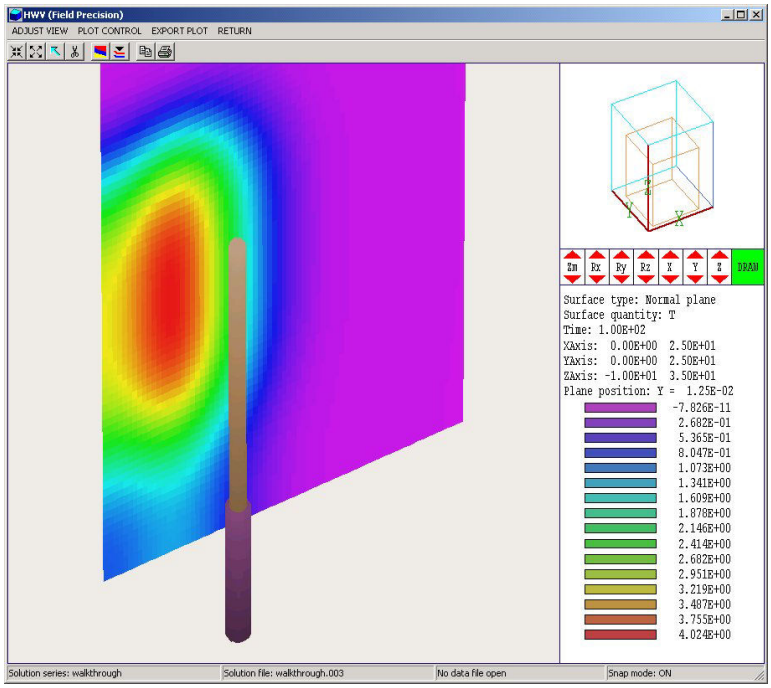


Figure 31: Normal plane plot with superimposed region boundaries of the probe and sheath in the `WALKTHOUGH` example.

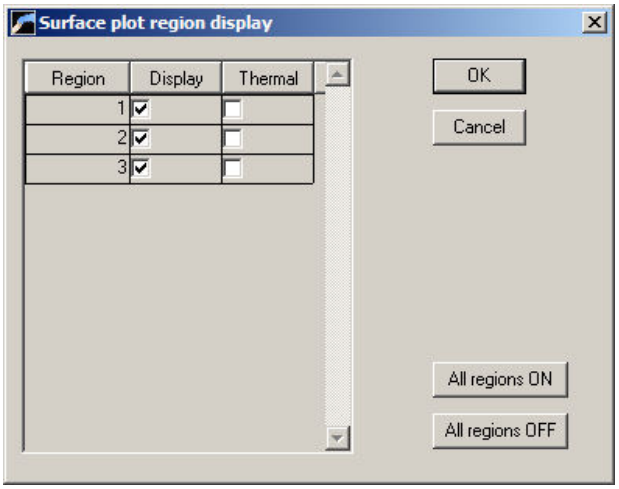


Figure 32: Dialog to control region display.

PLOT QUANTITY

Set the quantity that determines color coding in the normal plane or on region boundaries. Available quantities are 1) temperature (T), 2) the magnitude and components of thermal flux ($|\mathbf{F}|$, F_x, F_y and F_z), the source density (Q), the conductivity (k) and the specific heat (C_p).

PLOT LIMITS

Over-ride autoscaling and set limits for plotted quantities in the normal plane or on region boundaries. Facets where calculated quantities are outside the range are plotted in gray.

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 displayed area 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.

ORTHOGONAL/PERSPECTIVE VIEW

By default, **HWV** includes perspective correction in surface plots. Click on this command to move the view point to infinity. Click the command a second time to restore the view point.

13 Customizing HWV

You can create configuration files to customize the operation of **HWV** for your application. You can then switch between configurations for different types of analyses. **HWV** is supplied with two default configuration files in the same directory as the executable program: `hwv_standard.cfg` and `hwv_dynamic.cfg`. We recommend that you do not change these files. Instead, make a copy to use as a template and rename it. You can put your new configuration file anywhere, but we suggest you store all configurations in the same directory as `hwv.exe`. Note that **HWV** remembers the last configuration used and attempts to reload it at the next run.

When you have created a template file, open it in an editor. Note that it is divided into the ten sections shown in Table 13. With few exceptions, do not make changes to the sections `HEADER`, `STOREDQUANT` `REGIONQUANT`, `PROGPARAM`, `RUNPARAM` and `REALTIMEPARAM`. Entries in these sections coordinate the operation of **HeatWave** and **HWV**. Modifications may cause file load errors.

13.1 Defining calculated quantities

You can define custom quantities for display and analysis in the last four sections of the configuration file:

Interpolation. Scalar quantities for multiple uses: 1) displays in slice, plane and surface plots, 2) results of point and scan calculations and 3) values written to matrix output files.

Vector. Vector quantities for the display of field or flux lines in slice and surface plots.

Volume. Scalar quantities for volume integrals over regions.

Surface. Vector quantities for surface integrals over the boundaries of region sets.

Run speed is a critical issue in **HWV**. The generation of a single slice plot may require over 50,000 interpolations. To ensure fast operation, expressions for calculated quantities in the configuration file are written in reverse Polish notation (RPN). **HWV** parses the function strings once while loading the configuration file and saves them in a coded form. Thereafter, operations are performed at the speed of compiled code. The following section reviews the basics of RPN notation. Section 13.3 covers parameters and variables that may appear in expressions. Section 13.4 reviews the form of the data sections.

13.2 Introduction to RPN

Consider the expression

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

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

Table 14: General organization of the **HWV** configuration file

```
HEADER
  Control parameters (Required)
END
STOREDQUANT
  Stored quantities in the solution file (Required)
END
REGIONQUANT
  Physical properties of regions (Optional)
END
PROGPARAM
  Program parameters (Optional)
END
RUNPARAM
  Parameters set by the solution program (Optional)
END
REALTIMEPARAM
  Parameters set by the user (Optional)
END
INTERPOLATION
  Calculated quantities for plots and data output  (Required)
END
VECTOR
  Calculated quantities for vector display (Optional)
END
VOLUME
  Calculated quantities for volume integrals (Optional)
END
SURFACE
  Calculated quantities for surface integrals (Optional)
END
ENDFILE
```

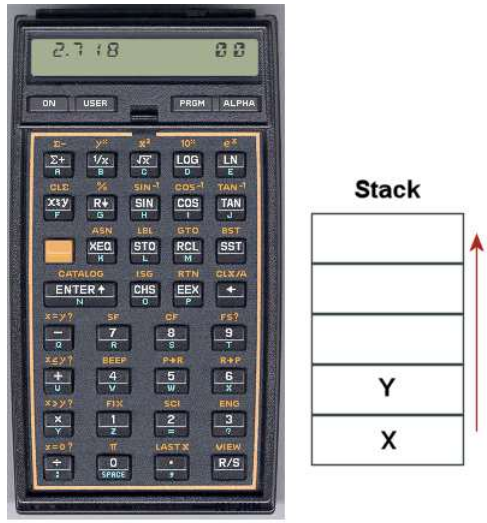


Figure 33:

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

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

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

The string is parsed in strict order from left to right for expressions of any degree of complexity. A calculator with RPN logic uses the stack shown in Fig. 33. Numbers are pushed and popped at the bottom. The first two stack registers have the special names X and Y. Three simple rules govern the evaluation of RPN expressions:

> If the entry is a number, push it on the stack.
>
> If the entry is a unary operator (exp, ln, sin,...), apply it to the number in the X register.
>
> If the entry is a binary operator (+,*,^,...) combine the numbers in the X and Y registers and move the stack down.

The final result is given by the number remaining in the X register. By convention, the binary operators act in the following way:

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

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

13.3 Variables, operations and parameters

Expressions to define quantities in **HWV** may contain numbers, unary and binary operations, standard variables, standard calculated quantities and parameters. The standard variables are

```
$X, $Y, $Z, $NREG, $DUNIT
```

Note that the name of variables and parameters must start with a dollar sign. Calls to defined functions for plots, line scans and other operations are always made at a specific location in the solution space. **HWV** sets the current position $[\$X, \$Y, \$Z]$ before any calls are made. To illustrate, the following expression gives the distance from the origin in meters:

```
RDIST = $X 2 ^ $Y 2 ^ $Z ^ 2 + + @SQRT
```

The standard variable `$DUNIT` is a dimension conversion factor passed from the solution program. For example, suppose dimensions in **MetaMesh** were defined in centimeters and converted to meters for use in the solution program. In this case, an entry in the header of the solution file would set `$DUNIT` = 100.0. The following expression returns the distance from the origin in centimeters:

```
RDIST = $X 2 ^ $Y 2 ^ $Z ^ 2 + + @SQRT $DUNIT *
```

Unary operators have names that begin with '@'. Table 15 lists the available set. The following considerations apply with respect to the `@CHS` operator:

> You may include negative numbers (such as `-3.1456` and `-8.9E-09`) in expressions. The number `-22.56` is equivalent to `22.56 @CHS`.
>
> Expressions like `-$Mu0` and `-&grady[3]` that mix the binary operator with a variable are invalid. Instead, use forms like `$Mu0 @CHS`.

The following quantities represent values from the thermal solution calculated at the current position:

```
&T
&dTdx
&dTdy
&dTdz
&gradT
&Q[N]
```

The names must begin with an ampersand. They are always followed by an integer in box brackets giving the number of the stored quantity. The quantity `&gradT` equals $|\nabla T|$. The quantity `&Q[n]` is the interpolated value of stored quantity n at the current position [`$X,$Y,$Z`]. **HeatWave** includes the following stored quantities:

> `&Q[1]` The temperature T in °C or °K (node).
>
> `&Q[2]` The thermal conductivity k in W/m-°C (element).

Table 15: Unary operators

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

&Q[3] The specific heat C_p in J/kg-°C (element).

&Q[4] The thermal source density q in W/m^3 (element).

Note that the specific heat is meaningful only for dynamic solutions.

The program performs the interpolation and replaces the symbol with its value before evaluating the expression. The interpolation method depends on whether the stored quantity is of type *Node* or *Element*. **HWV** performs detailed interpolations of *Node* quantities. In the LIN mode, the program interpolates within the target element using shape functions and values of the stored quantity at the eight nodes. In the LSQ mode, **HWV** collects data from surrounding elements of the same region type and makes a least-squares fit to a three-dimensional, second-order function.

HeatWave also transfers the following region variables:

```
$Rho
$EMiss
```

These variables assume values characteristic of the region at the current interpolation point. The variable $Rho is the mass density in kg/m^3 and $EMiss is the emissivity for regions with thermal radiation boundaries.

Optionally, expressions may include parameters defined in the PROGRARAM section. As an example, this entry defines the specific heat of water:

```
PROGPARAM
  $CpWater = 4186.0
END
```

You can define up to ten parameters in the PROGRARAM section. The name must begin with a dollar sign and may include up to 12 characters. A value in any valid real-number format follows the equal sign.

13.4 Setting up the calculated-quantity sections

The final sections of the configuration file define quantities for plots and calculations. There are four sections:

```
INTERPOLATION
  QuantityName = Definition
END
VECTOR
  QuantityName = Definition1;Definition2;Definition3
END
VOLUME
  QuantityName = Definition
END
SURFACE
  QuantityName = Definition1;Definition2;Definition3
END
```

The INTERPOLATION section may contain from zero to 50 quantity definitions. The VECTOR, VOLUME and SURFACE sections may contain a maximum of 20 quantity definitions each.

A definition line for a scalar quantity in the *Interpolation* section consists of a variable name, an equal sign and an RPN expression using the components discussed in the previous section. The following rules apply to quantity names in all sections:

> A name has a maximum length of 12 characters (the length is limited so that quantity names will fit in plots and formatted output lists).
>
> Names may include characters, numbers and the underscore symbol. Names may not include spaces and other punctuation marks.

The RPN expression is a string (maximum length of 300 characters) with entries separated by spaces. Here are some examples of quantity definitions:

```
EPERP = &EY @XSQ &EY @XSQ + @SQRT
LOGEXVCM = &EX 100.0 / @LOG
DX = &EX $MU0 * &Q[2] *
FIELDANGLE = &BY &BX / @ATAN $RADTODEG *
```

The last expression requires that the parameter

```
$RADTODEG = 57.2958
```

was defined in the PROGPARAM section. Note that the result of the division operator in the expression is 0.0 if &BX = 0.0.

The order in which quantities appear in the INTERPOLATION section affects some operations in **HWV**.

Table 16: Naming conventions in RPN expressions

Name form	**Functions**
`$NAME`	Program parameters
`$NAME`	Standard variables (*e.g.*, `$X`)
`&NAME`	Special quantities calculated from the solution (*e.g.*, `&Ex`)
`@NAME`	Unary operators in RPN expressions

The first listed quantity is the default choice for *Slice* plots You can change the choice by entering a value in the *SliceDisplay* command in the `HEADER`. The integer is the order of the default plot quantity as listed in the `INTERPOLATION` section.

In dialogs for picking a plot quantity, the options are listed in the same order that they appear in the configuration file. Therefore, you should put important quantities near the beginning and specialized quantities near the end.

The same order is followed in listings created by the *Matrix file* command.

Each quantity in the *Vector* section requires three expressions for the x, y and z components. The format for a definition line is

```
Name = (Expression x);(Expression y);(Expression z)
```

The line may contain a maximum of 500 characters. The component expressions are separated by semicolons. The following example defines the thermal flux:

```
Flux = &Q[2] &dTdx *;&Q[2] &dTdy *;&Q[2] &dTdz *
```

The volume integral of a quantity Q over a region N_R is given by

$$\int dx \int dy \int dz \; Q(x, y, z), \tag{10}$$

where only those elements with region number equal to N_R are included. You can define up to 20 quantities in the *Volume* section for integrals. The expressions should define quantities with dimensions of the form $(quant/m^3)$. Volume integrals use internal program coordinates in units of meters. Note that the quantity `$DUNIT` is used in **HWV** only for display labels.

The surface integral of a vector quantity $\mathbf{S}$ over a region or a set of regions is given by

$$\int \int dA \; \mathbf{S} \cdot \mathbf{n}, \tag{11}$$

where dA is an element of area on the surface of the region set and $\mathbf{n}$ is a unit vector normal to the surface pointing out of the region set. You can define up to 20 quantities in the `SURFACE` section for integrals. The data line for each quantity should contain the name, an equal sign, and three RPN expressions separated by semicolons for the vector components. The expressions should define quantities with dimensions of the form $(quant/m^2)$.

Note that quantity names must begin with special symbols if they are used in RPN expressions. Symbols for different quantity types are listed in Table 16. Therefore, quantity names in the `PROGPARAM` section must include the dollar sign. Quantity names that are defined only for display purposes need not include a special symbol. The includes quantities defined in the `INTERPOLATION`, `VECTOR`, `VOLUME` and `SURFACE` sections.

14 Probe – history file plot utility

14.1 Introduction

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

Table 17 shows the standard probe file format. The first section is a header that contains the following information:

Generating program name.

Dimensionality of the generating program (1, 2 or 3).

The spatial position of the probe (from 1 to 3 quantities).

The index of the element sampled by the probe.

The region number of the element.

Conversion factors for the probe position and the recorded quantities.

Labels for the recorded quantities.

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

14.2 Loading data files

When you start **Probe** the only active menu option is *Load probe*. Plotting and analysis functions become active when a probe file has been opened. The program displays a dialog showing all files with suffixes of the form `P01`,...,`P12`. Pick a file to analyze and click *OK*. Changing directories in the dialog will change the working directory of the program. If the load is successful, **Probe** creates a default plot of the data (Fig. 34).

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

Table 17: Example of the **Probe** file format

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

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

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

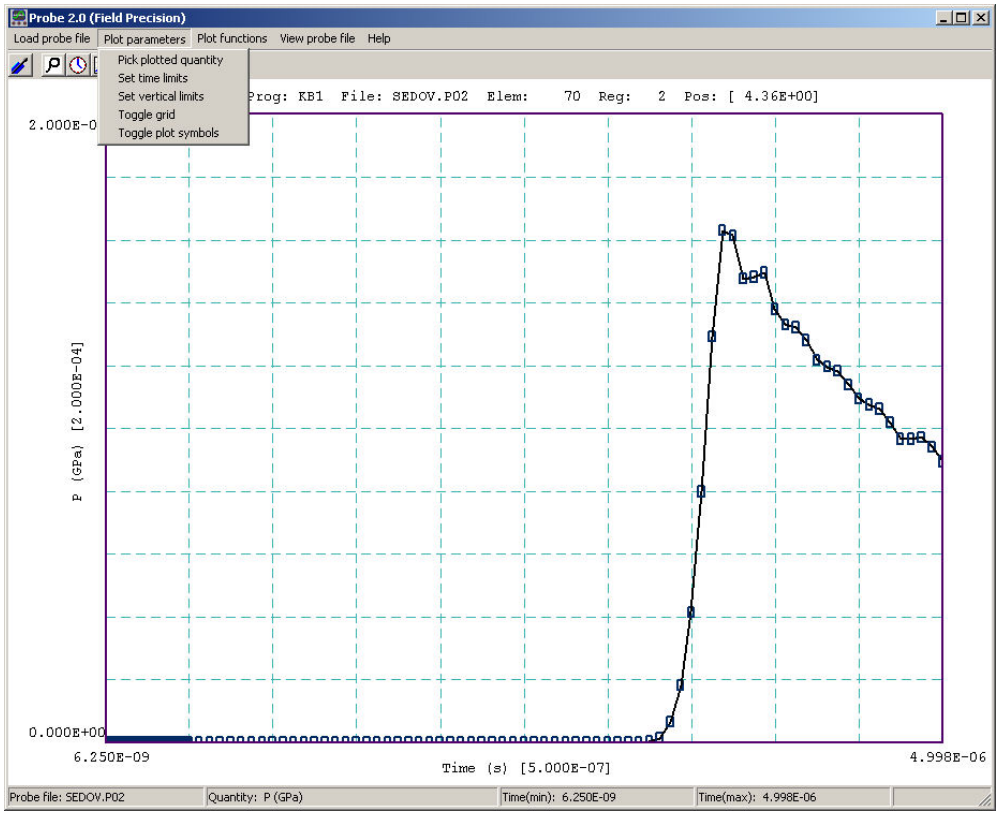


Figure 34: **Probe** screen shot.

The grid intervals are shown in parentheses next to the labels of the horizontal and vertical axes. The title line at the top of the plot shows the following information: generating program, probe file name, element number, region number and position. This information is recorded in hardcopy plots to help you archive your data.

14.3 Plot settings

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

PLOTTED QUANTITY

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

TIME LIMITS

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

VERTICAL LIMITS

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

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

TOGGLE GRID

Switch between grid and and fiducial lines in the plot.

TOGGLE PLOT SYMBOLS

Include or remove symbols to mark the recorded points.

14.4 Plot functions

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

OSCILLOSCOPE MODE

When you issue this command, **Probe** simulates a digital oscilloscope. As shown in Fig. 35, the mouse cursor changes to a cross-hair pattern when it is inside the plot window. The program adds movable fiducial lines to mark the current point. You can drag the fiducials along the time axis by moving the mouse. A box at the bottom of the plot shows values of the time and plotted quantity at the current position. If you press the left mouse button, the program displays a box with the following information about the current point:

Time, t.

Value of the plotted quantity, $V(t)$.

Derivative of the plotted quantity, $dV(t)/dt$.

Definite integral of the plotted quantity, $\int_0^t V(t')dt'$.

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

SMOOTH DISPLAY

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

DEFAULT PRINTER

Probe can port copies of the plot to any installed Windows printer. The program sends output

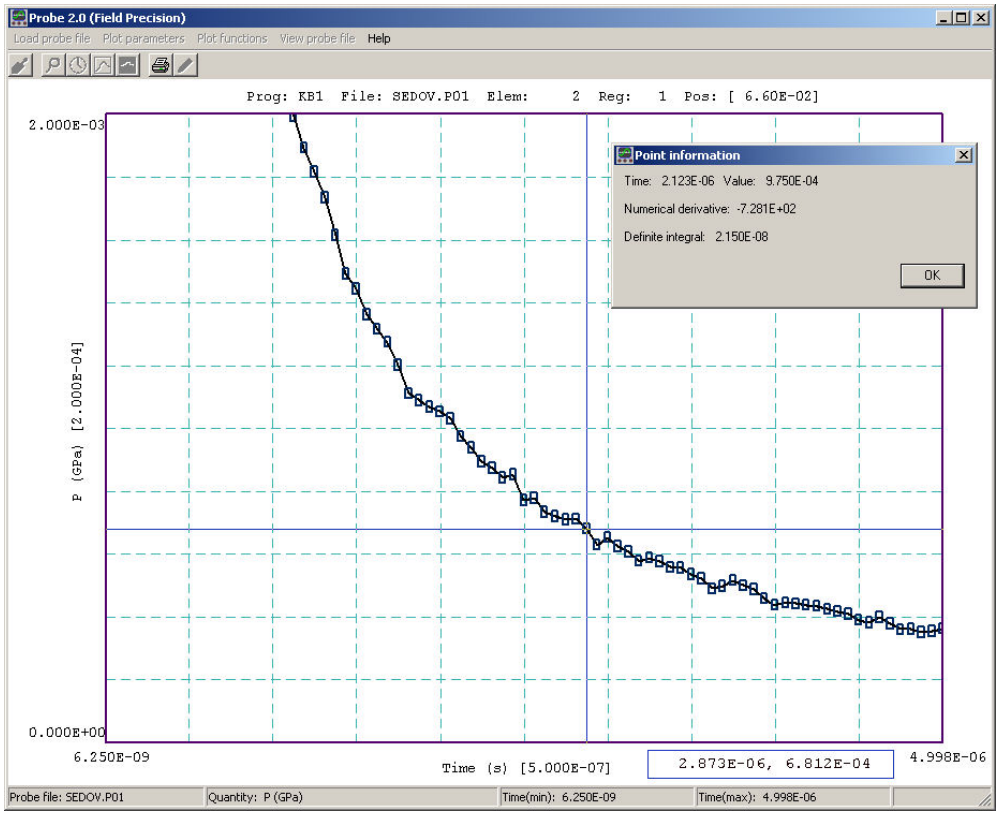


Figure 35: **Probe** in the *Oscilloscope mode*.

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

PLOT FILE (EPS)
PLOT FILE (BMP)
PLOT FILE (PNG)

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

COPY TO CLIPBOARD

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

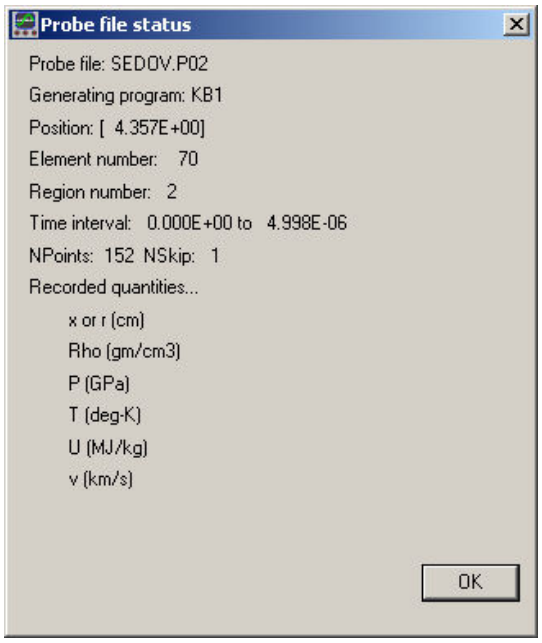


Figure 36: **Probe** file-information message box.

14.5 Information

PROBE FILE INFORMATION

Display information on the probe file in a message box (Fig 36). The quantity $NSkip$ in line 7 is used for long files. There is no reason to store more than 1000 points for plots on typical screens and hardcopy devices. When there are less than 1000 data lines, **Probe** loads all points ($NSkip = 1$). When the file contains 1000 to 2000 data lines, the program loads every second point ($NSkip = 2$), and so forth. In this way the **Probe** can handle probe files of any length without exceeding memory limits.

VIEW PROBE FILES

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

15 HeatWave file formats

HeatWave creates data and history files. Data files may be generated in binary or text format. The mode is controlled by the *Format* command (Sect. 3.1).

> The default binary mode should be used for most applications. It gives the minimum file size and preserves the full accuracy of the double precision numbers used in **HeatWave**.
>
> Use the text mode to port information to your own programs or if you want to inspect results with an editor.

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. The FORTRAN code extract shown in Table 18 comprises the entire output algorithm. The header contains the following elements:

> The 6-character string *BINARY*.
>
> The number of regions in the solution space, $NReg$ (4-byte integer).
>
> The data dump time in seconds for a dynamic solution. The number equals zero for static solutions (8-byte real).
>
> A total of $NReg$ 20-character strings giving the names of regions obtained from the **MetaMesh** input file.
>
> A total of $NReg$ single-character strings, either I or E. The character I designates a standard material region that should be included in a HWV analysis and E indicates a fixed-potential that should be excluded.
>
> The mesh size parameters I_{max}, J_{max} and K_{max} (4-byte integers).
>
> The quantity $DUnit$, the unit conversion factor associated with the solution (double precision, 8-byte real).

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 4-byte integers. The coordinates X,Y and Z (in meters) are 8-byte real numbers as are the following stored quantities:

> The temperature T at the node in units of °C.
>
> The thermal conductivity k in units of W/m-°C associated with the upper element.

Table 18: Code to create the **HeatWave** binary output file

```
! Header
    Character6 = 'BINARY'
    WRITE (OutField) Character6
    WRITE (OutData) t            ! Run parameter, time
    WRITE (OutField) NRegMax
    DO N=1,NRegMax
      WRITE (OutField) Reg(N).RName
    END DO
    DO N=1,NRegMax
      IF (Reg(N).Fixed) THEN
        Character1 = 'E'
      ELSE
        Character1 = 'I'
      ENDIF
      WRITE (OutField) Character1
    END DO
    WRITE (OutField) IMax,JMax,KMax
    WRITE (OutField) DUnit
! Node quantities
  DO K=0,KMax
    DO J=0,JMax
      DO I=0,IMax
        MC = M(I,J,K)
        STotal = C(MC).SUp + C(MC).SFileUp
        WRITE (OutData) &
        C(MC).RegNo,C(MC).RegUp,C(MC).x,C(MC).y,C(MC).z, &
        C(MC).T,C(MC).KUp,DZero,STotal
      END DO
    END DO
  END DO
! Region properties
  DO N=1,NRegMax
    MatNo = Reg(N).Mat
    WRITE (OutData) Mat(MatNo).Dens,Mat(MatNo).EMiss
  END DO
```

For dynamic solutions, the specific heat C_p in units of J/kg-°C associated with the upper element. The quantity equals zero for static solutions.

The thermal source density q in units of W/m^3 associated with the upper element.

Finally, HeatWave records region information, one data entry for each region in the solution volume. The following quantities are recorded:

The region mass density ρ in kg/m^3 for dynamic solutions. The quantity equals zero for static solutions.

The emissivity for regions with thermal radiation boundaries (dimensionless).

Table 19 shows the header and initial node data lines from an output file in text format (note that the node data lines have been broken for readability). The first line contains the string `TEXT` and the second contains the data dump time (real number)/ Four region names are shown in the example. The names are strings up to 20 characters in length with no spaces. Each name is followed by the character I (included) or E (excluded). The order of nodes is the same as that of Table tab:outputcode. Real numbers are recorded in format `(1P,E14.6)`.

Files created in response to the **HWV** *Matrix file* are in text format and have a structure similar to that of Table 19. The files are useful to transfer information to your own analysis programs and they are compatible with **OmniTrak** and **FEVision**. There are two differences from the `WOU` file in text format. First, calculations are performed over a regular mesh at locations

$$X(I) = X_{min} + I\Delta x, \tag{12}$$
$$Y(J) = Y_{min} + J\Delta y, \tag{13}$$
$$Z(K) = Z_{min} + K\Delta z. \tag{14}$$

where

$$\Delta x = (X_{max} - X_{min})/I_{max}, \tag{15}$$
$$\Delta y = (Y_{max} - Y_{min})/J_{max}, \tag{16}$$
$$\Delta z = (Z_{max} - Z_{min})/K_{max}. \tag{17}$$

Second, the listed quantities depend on definitions in the `INTERPOLATION` section of the configuration file and the value of the parameter NRecord in the header. The following quantities are included when operating under configuration `heatwave_standard.cfg` with NRecord = 6: temperature (T in °C or °K), x components of thermal flux ($F_x = -k\partial T/\partial x, F_y, F_z$ in W/m^2, magnitude of the thermal flux ($|F| = k|\nabla T|$), thermal conductivity (k in W/m-°C).

Table 19: Header and data extract showing the form of the **HeatWave** `WOU` file under the `Text` option and the **HWV** matrix file

```
TEXT
  2.009319E+01
NReg:   4
TISSUE I
SLEEVE I
PROBE I
BOUNDARY I
IMax:     92
JMax:     92
KMax:    102
DUnit:   1.000000E+03

 RegNo RegUp       X             Y             Z
=====================================================
     1     1  0.000000E+00  0.000000E+00 -1.000000E-02
     1     1  2.500000E-04  0.000000E+00 -1.000000E-02
     1     1  5.000000E-04  0.000000E+00 -1.000000E-02
     1     1  7.500000E-04  0.000000E+00 -1.000000E-02
     1     1  1.000000E-03  0.000000E+00 -1.000000E-02
     1     1  1.250000E-03  0.000000E+00 -1.000000E-02
     1     1  1.500000E-03  0.000000E+00 -1.000000E-02
     1     1  1.750000E-03  0.000000E+00 -1.000000E-02
    ...

            T             K            Cp            Q
       ==========================================================
         1.040439E-03  5.000000E-01  4.800000E+03  0.000000E+00
         1.244914E-03  5.000000E-01  4.800000E+03  0.000000E+00
         1.899021E-03  5.000000E-01  4.800000E+03  0.000000E+00
         3.119321E-03  5.000000E-01  4.800000E+03  0.000000E+00
         5.078186E-03  5.000000E-01  4.800000E+03  0.000000E+00
         7.967381E-03  5.000000E-01  4.800000E+03  0.000000E+00
         1.194936E-02  5.000000E-01  4.800000E+03  0.000000E+00
         1.711622E-02  5.000000E-01  4.800000E+03  0.000000E+00
          ...
```

Index