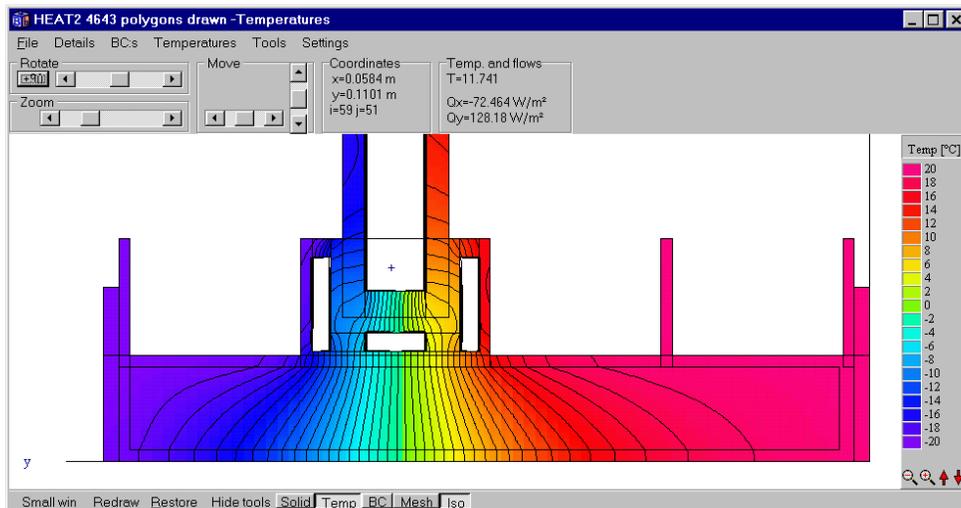


HEAT2

A PC-program for heat transfer in two dimensions.

Manual with brief theory and examples.

HEAT2 version 3.0 and HEAT2W version 1.01



October 26, 1998

(Revised November 25, 1998)

Thomas Blomberg

Contents

1. INTRODUCTION.....	7
1.1 HOW TO AVOID READING THIS MANUAL	7
1.2 SHORT DESCRIPTION OF HEAT2 AND HEAT2W	7
1.2.1 Applications	7
1.2.2 Features	7
1.2.3 Cartesian coordinates	8
1.2.4 Numerical method	8
1.2.5 Preprocessor HEAT2W	8
1.2.6 System requirements.....	10
1.2.7 Developers	10
1.2.8 Update info.....	10
1.3 ABOUT THIS MANUAL.....	11
1.4 INSTALLATION	11
1.5 TECHNICAL SUPPORT	11
2. MATHEMATICAL DESCRIPTION.....	13
2.1 GOVERNING DIFFERENTIAL EQUATIONS.....	13
2.2 BOUNDARY CONDITIONS	13
2.3 INITIAL CONDITIONS	14
2.4 FLUID REGIONS	14
2.5 HEAT CONDUCTION COUPLED TO RADIATION IN A CAVITY.....	15
3. NUMERICAL FORMULATION.....	17
3.1 INTRODUCTION	17
3.2 COMPUTATIONAL MESH	17
3.3 THERMAL CONDUCTANCES	17
3.4 HEAT FLOWS	19
3.5 NEW TEMPERATURES	20
3.6 CHOICE OF TIME-STEP,	21
3.7 ITERATIVE CALCULATION.....	21
3.8 STEADY-STATE CASE.....	21
3.9 INTERNAL REGIONS CONTAINING FLUID.....	22
4. OVERVIEW OF INPUT	23
4.1 INPUT MESH	23
4.2 NUMERICAL MESH	24
4.3 THERMAL PROPERTIES	26
4.4 BOUNDARY CONDITIONS	27
4.5 INITIAL TEMPERATURES	29
4.6 SOLVING THE PROBLEM FOR STEADY-STATE	30
4.7 SOLVING A TRANSIENT PROBLEM	31
4.8 INTERNAL MODIFICATIONS	35
4.9 INTERNAL RESISTANCES.....	36
4.10 SUMMARY.....	36
5. RADIATION AND VENTILATION IN CAVITIES	39
5.1 INTRODUCTION	39
5.2 INPUT DATA	39
5.3 OUTPUT DATA	40
5.4 NUMERICAL SOLUTION FOR THE CAVITY	41

5.4.1	<i>Radiation equations in temperature form</i>	41
6.	WORKING WITH HEAT2	43
6.1	LOADING AND SAVING INPUT DATA	43
6.2	INSERTING MESH COORDINATES	43
6.3	THERMAL PROPERTIES	46
6.4	BOUNDARY CONDITIONS	47
6.5	INITIAL TEMPERATURES	48
6.6	INTERNAL MODIFICATIONS	48
6.6.1	<i>Defining types</i>	48
	Note on order of approximation	51
6.6.2	<i>Defining areas</i>	51
6.7	FUNCTIONS	52
6.7.1	<i>Sinusoidal function</i>	53
6.7.2	<i>Step-wise constant and step-wise linear function</i>	53
6.7.3	<i>Example 1 - sinusoidal function with a time period of one year</i>	54
6.7.4	<i>Example 2 - sinusoidal function with a time period of one day</i>	55
6.7.5	<i>Example 3 - step-wise constant function</i>	56
6.7.6	<i>Example 4 - step-wise linear function</i>	56
6.8	INTERNAL RESISTANCES	57
6.9	SOLVING THE PROBLEM	57
6.9.1	<i>Steady-state stop criteria</i>	57
6.9.2	<i>Successive over-relaxation used in the steady-state calculations</i>	58
6.9.3	<i>Simulation time for transient calculation</i>	58
6.9.4	<i>Simulation window</i>	59
6.9.5	<i>Screen update</i>	59
6.9.6	<i>Reset</i>	60
6.10	OUTPUT	60
6.10.1	<i>Boundary flows</i>	60
6.10.2	<i>Heat flows for internal modifications</i>	60
6.10.3	<i>Results for radiation</i>	62
6.10.4	<i>Temp at point x,y</i>	63
6.11	THE RECORDER	64
6.12	INFO LOG	65
6.13	GRAPHICS	66
6.13.1	<i>Introduction</i>	66
6.13.2	<i>Basic commands</i>	67
6.13.3	<i>Menu item File</i>	67
6.13.3.1	<i>Saving images</i>	67
6.13.3.2	<i>Printing images from HEAT2</i>	68
6.13.4	<i>Menu item Options</i>	70
6.13.4.1	<i>Viewing thermal conductivities and heat capacities</i>	70
6.13.5	<i>Menu item Bounds</i>	71
6.13.6	<i>Menu item Temperatures</i>	71
6.13.7	<i>Viewing isotherms</i>	73
6.13.8	<i>Menu item Tools</i>	75
6.13.9	<i>Menu item Settings</i>	75
7.	HEAT2W - GRAPHICAL PREPROCESSOR FOR HEAT2	77
7.1	INTRODUCTION	77
7.2	DRAWING, MOVING, AND CHANGING OBJECTS	77
7.3	USING THE CLIPBOARD	80
7.4	EDITING THE MATERIAL LIST	80
7.5	WRITING THE INPUT FILE FOR HEAT2	81
7.6	MAPPING THE "REAL WORLD" INTO THE DRAWING AREA	81

7.7	DRAWING AREAS WITH INTERNAL MODIFICATIONS IN HEAT2W	81
7.8	LIMITATIONS IN HEAT2W VERSION 1.01	82
8.	CALCULATIONS USING PIPES.....	83
8.1	PIPE WITH A GIVEN HEAT FLOW	83
8.1.1	<i>Pipe approximated with a square</i>	83
8.1.2	<i>Pipe approximated with enclosing steps</i>	83
8.2	PIPE WITH A GIVEN TEMPERATURE	84
8.2.1	<i>Pipe resistance taken into account</i>	84
8.2.2	<i>Pipe resistance neglected</i>	84
9.	EXAMPLES	85
9.1	EXAMPLE 1 - A WALL WITH A JOIST	85
9.1.1	<i>Introduction</i>	85
9.1.2	<i>Input in HEAT2W</i>	85
9.1.3	<i>Input continued in HEAT2</i>	87
9.1.4	<i>Input using only HEAT2</i>	89
9.2	EXAMPLE 2 - HEAT LOSS FROM A HOUSE WITH A CELLAR	92
9.2.1	<i>Introduction</i>	92
9.2.2	<i>The input mesh</i>	92
9.2.3	<i>Material data</i>	95
9.2.4	<i>Boundary conditions</i>	96
9.2.5	<i>The simulation</i>	97
9.2.6	<i>Transient simulation</i>	99
9.2.7	<i>Input using HEAT2W</i>	104
9.3	EXAMPLE 3 - HEAT FLOW THROUGH A WALL WITH METAL STUDS	106
9.3.1	<i>Introduction</i>	106
9.3.2	<i>Input mesh</i>	106
9.3.3	<i>Calculations for different meshes</i>	107
9.3.4	<i>Conclusions</i>	109
9.4	EXAMPLE 4 - ROOF SECTION WITH HOLLOW BRICKS	110
9.4.1	<i>Introduction</i>	110
9.4.2	<i>Results for different numerical meshes</i>	110
9.4.3	<i>Choice of T_{so} and rate of convergence</i>	113
9.4.4	<i>Conclusions</i>	114
10.	A FEW TIPS.....	115
10.1	INTRODUCTION	115
10.2	NUMERICAL ASPECTS	115
10.2.1	<i>Numerical mesh</i>	115
10.2.2	<i>Expansive meshes</i>	115
10.2.3	<i>Steady-state calculations</i>	115
10.2.4	<i>Symmetric cases</i>	115
10.2.5	<i>Computational area for calculations of heat losses to the ground</i>	115
10.2.6	<i>Transient calculations for heat losses to the ground</i>	116
10.2.7	<i>Sloping boundaries</i>	116

REFERENCES.....	117
APPENDIX A. LIMITS AND RESTRICTIONS OF DATA.....	119
APPENDIX B. HOW TO ADD AND EDIT MATERIALS.....	121
APPENDIX C. INPUT DATA FILE EXAMPLE	123

1. Introduction

1.1 How to avoid reading this manual

For a quick start read Chapter 4 *Overview of input*.

1.2 Short description of HEAT2 and HEAT2W

1.2.1 Applications

HEAT2 is a PC-program for two-dimensional transient and steady-state heat conduction within objects that can be described in a rectangular grid. It belongs to the latest generation of computer models from the **Lund Group for Computational Building Physics**. It is well adapted to the following applications within building physics:

Analysis of thermal bridges

Calculation of U-values for building construction parts

Estimation of surface temperatures (surface condensation risks)

Calculation of heat losses to the ground from a house

Optimization of insulation fitting

Analysis of floor heating systems

There are also two related programs for heat conduction in three dimensions HEAT3 (Blomberg, 1998), and in cylindrical coordinates HEAT2R (Blomberg, 1994).

1.2.2 Features

The user works with an input mesh that facilitates and minimizes the input procedure. The time to generate the complete input for a reasonably complicated case is less than 10 minutes after a few hours' experience of the program. The following list shows some features:

- Extensive graphical capabilities: figures showing geometry, materials, numerical mesh, boundary conditions, temperature field, isotherms. Features: zoom, panning, rotation, color/gray-scale, high-resolution (600 DPI) printing. Arbitrary heat flows and temperatures can be recorded and shown during the simulation.
- Any structure consisting of adjacent or overlapping rectangles with any combination of materials may be simulated. Up to 62500 (250·250) nodes may be used.
- Boundary conditions may be a given heat flow, or a temperature with a surface resistance. Temperatures and heat flows may vary in time by one of the following functions: sinusoidal, stepwise constant or linear (data may be imported/exported from/to programs such as Excel).
- Available modifications: heat sources/sinks, internal boundaries of prescribed temperature, internal regions containing air or fluid of a single temperature, internal resistances, radiation inside cavities.

1.2.3 Cartesian coordinates

HEAT2 solves most of the heat conduction problems that can be described in Cartesian coordinates. This tie to rectangular coordinates may be a limitation in some cases, but it often reduces the amount of input needed from the user. In addition to this, building technology is often a “Cartesian world”. Sloped boundaries, such as sloping insulation under a concrete slab edge, have to be modeled in HEAT2 by steps. Figure 1.1 shows a Cartesian example.

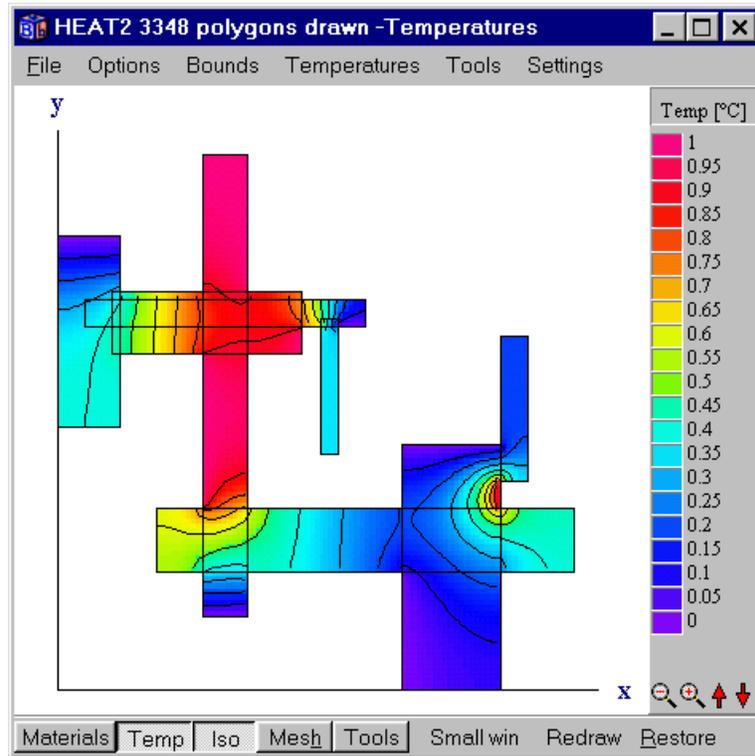


Figure 1.1: A Cartesian example. All boundaries are parallel to the x,y -axis.

1.2.4 Numerical method

HEAT2 solves the heat conduction equation by the method of explicit finite differences, as described in Chapter 3. The user chooses the computational mesh, and the stable time-step is calculated automatically. In the steady-state case, fast computation is achieved by over-relaxation.

The time required to solve a reasonably complicated steady-state problem with 10000 computational nodes is usually less than a minute on a Pentium.

1.2.5 Preprocessor HEAT2W

HEAT2W is a CAD-like drawing program that makes it even simpler to generate input for a wide range of heat transfer problems. The geometry is built using rectangles of different materials that may overlap each other. HEAT2W generates an input file that is imported into HEAT2. The user gives the boundary conditions in HEAT2 before solving the problem. Figure 1.2 shows a construction (sliding windows) drawn in HEAT2W. Figure 1.3 and Figure 1.4 show the numerical mesh and thermal conductivities, and the calculated temperatures and isotherms as drawn by HEAT2, respectively.

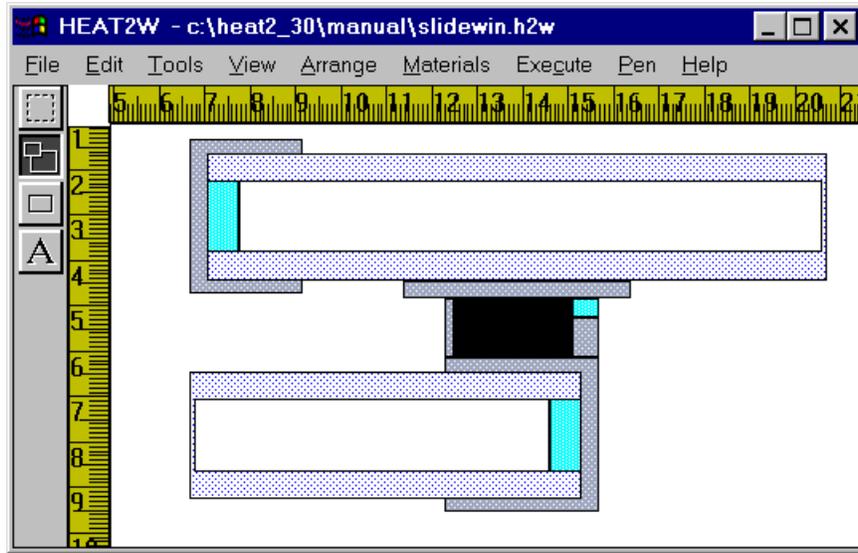


Figure 1.2: A construction (sliding windows) as drawn in HEAT2W.

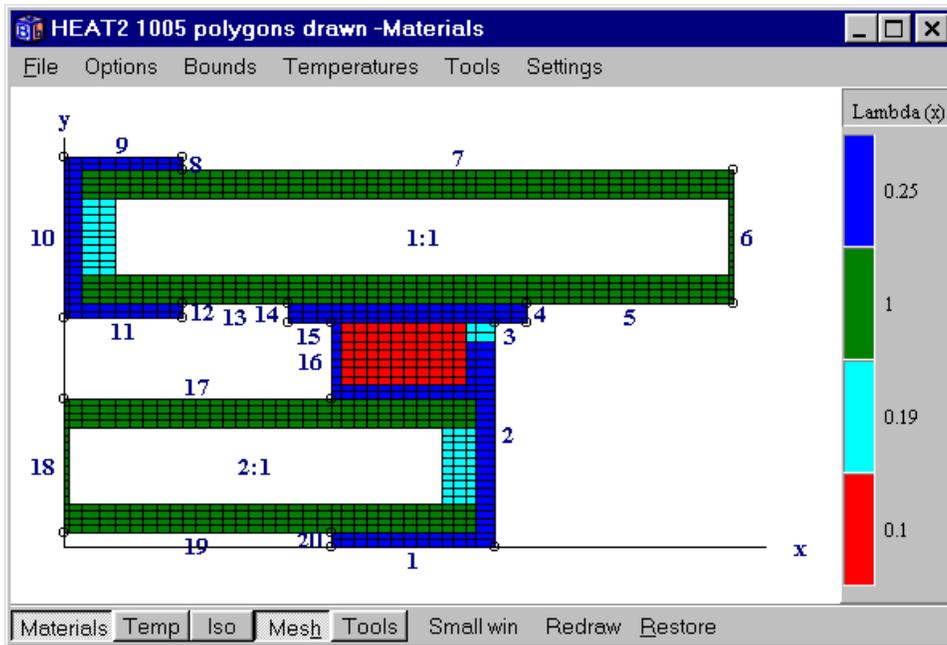


Figure 1.3: Numerical mesh and thermal conductivities as shown in HEAT2.

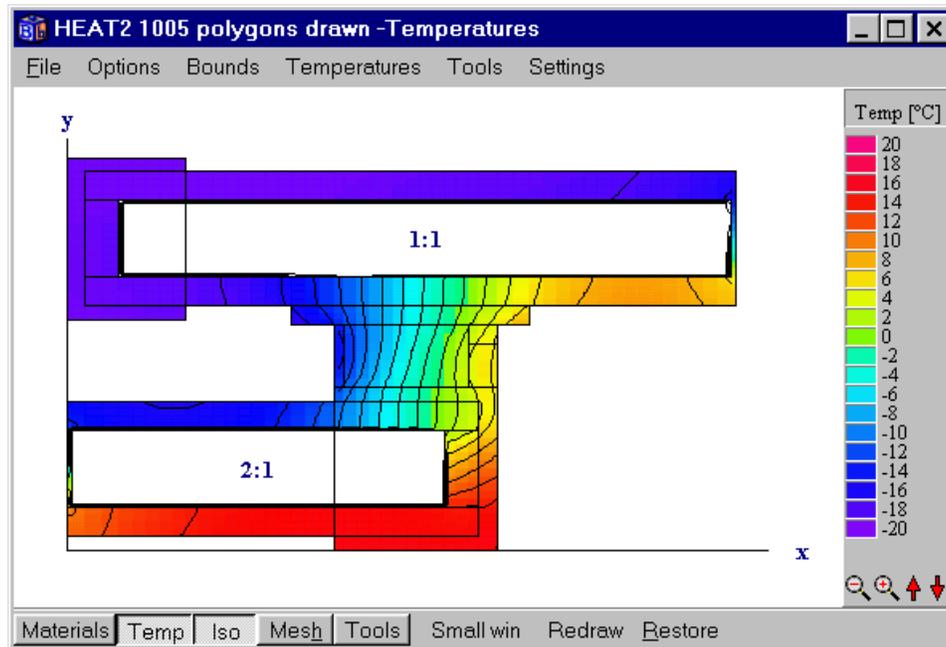


Figure 1.4: Calculated temperatures and isotherms (bottom) as shown in HEAT2.

1.2.6 System requirements

Windows 95/98/NT is required. A PC with a Pentium processor and 16 MB RAM is recommended (HEAT2 needs 4 MB RAM).

1.2.7 Developers

Lund Group for Computational Building Physics, Dept. of Building Physics, Lund University, Sweden.

1.2.8 Update info

HEAT2 version 3.0 for Windows 95/98/NT has the following new features compared to version 2.6 for DOS:

- HEAT2 3.0 is much faster than version 2.6. The improvement is up to 4-5 times for steady-state calculations and 2-3 times for transient simulations (the difference is smaller for problems with radiation inside cavities). This is due to better optimization by the compiler and some numerical improvements in code due to the 32-bits environment.
- The maximum number of nodes is increased 4 times from 15625 (125·125) to 62500 (250·250). The maximum number of radiation nodes is now 250 (126 in earlier version).
- Significant improvement of the graphical representation of output data, such as zoom, panning, rotation, color/gray-scale, fast printing of e.g. isotherms using high resolution (600 DPI).
- Temperatures and heat flows may be recorded in time (transient) or iteration by iteration (steady-state) and shown graphically by a chart even during the simulation. Each record, that can hold up to about half a million points, may be saved to file (*.REC). The chart may be saved or cut to the clipboard in Metafile (*.EMF, *.WMF) or Bitmap (*.BMP) format. Chart properties may be edited (axis, titles, legends, and so on).

- The function step list (linear or step-wise constant) can now hold up to about half a million points. The function step list may be saved to file (*.FUN) and used by other problems in HEAT2 and HEAT3. The data may be imported/exported via the clipboard to programs such as Excel.
- It is now possible to automatically insert input mesh coordinates to redefine existing problems.

1.3 About this manual

This manual is divided into two parts. The first part is about the mathematical description (Chapter 2) and the numerical formulation (Chapter 3). The second part deals with input data, the integrated environment, and gives a few examples (Chapter 4 and onwards). It is not necessary to read the first part to use the program.

1.4 Installation

Before using the installation diskette, it is recommended to make a backup copy. To install HEAT2 and HEAT2W run the SETUP.EXE program from Windows. The following steps describe this process in detail:

1. Start Windows (if not already started).
2. Insert the installation diskette
3. Start the setup program (SETUP.EXE).
4. Follow the instructions on the screen.

The following files will be installed:

- HEAT2.EXE is the main program
- HEAT2W.EXE is a pre-processor that can be used
- HEAT2W.INI is used by HEAT2W
- MTRL30.TXT is a file with thermal properties that may be edited (used by HEAT2)
- Files with extension DAT are HEAT2 input data files
- Files with extension H2W are HEAT2W input data files

1.5 Technical support

News are regularly updated on www.blocon.se. Questions and comments may be sent by email to info@blocon.se.

2. Mathematical description

2.1 Governing differential equations

The governing partial differential heat conduction equation in two dimensions for the temperature $T(x,y,t)$ is

$$\frac{\partial}{\partial x} \left(\lambda_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda_y \frac{\partial T}{\partial y} \right) + I(x, y, t) = C \frac{\partial T}{\partial t} \quad (2.1)$$

Here I , (W/m^3), is the rate of internal heat generation. The thermal conductivities in the x,y -directions are denoted by λ_x and λ_y , ($\text{W}/(\text{m}\cdot\text{K})$), respectively. The volumetric heat capacity is denoted by C , ($\text{J}/(\text{m}^3\cdot\text{K})$), which is the density ρ , (kg/m^3), times the specific heat capacity c_p , ($\text{J}/(\text{kg}\cdot\text{K})$), i.e. $C=\rho\cdot c_p$. The thermal conductivities in the two directions are usually the same ($\lambda_x = \lambda_y$). The internal heat generation is often zero. In the steady-state case, the right-hand side of Eq. (2.1) is zero.

2.2 Boundary conditions

There are two main types of boundary conditions that can be applied to the boundary segments ($b.s.$). The first type gives a prescribed temperature of the surrounding region, $T_{b.s.}(t)$, and a given surface resistance R , ($\text{m}^2\cdot\text{K}/\text{W}$):

$$T_{b.s.}(t) - T|_{surf} = R \cdot (-\lambda) \frac{\partial T}{\partial n} \Big|_{surf} \quad (^\circ\text{C}) \quad (2.2)$$

Here, $\partial T/\partial n$, is the derivative in the normal direction. The second type gives a prescribed heat flow into the region:

$$(-\lambda) \frac{\partial T}{\partial n} \Big|_{bs} = f(t) \quad (\text{W}/\text{m}^2) \quad (2.3)$$

Figure 2.1 shows an internal boundary separating two different materials of thermal conductivities λ_1 and λ_2 . The temperature is of course continuous at the boundary. The normal to the boundary is denoted by \hat{n} . There is a continuity of the heat flow across the boundary. The condition of continuous heat flow perpendicular to the boundary is

$$\lambda_1 \frac{\partial T}{\partial n} \Big|_1 = \lambda_2 \frac{\partial T}{\partial n} \Big|_2 \quad (2.4)$$

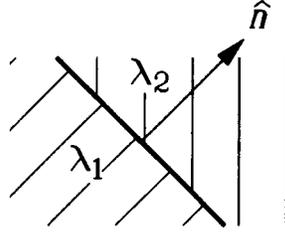


Figure 2.1: The normal \hat{n} at an internal boundary.

There may be a contact resistance R_{ins} , ($\text{m}^2\cdot\text{K}/\text{W}$), between two regions, see Figure 2.2. In this case, the temperature is different on the two sides of the contact resistance. The condition for continuous heat flow at this internal insulation is:

$$\lambda_2 \left. \frac{\partial T}{\partial n} \right|_2 = \frac{T|_2 - T|_1}{R_{ins}} = \lambda_1 \left. \frac{\partial T}{\partial n} \right|_1 \quad (2.5)$$

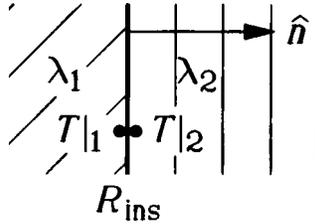


Figure 2.2: Case involving internal insulation.

2.3 Initial conditions

The initial temperature distribution at time $t = t_{start}$ is in two dimensions denoted by $T(x, y, t_{start})$. In the steady-state case, the initial temperatures are irrelevant to the solution.

2.4 Fluid regions

Consider Figure 2.3 in which S denotes the boundary surface of an internal region containing a fluid with a volumetric heat capacity C_{fluid} , ($\text{J}/(\text{m}^3\cdot\text{K})$). The inward normal for the region is denoted by \hat{n} , and the volume of the fluid by V_{fluid} , (m^3). The heat flow into the area Q_{in} , (W), is obtained by an integral over the boundary surface S . The flow equals the rate of increase of the heat content in the fluid:

$$Q_{in} = \oint -\lambda \frac{\partial T}{\partial n} dS = C_{fluid} \cdot V_{fluid} \frac{\partial T_{fluid}}{\partial t} \quad (\text{W}) \quad (2.6)$$

In the case involving a cavity filled with air, the heat capacity may be put to zero and the right-hand side of Eq. (2.6) becomes zero.

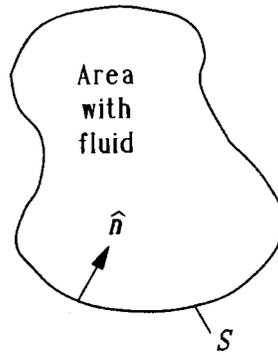


Figure 2.3: A fluid region enclosed by the boundary surface S .

2.5 Heat conduction coupled to radiation in a cavity

Heat conduction in solid building parts coupled to long-wave thermal radiation in air cavities is a frequently occurring and studied type of process in building physics. The numerical simulation of this coupling poses particular problems. An equation system for the radiative exchange coupled to convective heat transfer in the cavity and heat conduction to the nodes in the numerical mesh must be solved at each time.

A systematic analysis of this problem is presented in (Blomberg, 1996). The number of original $(2N+1)$ equations is reduced as much as possible (to N equations). The final formulas, and in particular the iterative process to account for the nonlinearity due to the forth-power radiation law, are formulated to suit computer modelling.

3. Numerical formulation

3.1 Introduction

In the numerical formulation, the partial differential equation is replaced by a discrete approximation. The temperature field is approximated by values at discrete points. This gives a computational mesh. The field is considered at consecutive time-steps with a time increment Δt . In this study, only Cartesian meshes will be used. The increments in the x -, and y -directions are denoted by Δx_i , and Δy_j , respectively. The smaller these increments are, the better is the agreement with the "true" temperature distribution.

3.2 Computational mesh

Consider a two-dimensional body that is divided into increments in the x - and y -directions as shown in Figure 3.1. This rectangular mesh may have computational cells of different sizes. The width and height of cell (i,j) are denoted by Δx_i and Δy_j , respectively. The temperature in the midpoint of cell (i,j) at the considered time-step is denoted by $T_{i,j}$.

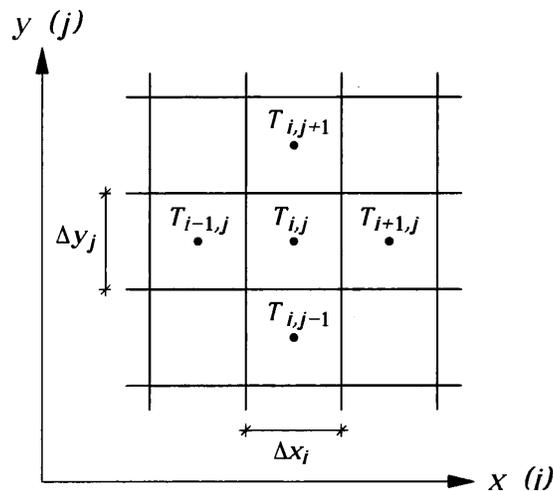


Figure 3.1: Choice of indices for the cells in the computational mesh.

3.3 Thermal conductances

The thermal coupling between the cells in the numerical mesh is described by thermal conductances. Figure 3.2 shows the notation of indices for the conductances.

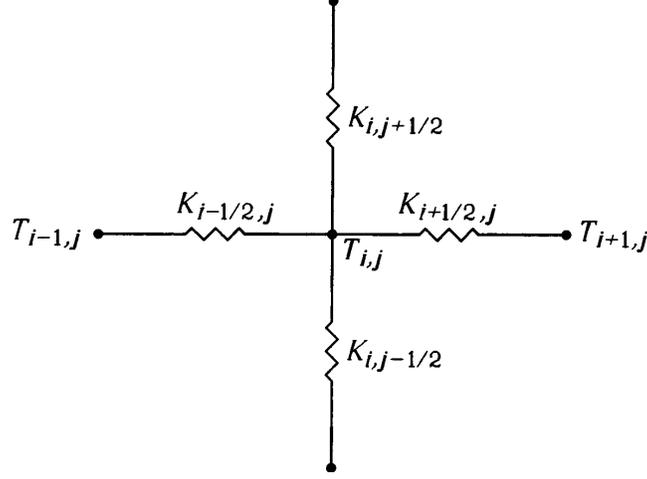


Figure 3.2: The thermal conductances between cell (i,j) and neighboring cells.

The conductance per unit length perpendicular to the (x,y) -plane between the two cells $(i-1,j)$ and (i,j) is denoted by $K_{i-1/2,j}$, (W/(m·K)). It is calculated as

$$K_{i-1/2,j} = \frac{\Delta y_j}{\Delta x_{i-1} / (2 \cdot \lambda_{x_{i-1,j}}) + \Delta x_i / (2 \cdot \lambda_{x_{i,j}}) + R_{i-1/2,j}} \quad (\text{W}/(\text{m} \cdot \text{K})) \quad (3.1)$$

Here $\lambda_{x_{i,j}}$ is the thermal conductivity in the x -direction for cell (i,j) . The conductance refers to the total heat flow through the side Δy_j per unit length. The first term in the denominator is the thermal resistance in the x -direction for half of the cell $(i-1,j)$, the second term being the resistance for half of the cell (i,j) . The third term $R_{i-1/2,j}$, ($\text{m}^2 \cdot \text{K}/\text{W}$), is an optional additional thermal resistance at the interface between the two cells $(i-1,j)$ and (i,j) as shown in Figure 3.3. For the cell $(1,j)$ lying at the boundary as shown in Figure 3.5, the conductance $K_{1/2,j}$ is calculated as

$$K_{1/2,j} = \frac{\Delta y_j}{\Delta x_1 / (2 \cdot \lambda_{x_{1,j}}) + R_{1/2,j}} \quad (3.2)$$

Here $R_{1/2,j}$, ($\text{m}^2 \cdot \text{K}/\text{W}$), is the boundary surface resistance.

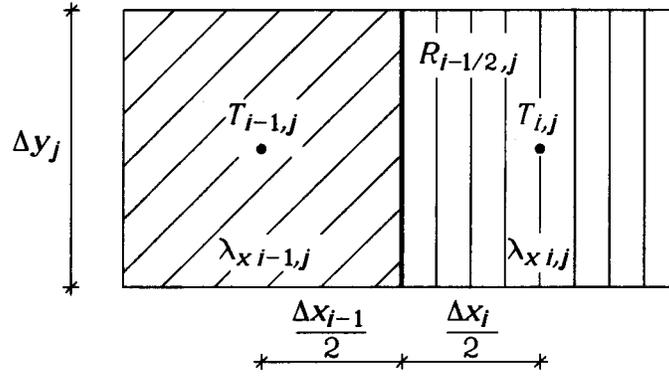


Figure 3.3: Thermal resistance between cell $(i-1, j)$ and cell (i, j) .

3.4 Heat flows

Figure 3.4 shows the four heat flows associated with an internal cell.

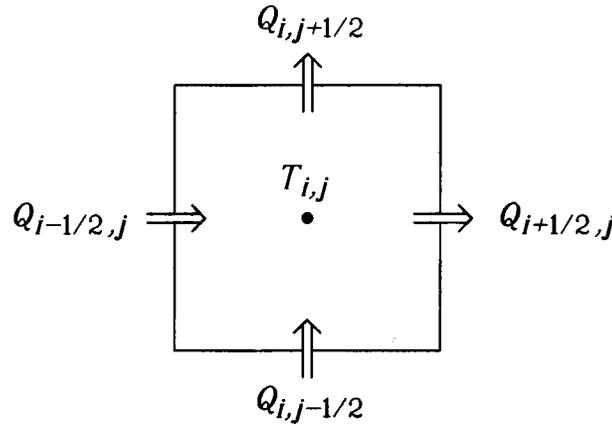


Figure 3.4: The four heat flows of cell (i, j) .

The heat flow through the left boundary $Q_{i-1/2, j}$, (W/m), is

$$Q_{i-1/2, j} = K_{i-1/2, j} \cdot (T_{i-1, j} - T_{i, j}) \quad (\text{W/m}) \quad (3.3)$$

The heat flow through the lower boundary $Q_{i, j-1/2}$ is

$$Q_{i, j-1/2} = K_{i, j-1/2} \cdot (T_{i, j-1} - T_{i, j}) \quad (\text{W/m}) \quad (3.4)$$

The heat flow through a boundary cell is determined by the boundary condition. Consider, for example, the outer boundary cell $(1, j)$ in Figure 3.5. If a temperature is given, Eq. (3.3) is modified to

$$Q_{1/2, j} = K_{1/2, j} \cdot (T_{b.s.}(t) - T_{1, j}) \quad (\text{W/m}) \quad (3.5)$$

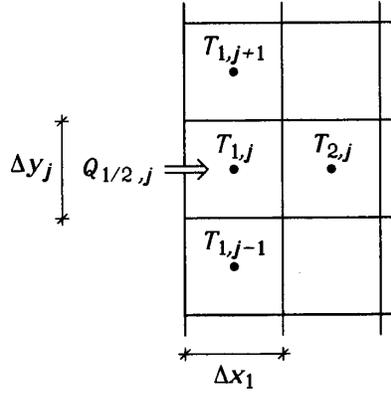


Figure 3.5: Heat flow $Q_{1/2,j}$ through the outer boundary of boundary cell (I,j) .

If a boundary heat flow $q_{b.s.}$ (W/m²), is given, the heat flow per unit length perpendicular to the (x,y) -plane is

$$Q_{1/2,j} = q_{b.s.}(t) \cdot \Delta y_j \quad (\text{W/m}) \quad (3.6)$$

The accumulated heat flow during a time-step Δt for the left boundary segment to which cell (I,j) belongs is

$$E_{b.s.}^{new} = E_{b.s.} + \Delta t \cdot \sum_j Q_{1/2,j} \quad (\text{J/m}) \quad (3.7)$$

where the summation involves all cells that belong to the boundary segment.

3.5 New temperatures

The heat capacity of cell (i,j) becomes $C_{i,j} \Delta x_i \Delta y_j$ per unit length perpendicular to the (x,y) -plane. Here $C_{i,j}$, (J/(m³·K)), is the volumetric heat capacity for cell (i,j) .

The increase of the energy for an internal cell during time-step Δt is given by the energy balance below. In the formula, $T_{i,j}$ is the old temperature of cell (i,j) , and $T_{i,j}^{new}$ the new one. The development in time is incremented by Δt , and the new time becomes $t^{new} = t + \Delta t$. The heat generation in cell (i,j) is denoted by $I_{i,j}$, (W/m³). A negative value indicates a heat sink. The heat generation is often zero.

$$C_{i,j} \Delta x_i \Delta y_j \cdot (T_{i,j}^{new} - T_{i,j}) = (Q_{i-1/2,j} - Q_{i+1/2,j} + Q_{i,j-1/2} - Q_{i,j+1/2} + I_{i,j} \Delta x_i \Delta y_j) \cdot \Delta t \quad (3.8)$$

The final equation for the new temperature becomes

$$T_{i,j}^{new} = T_{i,j} + \frac{\Delta t}{C_{i,j} \Delta x_i \Delta y_j} (Q_{i-1/2,j} - Q_{i+1/2,j} + Q_{i,j-1/2} - Q_{i,j+1/2} + I_{i,j} \Delta x_i \Delta y_j) \quad (3.9)$$

3.6 Choice of time-step,

The stable time-step Δt for cell (i,j) is determined from the following stability criterion:

$$\Delta t < \frac{C_{i,j}\Delta x_i\Delta y_j}{K_{i-1/2,j} + K_{i+1/2,j} + K_{i,j-1/2} + K_{i,j+1/2}} \quad \text{for all } i \text{ and } j \quad (3.10)$$

This criterion must be satisfied for all cells (i,j) . The smallest stable time-step obtained is used for all cells to guarantee stability. The analysis leading to this criterion is not given here. The reader is referred to (Efring, 1990).

3.7 Iterative calculation

Equation (3.9) gives the new temperature based on the change in energy during the time-step Δt . This calculation is made for all cells. The updated temperatures give the new heat flows between the cells according to Eqs (3.3)-(3.6), which in turn change the temperatures again, and so on.

3.8 Steady-state case

Transient problems are solved with the above method of explicit forward differences. This means that the old temperatures are used to calculate the heat flows. The new temperatures $T_{i,j}^{new}$ are calculated by Eq. (3.9).

The successive over-relaxation method is used in the steady-state case, see (Hirsch, 1992). Here, the temperatures are calculated in the same way as with explicit forward difference, but new temperatures are used in the formulas as they arise. The temperatures are calculated using an over-relaxation factor ω that lies in the range 1.0-2.0. An optimized ω may give calculation times between 1/50th and 1/10th of that required for a calculation not using over-relaxation ($\omega=1.0$). The optimized ω typically lies in the range 1.8-2. In HEAT2 (and in HEAT2R and HEAT3) this factor is initially set to 1.95. Equation (3.9) is modified to

$$T_{i,j}^{new} = T_{i,j} + \frac{\Delta t \cdot \omega}{C_{i,j}\Delta x_i\Delta y_j} (Q_{i-1/2,j} - Q_{i+1/2,j} + Q_{i,j-1/2} - Q_{i,j+1/2} + I_{i,j}\Delta x_i\Delta y_j) \quad (3.11)$$

The heat capacities of the cells do not matter in the steady-state solution. The stable time-step for each cell determines the time-scale for temperature changes within the cell. It is better if all cells have the same stable time-step, which means the thermal response time for each cell is the same. Accordingly, the heat capacities $C_{i,j}$ are chosen to give the same time-step for all cells. The heat capacities are determined by putting Δt equal to the right-hand side in (3.10). The chosen capacities are then

$$C_{i,j} < \Delta t \cdot \frac{K_{i-1/2,j} + K_{i+1/2,j} + K_{i,j-1/2} + K_{i,j+1/2}}{\Delta x_i\Delta y_j} \quad (3.12)$$

Actually, the choice of Δt does not matter since it cancels in Eq. (3.11).

3.9 Internal regions containing fluid

In some cases there is a need to model an internal area containing liquid fluid or air. An example of this is rock cavities with water used for heat storage.

Two cases are considered when modelling an internal region with a fluid: the fluid has heat capacity, or the fluid has negligible heat capacity.

For the given fluid volume V_{fluid} with the volumetric heat capacity C_{fluid} , the temperature of the fluid T_{fluid} is calculated from the energy balance involving the surrounding cells. Equation (2.6) becomes in a discrete approximation:

$$C_{fluid} \cdot V_{fluid} \cdot (T_{fluid}^{new} - T_{fluid}) = \sum_{n=1}^N K_n \cdot (T_n - T_{fluid}) \cdot \Delta t \quad (3.13)$$

The temperatures in the cells adjacent to the fluid region are denoted by T_n . The thermal conductance between the center of the cell in question and the fluid is denoted by K_n . The total number of cells, N , enclosing the fluid volume is taken into account in calculating the new fluid temperature, T_{fluid}^{new} .

Air may be approximated as a fluid with negligible heat capacity. The air temperature T_a inside an internal region becomes an average of the temperatures in the surrounding cells:

$$T_a = \frac{\sum_{n=1}^N (K_n \cdot T_n)}{\sum_{n=1}^N K_n} \quad (3.14)$$

4. Overview of input

4.1 Input mesh

An *input mesh* facilitates the description of the geometry, the numerical mesh, and the boundary conditions. Consider first as a rather simple example the L-shaped region in Figure 4.1. The required input mesh is shown. In general, I, J are the *input mesh coordinates*. The *mesh segment I* between the mesh coordinates $I-1$ and I has the length ΔX_I . Respectively, the segment J between coordinates $J-1$ and J has the length ΔY_J . The number of input mesh segments, I_{\max} and J_{\max} , respectively, is given in each direction. In Figure 4.1 there are two in both directions.

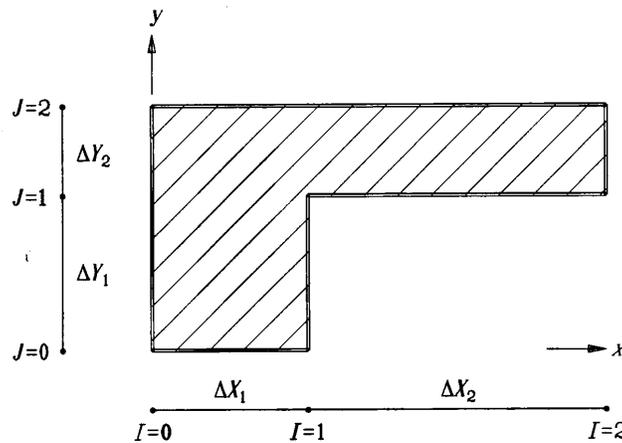


Figure 4.1: An simple example with a suitable input mesh.

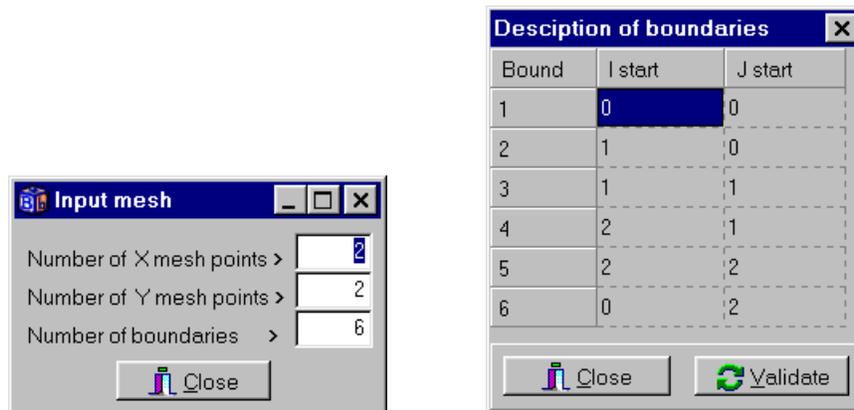


Figure 4.2: The number of mesh coordinates in each direction and the number of boundary segment have to be given first (left figure). The computational area is then defined by enumerating the boundary segments consecutively counterclockwise using the input mesh coordinates (right figure).

The computational region is enclosed by a number of straight lines, which are called *boundary segments*. The boundary segments are defined consecutively counterclockwise using the input mesh coordinates. The boundary segments will be numbered in the order that they are given. It makes no difference which one of the boundaries that is taken first. Starting at $(0,0)$, it would be

$(I,J)=(0,0), (1,0), (1,1), (2,1), (2,2), (0,2)$, and back to $(0,0)$, see Figure 4.2. Figure 4.3 shows the boundary segments 1..6.

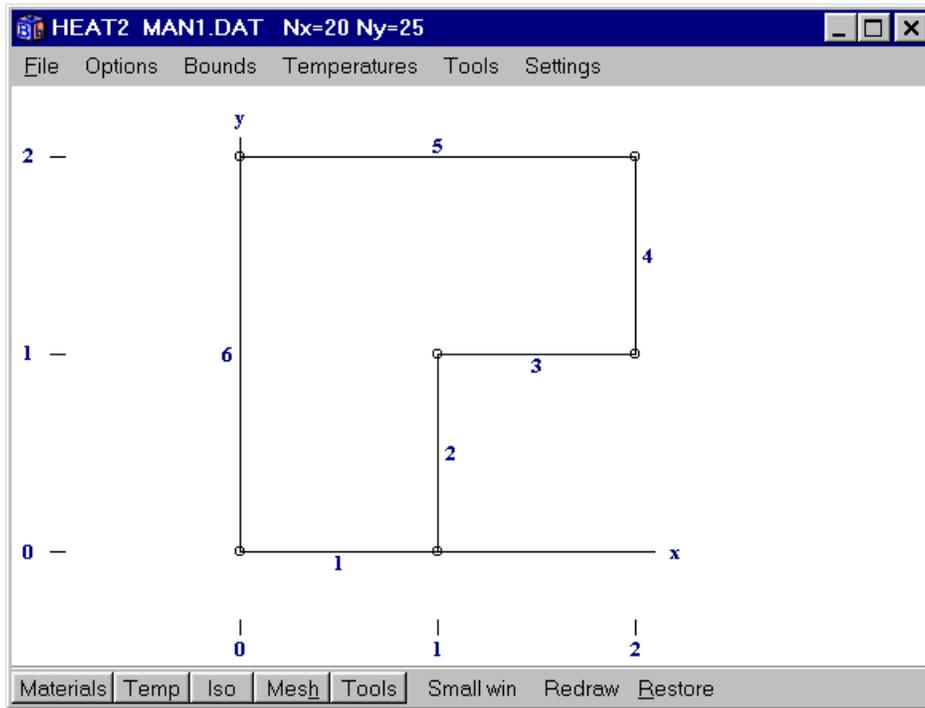


Figure 4.3: The six boundary segments defines the computational area.

4.2 Numerical mesh

The lengths ΔX_I and ΔY_J , and the number of computational cells N_I and N_J are specified for each mesh segment, see Figure 4.4. The cells are normally placed in an equidistant mesh. The computational cells in each segment have the same size, see Figure 4.5. In general, capital letters refer to the input mesh (e.g. ΔX_I), while small letters refer to the numerical mesh (e.g. Δx_i). There must be at least one numerical cell in each input mesh segment. In Figure 4.5 there are 10 cells in both segments in the x -direction. In the y -direction, there are 15 cells in the first segment, and 10 cells in the second one. The length of each segment is also shown in the figure.

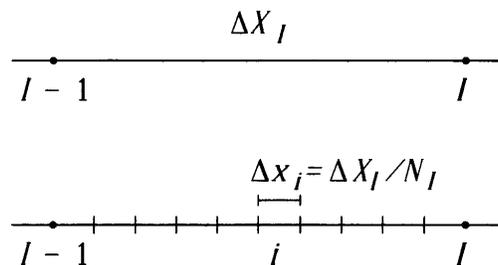


Figure 4.4: The length Δx_i in the computational mesh.

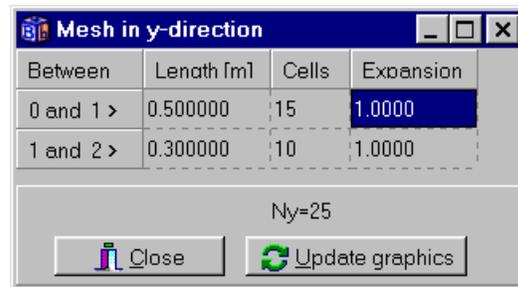
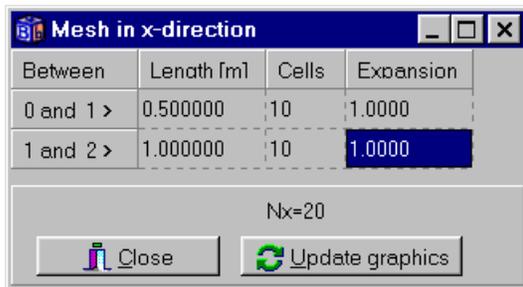
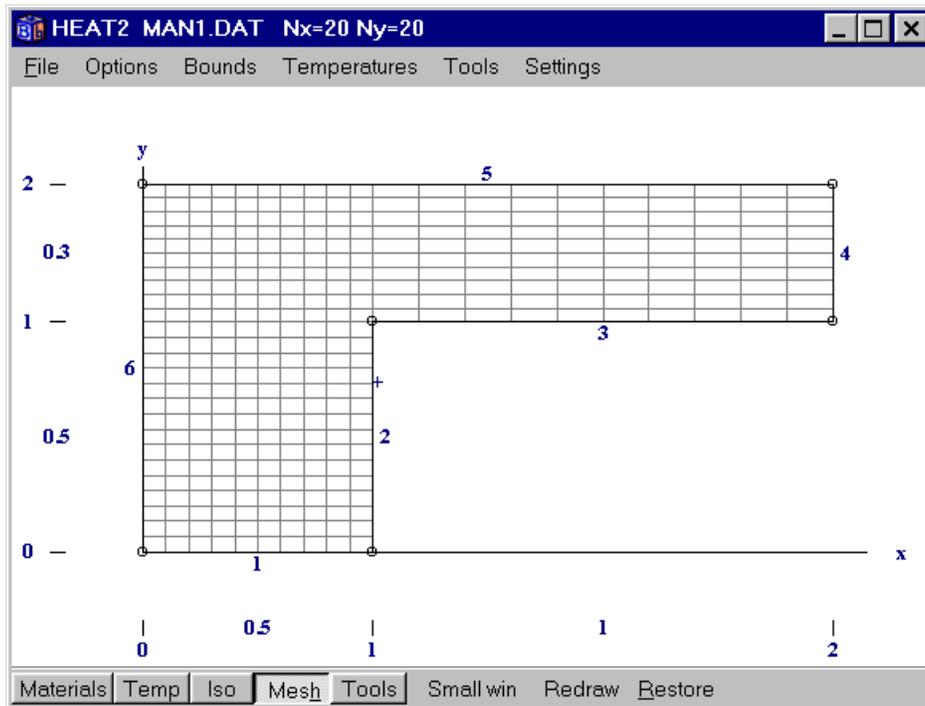


Figure 4.5: A numerical mesh is shown here with 20·25 cells (equidistant in each mesh segment).

Expansive meshes may be used to concentrate the cells towards areas with large temperature gradients. An expansion coefficient is given for each mesh segment. The lengths of the successive computational cells in each segment will be increasing or decreasing by this factor. For example, if the length in a segment with three cells is 21 m, the length of each computational cell will be 7 m in an equidistant mesh (the expansion coefficient is 1). A coefficient of 2.0 will give lengths of 3.0, 6.0 and 12.0 m. Using 0.5 will give a decreasing mesh with 12.0, 6.0 and 3.0 m, respectively.

Be aware that an expansive mesh may lead to small computational cells causing the time-step to be very short. A badly chosen computational mesh can increase the computational time dramatically, especially for transient analyses. Do not use extreme expansion coefficients with a large number of cells that would cause very small cells in a segment. "Normal" values are in most cases those between 0.8-1.25.

Figure 4.6 shows an expansive mesh. The first segment in the x -direction has an expansion coefficient of 0.9, i.e. the cell sizes are decreasing in the x -direction. The coefficient is 1.2 in the second segment yielding increasing sizes. The expansion in the y -direction is 0.95 and 1.2,

respectively. A proper mesh can usually be obtained by trying a few different expansion factors and visually checking the generated mesh grid.

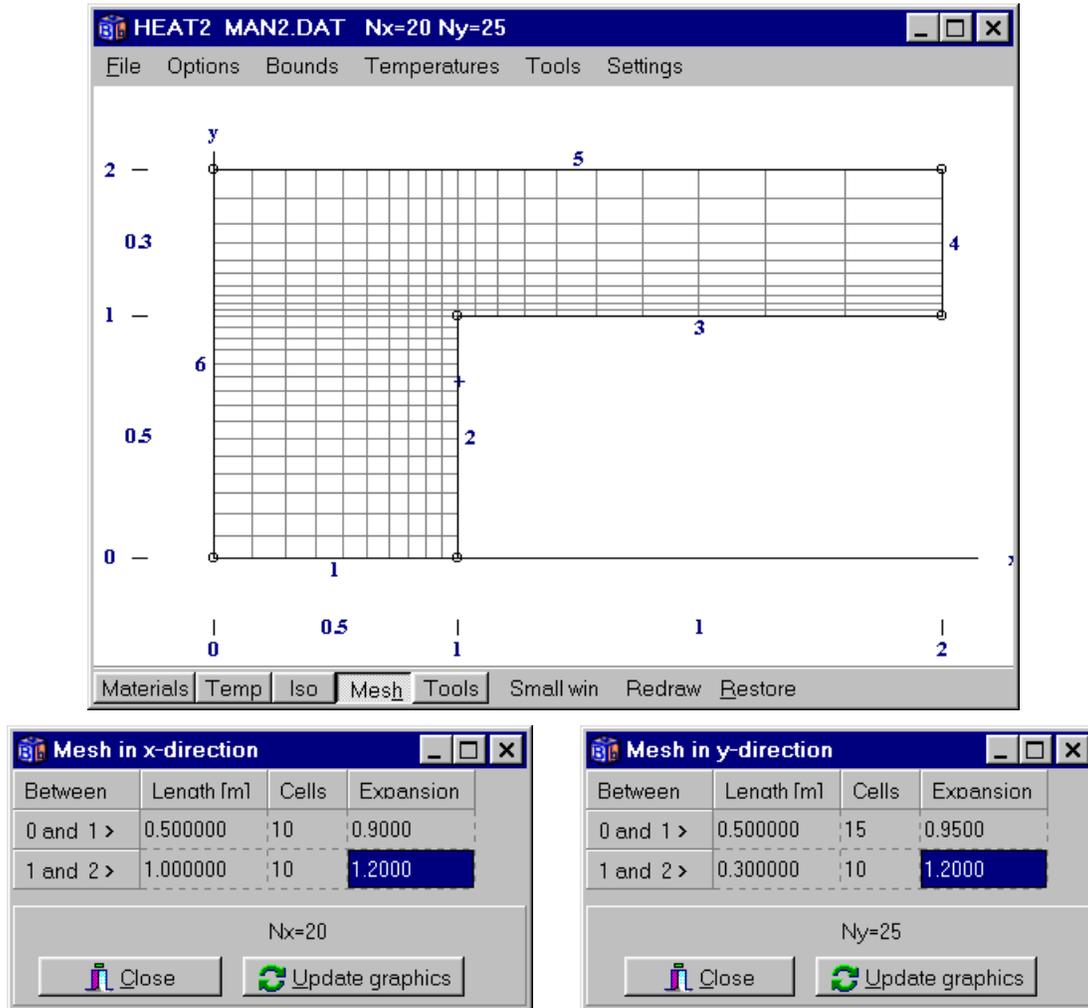


Figure 4.6: An expansive mesh with smaller cells near the inner corner (the temperature gradient is higher in the corner requiring smaller cells to decrease the numerical error).

4.3 Thermal properties

The thermal properties are defined by the thermal conductance λ , (W/(m·K)), and the volumetric heat capacity C , (J/(m³·K)), which is the density times the specific heat capacity ($C=\rho \cdot c_p$). The thermal conductivity may differ in the x - and y -directions (λ_x, λ_y).

The thermal properties are specified in the following way. Default thermal properties are first given (λ_x, λ_y, C) that will be valid for the whole region. Then, rectangular regions with other thermal properties are defined by the lower left and upper right input mesh coordinates. The rectangles may overlap each other.

Consider the problem in Figure 4.7 where two materials are specified. The green (upper right) area shows a rectangle with lower left coordinate (1,1) and upper right coordinate (2,2).

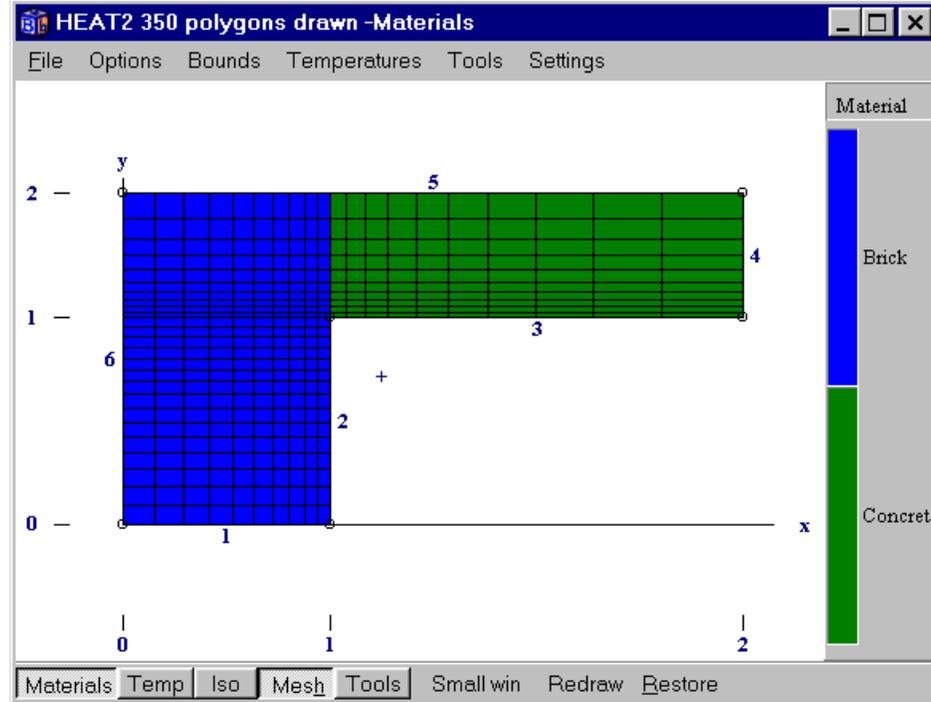
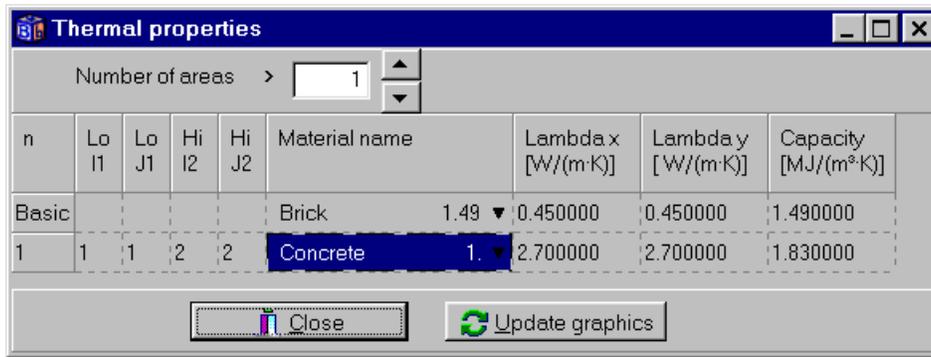


Figure 4.7: The materials are defined by giving default thermal properties that are valid for the whole computational area, and then using overlapping rectangles of different properties. The green (upper right) area shows a rectangle with lower left coordinate (1,1) and upper right coordinate (2,2).

4.4 Boundary conditions

Different *types* of boundary conditions may be defined. A boundary condition type may either be a temperature T , ($^{\circ}\text{C}$), with a surface resistance between the air and the surface R , ($\text{m}^2\cdot\text{K}/\text{W}$), or a given heat flow q , (W/m^2). The temperatures and the heat flows may be constant in time, or time-dependent using a sinusoidal, a step-wise constant, or a step-wise linear function. See definitions in Section 6.7.

The user defines the boundary conditions in a list and assigns them to the boundary segments. Consider Figure 4.8. There are in this case three boundary conditions with list numbers $n=1, 2$, and 3 in the first column. The second column shows what boundary segments that are assigned to a certain type (the boundary segment numbers 1..6 are shown outside the computational area in the figure). By default, all boundary segments will use the first type. Boundary segments 2 and 3

are assigned boundary condition type 2 ($T=20$, $R=0.13$). Boundary segments 5 and 6 are assigned type 3 ($T=0$, $R=0.04$). All other boundary segments (in this case 1 and 4) will have the default first type, i.e. an adiabatic boundary condition ($Q=0$).

If a boundary segment number appears in more than one type list, the latter will prevail. The boundary conditions may also be shown graphically, see Figure 4.9.

n	type	Bounds	function	Q [W/m ²]	Temp [C]	Res. [m ² K/W]
1	Q=const	Default	0			
2	T=const	2 3			20	0.130000
3	T=const	5 6			0	0.040000

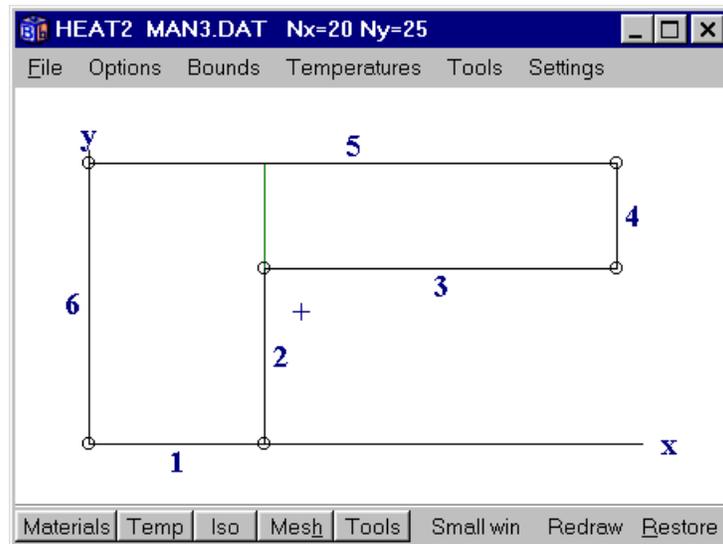


Figure 4.8: The boundary conditions are assigned to the corresponding boundary segments numbers 1..6.

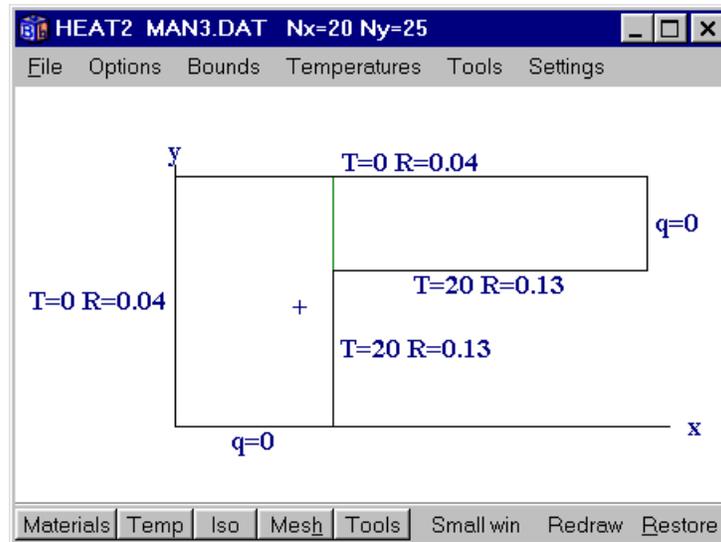


Figure 4.9: The boundary conditions may also be shown graphically by enabling item *Bounds/BC type values*.

4.5 Initial temperatures

The initial temperatures are given in a similar way as the thermal properties. A default temperature is first given that will be valid for the whole region. Overlapping rectangular regions with other temperatures are defined by the lower left and upper right input mesh coordinates.

Consider Figure 4.10. In this case, two different temperatures are used. The default (basic) temperature 20 °C is first assigned the whole area. Secondly, a rectangle defining a temperature of 10 °C for the upper (green) part is given.

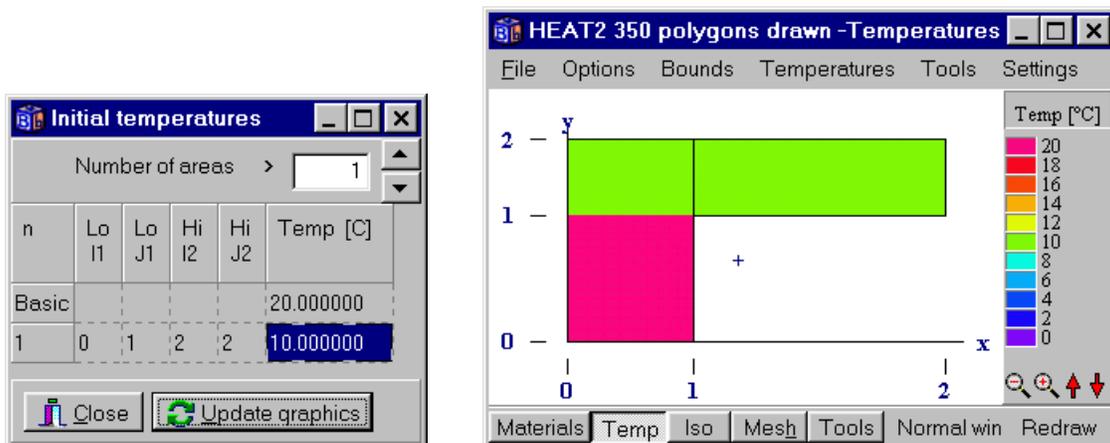


Figure 4.10: Initial temperatures are defined by giving a default (basic) temperature that is valid for the whole computational area, and then using overlapping rectangles assigning different temperatures. In this case, two different temperatures are used.

4.6 Solving the problem for steady-state

To start a steady-state simulation select *Solve/Start steady-state calculation*. A window, see Figure 4.11, will be displayed showing information such as the chosen stop criterion, number of iterations, and errors for temperature and flow, respectively. This problem is solved in about a second. Figure 4.12 shows the calculated temperature field and isotherms. The heat flows may be viewed by checking *Bounds/Heat flows* in the graphics window. A list is shown in Figure 4.13 (*Output/Boundary flows*).

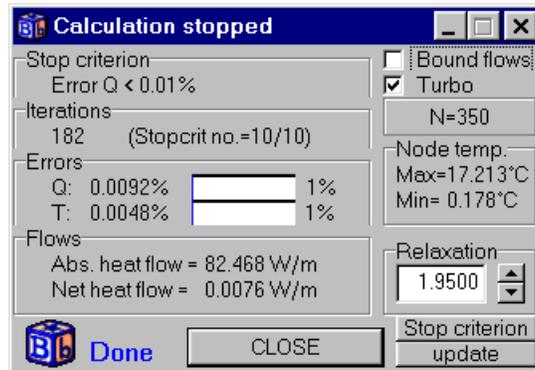


Figure 4.11: Window shown during steady-state calculation.

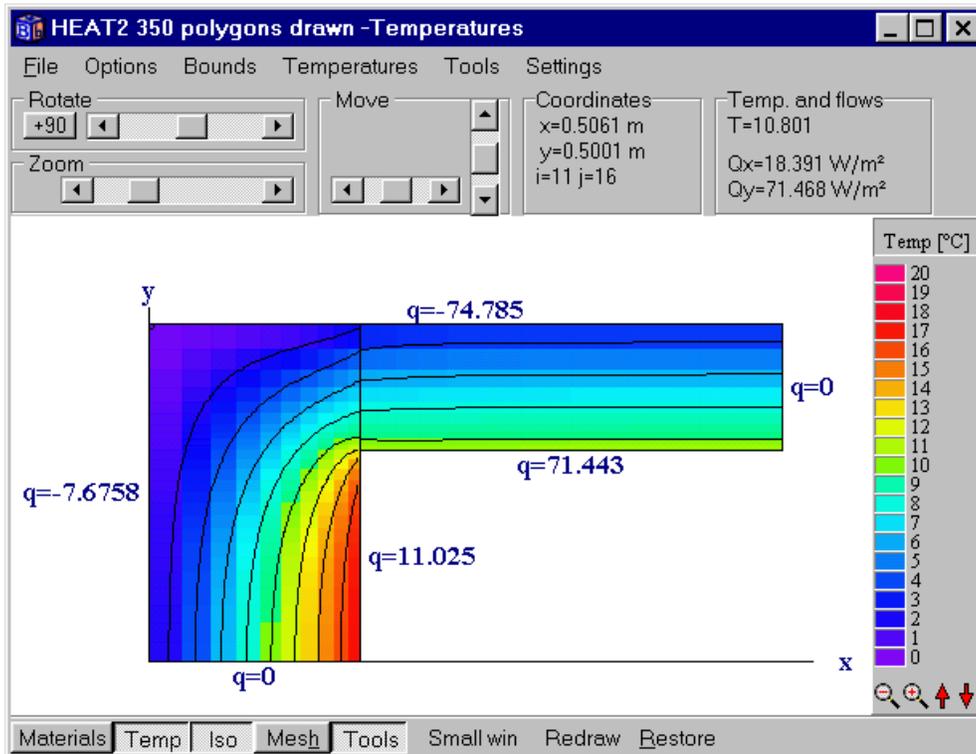


Figure 4.12: Calculated temperature field and isotherms.

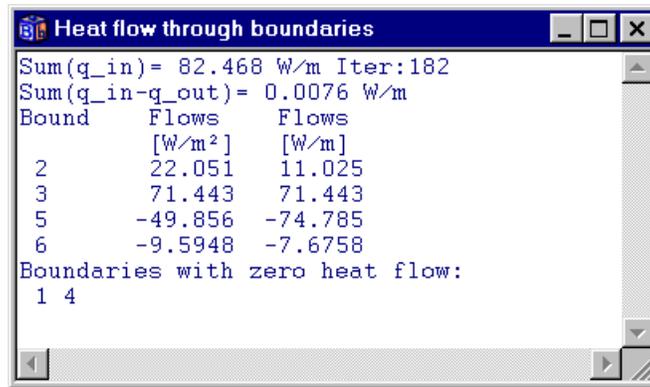


Figure 4.13: Heat flows through the boundaries.

4.7 Solving a transient problem

In the following example, a transient calculation using a sinusoidal variation on the external side of the corner will be carried out. First, define a sinus function (item *Functions/Function 1*) with a daily variation between 0 and 20 °C with maximum at noon, see Figure 4.14.

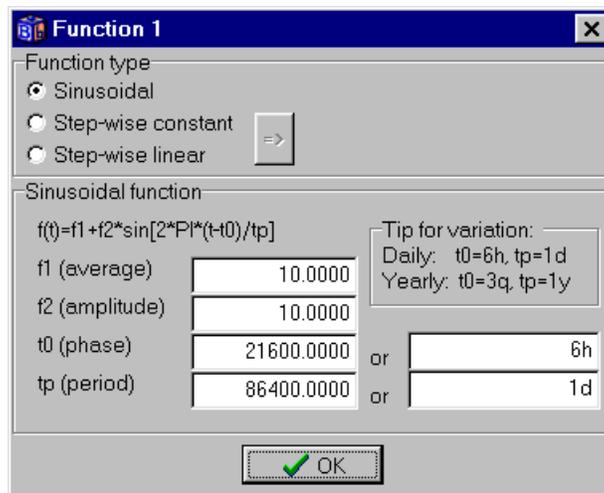


Figure 4.14: A sinus function with a daily variation between 0 and 20 °C with maximum occurring at noon. The time for the phase and period may be given either in seconds or a time string.

Next step is to couple the function to a boundary condition type (*Input/Boundary conditions*), see Figure 4.15. Give the stop time 4 days (Solve/Options for transient), see Figure 4.16.

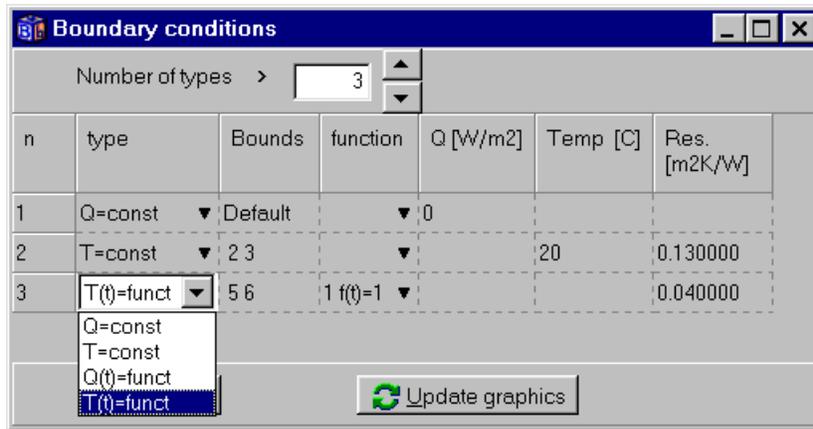


Figure 4.15: A temperature function is chosen in the pick list for the boundary condition type. The function number is selected in the function pick list (number 1 is default).

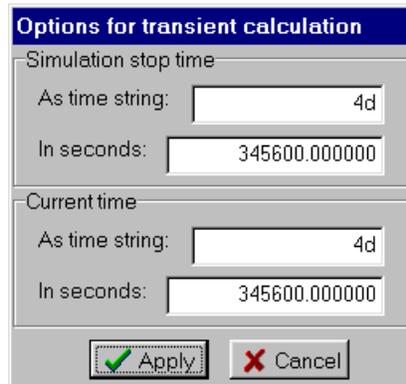


Figure 4.16: The stop time is 4 days.

In this example we will also look at the flows through the internal boundaries during each hour. Enable the record (*Output/Recorder*) and pick the second item in the list as in Figure 4.17 (left). Give the numbers 2 and 3 for the internal boundaries as in Figure 4.17 (right). This will record the sum of the two boundary heat flows at each screen update (*Solve/Update*) as given in Figure 4.18, in this case every hour. Press the item *Graphics* in the record window to bring up a chart window (this will be empty since no data is yet recorded).

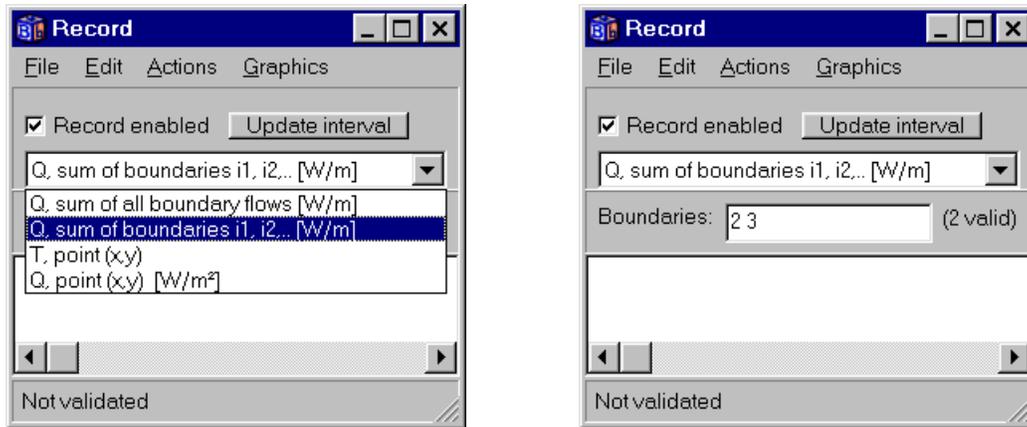


Figure 4.17: Flows and temperatures may be saved to a list during the simulation. In this case the sum of the heat flows through boundaries 2 and 3 will be recorded.

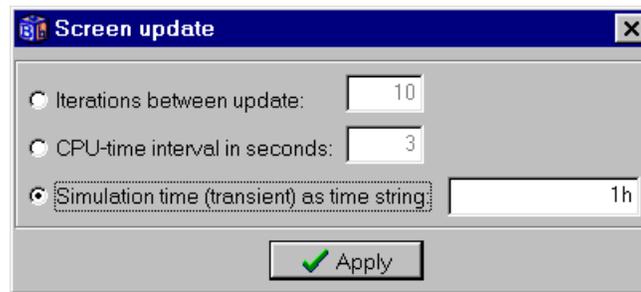


Figure 4.18: The time interval for recorded data is given in item *Solve/Update*.

Start the transient simulation by selecting *Solve/Start transient calculation*. Consider Figure 4.19. A window (top left) will be displayed showing information such as the chosen stop criterion, number of iterations, and error for temperature and flow. The record window (bottom left) and the chart window (bottom right) show the internal heat flows during each hour.

Note the large heat flow during the first hours at boundary 3 due to the initial temperature (10 °C) as given in Figure 4.10.

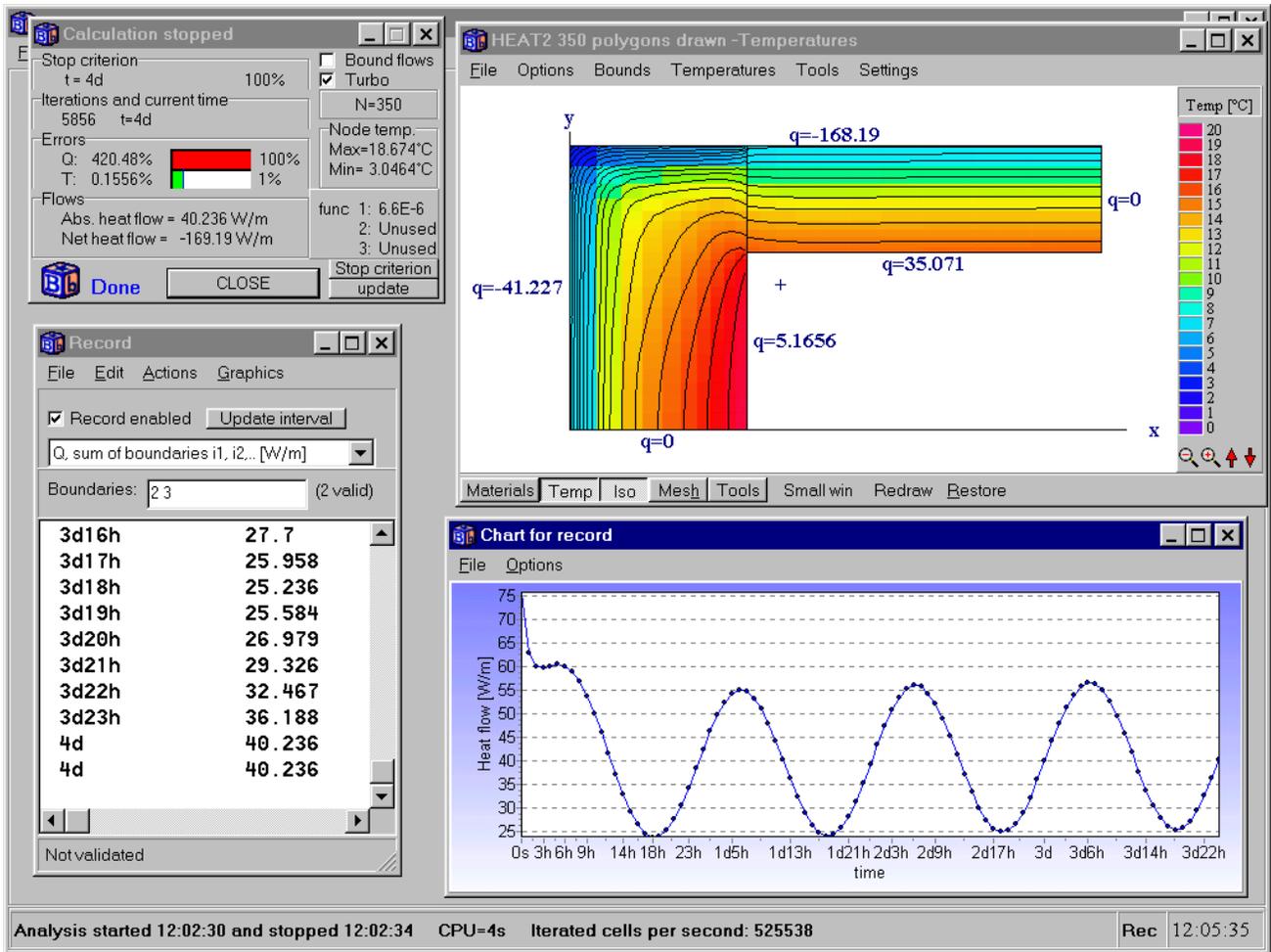


Figure 4.19: The solve window (top left) shows chosen stop criterion, number of iterations, and error for temperature and flow. The record window (bottom left) and the chart window (bottom right) show the internal heat flows during each hour.

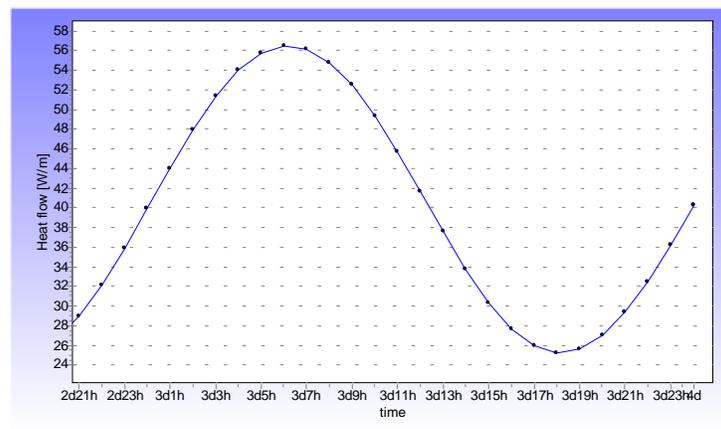


Figure 4.20: The maximum heat flow through the internal boundaries into the wall occurs a six-o'clock in the morning and the minimum at six-o'clock in the afternoon.

4.8 Internal modifications

Internal modifications may be used, such as regions with heat sources (constant or time-dependent), given temperatures (constant or time-dependent), air-holes, fluid with capacity, and radiation inside cavities that may be ventilated (see Chapter 5). Approximated circular pipes with a given heat source or temperature (constant or time-dependent) may be used, see Chapter 8. Figure 4.21 shows a calculation for stairs heated by pipes with a temperature of 30 °C. The external temperature is -15 °C. The color for $T=0$ is disabled here, see scale to the right. Even though the stairs are heated by pipes, the edge of each step will still be below zero and can be slippery.

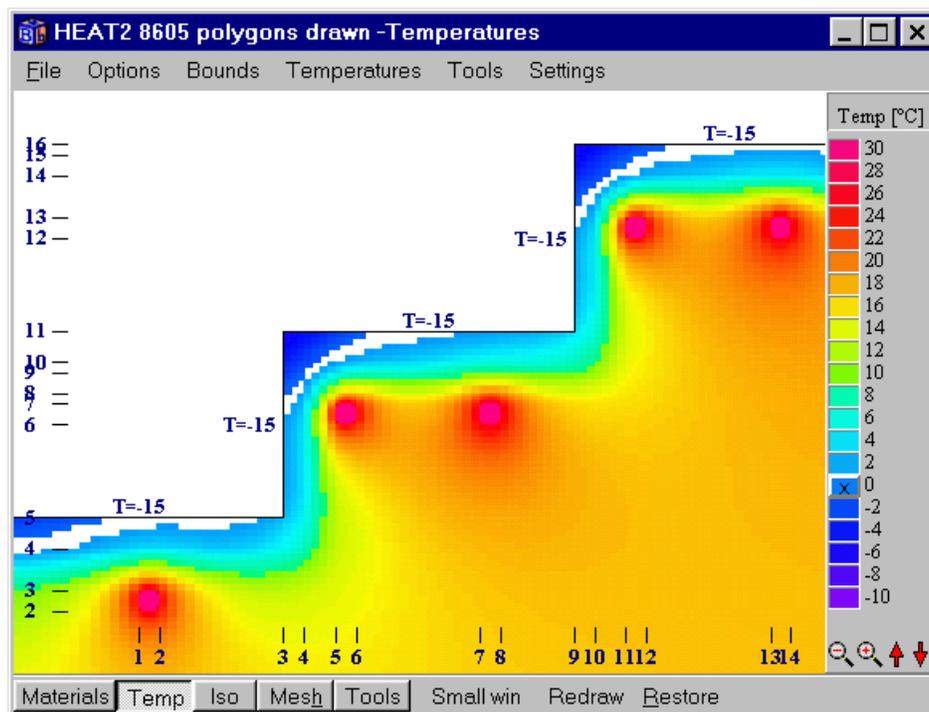


Figure 4.21: Stairs heated by pipes with a temperature of 30 °C. The external temperature is -15 °C. The color for zero temperature is disabled, see scale to the right. The edge of each step will still be below zero and can be slippery.

The internal modifications are placed as rectangular regions using the lower left and upper right input mesh coordinates. They must be placed according to some specific rules, see Figure 4.22. There must be at least one numerical cell between the areas (e.g. between area 2 and 3). There must be at least one cell between an area and a boundary (e.g. areas 3 and 4). Two areas may meet at corners (areas 1 and 2). The above rules apply to all kinds of areas with an internal modification, such as areas with a given temperature, heat source, and so on. HEAT2 automatically checks that the given regions are consistent.

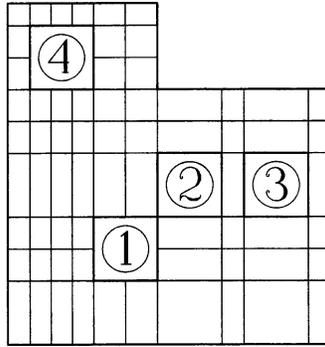


Figure 4.22: An example of how four internal modifications may be placed.

4.9 Internal resistances

Internal resistances R , ($\text{m}^2\text{K/W}$), may be given by specifying lines, perpendicular to the actual resistances, either in the x -direction or in the y -direction. Consider Figure 4.23. The vertical line located between the input mesh points $(1,1)$ and $(1,4)$ implies a resistance R_x in the x -direction.

4.10 Summary

A suitable input mesh has to be defined taking the geometry, materials, initial temperatures, boundary segments with different boundary conditions, internal modifications, internal resistances and numerical mesh into account.

Consider Figure 4.23. The region contains two materials A and B . A suitable input mesh generation has been made. There are five segments in both directions. The region A has the input mesh coordinates lower left $0,2$ and upper right $5,5$. The rectangle defined by the lower left coordinate $3,3$ and the upper right coordinate $4,4$ marks material B . The material B is also defined with the rectangle coordinates $0,0$ and $2,2$. A resistance R_x in the x -direction is located between the input mesh coordinates $1,1$ and $1,4$.

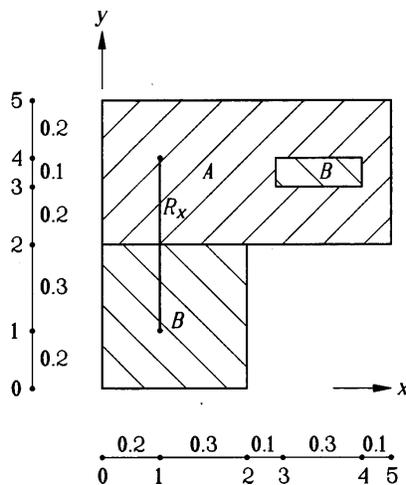


Figure 4.23: A suitable input mesh has to be defined taking the geometry, materials, initial temperatures, boundary segments with different boundary conditions, internal modifications, internal resistances, and numerical mesh into account.

It makes no difference which one of the boundaries that is given first, but the enumeration must be done counterclockwise. Consider Figure 4.23. Here, the boundary between $0,0$ and $2,0$ is chosen as the first one, hence the boundary between $5,5$ and $0,5$ becomes number 5. The boundaries of the computational region will here be given as $0,0$ $2,0$ $2,2$ $5,2$ $5,5$ $0,5$ $0,0$. The lengths in the x -direction and y -direction are: 0.2 0.3 0.1 0.3 0.1 and 0.2 0.3 0.2 0.1 0.2 .

5. Radiation and ventilation in cavities

5.1 Introduction

Heat conduction in solid building parts coupled to long-wave thermal radiation in air cavities is a frequently occurring and studied type of process in building physics. The cavity, which can be ventilated, may be a crawl-space, the space between glass panels in a window, closed air volumes in brick, and cracks and other voids in building thermal envelopes.

Fast and accurate routines for calculation of radiation (gray-diffuse) in rectangular cavities are developed. The theory for the method used in this section is presented in (Blomberg, 1996).

5.2 Input data

Figure 5.1 shows a cavity with the emissivity ϵ_{lower} and the convective surface heat transfer coefficient α_{lower} at the lower side. The three other sides may in the same way have their own properties. The cavity may be ventilated with a constant inlet temperature T_v at the ventilation rate n (h^{-1}). In most cases the cavity is unventilated ($n=0$). The gas in the cavity (e.g. air or krypton) has the volumetric heat capacity C , ($\text{J}/(\text{m}^3\cdot\text{K})$), which is the density times the specific heat capacity ($C=\rho\cdot c_p$).

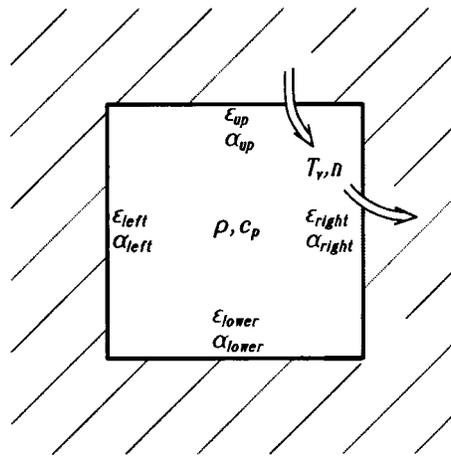


Figure 5.1: Input parameters for radiation inside a cavity.

Each cavity is defined as an internal modification in the input mesh. The numerical grid defines the number of surface elements N_R for which radiative heat exchange is considered. Figure 5.2 shows the minimal case with one surface element on each side. The cavity has one cell (1·1). There are four elements with temperatures T_{s1} , T_{s2} , T_{s3} , and T_{s4} . Figure 5.3 shows two other examples: a cavity with 2·3 cells ($N_R=10$) to the left, and 10·10 cells ($N_R=40$) to the right. The maximum number of surface elements that is allowed is $N_R=250$. HEAT2 will automatically check that this limit is followed.

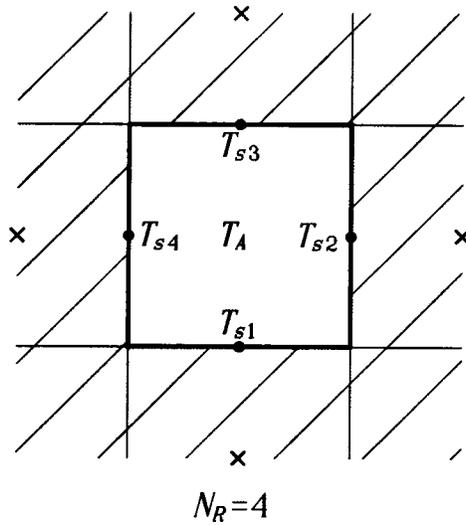


Figure 5.2: Four radiation surface elements in a cavity.

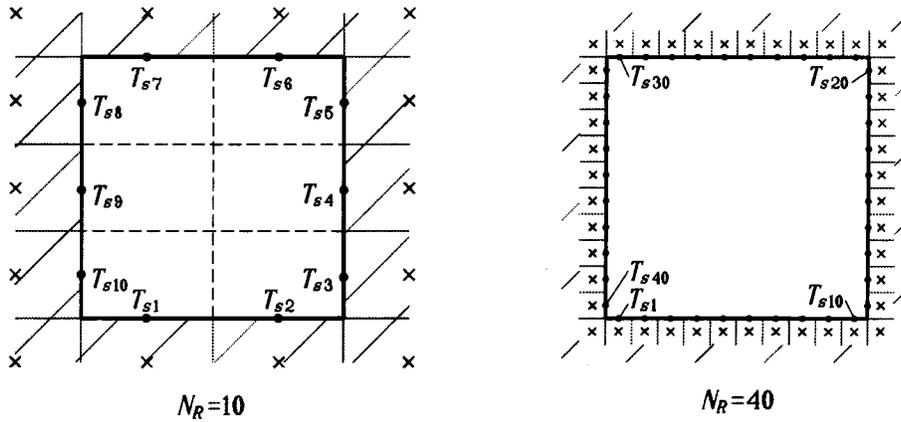


Figure 5.3: The number of radiation elements N_R is defined by the numerical mesh.

The program may consider several cavities with radiation exchange. The maximum number of cavities with radiation is 20. Each cavity may have its own set of data (ϵ and α on the four sides, T_v , n , and C). See 6.6.1 for more info.

5.3 Output data

The following results are shown for each cavity: calculated surface temperatures T_{s_j} for $j=1..N_R$, see Figure 5.4; temperatures T_{b_j} in the cells closest to the cavity, the conductances K_{b_j} between the center of the cell and the surface of the enclosing cells, the heat flows q_{b_j} (W/m), and the gas temperature T_a . The volume of the cavity per meter (m^3/m), the number of radiation elements N_R , and the ventilation conductance (or convectance) K_v (W/K), see (Blomberg, 1996), are also given.

Internal flows in (W/m) and (W/m²) through each of the four sides for the cavity is also given.

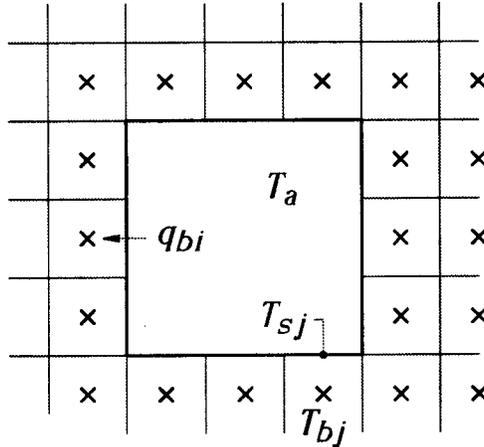


Figure 5.4: Calculated results for each cavity with radiation.

5.4 Numerical solution for the cavity

The method of the numerical solution for the heat balance for the cavity is given in (Blomberg, 1996). A number of equations has to be satisfied. The first one is a heat balance equation for the gas in the cavity. The second set of equations concerns balances for the surface temperatures. The third set is the radiation heat balance equations (Holman, 1986):

$$\frac{\varepsilon_i A_i}{1 - \varepsilon_i} \cdot (\sigma \tilde{T}_{si}^4 - J_i) = \sum_{j=1}^N A_i F_{ij} (J_i - J_j) \quad i = 1, \dots, N_R \quad (5.1)$$

Here, J_i (W/m²) is the radiosity of surface element i , i.e. the total radiation which leaves surface i per unit area and unit time. The emissivity of the gray and diffuse surface i is denoted ε_i , and σ is Stefan-Boltzmann's constant. The tilde sign \sim denotes temperature in Kelvin. The view factors are denoted by F_{ij} .

5.4.1 Radiation equations in temperature form

The radiation balance equations are reformulated to involve temperatures only. The temperatures are linearized around a suitable temperature level \tilde{T}_{so} , see (Blomberg, 1996). In building physics applications (except fire), the surface temperatures T_{si} vary in a rather limit interval. The differences $T_{si} - T_{so}$ (and $T_{sj} - T_{so}$) are small compared to \tilde{T}_{so} . Then the error in the linearization becomes rather small.

HEAT2 may perform the calculations using the linearized form, or using a higher order of approximation. This may be specified in menu item *Radiation settings*, where "1" means linear approximation and "2" a second order of approximation. The maximum order that may be given is 5. The calculations will take somewhat longer time with higher orders of approximation. For most "normal" building physics applications, however, the linear approximation should be sufficient.

For each cavity, a suitable level of T_{so} (°C) is specified. The value of T_{so} should be equal to an estimated cavity temperature. As an example consider a window with the exterior temperature 5 °C and the interior temperature 15 °C. Here, T_{so} should be put to 10 °C. It may be noted that the

choice of T_{so} is not particularly critical, especially with higher orders of approximation. Even in the case with a linear approximation, the error turns out to be relatively small.

6. Working with HEAT2

6.1 Loading and saving input data

The data file (*.DAT) contains the input data for a problem.

The *PSE* file contains input data, the temperature field and, if solving a transient problem, the accumulated boundary heat flows and the point of time when the simulation was interrupted. This makes it possible to stop a simulation (steady-state or transient) and quit HEAT2, and later revert the simulation from where it was interrupted. To do this, save a *PSE* file (*PSE* is short for *pause*) for the problem before quitting HEAT2. When restarting HEAT2 later, open the *PSE* file and continue the simulation.

6.2 Inserting mesh coordinates

Suppose that we wish to add an area somewhere in an already defined problem. New mesh coordinates must probably be defined. It is likely that many of the already defined areas, boundary conditions, resistances and heat sources, and so on, have to be redefined in the new coordinates.

Input mesh coordinates may be inserted automatically (*Edit/Insert mesh coordinate*, see Figure 6.1). The new coordinate will be inserted in the middle of the old segment, which will be divided into two new segments (each with half the length of the old one). The number of cells in each of the two new segments will be the number of cells in the old one divided by two. If the number of cells in the old segment is odd, the number will be one even and one odd, e.g. seven cells will split into 3 and 4 cells. When a coordinate is inserted, the only thing the user probably has to change is the lengths, the number of cells for the new segments and the expansion coefficients.

Note that there is no undo command available (it may be wise to save the data before this operation is done so that it can be restored if it did not turn out the way it was suppose to be). Neither can mesh points be removed.

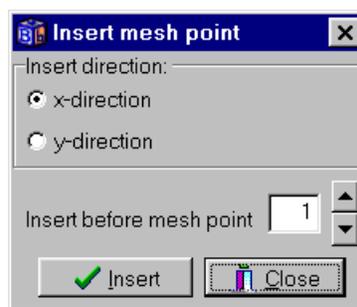


Figure 6.1: Menu for inserting mesh coordinates.

Consider Figure 6.2. The top figures show the originally described problem. The bottom figures show the coordinates when insertion is made in the *x*-direction before mesh point 1.

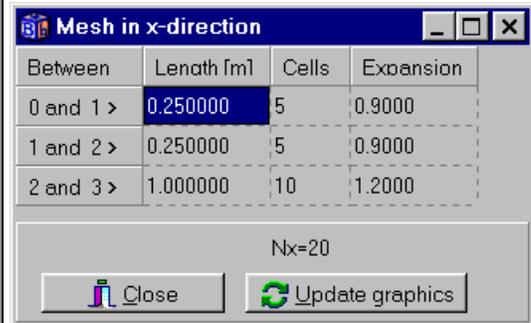
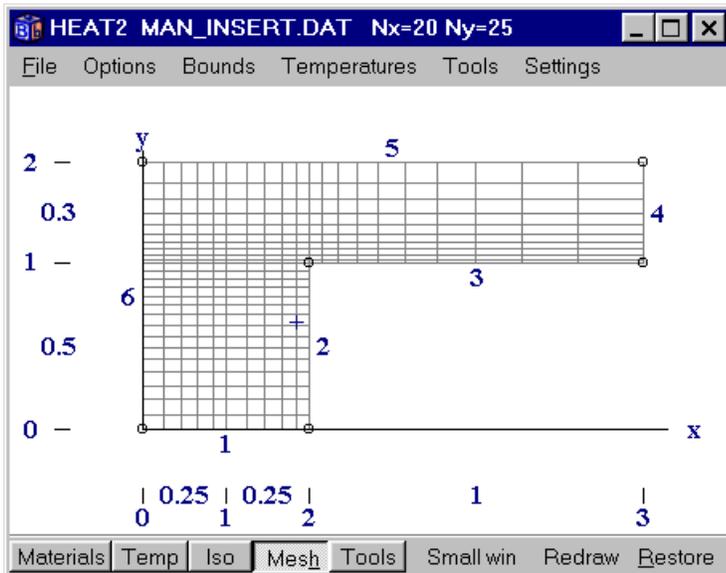
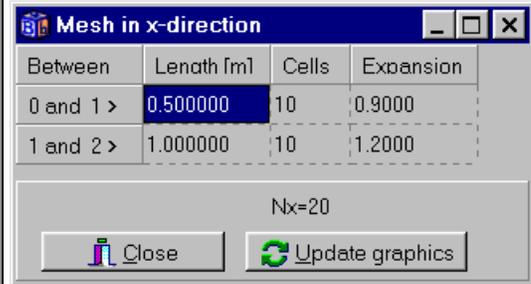
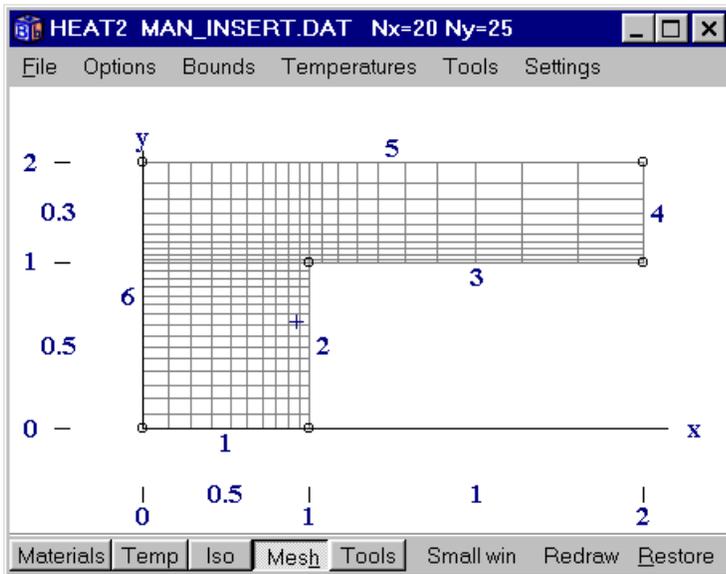


Figure 6.2: A mesh point has been inserted in the bottom figures in the x -direction before mesh point 1.

Suppose that we want two different boundary conditions at the upper boundary ($y=0.8$) at $0 \leq x < 1$ and $1 \leq x < 1.5$. In case that a proper input mesh was not initially made, the existing mesh has to be redefined. A new mesh point is inserted in the x -direction before mesh point 3, see Figure 6.3.

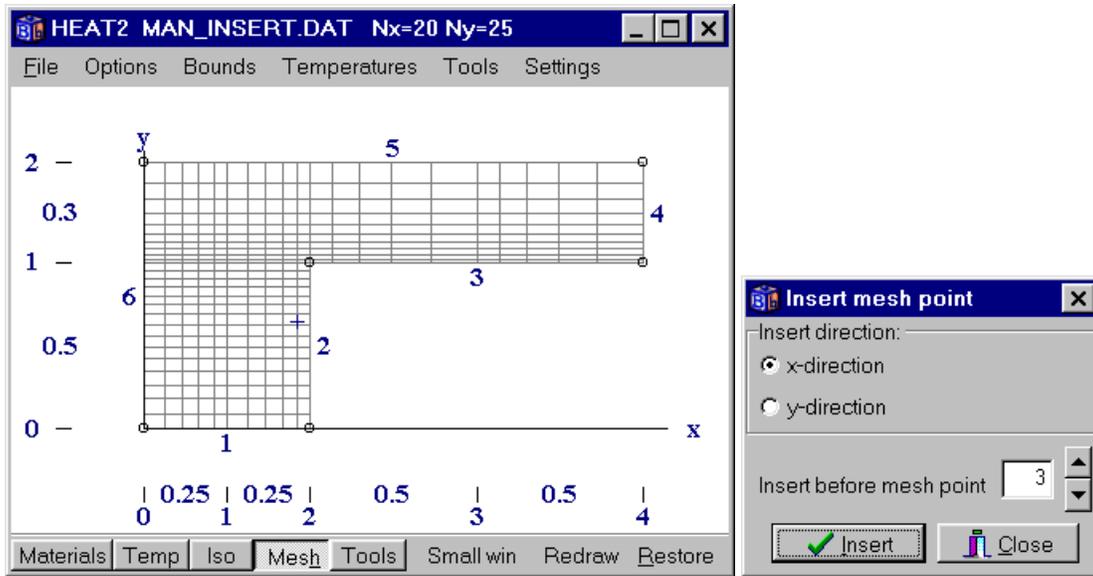


Figure 6.3: New mesh point inserted in the x-direction before mesh point 3.

Change the number of boundaries (*Input/Input mesh*) from 6 to 7 as in Figure 6.4 (top). Change the description of boundaries for boundaries 6 and 7 as in Figure 6.4 (bottom). There are now four segments in the x-direction, and seven boundary segments (delimited by small circles), see Figure 6.5. Segments 5 and 6 can now be assigned different boundary conditions.

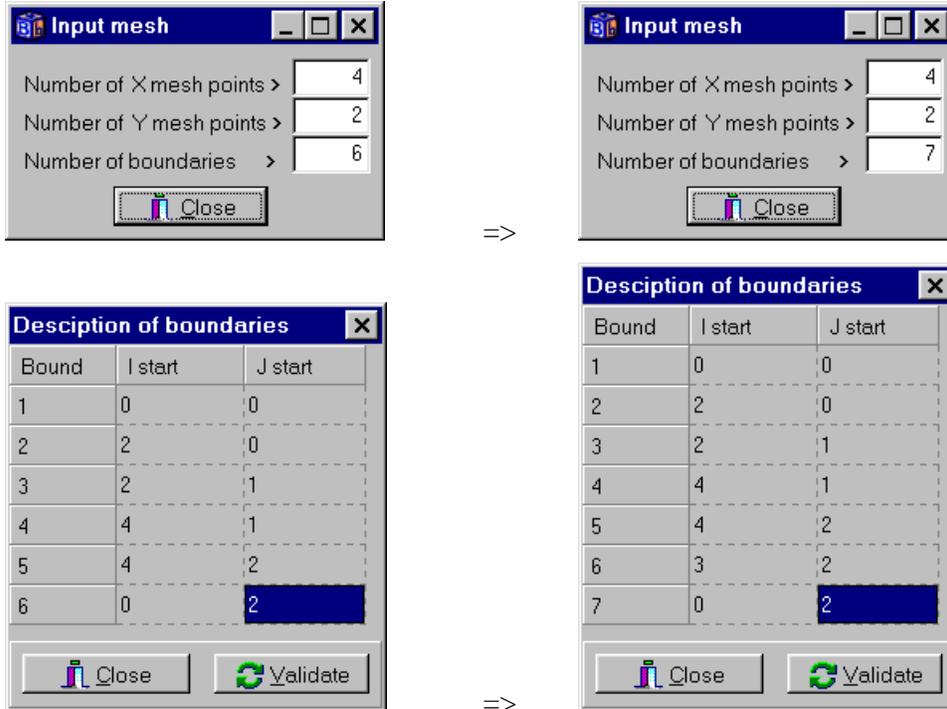


Figure 6.4: The number of boundaries is changed from 6 to 7. New descriptions of boundaries 6 and 7 are made.

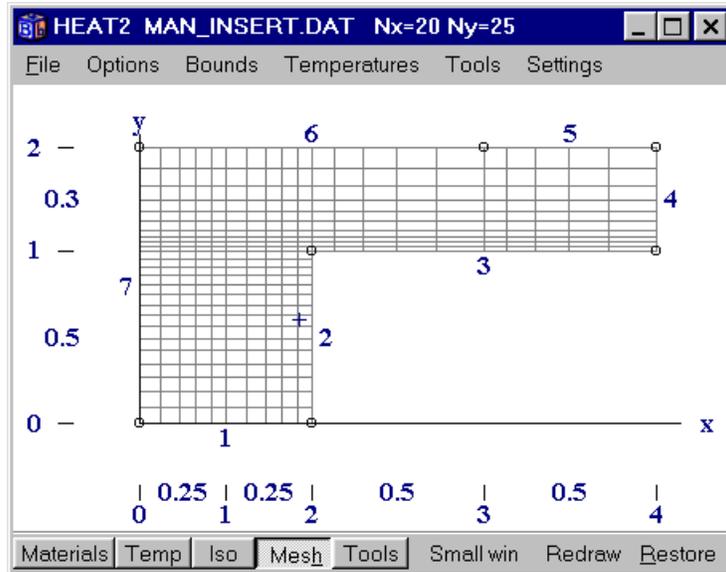


Figure 6.5: Segments 5 and 6 can now be assigned different boundary conditions.

6.3 Thermal properties

Figure 6.6 shows the window for thermal properties. See Section 4.3 for more information. The basic thermal properties, λ_x , λ_y , and C will be valid for the whole area. Areas defined in a list will overlap the region with new materials.

An optional material name can be given. This name will be saved to the input data file. There is no direct connection between the name and the thermal properties. They can be edited separately. The name will be displayed on the thermal property scale in the graphic window.

It is possible to pick a material from the material list that may be edited (MTRL30.TXT), see Appendix B. When a material is picked, the name and the thermal properties will be loaded for the current area.

The volumetric heat capacity is denoted by C , ($\text{J}/(\text{m}^3\cdot\text{K})$), which is the density ρ , (kg/m^3), times the specific heat capacity c_p , ($\text{J}/(\text{kg}\cdot\text{K})$), i.e. $C=\rho\cdot c_p$. Note that C is given in the unit ($\text{MJ}/(\text{m}^3\cdot\text{K})$), that is $1.0 \cdot 10^6 \text{ J}/(\text{m}^3\cdot\text{K})$. The default value of C is $1.0 \text{ MJ}/(\text{m}^3\cdot\text{K})$. For the steady-state simulation the value of C is neglected.

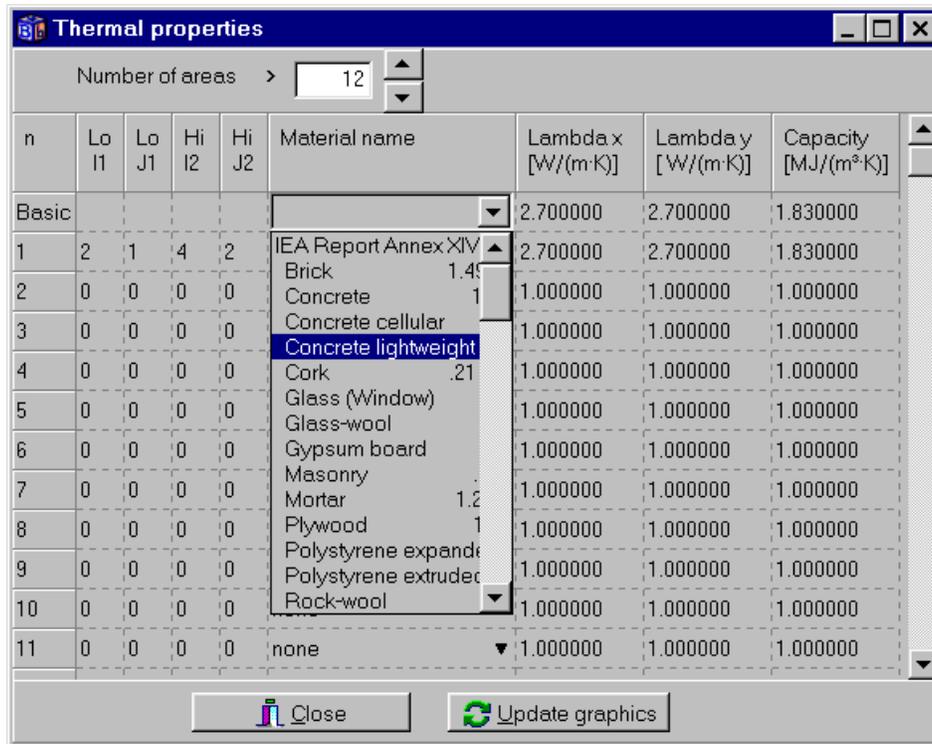


Figure 6.6: Data entry for thermal properties.

6.4 Boundary conditions

A boundary condition type may either be a temperature T , ($^{\circ}\text{C}$), with a surface resistance R , ($\text{m}^2\cdot\text{K}/\text{W}$), or a given heat flow Q , (W/m^2). The temperatures and the heat flows may be constant in time, or time-dependent using a sinusoidal, a step-wise constant, or a step-wise linear function. See definition in Section 6.7.

The user defines the boundary conditions in a list and assigns them to the boundary segments. Consider Figure 6.7. There are in this case four boundary conditions with list numbers $n=1-4$ in the first column. The second column shows boundary segments assigned to a certain type. By default, all boundary segments that is not defined here will use the first type.

The third column links to a function (sinusoidal, stepwise constant or step-wise linear). The type may be either a function for temperature or heat flow.

The last column gives the surface thermal resistance (applies to a type with temperature).

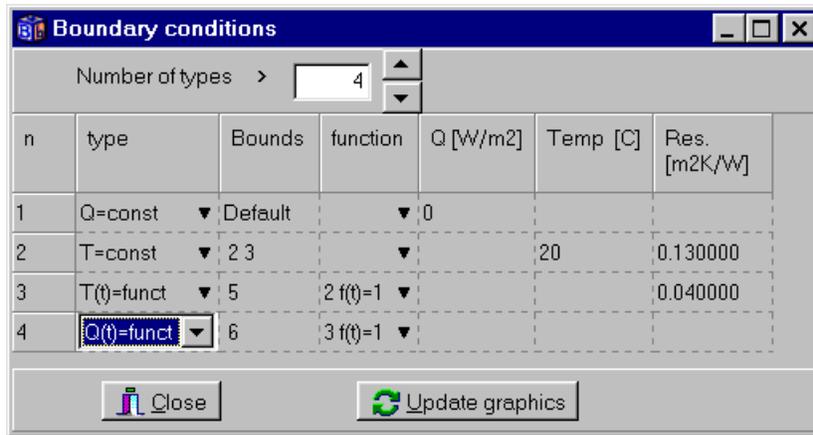


Figure 6.7: The boundary types are linked to each boundary segment.

6.5 Initial temperatures

See Section 4.5. In order to initialize the temperatures use *Solve/Reset Temperatures* or click on the *update graphics* button, see Figure 6.8.

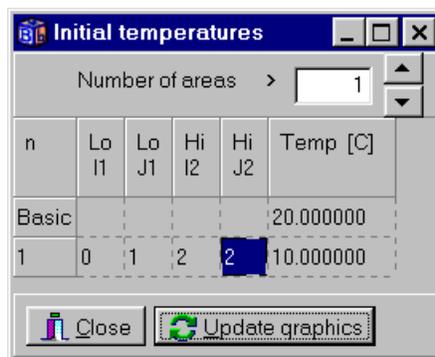


Figure 6.8: Initial temperatures.

6.6 Internal modifications

6.6.1 Defining types

Figure 6.9 shows the input window for the internal modification types. Each type is picked from a list and the data are given in the column “Data”. If a modification depends on a function, the function is chosen in the last column. Table 6.1 shows the input data for the available modifications.

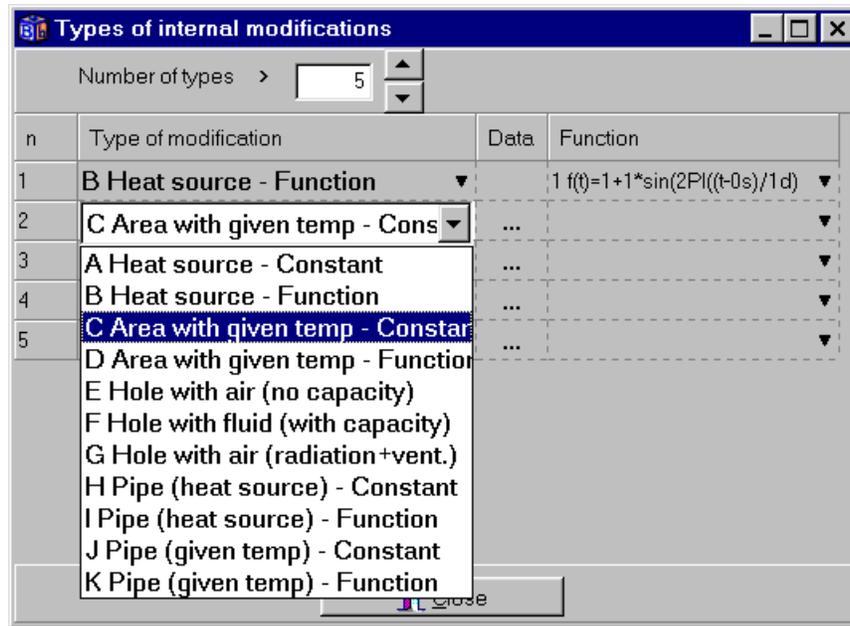
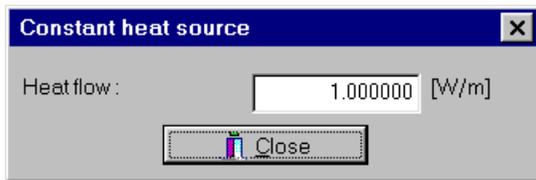


Figure 6.9: Internal modifications menu.

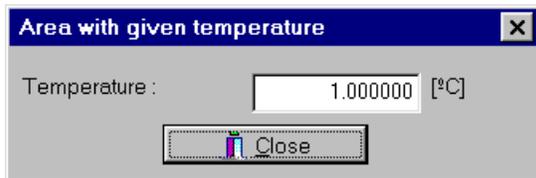
Type	Description
A	Constant heat source, (W/m)
B	Heat source that varies in time (a link is made to a pre-defined function)
C	Constant temperature
D	Temperature that varies in time (a link is made to a pre-defined function)
E	Air hole, see Sections 2.4 and 3.9. Input is the surface resistance, R , ($\text{m}^2\cdot\text{K}/\text{W}$).
F	Hole with a fluid with capacity, input is the surface heat transfer coefficient α , ($\text{W}/(\text{m}^2\cdot\text{K})$), the fluid's volumetric capacity C , ($\text{MJ}/(\text{m}^3\cdot\text{K})$), and the initial temperature of the fluid, T_0 , ($^\circ\text{C}$).
G	Radiation inside cavities which may be ventilated. Input are emissivity ϵ and the convective surface heat transfer coefficient α at the four sides. The cavity may be ventilated with a constant inlet temperature T_v at the ventilation rate n (h^{-1}). In most cases the cavity is unventilated ($n=0$). The gas in the cavity has the volumetric heat capacity $C=\rho\cdot c_p$, ($\text{kJ}/(\text{m}^3\cdot\text{K})$). The default value of $C=1.239$ $\text{kJ}/(\text{m}^3\cdot\text{K})$ is for air in room temperature. The temperatures are linearized around a suitable temperature level \tilde{T}_{so} , see Chapter 5.
H	Approximated circle with constant heat source (pipe), see Chapter 8.
I	Approximated circle with non-constant heat source (pipe), see Chapter 8.
J	Approximated circle with constant temperature (pipe), see Chapter 8.
K	Approximated circle with non-constant temperature (pipe), see Chapter 8.

Table 6.1: Available internal modifications.

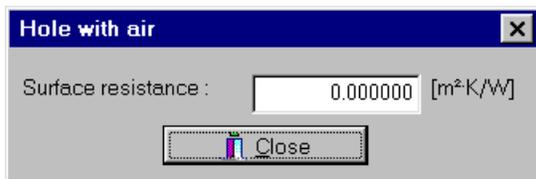
The data entry for all constant modifications is shown below. In the non-constant cases, a link to a function is made.



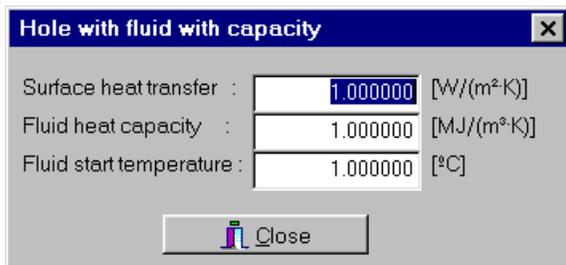
A: Heat source constant



C: Temperature constant



E: Air hole



F: Hole with fluid with capacity

Hole with air - radiation and ventilation		X
Gas heat capacity	<input type="text" value="1.239000"/>	[kJ/(m³K)]
Ventilation rate	<input type="text" value="0.000000"/>	[h⁻¹]
Ventilation temperature	<input type="text" value="20.000000"/>	[°C]
Emissivity at lower side	<input type="text" value="0.900000"/>	[-]
Emissivity at right side	<input type="text" value="0.900000"/>	[-]
Emissivity at upper side	<input type="text" value="0.900000"/>	[-]
Emissivity at left side	<input type="text" value="0.900000"/>	[-]
Conv. coeff. lower side	<input type="text" value="1.000000"/>	[W/(m²K)]
Conv. coeff. right side	<input type="text" value="1.000000"/>	[W/(m²K)]
Conv. coeff. upper side	<input type="text" value="1.000000"/>	[W/(m²K)]
Conv. coeff. left side	<input type="text" value="1.000000"/>	[W/(m²K)]
Reference temperature	<input type="text" value="10.000000"/>	[°C]
Order of approximation (1-5) (this will be used for all areas with radiation)	<input type="text" value="1"/>	
<input type="checkbox"/> Show matrices in info log		
<input type="button" value="Close"/>		

G: Hole with air, radiation and ventilation

Note on order of approximation

The order of approximation for the radiation calculation may be given here, see Chapter 5. A value of 1 specifies a linear approximation (default value). The calculation will take a little bit longer time with a higher order of approximation. For most cases the linear approximation is sufficient (i.e. the same results are obtained regardless of degree of approximation).

Constant heat source in pipe		X
Heat flow :	<input type="text" value="1.000000"/>	[W/m]
<input type="button" value="Close"/>		

H: Pipe heat source

Constant temperature in pipe		X
Temperature :	<input type="text" value="1.000000"/>	[°C]
<input type="button" value="Close"/>		

J: Pipe given temp

6.6.2 Defining areas

Figure 6.10 shows the entry for the areas with internal modifications. The lower left and upper right input mesh coordinates are given for each area. The corresponding type is chosen from a pick list. Notice that an internal area must not be adjacent to an outer boundary, nor to another internal area.

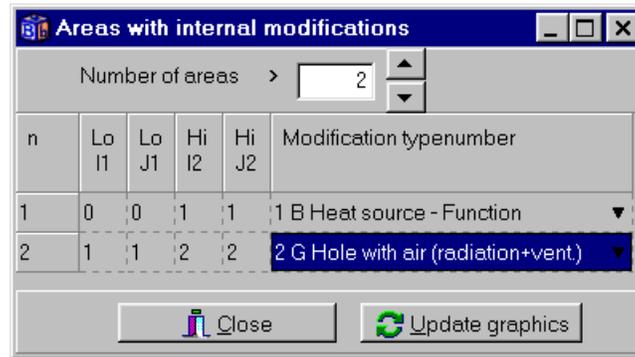


Figure 6.10: Each area is defined by the lower left and upper right input mesh coordinates. A link is made to one of the earlier defined modifications available in a pick list.

6.7 Functions

Functions may be used to describe changes of temperatures or heat flows in time for boundary conditions. Functions may also apply to internal modifications, such as heat sources and areas with given temperatures. Three independent functions can be used in each problem. The following functions are available:

- $$f(t) = f_1 + f_2 \cdot \sin\left(\frac{2\pi(t - t_0)}{t_p}\right) \quad t_p > 0 \quad (\text{sinusoidal})$$
- $$f(t) = \begin{cases} 0 & t < t_1 \\ f_1 & t_1 \leq t < t_2 \\ f_2 & t_2 \leq t < t_3 \\ \vdots & \\ f_n & t_n \leq t \end{cases} \quad (\text{step-wise constant})$$
- $$f(t) = \begin{cases} f_1 \cdot t_1 & t < t_1 \\ f_1 + (f_2 - f_1) \cdot \frac{(t - t_1)}{(t_2 - t_1)} & t_1 \leq t < t_2 \\ f_2 + (f_3 - f_2) \cdot \frac{(t - t_2)}{(t_3 - t_2)} & t_2 \leq t < t_3 \\ \vdots & \\ f_n & t_n \leq t \end{cases} \quad (\text{step-wise linear})$$

6.7.1 Sinusoidal function

Figure 6.11 shows the entry for the sinusoidal function. Input is average f_1 , amplitude f_2 , phase time t_0 and period time t_p . The phase and period time is given in seconds or with a time-string.

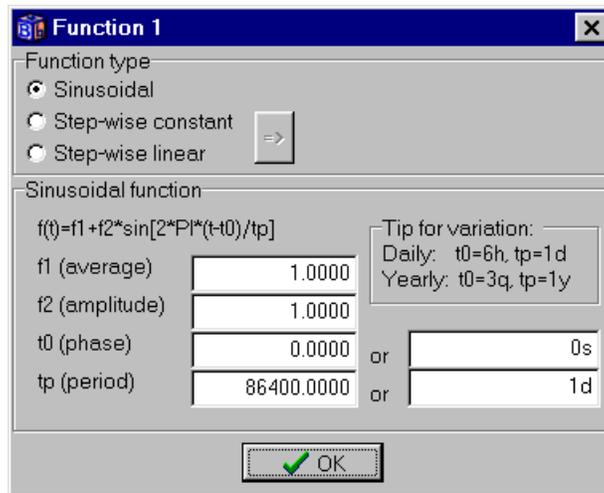


Figure 6.11: Functions are given in item *Solve/Function*.

The time-string is a sequence of pairs with a number and one of the following letters:

<i>y</i>	year (365 days, 31536000 seconds)
<i>q</i>	month (year/12, 30.417 days, 2628000 seconds)
<i>d</i>	day (86400 seconds)
<i>h</i>	hour (3600 seconds)
<i>m</i>	minute (60 seconds)
<i>s</i>	second (can be omitted)

Please note that *m* is minute and *q* is month. Here are some examples:

1y14h	1 year and 14 hours
14q3d15	1 year, 2 months, 3 days and 15 seconds (same as 1y2q3d15)
86400	1 day
1h2d	Not valid. The expression must be in descending order, see next row.
2d1h	This string is OK, meaning 2 days and 1 hour

6.7.2 Step-wise constant and step-wise linear function

The step-wise constant and step-wise linear values are given in an editor, see Figure 6.12. Each row should contain the time (in seconds or using a time-string) and the function value. Comments may be written after the data on each row or after a '%', see Figure 6.12 (right).

The data will be validated if the *Validate* menu item is pressed (or when a simulation is started). The right figure shows an error on the third row, see message at the bottom. The function values for the data given in the left figure are shown in Figure 6.16 and Figure 6.17. The maximum number or values is constrained by the 16 MB of text that the editor can hold. Using e.g. 16 characters per row would give a maximum number of one million function values.

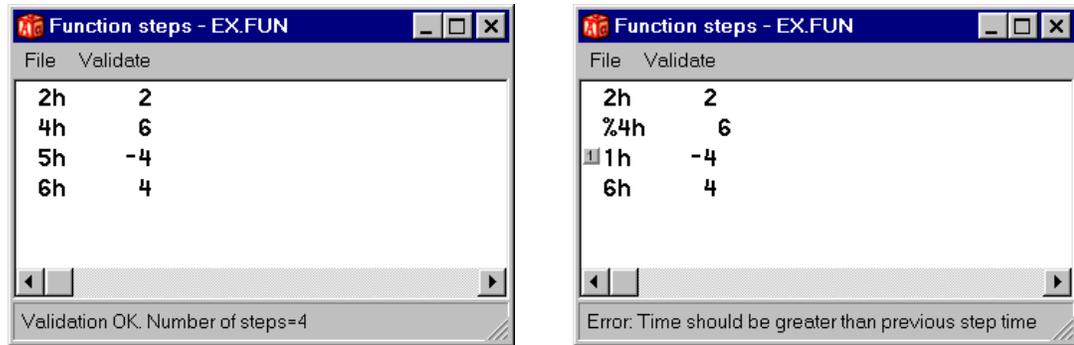


Figure 6.12: Data for function steps are given in an editor. The input is validated and a marker will be shown if there is an error.

The function values may be generated in other programs and pasted into the function editor. Figure 6.13 shows the function $\ln(x)$ created in Microsoft Excel for a few values that have been pasted into the function editor. The data for the function steps may be saved (*.FUN) and later be used by other problems in HEAT2 and HEAT3.

During the simulation, the function value will be shown in the simulation window. Note that it is possible to open function values (*.FUN) in the record list and graphically see the variation as stepwise-linear values, see Section 6.11.

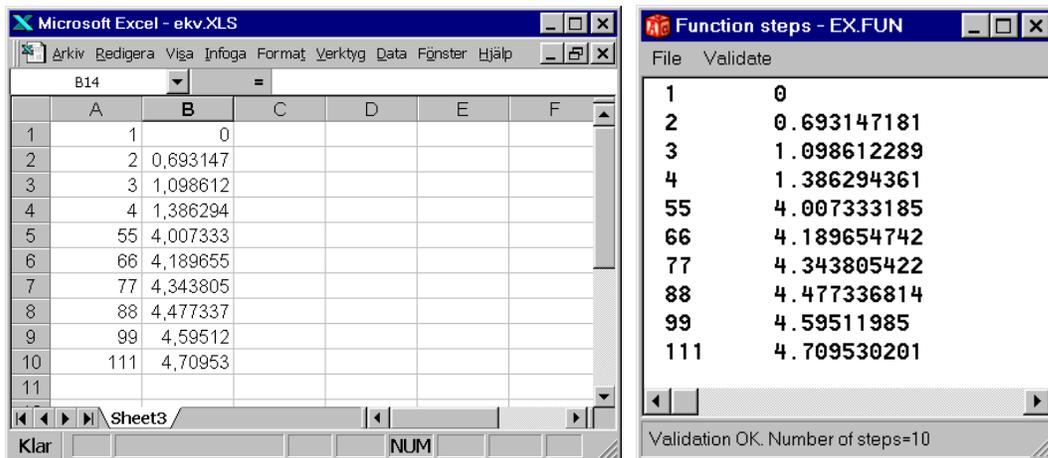


Figure 6.13: Arbitrary function step values may be pasted from other programs. Here is the function $\ln(x)$ created in Microsoft Excel for a few values that have been pasted into the function editor.

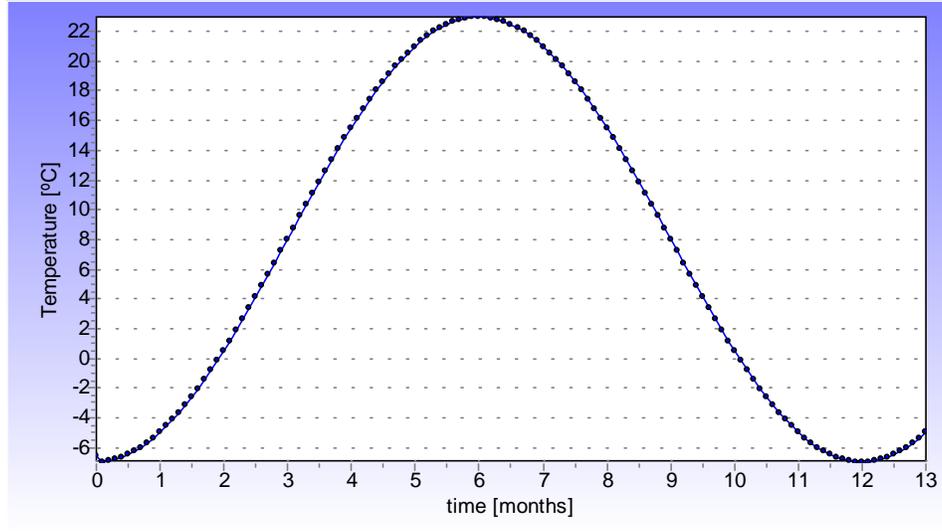
6.7.3 Example 1 - sinusoidal function with a time period of one year

Figure 6.14 shows a sinusoidal function with the average value $8\text{ }^{\circ}\text{C}$, the amplitude $15\text{ }^{\circ}\text{C}$, the time phase three months ($-3q$), and the time period one year ($1y$). With this time phase ($-3q$ or $+9q$), the coolest temperature ($-7\text{ }^{\circ}\text{C}$) will occur after $0, 12, \dots, 12n$ ($0 \leq n < \infty$) months, i.e. on the 1:st of January every year. The warmest temperature ($23\text{ }^{\circ}\text{C}$) will occur after $6, 18, \dots$, months, i.e. on the 1:st of July every year¹.

¹ Using time phase $-4q$ instead will give min and max at Feb 1 and Aug 1, respectively.

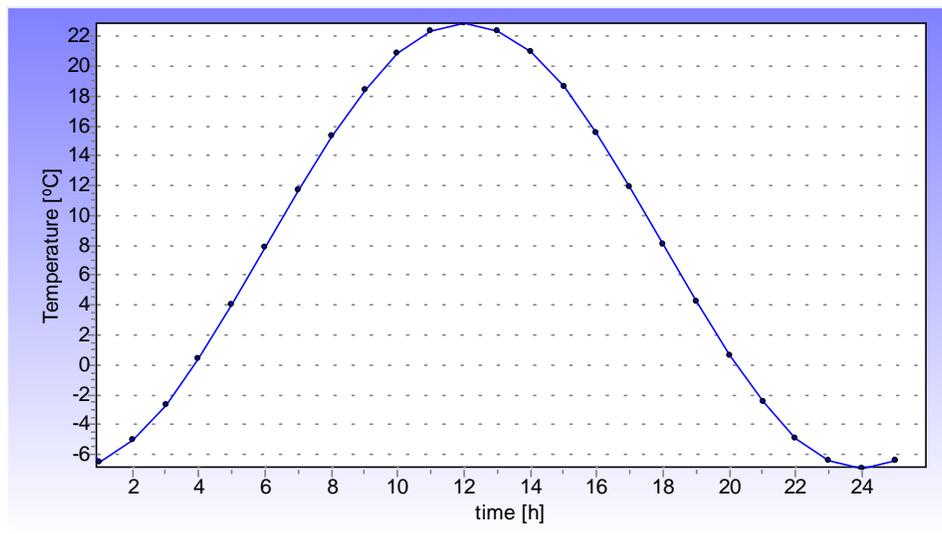
6.7.4 Example 2 - sinusoidal function with a time period of one day

Figure 6.15 shows a sinusoidal function with the average value 8 °C, the amplitude 15 °C, the time phase six hours (-6h), and the time period one day (1d). With this time phase (-6h or +18h), the coolest temperature (-7 °C) will occur every day at midnight, and the warmest temperature (23 °C) will occur at noon.



$$f(t) = 8 + 15 \cdot \sin\left(\frac{2\pi(t - 3q)}{1y}\right)$$

Figure 6.14: Sinusoidal function with $f_1=8$, $f_2=15$, $t_o=3q$, and $t_p=1y$.



$$f(t) = 8 + 15 \cdot \sin\left(\frac{2\pi(t - 6h)}{1d}\right)$$

Figure 6.15: Sinusoidal function with $f_1=8$, $f_2=15$, $t_o=6h$, and $t_p=1d$.

6.7.5 Example 3 - step-wise constant function

Figure 6.16 shows a step-wise constant function with four steps using the data as given to the right. Note that the value of the function is zero before the first defined step. If $t_1=0$, instead of $t_1=2h$, the first value f_1 will be used from start. Also note that the function value for the last step will sustain after this point of time.

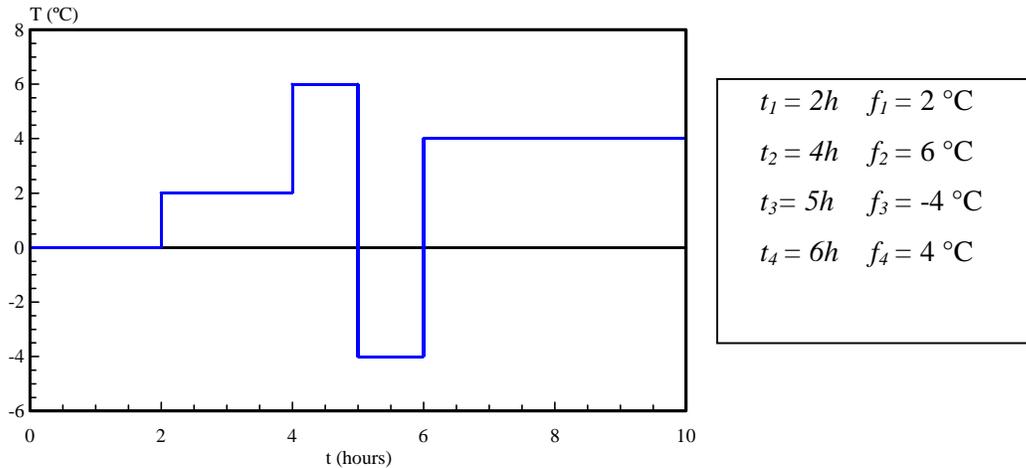


Figure 6.16: Step-wise constant function for the four steps given to the right.

6.7.6 Example 4 - step-wise linear function

Figure 6.17 shows the step-wise linear function for the four steps. Note that the function value is zero for $t=0$ (may be changed by giving a function value for $t_1=0$). The function value for the last step will be sustained.

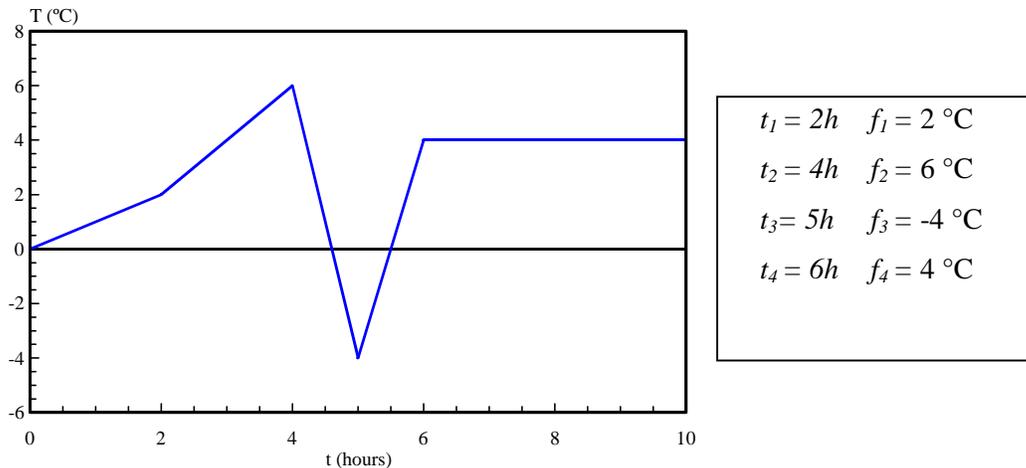


Figure 6.17: Step-wise linear function for the four steps given to the right.

6.8 Internal resistances

Figure 6.18 shows the input window for internal resistances. It is allowed to give a boundary resistance as an internal resistance line. Note however that this value will be added to the boundary resistance if this is given in a boundary condition.

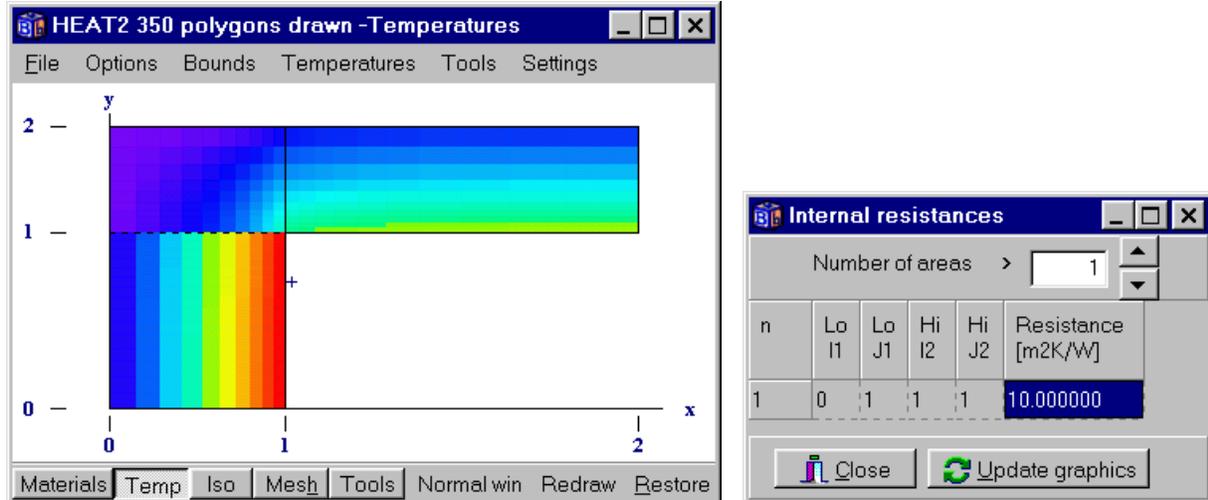


Figure 6.18: A vertical internal resistance of $10 \text{ m}^2\cdot\text{K}/\text{W}$ has been inserted at the horizontal line defined by lower left (0,1) and upper right (1,1). The calculated temperature field is shown to the left.

6.9 Solving the problem

6.9.1 Steady-state stop criteria

There are three ways to give a stop criterion for a steady-state simulation, see Figure 6.19. The first way considers flows. The sum of all heat flows (positive and negative) entering the boundary surfaces Q_b , divided by the sum of the absolute values of all these heat flows, must be less than a given value F , see Eq. (6.1). This value is recommended in the European Standards to be 0.001 (CEN, 1995). The heat flow due to internal modifications, such as heat sources, will also be accounted for when F is calculated.

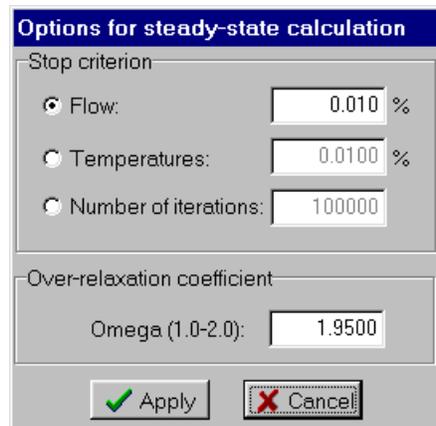


Figure 6.19: Steady-state stop criteria.

$$\frac{\sum Q_b}{\sum |Q_b|} \leq F \quad (6.1)$$

The second possibility concerns temperatures. The calculation is interrupted when the relative difference of the cell temperature of two succeeding iterations, valid for the whole computational area, is less than this value R , see Eq. (6.2).

$$\left| 1 - \frac{T_{i,j,k}}{T_{i,j,k}^{new}} \right| \leq R \quad (6.2)$$

Smaller values of F and R give in general smaller numerical errors, but longer computational run-time. It is normally sufficient to put F to values between 10^{-5} and 10^{-3} . The default value is 10^{-4} (0.01%). In some cases, the criteria R and F can be "accidentally" fulfilled during the convergence of the numerical solution. This is avoided by letting the criterion be valid at ten *consecutive*² time-steps before the simulation stops.

The third way is simply to give a maximum number of iterations.

6.9.2 Successive over-relaxation used in the steady-state calculations

Different numerical techniques can be used to decrease the CPU-time in the steady-state case. One example is the successive over-relaxation method (Hirsch, 1992), see Section 3.8. The optimal relaxation factor ω lies typically in the range 1.8-2. The default value is $\omega=1.95$.

6.9.3 Simulation time for transient calculation

Simulation stop time is given in *Solve/Options for transient calculation*, see Figure 6.20. The start time, also denoted by *current time* since it changes with the simulation, may also be arbitrarily chosen. A reason for changing the start time from zero may be to fit the start time to a certain function time step. As an example, assume there is a function with weekly step-wise linear values during 10 years. If you only want to use the last years' data, set the start time to 9 years. A special format for time input may be used (instead of using seconds), see Section 6.7.1. The current simulation time will be put to zero when a reset is made, see Section 6.9.6.

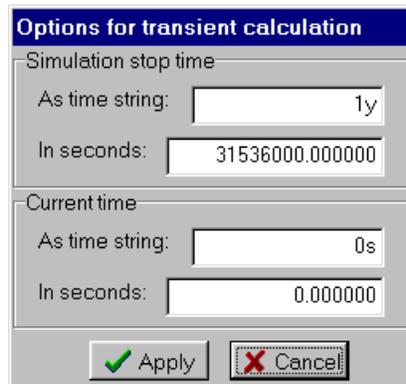


Figure 6.20: Simulation time and current time.

² It was sufficient if the criterion was fulfilled ten times, not necessarily consecutive, in earlier versions of HEAT2.

6.9.4 Simulation window

A window with information about the calculation is shown during the simulation. Figure 6.21 shows the steady-state and the transient cases, respectively. Information is given such as stop criterion, iteration, errors, heat flows, maximum and minimum node temperatures, and number of cells. There are short-cuts to *boundary flows*, see section 6.10.1, *Stop criterion*, see Sections 6.9.1 and 6.9.3, and to *Screen update*, see Section 6.9.5.

In the steady-state case (left), the relaxation coefficient may be changed.

In the transient case (right), the stable time-step is shown in the lower left side. If functions are used, the function values will be shown in the lower right side.

The “**Turbo**” button enables more CPU-time to the actual iteration (solving) process. The performance of other processes (so-called threads), such as moving or rotating the graphics, (or even working with other programs simultaneously), may then be slow. A disabled “**Turbo**” will provide more CPU-time to other processes, but will on the other hand increase computational time for the problem.

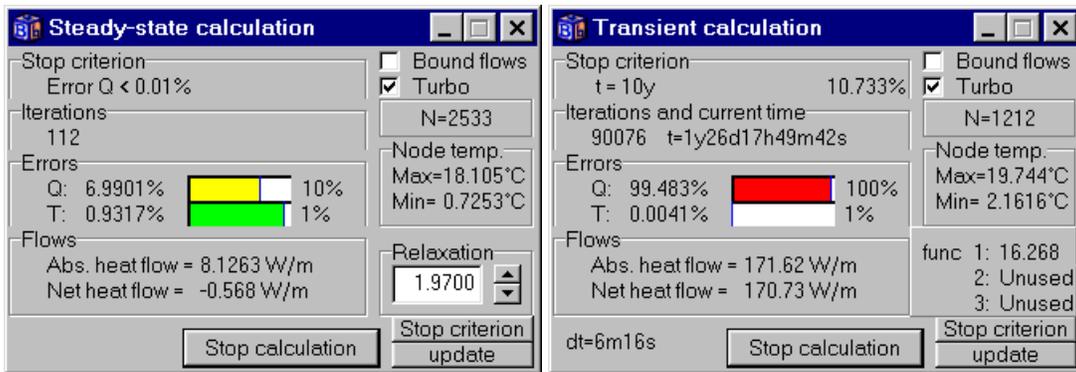


Figure 6.21: Windows shown during steady-state and transient simulations.

6.9.5 Screen update

Data shown during the calculation are updated as defined in item *Solve/Screen update*, see Figure 6.22. The update interval may depend on number of iterations, on CPU-time (real time), or on simulation time in transient cases. The default value for the screen update is set to every 3:rd second. Note that the update interval also affects the recording interval, see Section 6.11.

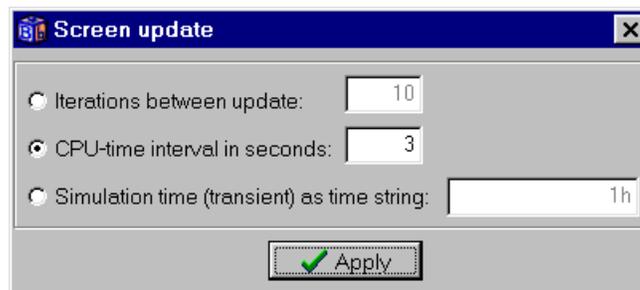


Figure 6.22: Data shown during the calculation are updated as defined here.

6.9.6 Reset

There are three options for resetting a calculation. The first one *Solve/Reset calculation* sets the iteration number and the current simulation time to zero. The second one *Solve/Reset temperatures* initializes the temperature field. The third one does both.

6.10 Output

6.10.1 Boundary flows

Figure 6.23 shows the windows for boundary flows for steady-state (left) and transient problems (right). The top row shows the sum of all heat flows through the boundary segments in (W/m). The second row shows the sum of all the positive flows. *All flows entering the solid material are positive, all flows from the solid material are negative.* The first row shows the number of iterations that has been made. In the transient case, the current simulation time is shown on the second row to the right. The flows through each boundary segment are given in units (W/m²) and (W/m). The accumulated heat flows (kWh/m) are given in the transient case. All boundaries with zero flow are shown in the last row.

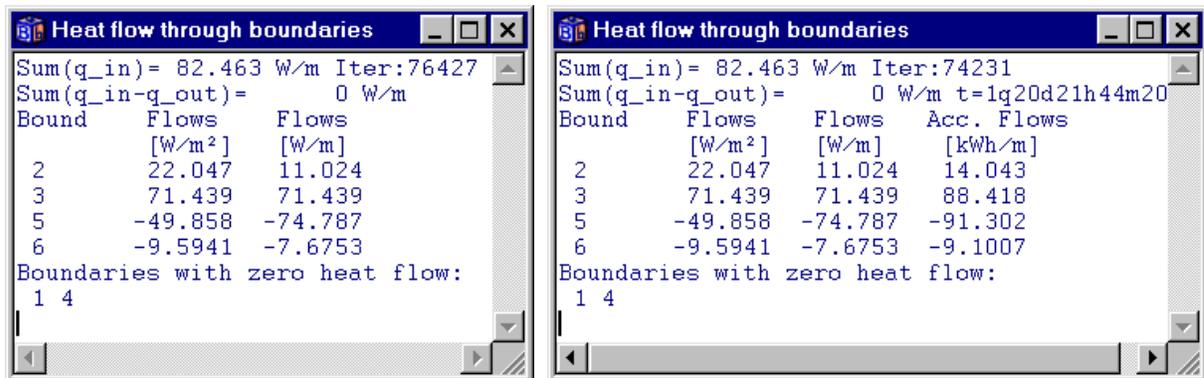


Figure 6.23: Windows showing boundary flows for steady-state (left) and transient problems (right). Data may be cut and pasted by right-clicking in the window.

6.10.2 Heat flows for internal modifications

The heat flow through the four boundaries of an internal modification may be viewed. As an example, consider Figure 6.24 showing a problem with two internal modifications. The first one is a heat source of 3 W/m, the second one is a cavity with radiation (the white field with code 2:2).

The boundary condition on the left and right sides are $T=0$ with $R=0.04$ and $T=1$ with $R=0.13$, respectively. The upper and lower boundaries have an adiabatic ($q=0$) boundary condition.

Note the boundary flows. The heat flow at the right boundary will be positive (directed into material) above the isotherm $T=1$ (the horizontal line in the middle to the right), and negative below it.

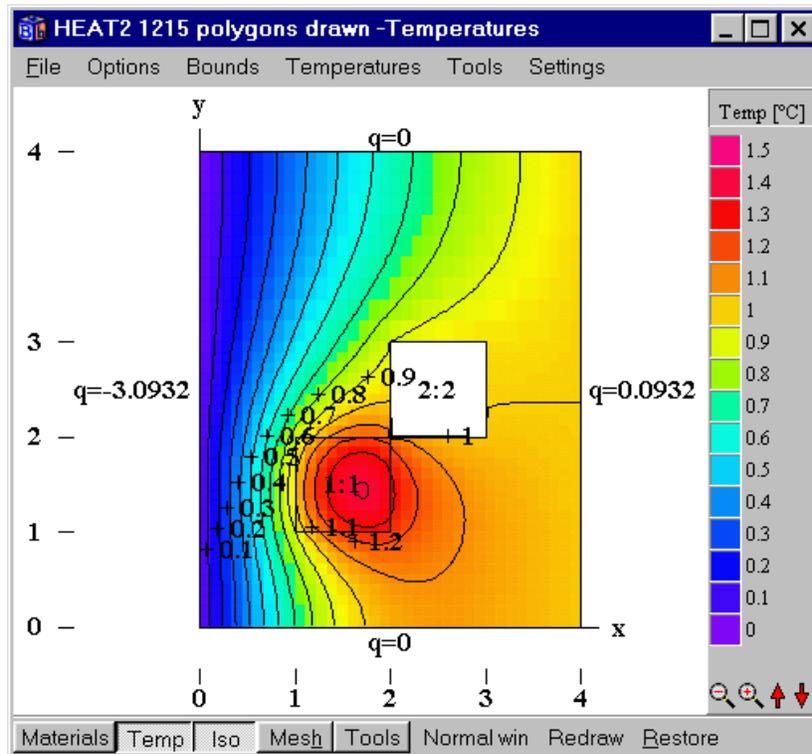


Figure 6.24: Problem with two internal modifications. The first one is a heat source of 3 W/m, the second one is a cavity with radiation (the white field). The boundary condition on the left and right sides are $T=0$ with $R=0.04$ and $T=1$ with $R=0.13$, respectively.

Figure 6.25 shows the steady-state heat flows through the four sides and the total flow of each area (a positive flow is directed from the cavity into the solid material). All four flows are positive from the first area with the heat source. The heat will flow into the cavity on the lower and right sides. Since no effect is given (ventilation is zero) in the cavity, the net heat flow is zero. The net positive flow (sum of all positive flows) and the net flow for all areas are shown at the bottom.

The flows for each area can be given in units (W/m²) or (W/m), see top row.

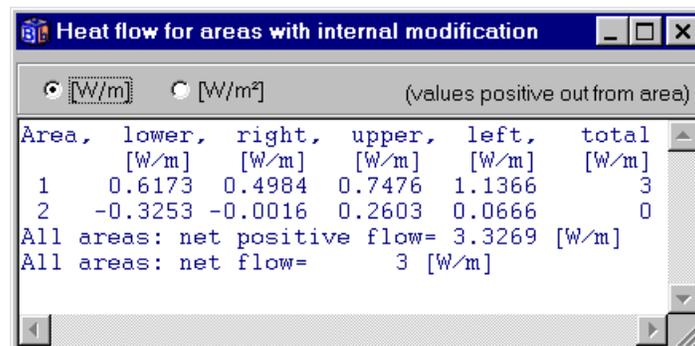


Figure 6.25: Heat flows for internal modifications.

Figure 6.26 shows the boundary flows. Note the absolute heat flow in the simulation window, see Figure 6.27. The absolute heat flow ($q=3.4201$) involves the net positive flow ($q=3.3269$) from the internal modifications and positive boundary flows ($q=0.0932$). The net heat flow at the bottom in Figure 6.27 relates to the boundaries, see Figure 6.24, (i.e. $q=0.0932-3.0932=-3$ W/m).

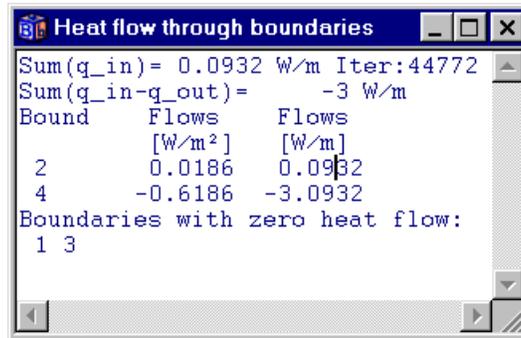


Figure 6.26: Boundary flows.

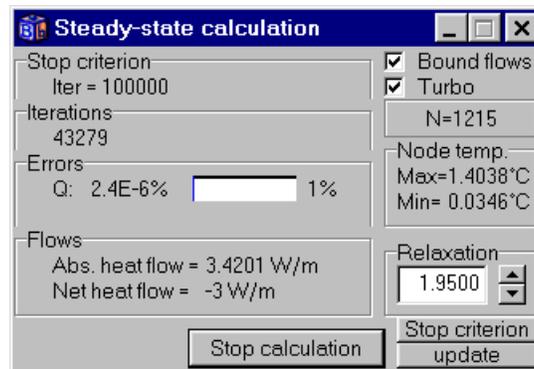


Figure 6.27: Heat flows in the simulation window.

6.10.3 Results for radiation

Figure 6.28 shows the results for area 2 for the problem in the last section. The second row shows the gas temperature T_a , the ventilation conductance (or convectance) K_v (W/K), the area of the cavity (m²), and the number of radiation nodes N_R .

The following results are given for each radiation node if box *Details for area* is checked: calculated surface temperatures T_{sj} for $j=1..N_R$ (see Figure 5.3), the temperatures in the boundary cells T_{bj} , the conductances K_{bj} between the center of the cell and the surface of the enclosing cells i,j , and heat flows q_{bj} (W/m).

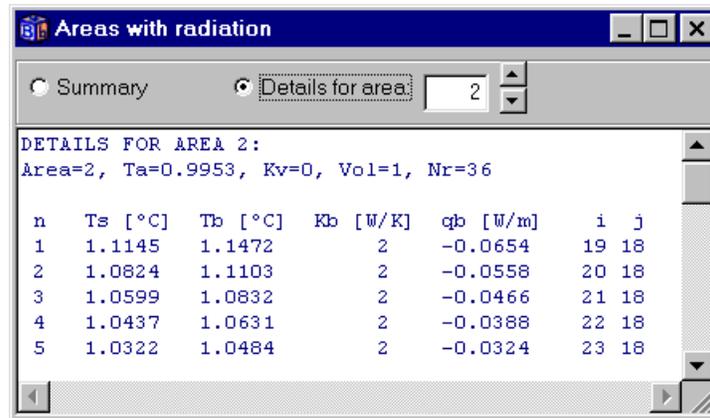


Figure 6.28: Results for areas with radiation.

6.10.4 Temp at point x,y

The temperature T , ($^{\circ}\text{C}$), and the heat flow components Q_x , Q_y , (W/m^2), may be obtained for any arbitrarily chosen point (x,y) by positioning the cursor in the desired location, see Figure 6.29. The coordinates and indices (i,j) for the numerical cell (where the point lies) are also shown. Areas of particular interests may be enlarged by left-clicking and moving the mouse up and down.

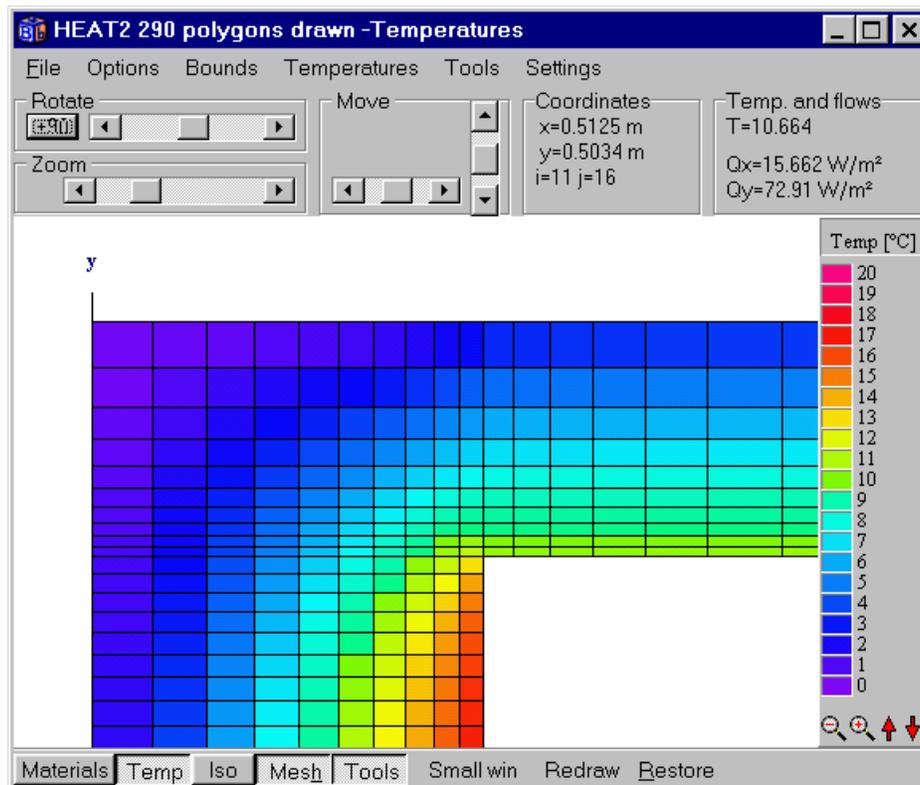


Figure 6.29: Temperatures and flows may be obtained for a point (x,y) .

6.11 The recorder

Flows and temperatures may be saved to a list during the simulation. Figure 6.30 shows the recorder window (*Output/Recorder*). Recording will be made during the simulation if **Record enabled** is checked. The pick list shows four options. The first option will record the sum of all boundary flows (W/m). The second will record the sum of heat flows through arbitrarily chosen boundaries. Figure 6.30 (right) shows a case where the flows for boundaries 2 and 3 will be recorded. The third and the fourth options will record temperatures or heat flux (W/m²) for an arbitrarily chosen point (x,y).

The values will be recorded at each screen update (*Solve/Update* or short cut button **Update interval**) as shown in Figure 6.31, in this case every simulated hour. Each row in the list will contain the time (in time string format or in seconds) and the actual value. The update interval may be changed during the simulation (or between two simulations).

A list may be saved (*.REC) and later be opened. Data may be cut and pasted from the *Edit* menu.

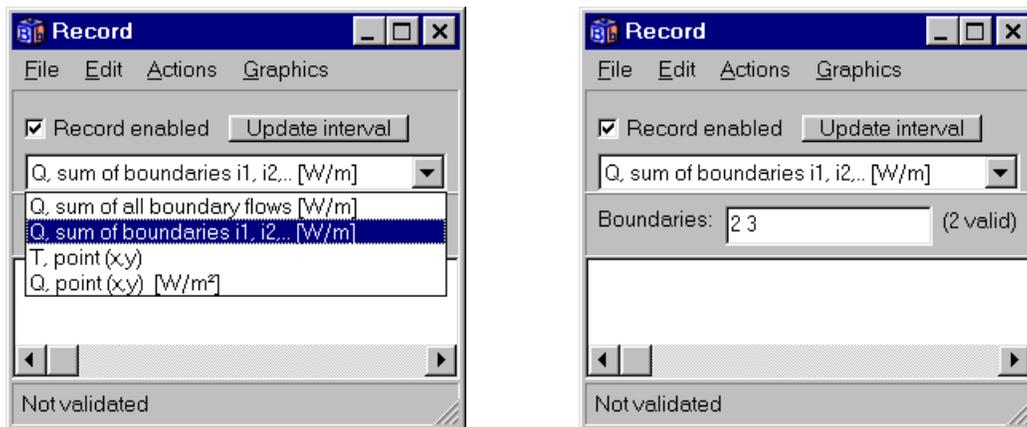


Figure 6.30: Flows and temperature may be saved to a list during the simulation. In this case it is the sum of the heat flows for boundaries 2 and 3 that will be recorded.

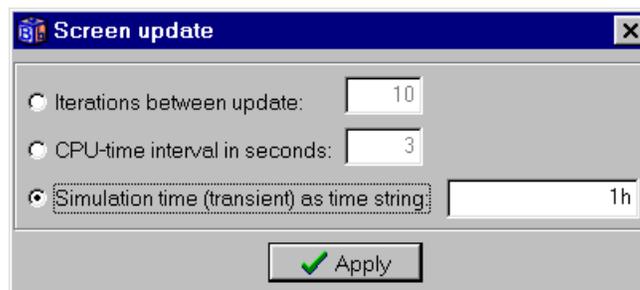


Figure 6.31: The time interval for recorded data is given in item *Solve/Update*.

A recording will start whenever the simulation starts and stop whenever the simulation stops if the recorder is enabled (even though the recorder window is closed). An indication is shown (**REC** will be displayed on the bottom row in the main window) if the recorder is active. Recorded values will be appended to the end of the list. The list may be cleared in the *Actions* menu. A chart of the recorded values may be viewed by pressing the item *Graphics* in the record window.

Figure 4.19 and Figure 4.20 shows the heat flow through the internal boundaries in a transient case with a periodic variation on the external sides. Figure 6.32 shows the heat flow through the external boundaries for a steady-state case. The number of iterations is shown on the x -scale. The update interval (*Solve/Update*) is set to each iteration step in this case. The “rollercoast” behaviour is due to the relaxation coefficient ($\omega=1.95$). Note that the label on the x -axis is “iterations” in the steady-state case. In the transient case, the label will be an optional time string (seconds, minutes, hours, days, months, years, or a mix of these).

The chart will be automatically updated if visible during the simulation. This will increase the run time, so it may be a good idea to close the window (and the recording window) if speed is crucial.

It is possible to view the complete history, or just the last 20 points (*Options/Show only 20 last points*).

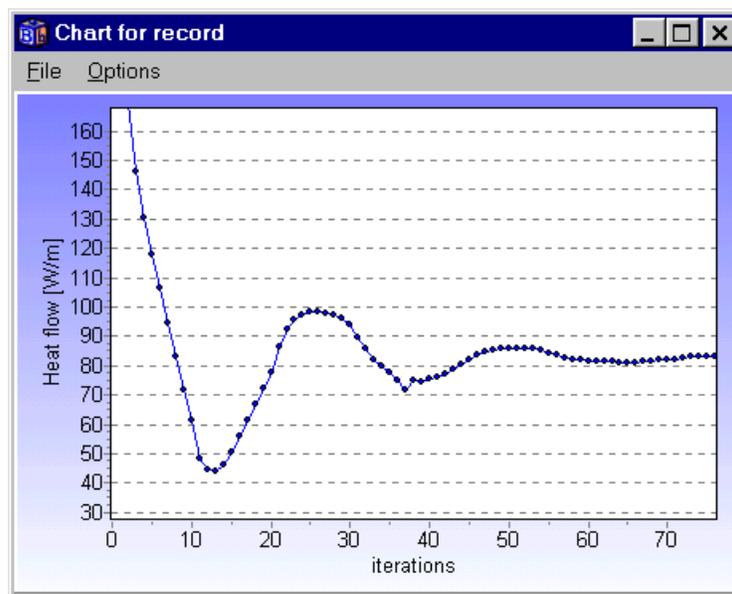


Figure 6.32: Chart showing recorded values.

The chart may be saved to the clipboard or to a file in Metafile (EMF, WMF) or in bitmap (BMP) formats. Printing from HEAT2 directly is also possible.

Areas in the chart window can be enlarged by left-clicking the mouse cursor in the upper left corner and moving it to the lower right one. The area will be restored by making the opposite move by starting in the lower right corner, or by clicking menu item *Options/Reset axis*. The area may be moved by right-clicking. The chart may be edited (*Options/Edit chart*). Two of the features that might be mentioned here are changing the titles and adding legends.

Note that it is possible to open function values (*.FUN) in the record list and graphically see the variation as stepwise-linear values.

6.12 Info log

The info log (item *Output/Info log*) contains information of input data and generated numerical mesh for the considered problem, see Figure 6.33. The stable time step is shown, and the location for the cell that determines it. The info log is automatically created when a simulation is started or a problem is loaded.

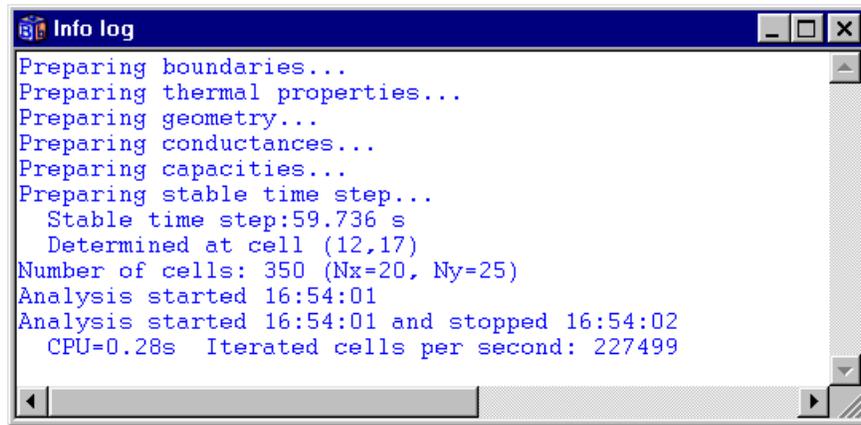


Figure 6.33: The info log.

6.13 Graphics

6.13.1 Introduction

Figure 6.34 shows the “graphic window” that can display temperatures, isotherms, materials, numerical mesh, boundary conditions, and more. The graphic window will be updated whenever new input data are given. Pictures showing temperatures and flows are updated during the simulation (may be changed in item *Temperatures/Update figure on solve*). Colors and fonts may be changed in item *Settings*.

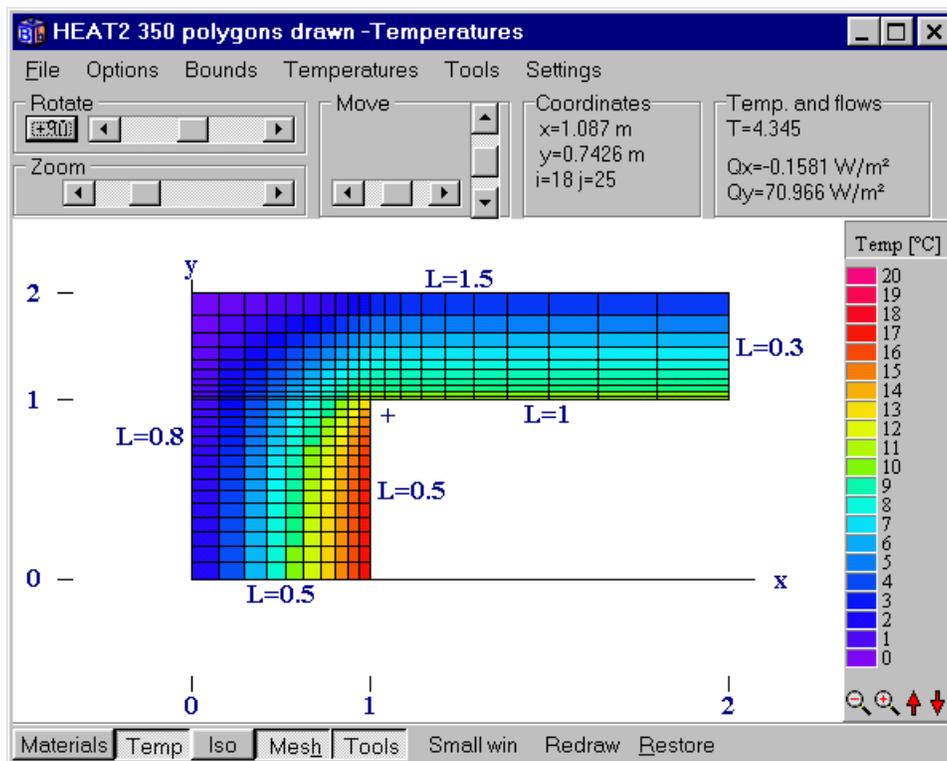


Figure 6.34: The graphic window.

6.13.2 Basic commands

The picture may be rotated, moved, or zoomed using the top tool-bar, the mouse, or directly from the keyboard, see Figure 6.35 for mouse and keyboard commands. The list is also displayed in item *Tools/Help shortcuts*.

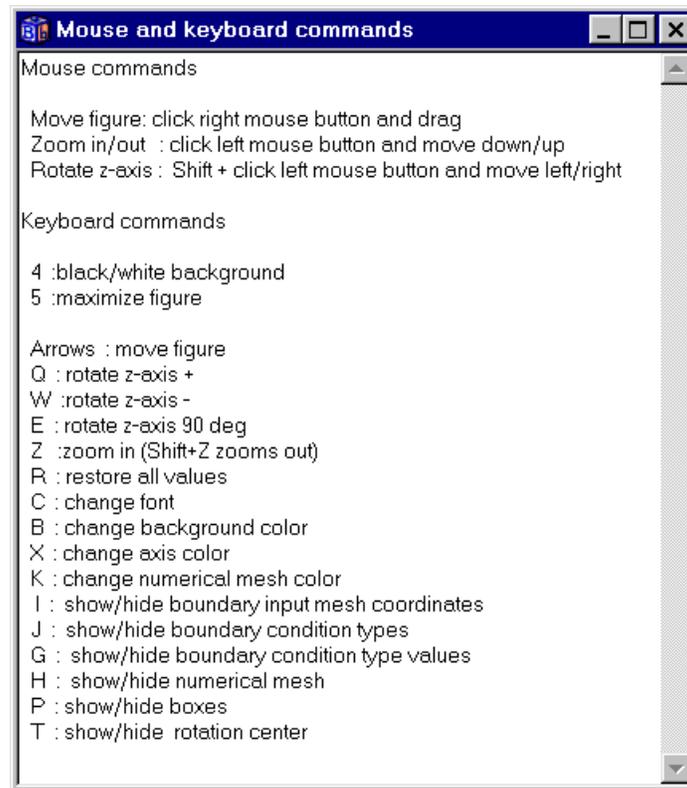


Figure 6.35: Shortcuts for basic commands.

6.13.3 Menu item *File*

6.13.3.1 *Saving images*

The images may be saved to a bitmap format (*File/save image as bitmap*) or cut to the windows clipboard (*File/Cut image to clipboard*).

The image may be cut and pasted into another program, e.g. Microsoft Word (as used in this manual). Using the cut option in *File/Cut image to clipboard* will give an image without the top window frame, see Figure 6.36 (left). Using the two keys **Alt-Print Screen** will capture the current active window (the **Print Screen** key will capture the whole screen, see Figure 4.19). The images can be re-scaled in e.g. Word. Note that the resolution will be the one on your screen (1024-768 will give better quality than 640-480). Printing directly from HEAT2 will produce better quality with the printer's resolution (a 15" screen with a resolution of 1024-768 has about 75 DPI as a comparison to a 600 DPI printer), see Figure 6.39.

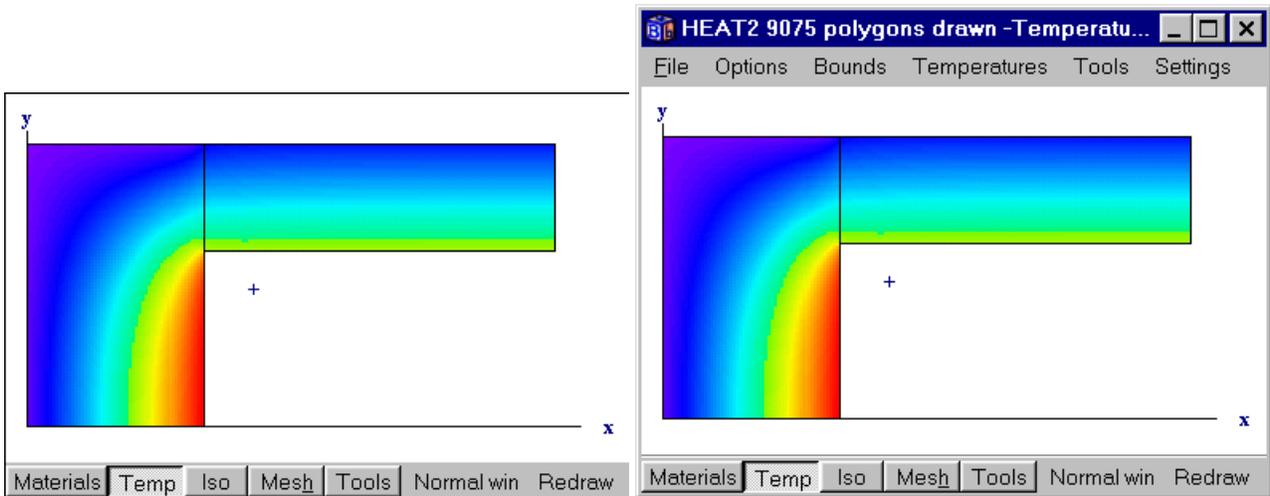


Figure 6.36: Use the cut option in *File/Cut image to clipboard* to get an image without the top window frame (left). Press **Alt-Print Screen** to capture the current active window (right). The active window has often a blue top menu row.

6.13.3.2 Printing images from HEAT2

Figure 6.37 shows the print options where the pen thickness can be changed.

There is an option for a simple preview with a green frame that outlines the paper, see Figure 6.39. The image can be positioned and enlarged or reduced in size.

Directly printing by the print command will use the resolution of the current printer, see Figure 6.38. It is also possible to print a screen dump (with poor quality).



Figure 6.37: Print options.

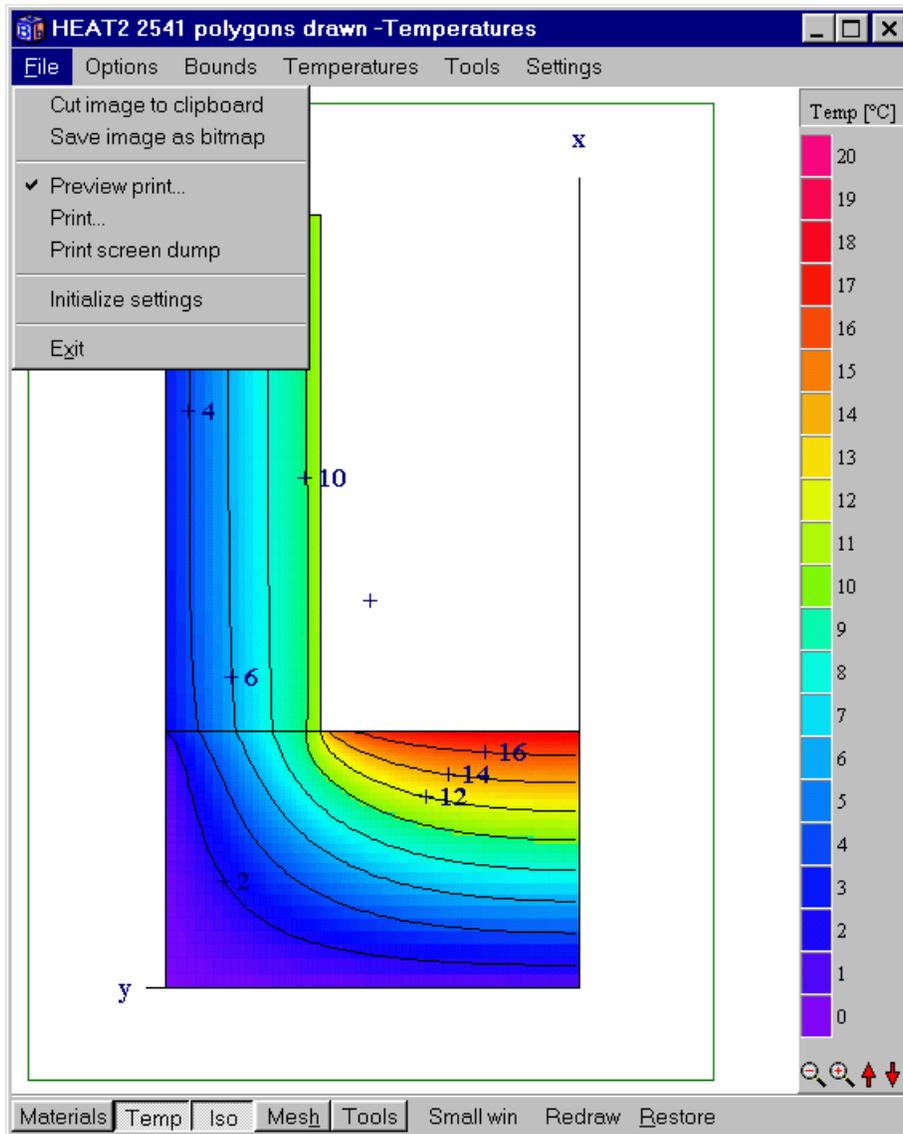


Figure 6.38: A simple preview with a green frame that shows the paper borders.

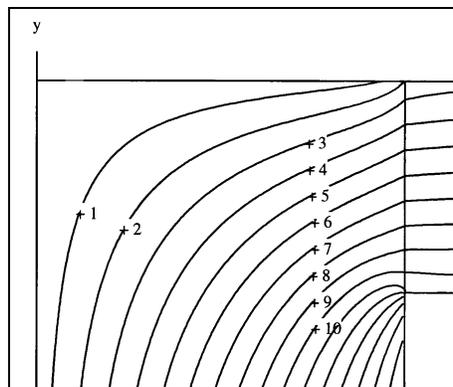


Figure 6.39: Isotherms printed directly from HEAT2. The printer's resolution will be used, in this case 600 DPI. This picture has then been scanned into this document.

6.13.4 Menu item *Options*

Figure 6.40 shows the menu item *Options*.

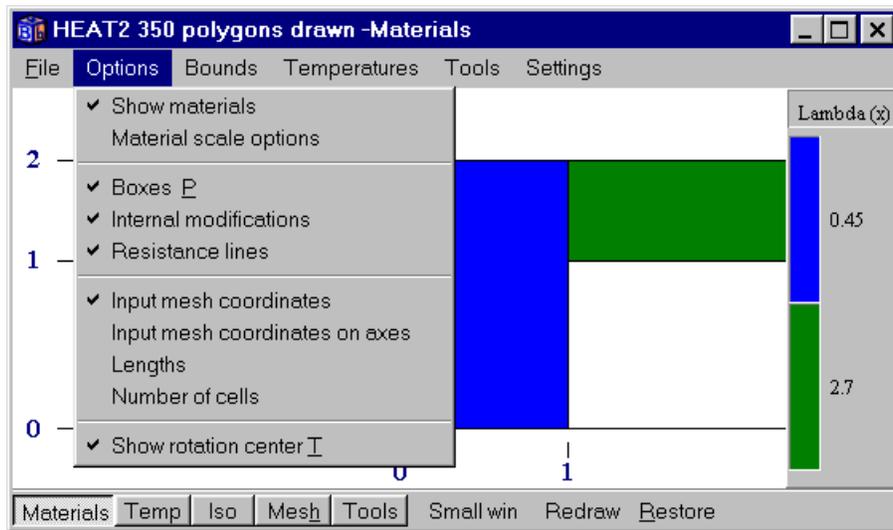


Figure 6.40: Menu item *Options*.

6.13.4.1 Viewing thermal conductivities and heat capacities

Click "Materials" on the bottom tool bar (or check item *Options/show materials*) to view thermal conductivities, see Figure 6.40. The color and the scale gives the thermal conductivity.

Clicking the scale area to the right (or *Options/Material scale options*) brings up the window shown in Figure 6.41. Figure 6.42 (left) shows colors sorted by material names (two material may have the same thermal properties but different names). Figure 6.42 (right) shows sorting by defined areas (white indicates default value), see Section 4.3.

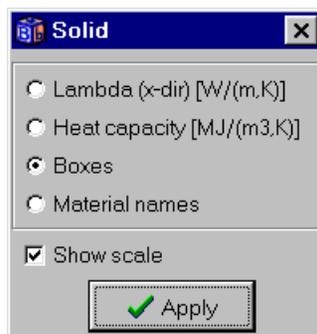


Figure 6.41: Click the scale bar (or *Options/Material scale options*) for options.

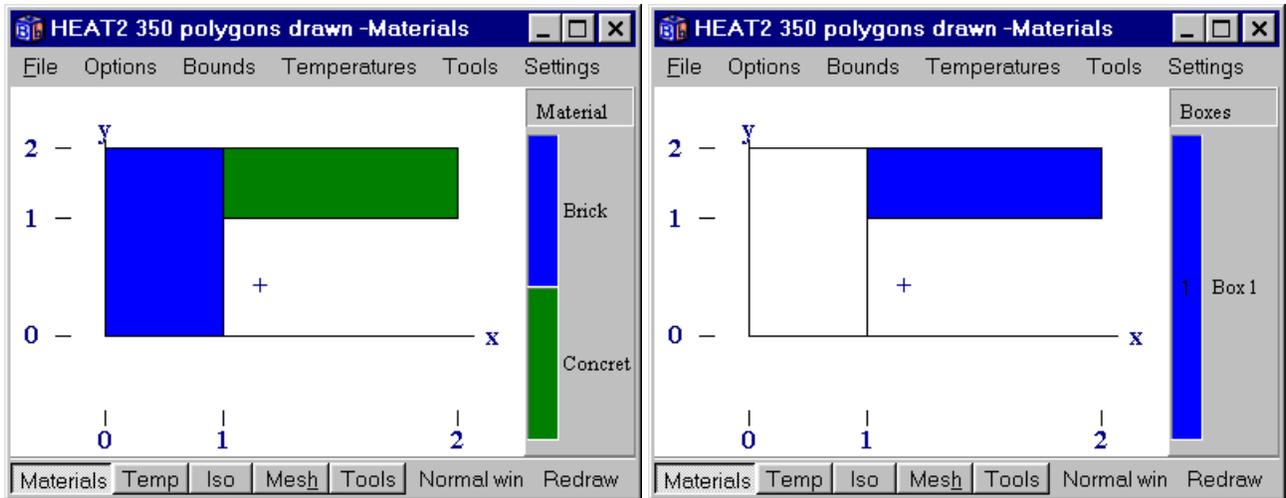


Figure 6.42: Colors indication material names (left) and boxes (right).

6.13.5 Menu item *Bounds*

Figure 6.43 shows the menu item *Bounds*. In this case, the outlined boundaries are shown and the boundary conditions. It is also possible to view current heat flow in (W/m²) or in (W/m), and the boundary air temperature (this will change if a temperature function is used).

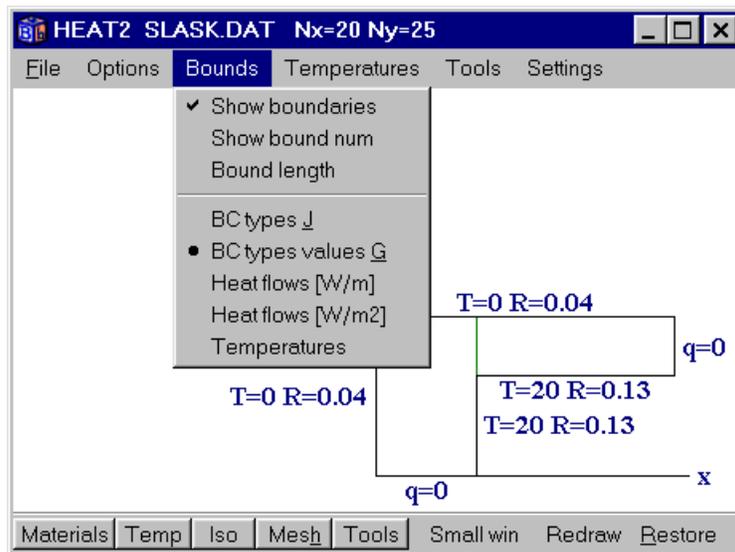


Figure 6.43: Boundary conditions as shown by *BC types values*.

6.13.6 Menu item *Temperatures*

Figure 6.44 shows the *Temperatures* menu. The picture is updated (unless *Temperatures/Update figure on solve* is disabled) during the simulation at time interval given in *Solve/Screen update*.

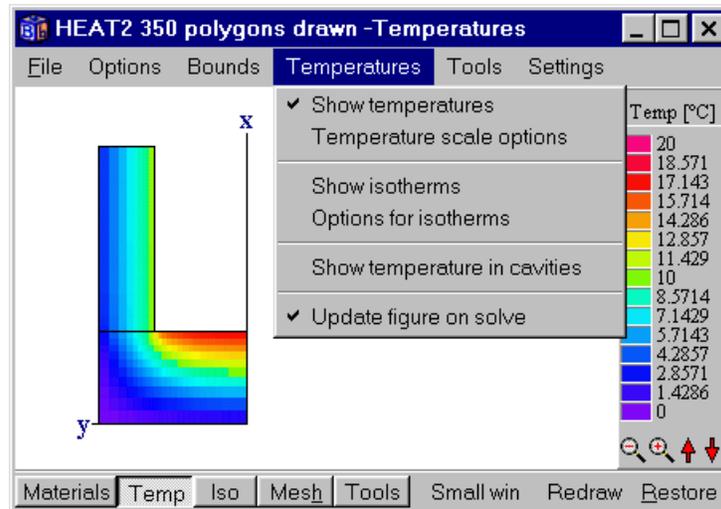


Figure 6.44: Temperatures (rotated figure). Note that the number of bars in the scale will be reduced if the window is too small to hold all (in this case the number of bars is reduced automatically from 21 to 15).

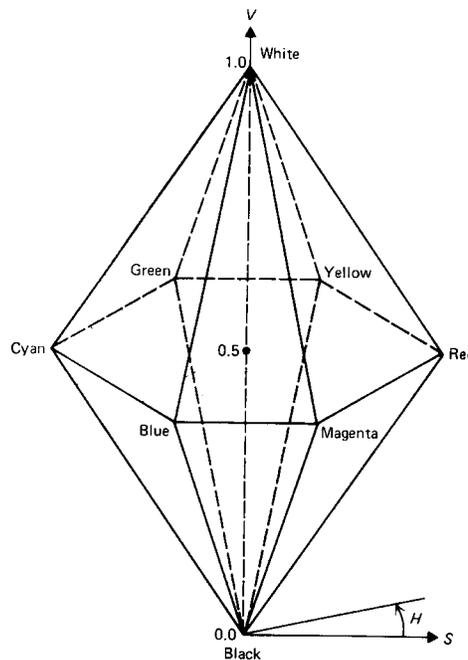
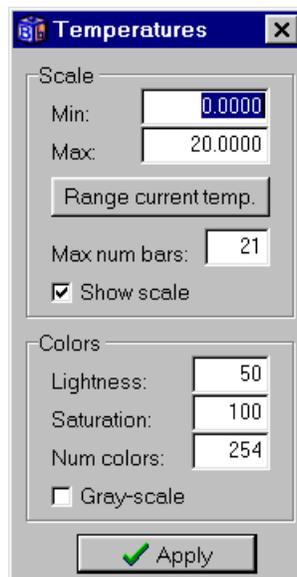


Figure 6.45: Options for temperature scale (left). The hue-lightness-saturation color model (right).

Click the scale bar (or *Temperatures/options for scale*) for options, see Figure 6.45 (left). The minimum and maximum temperatures for the scale may be given here. Option **Range current temp** sets the scale to the maximum range accounting for all surfaces within the drawing window.

The number of bars in the scale may be changed. The temperature may be shown in colors or in gray-scale. The colors are calculated according to the hue-lightness-saturation model used by Tektronix and based on the Ostwald color system, see Figure 6.45 (right). Hue is the angle

around the double hexcone. Saturation is measured radially from the vertical axis, from 0 on the axis to 1 on the surface. Lightness is 0 for black (at the lower tip of the double hexcone) and 1 for white (at the upper tip).

The hue variable is defined from magenta to blue using a certain number of colors (**Num colors**).

A certain temperature interval may be disabled by clicking the temperature scale. Figure 6.46 shows two disabled temperatures, $T=11$ and $T=16$ °C. This means that the surfaces with temperatures between 10.5-11.5 and 15.5-16.5 will not be drawn (the temperature step is 1 in this case since there are 21 bars). A click with the left mouse button on the scale will enable/disable an interval. A click with the *right* mouse button will enable/disable all intervals.

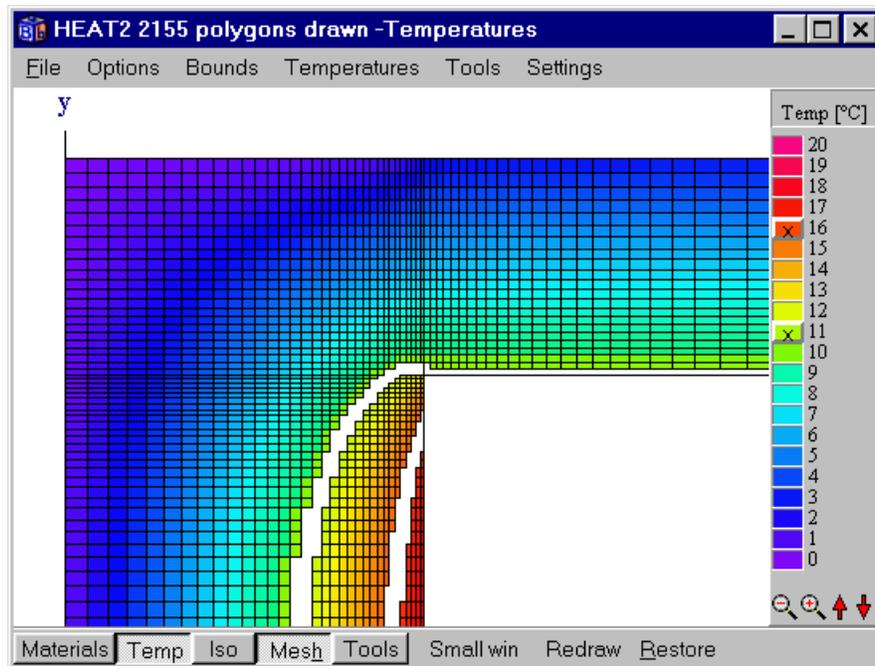


Figure 6.46: Temperature intervals may be hidden by pressing the scale color, in this case $T=16$ and $T=11$ °C.

6.13.7 Viewing isotherms

Figure 6.47 shows options for isotherms (*Temperatures/Options for isotherms* or short-cut by right-clicking on the menu bar item **Iso**). The result is shown in Figure 6.48.

The range may be set automatically to the min and max temperature for each screen update, or to a fixed range. The *Temperature step* is the interval in degrees for the isotherms between the given minimum and maximum temperatures.

A calculation of isotherms is made for an arbitrarily chosen discretation in pixels. A smaller value will give a denser grid resulting in smoother isotherm curves, but will on the other hand increase the time it takes to show them. The smallest value that may be given in *pixel step* is 1. The grid may be shown by checking *Show pixel step*.

Isotherm curves may be labeled with the temperature values. There is an easy way to spread these values along the curves and avoiding them to clump together. There is a number of short lines that are put together in every isotherm. When the number of lines that have been drawn equals the value given in *Configuration*, the isotherm will be labeled at that position. Note that the pixel

step affects the position. A value of the configuration may typically be about 10-150. The best way to get a clearly labeled figure is simply to try few number of combinations. The labels is shown when *Show labels* is checked.

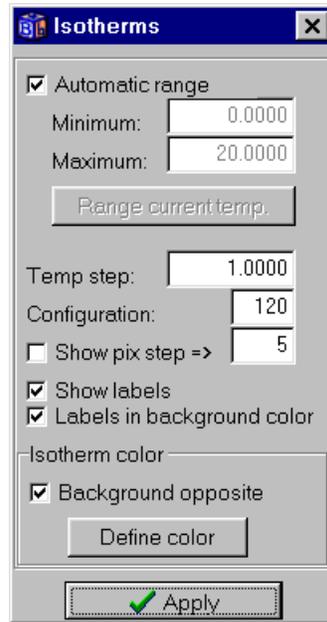


Figure 6.47: Isotherm settings.

The color of the isotherms may be altered in *Color for isotherm*. The check box *background opposite* applies to white and black (if the background is black then the isotherm color is white, and vice versa). The thickness of the curves may be increased in menu item *Settings/Thick pen*.

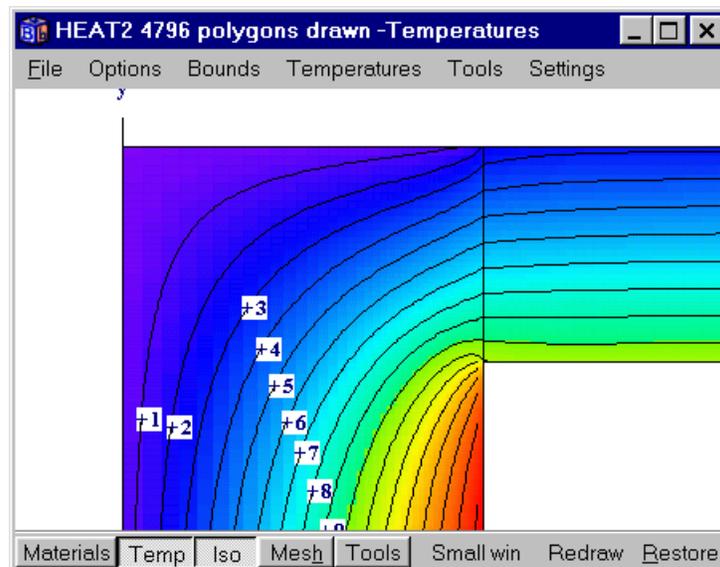


Figure 6.48: Isotherms with interval of 1 °C labeled in background color.

6.13.8 Menu item *Tools*

It is possible to 'maximize' the area (item *Tools/maximize*). This means that the scale in each of the two directions will be adjusted to the maximum length in its own direction, see Figure 6.49. As a result, the picture will be stretched in one direction. In some cases, this will make it easier to see details without having to enlarge areas (see the example in Section 9.3 and input file *cenex1.dat*). If no 'maximizing' is chosen (default), the scale will be adjusted to the largest length of the two directions. In this case, the scale will be the same in all directions.

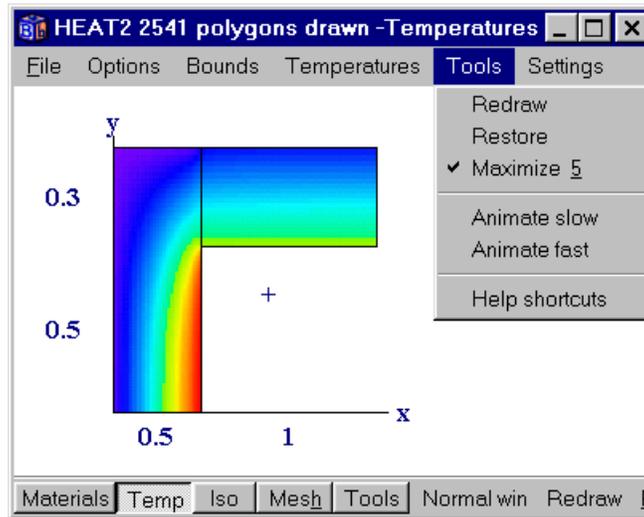


Figure 6.49: The scale in each of the three directions is adjusted to the maximum length in its own direction ('maximized area').

6.13.9 Menu item *Settings*

Fonts and colors may be changed in item *Settings*, see Figure 6.50.

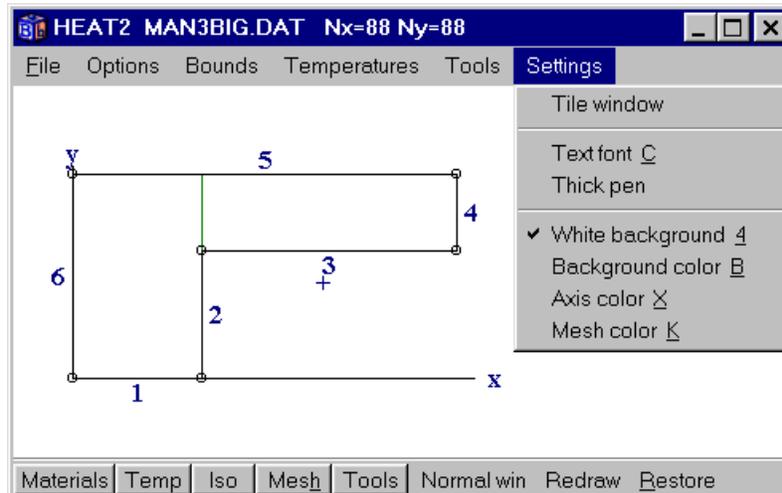


Figure 6.50: Fonts and colors may be changed in item *Settings*.

7. HEAT2W - graphical preprocessor for HEAT2

7.1 Introduction

HEAT2W is an interactive drawing program that serves as a preprocessor to HEAT2. With this program it is even more simple to describe the geometry and materials for a large range of heat transfer problems. The program works as common Windows programs for creating drawings.

The construction is drawn in HEAT2W using overlapping rectangles of different materials. A file, always named HEAT2W.DAT, is generated that may be read by HEAT2, see Figure 7.1. Boundary conditions have to be specified in HEAT2 before the problem can be solved. Note that the programs work with different input files. HEAT2W can save and open drawn construction as H2W files. HEAT2 works with .DAT files. It is easy to toggle between the two programs by using the Windows commando <Alt><TAB>.

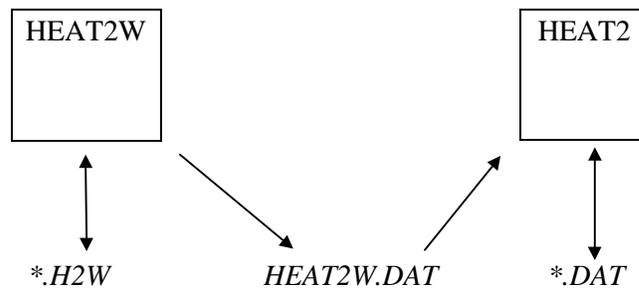


Figure 7.1: Schematic figure how HEAT2 and HEAT2W work. HEAT2W may be used as a preprocessor to HEAT2. The construction is drawn in HEAT2W using overlapping rectangles of different materials. A file, always named HEAT2W.DAT, is generated that may be read by HEAT2. It is easy to toggle between the two programs by using the Windows commando <Alt><TAB>.

HEAT2W provides a few advantages being a program under the Microsoft Windows environment, such as exporting (or importing) objects (e.g. pictures or text) via the clipboard to other programs for drawing or editing. It is also possible to scan a building design drawing and import it via the clipboard into HEAT2W. The actual heat transfer problem can be described by drawing rectangles "on" the imported design drawing.

Note that it is possible to do all input in HEAT2. For simpler cases, it is often quicker to work only with HEAT2.

7.2 Drawing, moving, and changing objects

The tool bar shown to left of the drawing area, see Figure 7.2, has four icons: 'Select', 'Zoom', 'Rectangle', and 'Text'. The tools are chosen by clicking on them with the mouse or picking the item in menu item 'Tools'. To draw a rectangle, choose the 'Rectangle' (third icon from the top). Put the cursor somewhere in the drawing area and press the left mouse button. Then, move the mouse and release the button. There will now be a rectangle in the drawing area.

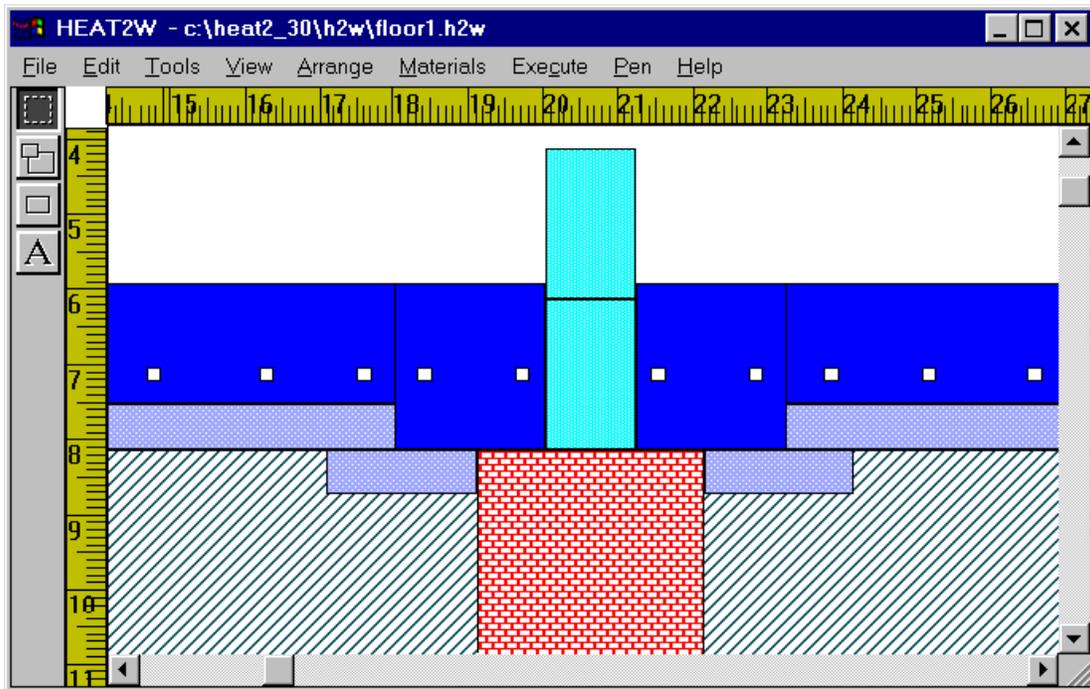


Figure 7.2: HEAT2W has a menu system, a drawing area, a tool bar on the left, and vertical and horizontal 'scrollers' and rulers. This example shows floor heating in a concrete slab.

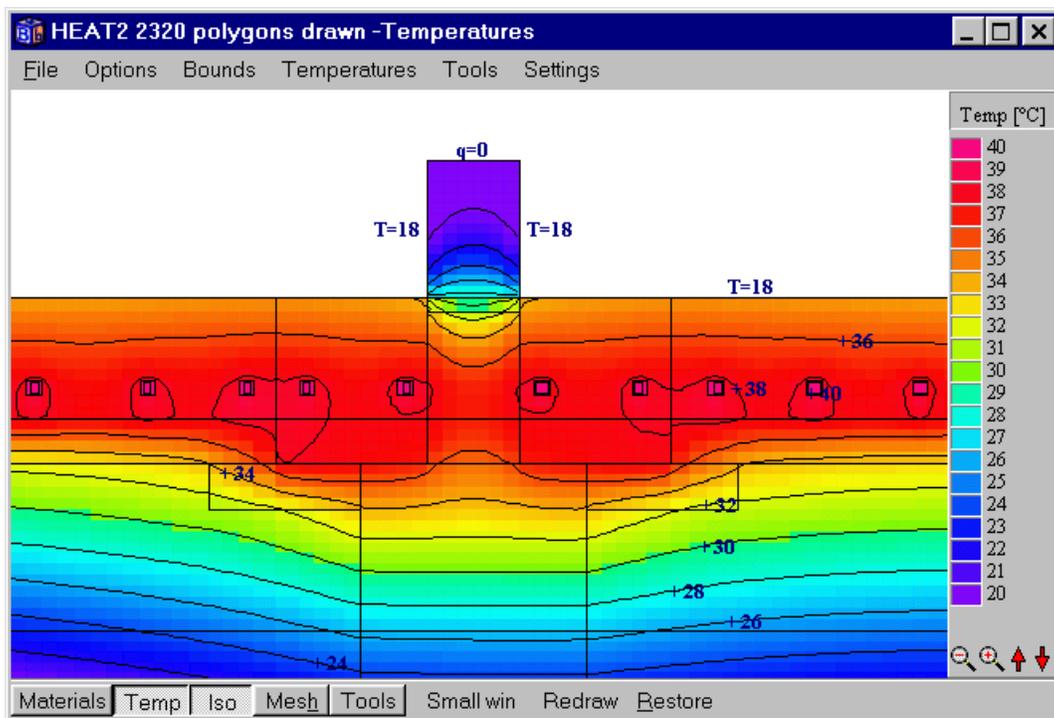


Figure 7.3: Calculated temperature field in HEAT2.

To choose another material, press the TAB key (shortcut for item 'Pick' in menu item 'Materials'). Every material has its own unique brush. The brush is defined by color and pattern. It is discussed later how to define a new material and brush.

Now, choose 'Select' at the top of the tool bar and click once on the rectangle that you drew. The selected rectangle will now get marks on each corner. Move the rectangle around in the drawing area with the mouse. The rectangle will be copied instead of dragged if the <Ctrl> key is pressed and held down while moving the mouse.

It is possible to select multiple rectangles, see Figure 7.4. These objects may be moved, re-sized, etc. Rectangles that are hidden under other rectangles may be put on the top with the commando 'Send to Front' in 'Arrange'. To put it back use 'Send to Back'.

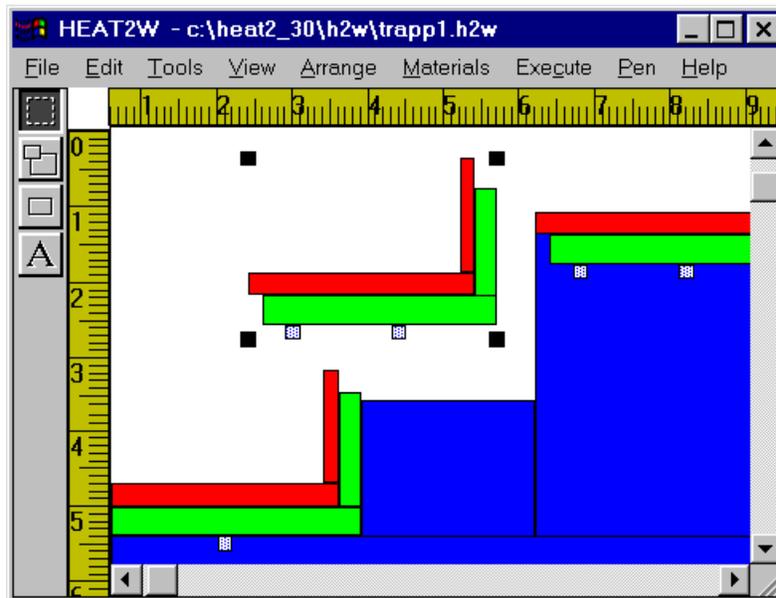


Figure 7.4: Multiple objects may be selected, copied, and moved together. In this case, a stairway heated by pipes is built by copying and pasting of one stair.

Any object can be increased or decreased in size by stretching any of its corner. The length and width can be specified by a double click on the object. To un-select any object, move the cursor to an empty area and press the left button.

The material of the rectangle may be changed by picking the desired item in the material list (TAB key). The material list will give you information on every rectangle that you click on.

Tip If the button is pressed and held down for about a second (and the tool is 'Select') you will get a bitmap instead of a region. Dragging a bitmap provides much better feedback than dragging a region. It may however be 'heavy' to drag a bunch of bitmapped rectangles on a slow computer.

Any area can be enlarged by selecting the tool 'Zoom'. To return to the ordinary picture go to item 'View' and 'Actual Size'. The item 'Fit in Window' shows the maximum area for the drawing space. The icon marked with an 'A' at the bottom of the tool bar is used for putting text in the drawing area. Text can be selected and moved around as a rectangle may be moved. The color, height, and font, can be change in menu item 'Pen'.

A grid of a certain step may help you in drawing the rectangles. This step may be given in item 'Grids...' under the menu item 'View'. Choose item 'Show Grid' to view the grid.

7.3 Using the clipboard

Any drawing may be copied to the clipboard and pasted to other instances of HEAT2W. A drawing can also be pasted into other Windows programs, e.g. Microsoft WORD (see Figure 7.2).

Earlier drawn constructions may be imported using the clipboard. For example, assume that you have just drawn a window that you want to put in a wall earlier saved in file *MyWall.H2W*. First, save the current file e.g. *MyWindow.H2W*, then load *MyWall.H2W* and select the wall you want to copy and export it to the clipboard. Load *MyWindow.H2W*, and paste the wall into the drawing. This allows you to build collections of construction details. Note that material list must contain all used materials in both files.

7.4 Editing the material list

There are about 20 materials defined in HEAT2W. Press the TAB key to list them. The thermal properties are taken from 'Catalogue of Material Properties', International Energy Agency, report Annex XIV, Volume 3. Average data is taken for density, specific heat capacity, and thermal conductivity. The materials can be viewed by loading the file *MAT.H2W*.

The materials may be edited (name, thermal properties, and brush) in menu item 'Materials'. Materials that currently are used in the drawing area are shown in capital letters in the list. You can only delete a material that is not in current use (shown in small letters).

To add a new material to the list is a bit tricky. Since every material has its own unique brush (defined by color and pattern) a new brush must be defined for this material. The procedure is as follows. Go to menu item 'Brush'. Choose a color in item 'Color'. As you can see the color can be given as primary colors or values in RGB coordinates. After deciding the color, choose the pattern for the brush. The pattern may be solid, brick, hatch, or degree of color saturation, to mention a few of the available patterns. The best way to obtain a 'nice' brush is to draw a rectangle, try a new brush, draw another rectangle, and so on. After this is done, enter item 'Edit' and give name and properties for the new material. Press insert and the new material is defined with the current brush.

Notice that the rectangles that was drawn during the step when you tested different brushes do not have any material name connected to them. These rectangles are invalid and if you do not delete them yourself, HEAT2W will later on when the output file will be written.

Also note that the program checks that the new brush is unique (not already in the material list). However, since more than 16 000 000 colors may be defined there is a risk that you choose a color that looks identical to a color that already exists. This is on your own responsibility. It may be good idea to define a brush for a material that are similar to other brushes of materials of the same kind. E.g. concrete, lightweight concrete and cellular concrete is defined in HEAT2W with a bluish nuance.

The material list will be saved with the current drawing to ensure that the drawn rectangles always have their materials defined. It is of course possible to save just the materials without any rectangles drawn.

It may be a good idea to add materials to the same file e.g. 'MyMat.H2W', and then load the file for every new drawing, and save it with a new filename. If a drawing is copied to the clipboard and pasted between different instances of HEAT2W make sure that the materials are always defined in the current material list.

It is also possible to start HEAT2W with a user-defined material list. Save the list as an ordinary data file (*.H2W) and copy it to HEAT2W.INI.

7.5 Writing the input file for HEAT2

As pointed out before, it is not possible to solve the problem in the current version of HEAT2W. Instead, an input file to HEAT2 is generated by choosing the item 'Process File' in menu item 'Execute'. The name of this input file is always 'HEAT2W.DAT'. Several checks are made to ensure that the drawing is valid. The area must be continuous, to mention one of the restrictions. HEAT2W will validate the area itself. The only thing you must keep in mind when working with this version is that there are not any holes in the drawing. If there are holes, these will be filled with the material you started to draw with.

The item 'Numerical Grid' lets you specify the number of computational cells in each direction. These will be spread out in an equidistant mesh. The default values are 18x18 cells. Note that there always has to be a minimum number of cells for the computational region. However, this is automatically taken care for.

7.6 Mapping the "real world" into the drawing area

The scale factor has a default value of 100:100. This means that 1 cm in the drawing area (according to the rulers) is 1 cm in the real world. Choosing 100:1 means that 1 cm is 0.1 mm, and 100:100 000 means that 1 cm in the drawing area is 10 m in the real world. This factor is set in item 'Scale Factor' in menu item 'Execute'.

The smallest rectangle that can be drawn in the drawing area is 1x1 mm². The maximum drawing area has the height and width of 1 m and is chosen by selecting menu item 'View' and 'Fit in window'. This means that the ratio between the smallest and largest rectangle you can draw is 1:1000. You can for example analyze a wall that is 500 mm thick with a thermal bridge effect due to a 0.5 mm thick steel bar across the wall. Another example is calculation of the heat loss from a house to the ground where the computational area includes soil within 50 m downwards the house. So, the smallest possible thickness of e.g. the slab will in this case be 5 cm. Note that you can change all lengths arbitrarily in HEAT2 afterwards.

7.7 Drawing areas with internal modifications in HEAT2W

Do as follows to draw areas with internal modifications in HEAT2W. Choose *heat flow* in the material list, see Figure 7.5, and draw the areas in the same way as with materials. Note that the area must be placed within the computational area (and not adjacent to any boundary, see Figure 4.22). By default, the heat flow is 1 W/m which may be changed in HEAT2. The type *heat flow* may also be changed in HEAT2 to any other type (given temperature, radiation inside cavities, etc.), see Section 6.6. The modification is later changed in HEAT2 from heat flow area to internal pipe with constant effect for the two areas, see Figure 7.6. Note the generated mesh that will approximate the pipes (12-30 cells for each pipe). Figure 7.6 shows an example of isotherms for three pipes with given heat sources. *Note that the thermal conductivity and the heat capacity for the area with an internal modification is 1 by default. Please change these values to correct ones in HEAT2.*

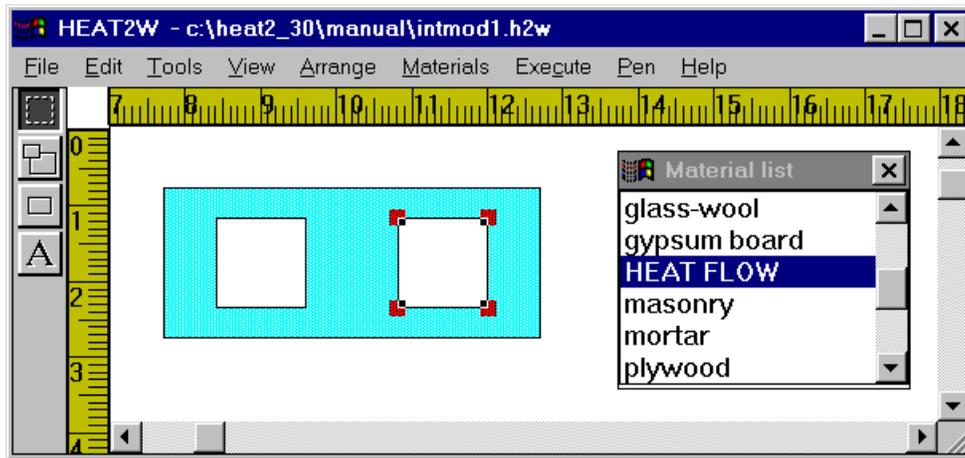


Figure 7.5: Internal modifications may be drawn in HEAT2W by using “Heat flow” in the material pick list.

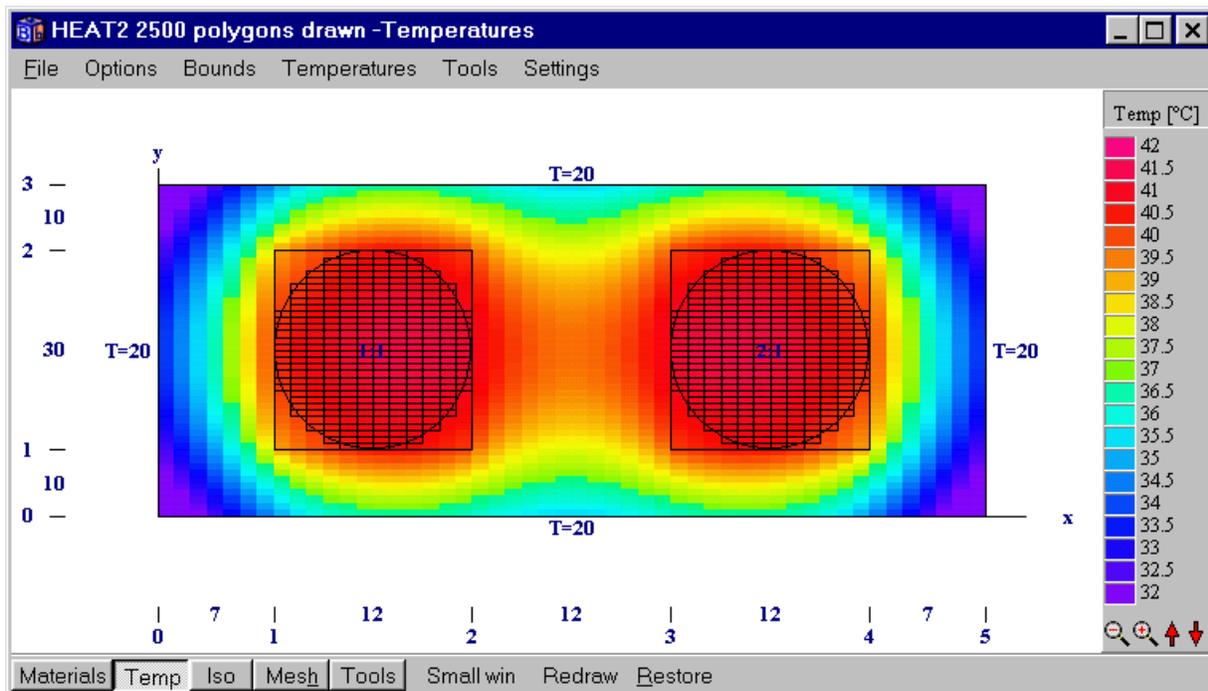


Figure 7.6: The modification is changed from heat flow area to internal pipe with constant effect for the two areas.

7.8 Limitations in HEAT2W version 1.01

The most imported restrictions are as follows. The maximum number of rectangles is 100. The maximum number of materials is 50. The maximum number of boundaries is 250.

Two ‘peculiarities’ may be mentioned. The first one is that when an area is enlarged (*Zoom*), the corners will sometimes not be marked properly. This is due to a bug in the program not yet found. The second one is that decimal numbers may not be entered for an object, only whole numbers may be used. If there is a wall with a wooden stud of 45.5 mm thickness, 45 or 46 mm must be entered (if the scale factor is 1). This value may however be changed in HEAT2 later.

8. Calculations using pipes

8.1 Pipe with a given heat flow

There are two ways to model a pipe with a given heat flow, see below.

8.1.1 Pipe approximated with a square

The pipe may be approximated with a square with the same area as the pipe, i.e. each of the four sides has the length $D = \sqrt{\pi} \cdot r \approx 1.77 \cdot r$, where r is the pipe radius. The effect for the pipe (W/m) will be uniformly distributed within the area. Use the internal modification *heat source* (the heat source may be constant or time-dependent), see Section 6.6.

8.1.2 Pipe approximated with enclosing steps

The modification *pipes (heat source)* may be used which will approximate the boundary of the pipe by small steps. Consider Figure 8.1. A rectangle with an internal modification is put in the input mesh with coordinates 1,1 (lower left) and 2,2 (upper right). A circle (or ellipse if it is not a square with equal side lengths) will be fitted inside the rectangle. The given effect in (W/m) will be uniformly distributed in all the cells that has the center inside the circle, as shown by the numerical cells in Figure 8.1. The number of cells is dependent on the defined numerical mesh. The number of specified cells for the mesh segment is $8 \cdot 8 = 64$ of which 52 will be used to approximate the pipe.

This method will give a more correct solution close to the pipe compared to using approximation with a square as described in the previous section. The draw-back is that many cells are needed for the calculations. The square approximation is sufficient in most cases.

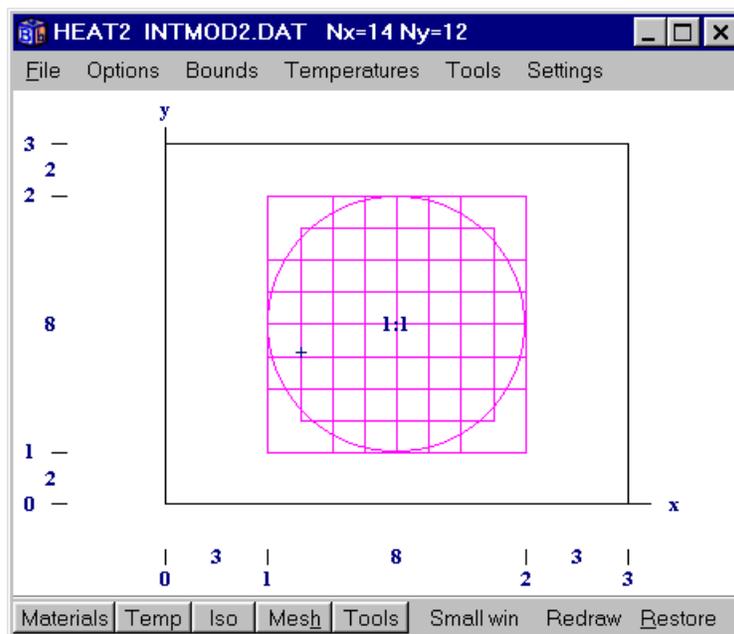


Figure 8.1: The modification *pipes (heat source)* may be used to approximate the boundary of the pipe by small steps depending on the numerical mesh.

8.2 Pipe with a given temperature

This case is more difficult to model. The heat flow from/to the pipe depends on the heat resistance for the pipe (total resistance between the fluid inside the pipe and the material/fluid outside the pipe).

8.2.1 Pipe resistance taken into account

If the pipe is insulated (or if the resistance for the pipe cannot be neglected), the pipe may be approximated by a square [Claesson et al, 1983], see Figure 8.2.

$$D = \sqrt{\pi} \cdot r \approx 1.77 \cdot r \quad (8.1)$$

Calculate a resistance R_c according to Eq. (8.2). The resistance R_c is then put on each of the four sides of the square. This is done in item *internal resistances*, see Section 6.8.

$$R_c = \frac{2 \cdot D \cdot \ln(r / (r - d_i))}{\pi \cdot \lambda_i} \quad (8.2)$$

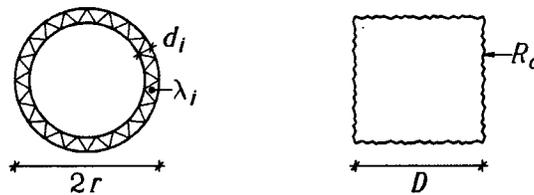


Figure 8.2: An insulated pipe may be approximated by a square and a resistance.

8.2.2 Pipe resistance neglected

If the resistance for the pipe can be neglected, the modification *pipes (given temp)* may be used. This will approximate the boundary of the pipe by small steps depending on the numerical mesh as in Section 8.1.2. The temperature will then be given in all the cells that has the center within the circle.

As an alternative, a square with the same area may be used ($D = \sqrt{\pi} \cdot r \approx 1.77 \cdot r$). This will give a somewhat larger error.

9. Examples

9.1 Example 1 - A wall with a joist

9.1.1 Introduction

Figure 9.1 shows a wall and a joist. We want to calculate the heat loss (W/m) and the lowest surface temperature on the floor. The indoor temperature is 20 °C and the outdoor temperature is 0 °C. The indoor and outdoor surface resistances are 0.13 and 0.04 m²·K/W, respectively. The thermal conductivities (W/m·K) and heat capacities (MJ/(m³·K)) are given in the table.

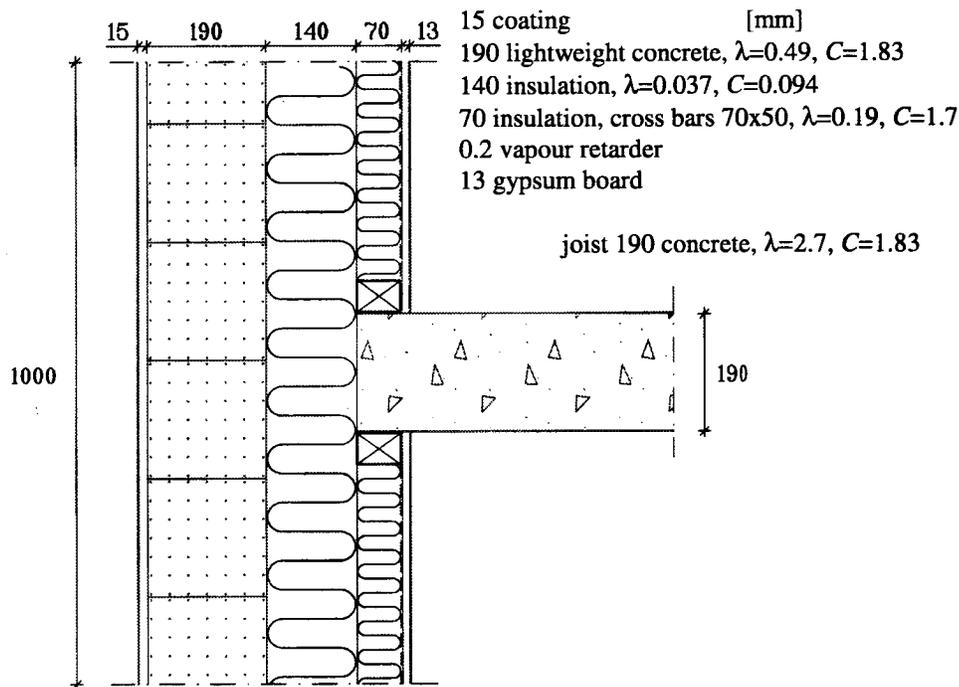


Figure 9.1: A non-supportive wall with a thermal bridge.

9.1.2 Input in HEAT2W

The instructions below show how input can be made using HEAT2W. If you have any problem or do not get the same answer, look at the input file *h2wex.h2w*.

1. Start HEAT2W. Press the <Tab> key. A list of materials is shown. Choose 'concrete lightweight'. Draw a rectangle somewhere in the drawing area.
2. Choose *Select* (the topmost icon on the tool-bar to the left). Click on the rectangle (which now will be marked on the corners). The rectangle may now be dragged by pressing the left mouse button and moving the mouse.
3. Double click (fast) on the rectangle. A window is shown with the width and height (Δx , Δy) of the rectangle. Give (19,100), i.e. 19x100 cm².
4. Press *Rect* on the tool-bar. Press the <Tab> key and choose 'rock-wool'. Draw another rectangle next to the first one. Press *Select*. Double click on the insulation

and give the measures (21,100). The width 21 is calculated as $14+7$. Move the insulation next to the concrete (without overlapping and without a space between the two rectangles).

5. Press *Rect.* Press the <Tab> key and choose 'wood (oak)'. Draw a third rectangle. Give the dimensions (7,5). Place the wooden piece as the top or the bottom one in Figure 9.2 (the vertical location does not have to be exact).
6. Draw and give the dimensions (50,19) for the concrete (pick 'concrete' in the material list) as described above. Place the concrete rectangle next to the wooden piece as in the figure.
7. The second wooden piece is yet to be drawn. The first may be copied as follows. Press *Select*. Hold down the key <Ctrl> and move the first wooden piece.
8. In this example we neglect the small thermal resistance of coating and gypsum boards.
9. Press *Execute* in the menu and then *Scale factor*. Give the value 1000 (100:1000 means that 1 cm on the drawing board is 10 cm in reality). The construction may be saved by pressing *File* and *Save* in the menu system.

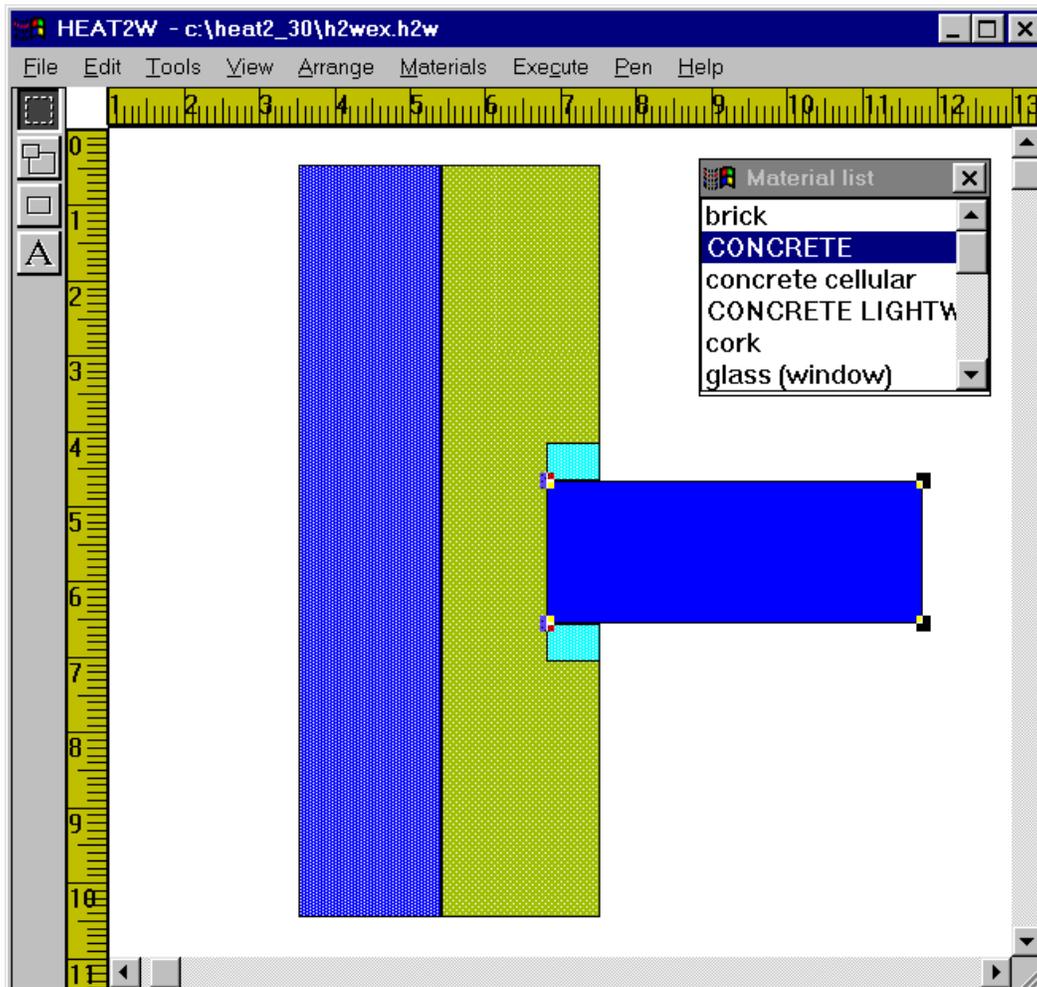


Figure 9.2: The construction drawn in HEAT2W.

9.1.3 Input continued in HEAT2

Go to the program manager in Windows by pressing <Alt> and <Tab> simultaneously. Start HEAT2 by clicking on its icon (if it is not already started). Follow the list below.

1. Load the generated input file *HEAT2W.DAT* in the menu item *File-Load*.
2. Figure 9.3 shows input mesh (check *Options/Input mesh coordinates*), lengths (*Options/Lengths*) numerical mesh (click **Mesh** on the bottom tool bar), boundary numbers (*Bounds/Show bound num*), and thermal conductivity (click **Materials** on the bottom tool bar).
3. The boundary conditions are now to be specified. Figure 9.4 shows the data (menu item *Input/Boundary conditions*). Three boundary conditions are defined. The second type is linked to boundaries 2, 3, 5 and 6, see Figure 9.3. The third type is linked to boundary 8. The other boundaries (1,4 and 7) will have the default (first) type which is adiabatic.

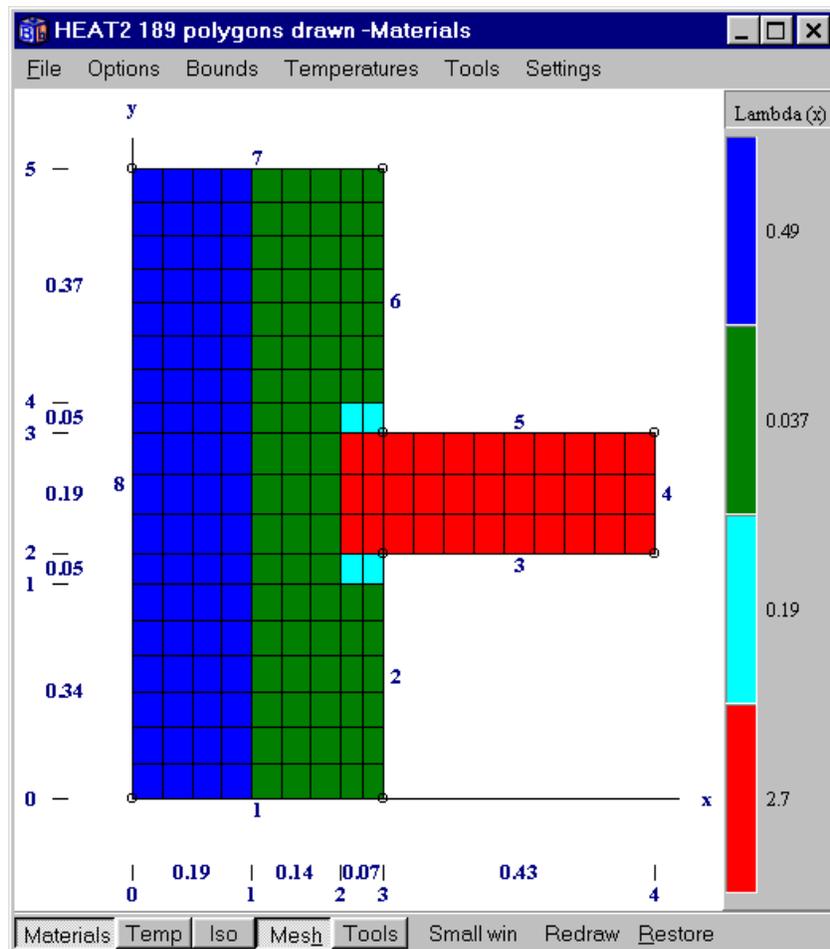


Figure 9.3: Input mesh, lengths, numerical mesh, boundary segments, and thermal conductivity as shown in HEAT2.

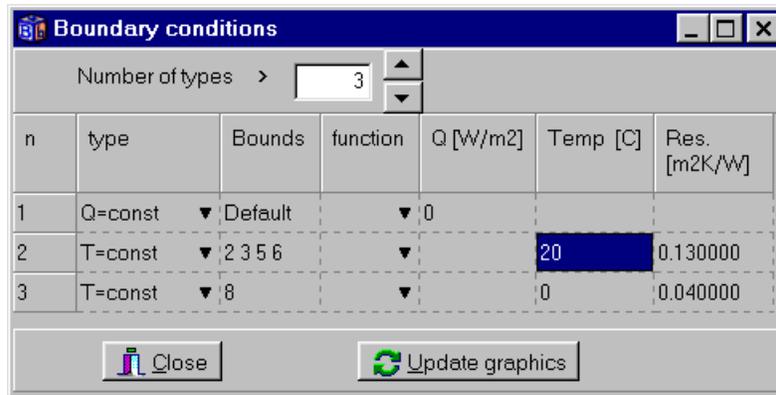


Figure 9.4: Three boundary conditions are defined. The second type is linked to boundaries 2, 3, 5 and 6, see Figure 9.3. The third type is linked to boundary 8. The other boundaries (1,4 and 7) will have the default (first) type which is adiabatic.

Start the steady-state calculation in *Solve-Steady state*. The problem has reached steady-state when the sum of the flows is zero, see Figure 9.5. This will take about a second on a Pentium.

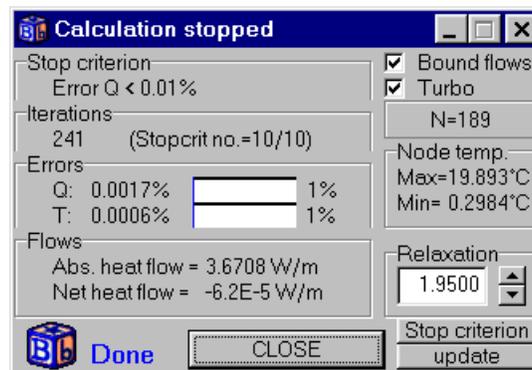


Figure 9.5: Steady-state is reached within a few seconds.

Go to *Output-Boundary Flows*. A list shows the heat flow through the 8 boundary segments, see Figure 9.6. The heat flow for boundary 8 is $Q=-3.67$ W/m (a negative sign means out from the material). The heat flow through segment 6 is larger compared with segment 2 due to a longer boundary. It may be interesting to compare the results with and without the joist. The heat flow is 3.2 W/m through the wall without the joist.

There is a numerical error due to the mesh. The number of numerical cells is here 189. Using 1000 (or even 20000) cells will give a heat flow of 3.69 W/m, i.e. using 189 cells gives in this case a relative error of less than 1%, $(3.69-3.67)/3.69=0.005$. A simulation with 20000 cells will take less than half a minute on a Pentium. It may be a good idea to use an expansive mesh with small cells where the temperature gradients are high, i.e. near the joist.

Figure 9.7 shows the calculated temperature field. To obtain the temperature at a certain point click on **Tools** on the menu bar and move the cursor to the desired location. The temperature and the coordinates are shown in the upper right corner.

What happens if the joist is pushed outwards next to the lightweight concrete? Try this on your own. The heat flow becomes 12.4 W/m, and the lowest floor temperature is $T=16.6$ °C.

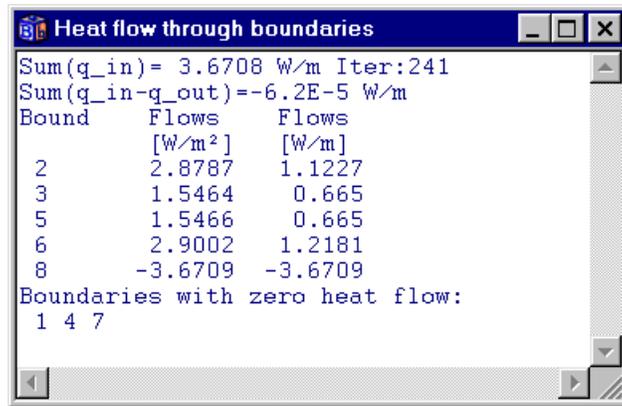


Figure 9.6: Heat flows through the boundaries.

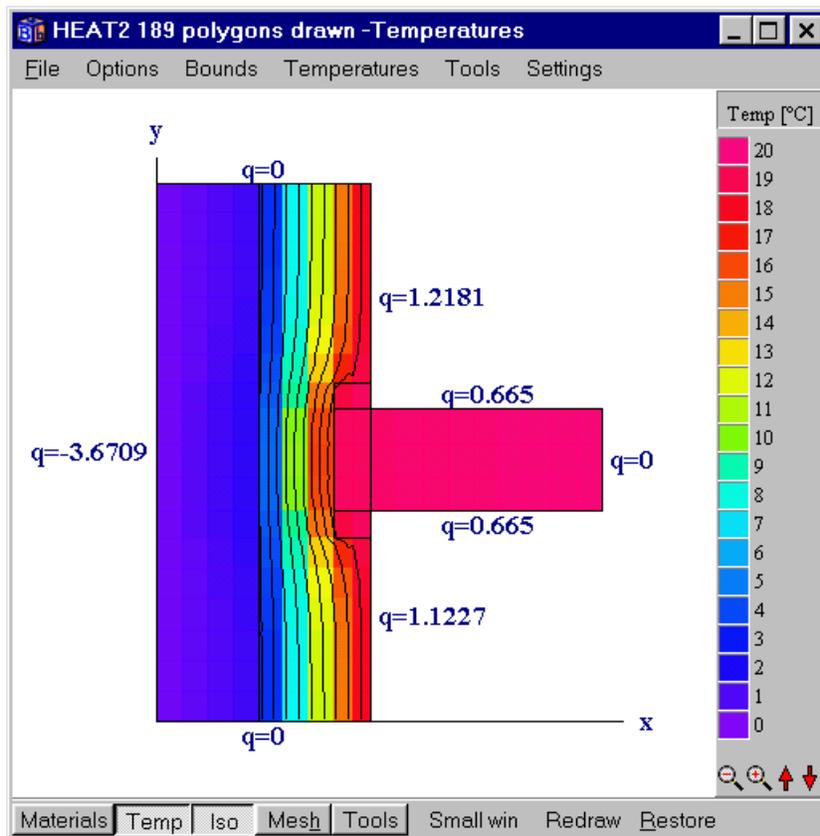


Figure 9.7: Temperature field.

9.1.4 Input using only HEAT2

Note that all the input could have been made using HEAT2 only.

Figure 9.8 shows the input mesh, boundaries, numerical mesh, and materials sorted by name (A, B, C, and D). The input for the data as shown in this figure is as follows:

1. Go to menu item *Input/Input mesh*. Give [4] and [5] number of mesh points in the two directions, respectively. Give [8] boundary segments. See Figure 9.9 (left).
2. Go to *Description of boundaries*. Give the coordinates for the boundary segments, see Figure 9.9 (right).
3. Go to *Mesh in x-direction*. Give the data as in Figure 9.10 (left).
4. Go to *Mesh in y-direction*. Give the data as in Figure 9.10 (right).
5. Go to *Thermal properties*. Give the data as in Figure 9.11.
6. Continue the input from point 3 in Section 9.1.3.

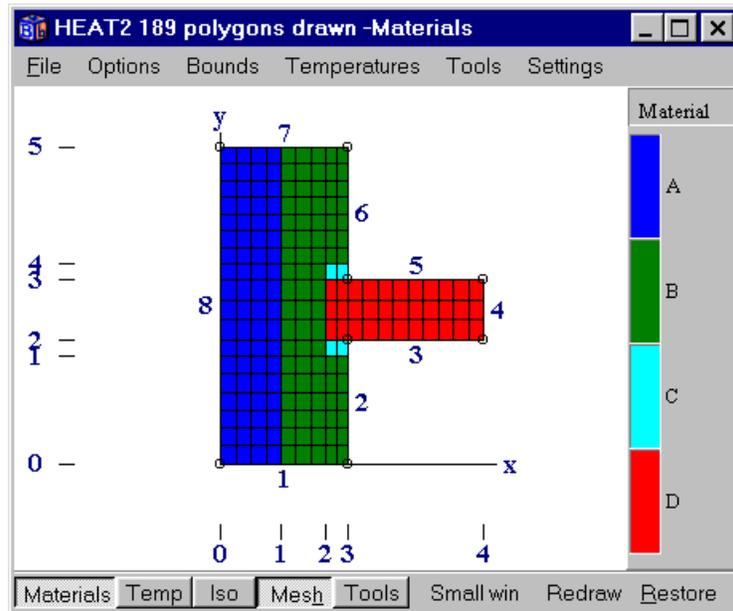


Figure 9.8: Input mesh, boundaries, numerical mesh, and materials sorted by name (A, B, C, and D).

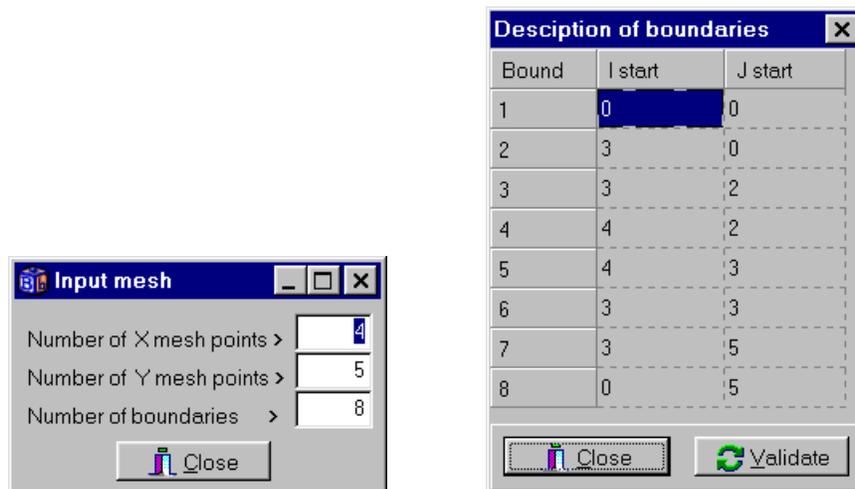


Figure 9.9: Data for input mesh and boundaries.

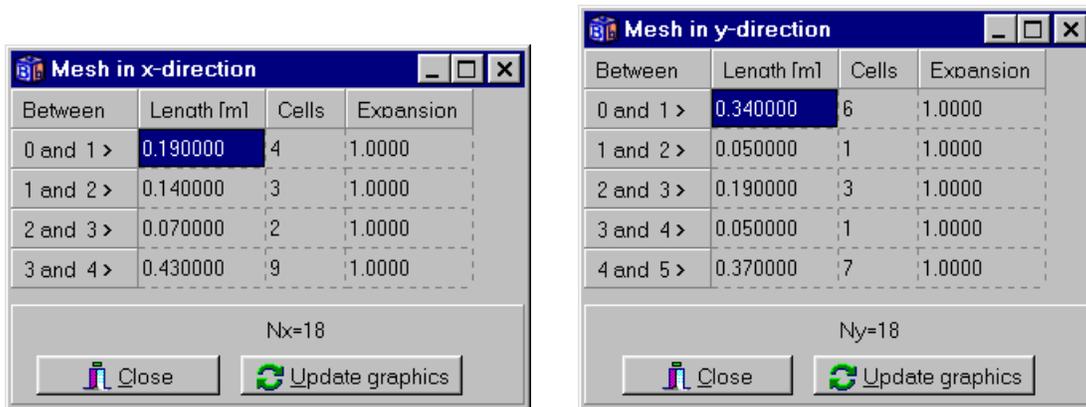


Figure 9.10: Data for lengths and numerical mesh in both directions.

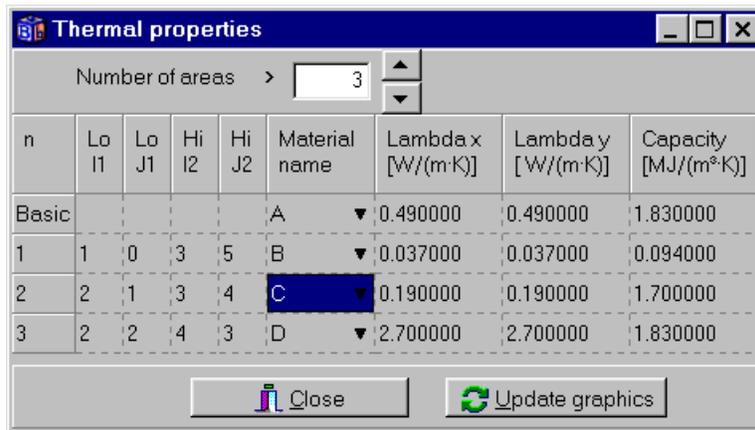


Figure 9.11: Data for thermal properties.

9.2 Example 2 - heat loss from a house with a cellar

9.2.1 Introduction

Figure 9.12 shows the cellar compartment of a house. The basement floor consists of 120 mm concrete with 100 mm underlying mineral wool. The wall is constructed by 300 mm haydite and 100 mm mineral wool. The size of the house is 10·10 m² (internal dimensions). Thermal conductivities are given in the figure. The indoor and outdoor surface resistances are 0.13 and 0.04 m²·K/W, respectively.

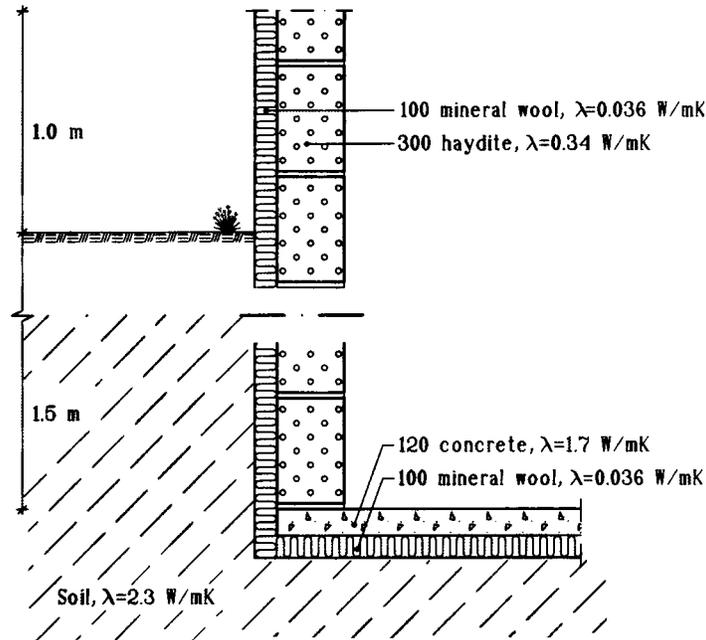


Figure 9.12: The cellar construction.

We are interested to know the heat loss from the house to the ground and the floor temperatures. This is a steady-state case. The indoor temperature is 20 °C and the outdoor temperature is 0 °C. Follow the instructions below. The input procedure is described with a few comments. All input is made using HEAT2. See input file *cellar.dat*.

9.2.2 The input mesh

The axes of a chosen input mesh for the example are shown in Figure 9.13. We have a symmetrical case, thus only half of the house is considered.

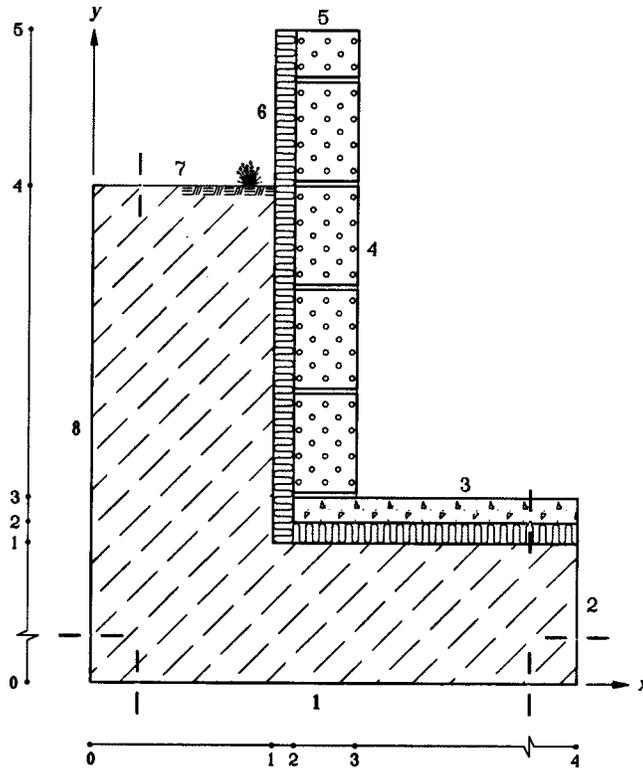


Figure 9.13: An input mesh with enumerated boundary segments.

1. Go to menu *Input/Input mesh*. Give the number of input mesh segments in the x - and y -directions, [4 respectively 5]. Give the number of boundary segments, [8]. See Figure 9.14 (left).
2. Go to *Description of boundaries*. The boundary segments are given counterclockwise by the coordinates in the input mesh. The first boundary starts at $0,0$. Give the data shown in Figure 9.14 (right).
3. The next step is to specify the dimensions, see Figure 9.15. We choose the computational area to extend 20 m vertically downwards and 20 m horizontally outwards from the house. The number of computational cells is now to be specified for each segment. Selecting an appropriate computational mesh needs careful consideration. Too few cells may lead to inaccurate results. On the other hand, using too many increases the computational time. It is obvious that the computational cells should be concentrated toward the house, where larger temperature gradients occur. A computational mesh with cells of decreasing or increasing sizes is called an expansive mesh. Figure 9.16 shows the numerical mesh and lengths of boundaries (check *Bounds/Bound length* in the graphics window). Figure 9.17 shows the expansive numerical mesh around the cellar.

Input mesh

Number of X mesh points >

Number of Y mesh points >

Number of boundaries >

Description of boundaries

Bound	I start	J start
1	0	0
2	4	0
3	4	3
4	3	3
5	3	5
6	1	5
7	1	4
8	0	4

Figure 9.14: Data for input mesh and boundaries.

Mesh in x-direction

Between	Length [m]	Cells	Expansion
0 and 1 >	20.000000	20	0.8000
1 and 2 >	0.100000	2	1.0000
2 and 3 >	0.300000	4	1.0000
3 and 4 >	5.000000	10	1.3000

Nx=36

Mesh in y-direction

Between	Length [m]	Cells	Expansion
0 and 1 >	20.000000	20	0.8000
1 and 2 >	0.100000	2	1.0000
2 and 3 >	0.120000	2	1.0000
3 and 4 >	1.500000	12	1.0000
4 and 5 >	1.000000	6	1.2000

Ny=42

Figure 9.15: Data for lengths and numerical mesh in both directions.

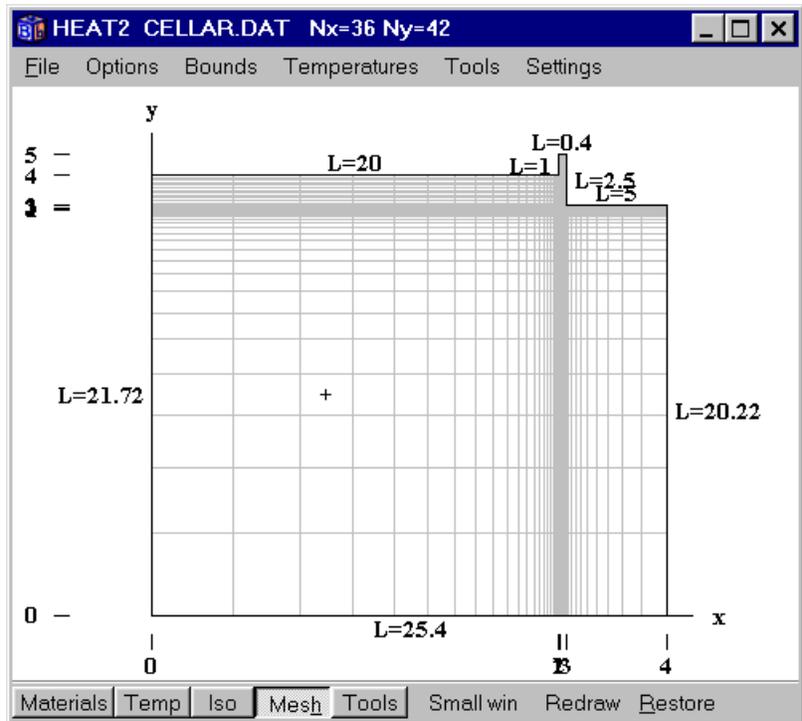


Figure 9.16: Numerical mesh and lengths of boundaries.

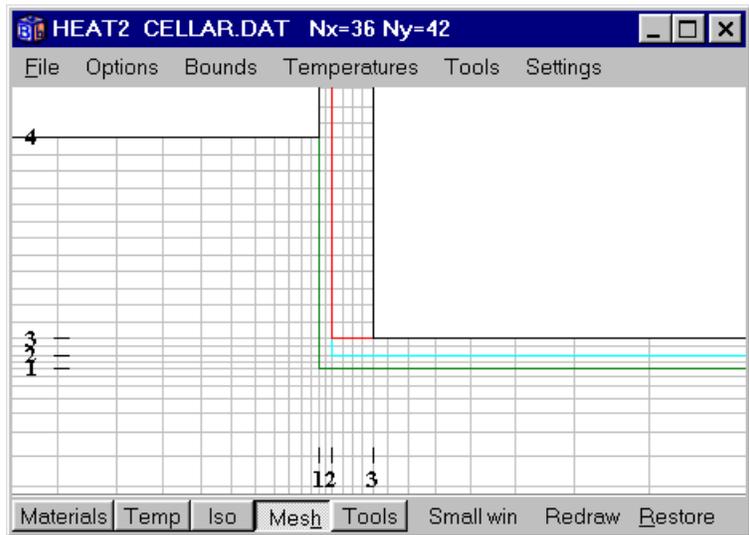


Figure 9.17: Numerical mesh around the cellar.

9.2.3 Material data

Thermal conductivities and volumetric heat capacities are now to be specified for different rectangular areas. Each area is represented by the lower left and upper right coordinates of the input mesh. However, basic data valid for the whole computational area is first given. New input data for a rectangular region replace the old ones. Note that the heat capacity is not used in the steady-state calculation, and may therefore not have to be changed from the default value 1.0. The thermal conductivities in the x - and y -directions are now to be specified. These have the same value in our example since only isotropic materials are used.

Thermal properties								
Number of areas > 3								
n	Lo I1	Lo J1	Hi I2	Hi J2	Material name	Lambda.x [W/(m·K)]	Lambda.y [W/(m·K)]	Capacity [MJ/(m³·K)]
Basic					Soil	2.300000	2.300000	2.300000
1	1	1	4	5	Mineral wool	0.036000	0.036000	0.062000
2	2	2	4	3	Concrete	1.700000	1.700000	1.830000
3	2	3	3	5	Haydite	0.340000	0.340000	1.490000

Figure 9.18: Data for thermal properties.

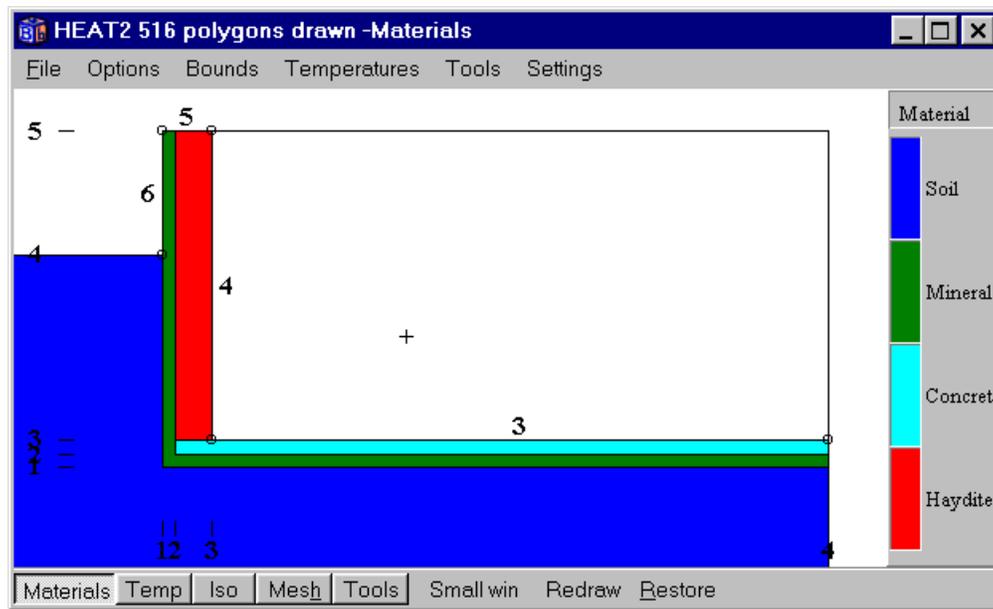


Figure 9.19: Material names.

9.2.4 Boundary conditions

There are three types of boundary conditions:

- Adiabatic surface on boundaries 1, 2, 5 and 8, see Figure 9.13
- $T=20\text{ }^{\circ}\text{C}$ with a thermal surface resistance $0.13\text{ m}^2\cdot\text{K}/\text{W}$ on boundary 3 and 4
- $T=0\text{ }^{\circ}\text{C}$ with a thermal surface resistance $0.04\text{ m}^2\cdot\text{K}/\text{W}$ on boundary 6 and 7

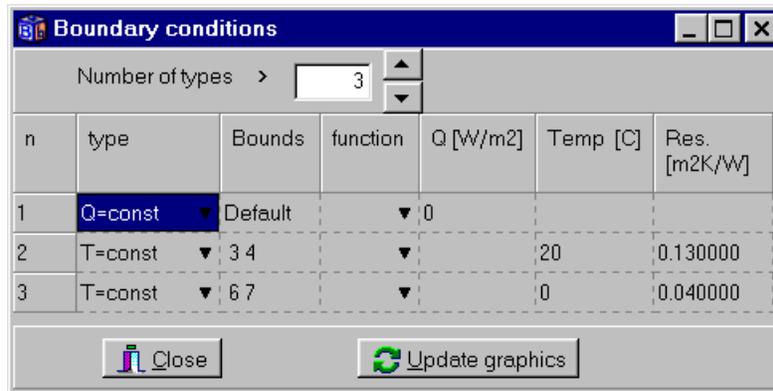


Figure 9.20: Three boundary conditions are defined. The second type is linked to boundaries 3, and 4, see Figure 9.21. The third type is linked to boundary 6 and 7. The other boundaries (1, 2, 5 and 8) will have the default (first) type which is adiabatic.

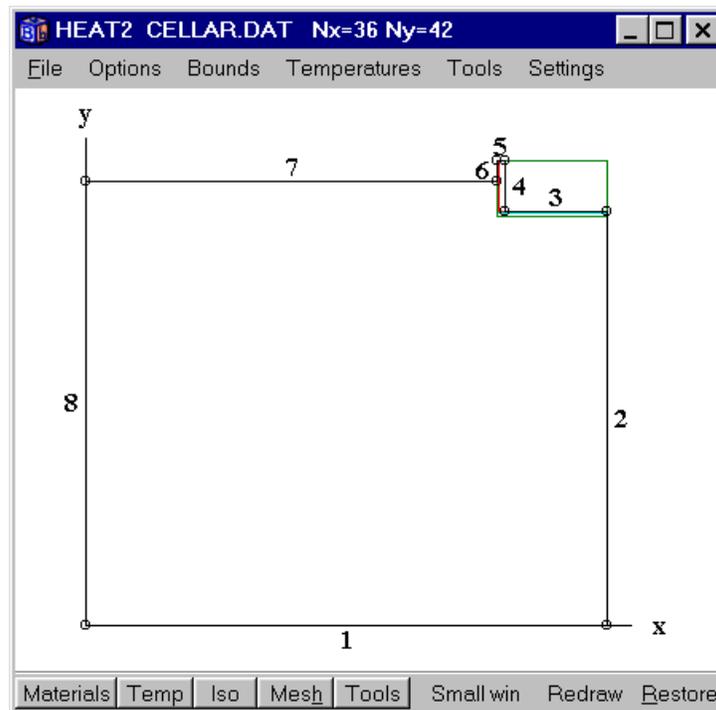


Figure 9.21: The boundary numbers 1..8 are linked to their boundary condition.

9.2.5 The simulation

The steady-state simulation can now be started by pressing F9, see Section 6.9. The simulation takes a few seconds on a Pentium. Go to *Output-Flows* where the boundary flows are shown. The net flow through the boundaries to the computational area is an indication of whether the problem has reached steady-state or not.

The heat flow from half of the house (symmetric case) to the ground is equal to the flow that passes through boundary 7. This is 26.1 W/m, see Figure 9.23. The temperature field, isotherms, and boundary heat flows are shown in Figure 9.24.

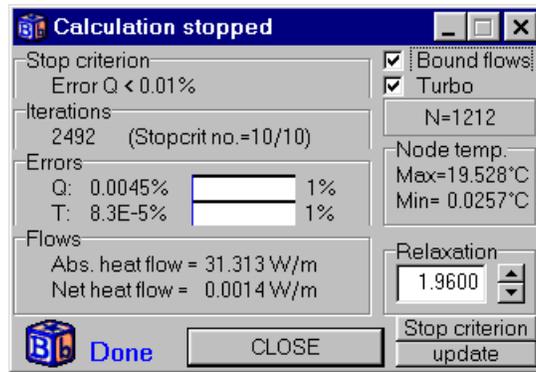


Figure 9.22: The simulation takes a few seconds on a Pentium.

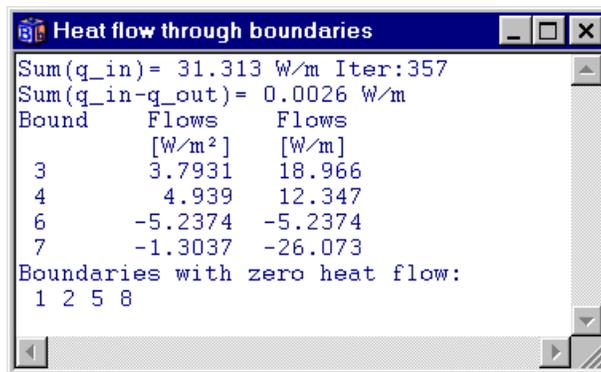


Figure 9.23: Heat flow through boundary segments.

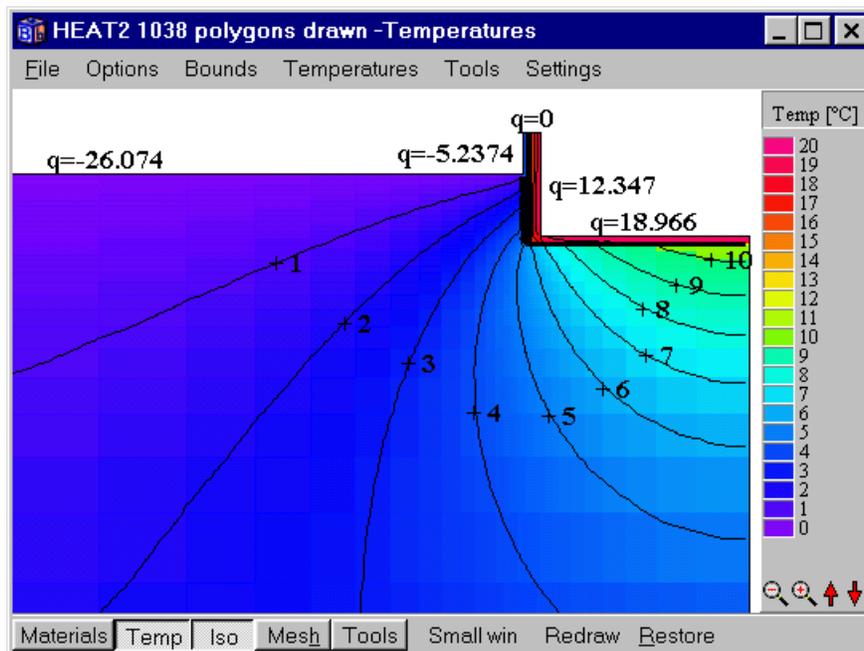


Figure 9.24: Temperature field, isotherms, and boundary heat flows.

9.2.6 Transient simulation

In the following example, a transient calculation using a sinusoidal variation on the external side of the corner will be carried out. First, define a sinus function (item *Functions/Function 1*) with a daily variation between -10 and +20 °C with maximum at noon, see Figure 9.25. Give initial temperatures (average, undisturbed temperature deep in the ground) of +5° C, see Figure 9.26.

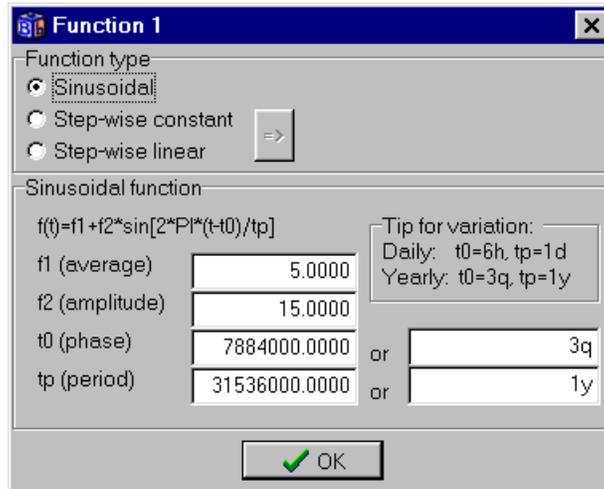


Figure 9.25: A sinus function with a daily variation between -10 and +20 °C with maximum occurring at noon. The time for the phase and period may be given either in seconds or with a time string.

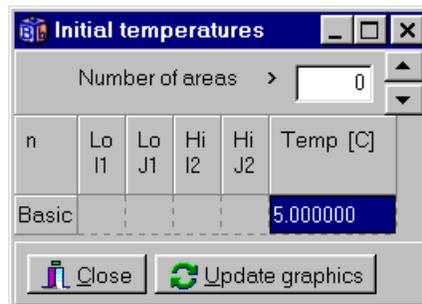


Figure 9.26: Initial temperatures.

Next step is to couple the function to a boundary condition type (*Input/Boundary conditions*), see Figure 9.27. Give the stop time 5 years (*Solve/Options for transient*), see Figure 9.28.

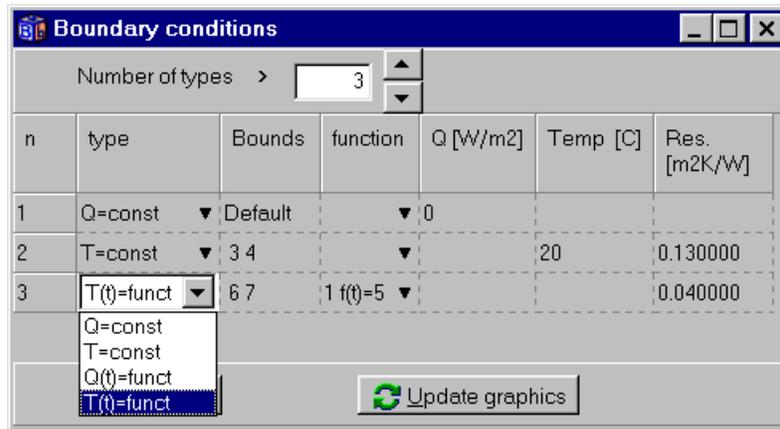


Figure 9.27: A temperature function is chosen in the pick list for the boundary condition type. The function number is selected in the function pick list (number 1 is default).

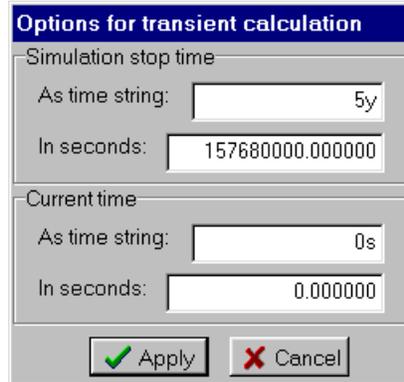


Figure 9.28: The stop time is 5 year.

In this example we will also look at the flows through the internal boundaries during each hour. Enable the record (*Output/Recorder*) and pick the second item in the list as in Figure 9.29 (left). Give the numbers 3 and 4 for the internal boundaries as in Figure 9.29 (right). This will record the sum of the two boundary heat flows at each screen update (*Solve/Update*) as given in Figure 9.30, in this case once every day. Press the item *Graphics* in the record window to bring up a chart window (this will be empty since no data is recorded yet).

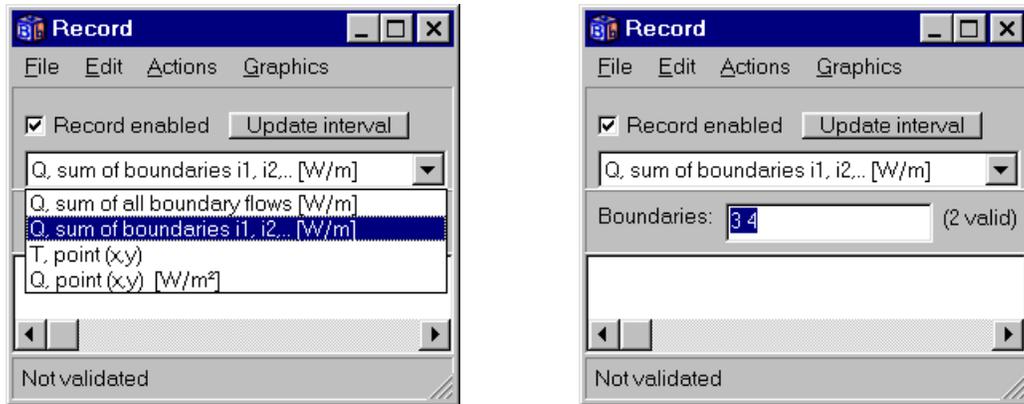


Figure 9.29: Flows and temperature may be saved to a list during the simulation. In this case it is the sum of the heat flows for boundaries 3 and 4 that will be recorded.

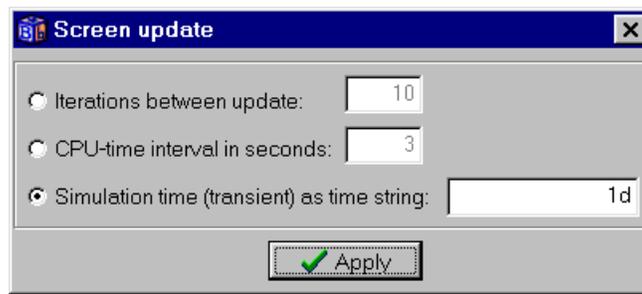


Figure 9.30: The time interval for recorded data is given in item *Solve/Update*).

To start the transient simulation select *Solve/Start transient calculation*. Consider Figure 9.31. A window (top left) will be displayed showing information such as the chosen stop criterion, number of iterations, and error for temperature and flows. The record window (bottom left) and the chart window (bottom right) show the internal heat flows during each day.

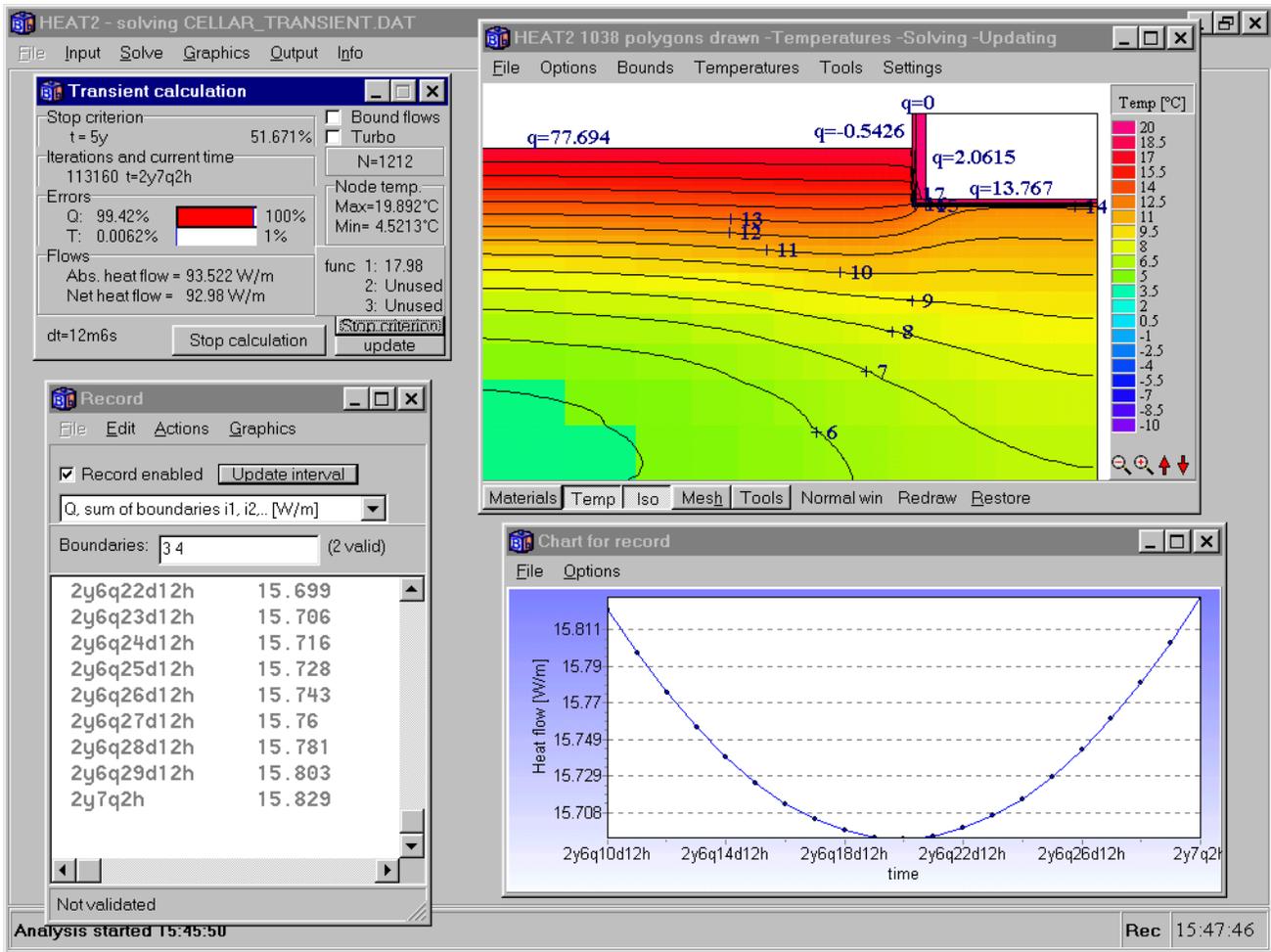


Figure 9.31: A window (top left) will be displayed showing information such as the chosen stop criterion, number of iterations, and error for temperature and flows. The record window (bottom left) and the chart window (bottom right) show the internal heat flows during each day. The minimum heat flow occurs in the end of July (6q20d). See also Figure 9.32. The time to simulate 5 years should be less than 5 minutes on a Pentium (it will take longer if updating of graphics is enabled).

Note that the heat flow through the cellar wall one meter *above* the ground is included. What is the heat flow through the floor only? Change the simulation stop time to 6 years, and remove boundary 4 from the recorder, see Figure 9.29 (right). Continue the simulation for the last year, see Figure 9.33.

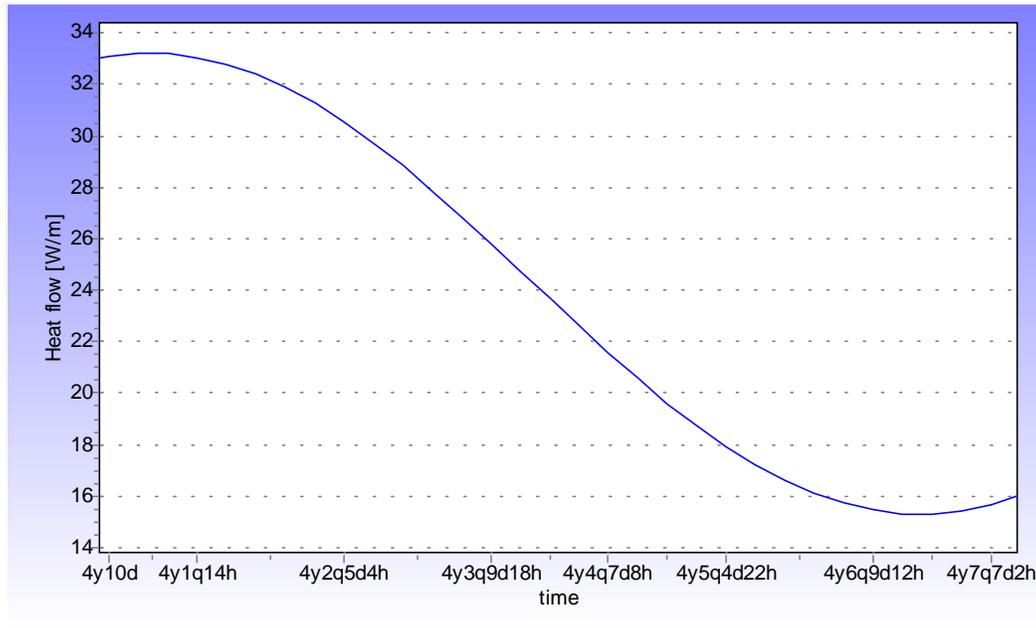


Figure 9.32: The max and min heat flow through the internal boundaries occur in the middle of January "4y20d" (33 W/m) and in the middle of July "4y6q20d" (15 W/m), respectively. Note that the string "4y6q20d" means that 6 months and 20 days has elapsed on the fourth year, i.e. the middle of July. The max and min outdoor temperature occur at $t=0$ (Jan 1) and at $t="6q"$ (July 1), respectively.³

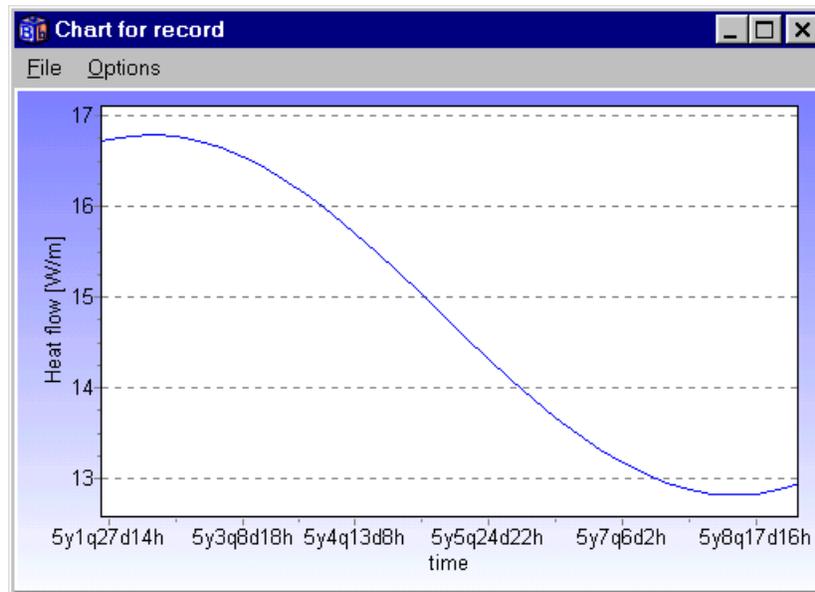


Figure 9.33: The maximum and minimum heat flow through the floor occurs in the middle of March "2q10d" (16.8 W/m) and in the middle of September "8q10d" (12.8 W/m), respectively. Note the time delay between extreme outdoor temperatures and extreme heat flows through the floor of about 2.5 months.

³ The chart in Figure 9.32 was copied to the clipboard in metafile format (EMF). The chart in Figure 9.33 was copied to the clipboard in bitmap format using Alt-Print Screen, see Section 6.13.3.1.

9.2.7 Input using HEAT2W

The program HEAT2W could have been used in this case but it would probably have taken a longer time compared with using only HEAT2. There would also have been a few problems involved. Materials with the given thermal conductivities must be defined since they do not exist. Alternatively, one could use similar materials and later change the thermal conductivities in HEAT2. Another problem is to draw the soil that extends 20 m vertically and horizontally. The best way is probably to just draw small rectangles defining the soil (shown in a darker shade in Figure 9.34) and later change the size for it in HEAT2, see Figure 9.35.

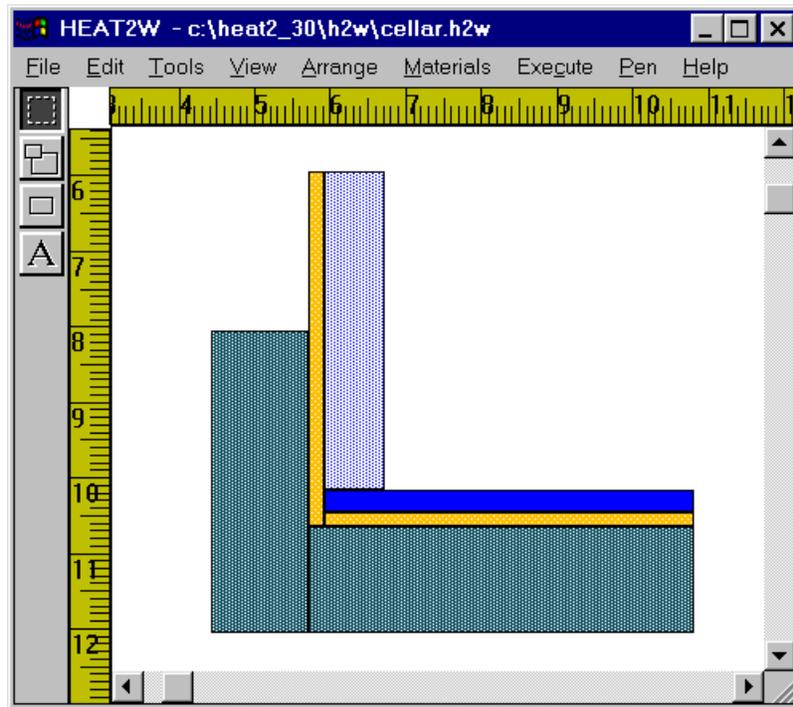


Figure 9.34: The cellar problem drawn in HEAT2W.

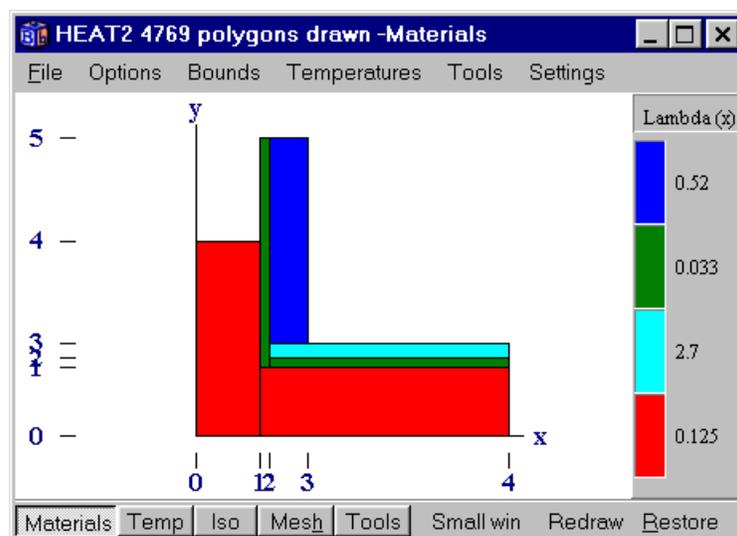


Figure 9.35: Lengths and mesh have to be adjusted in HEAT2.

It may be better to use HEAT2W for complicated geometry, such as that shown in Figure 9.36. Note that the sloping insulation at the edge is approximated by steps using small rectangles. The result is shown in Figure 9.37.

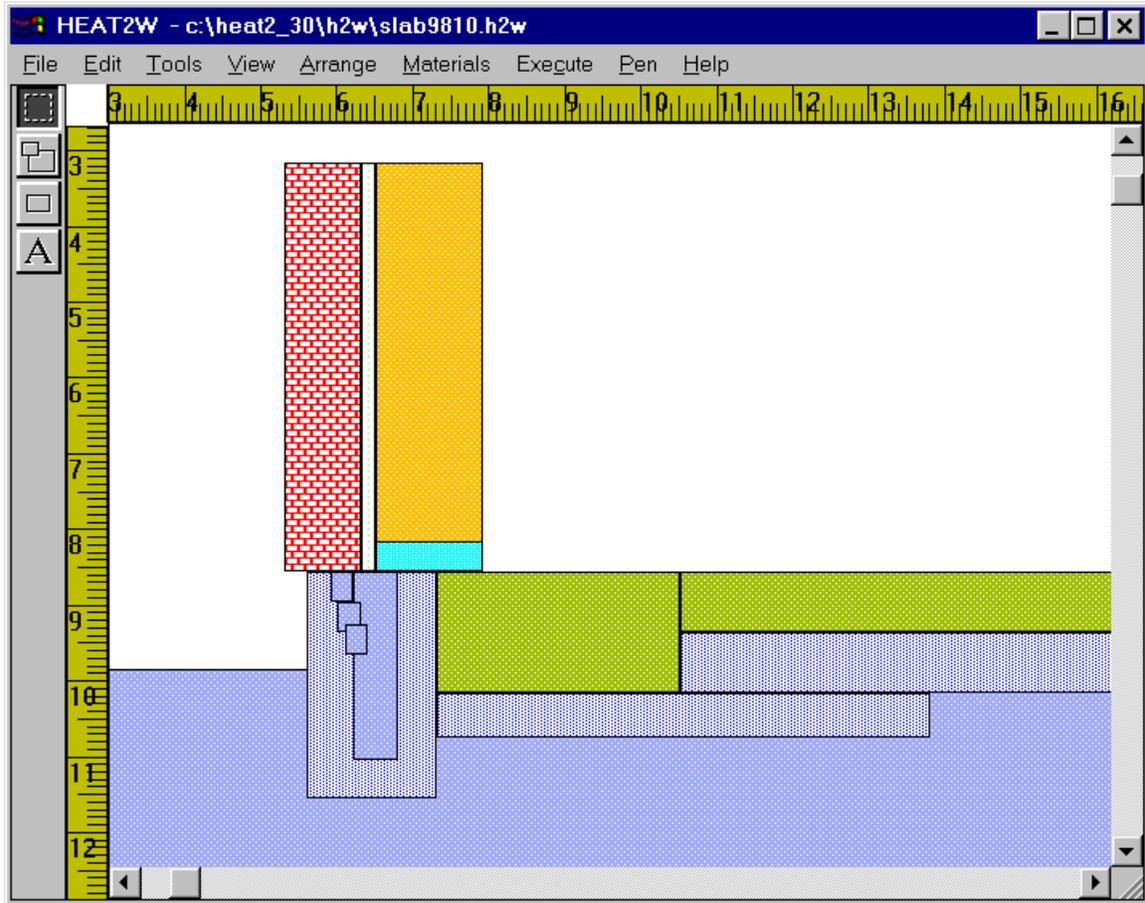


Figure 9.36: A more complicated geometry drawn in HEAT2W. Note that the sloping insulation at the edge is approximated by steps using small rectangles.

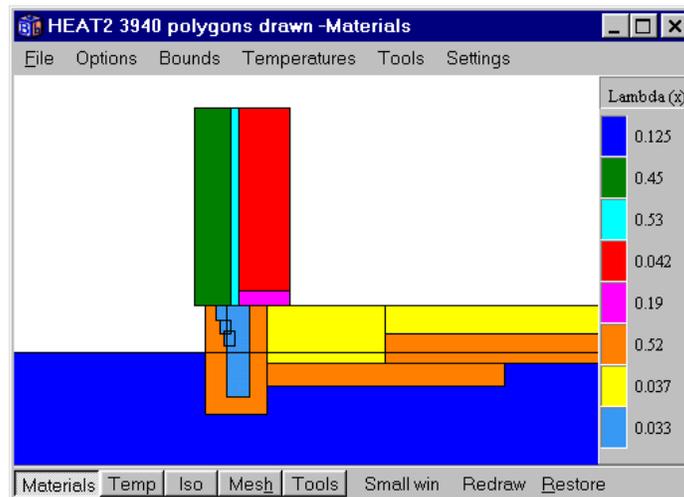


Figure 9.37: Results as shown in HEAT2.

9.3 Example 3 - Heat flow through a wall with metal studs

9.3.1 Introduction

Heat conduction in constructions with both insulation and metal parts often causes particular computational problems because of the wide difference in thermal conductivities.

The following example is a test reference case in the European standards (CEN, 1995) for heat conduction in two dimensions. Figure 9.38 shows a cross-section of the wall. The width and height is 0.5 m and 0.0475 m, respectively. There are four different materials with thermal conductivities between 0.029 W/(m·K), and 230 W/(m·K). It is a difficult numerical problem due to the high ratio of the maximum and minimum thermal conductivity. In this case the ratio becomes almost 8000 ($230/0.029=7900$).

The temperature is 20 °C and the surface resistance is 0.11 m²·K/W at the bottom. The temperature is 0 °C and the surface resistance is 0.06 m²·K/W at the top. The heat flow is zero through the two vertical boundaries due to symmetry.

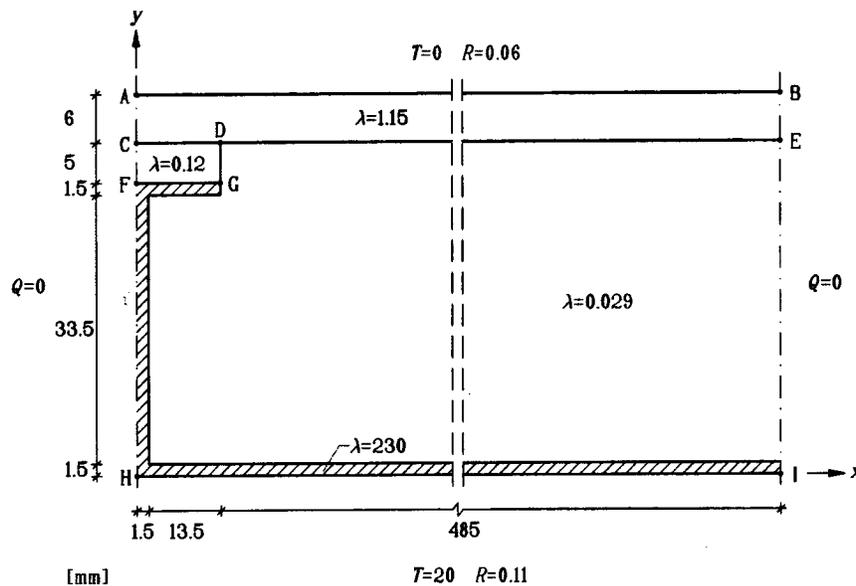


Figure 9.38: Test reference case in the European standards.

9.3.2 Input mesh

Figure 9.39 shows an example of a suitable input mesh. Consider the mesh in the x -direction. An extra mesh coordinate has been inserted between the mesh points 1 and 3. This allows an expansive mesh to be defined with smaller cells concentrated to areas where the temperature gradients are large, i.e. around point G and where the steel bends, see Figure 9.40. In the y -direction, three coordinates have been inserted in the same way (coordinates 2, 5 and 7). It would of course have been possible to use only three segments in the x -direction, and five segments in the y -direction. But in this way we can specify a better mesh with fewer cells.

Directions for how to give input is not given here. Please look at the input file *cenex1.dat* which is also listed in Appendix C.

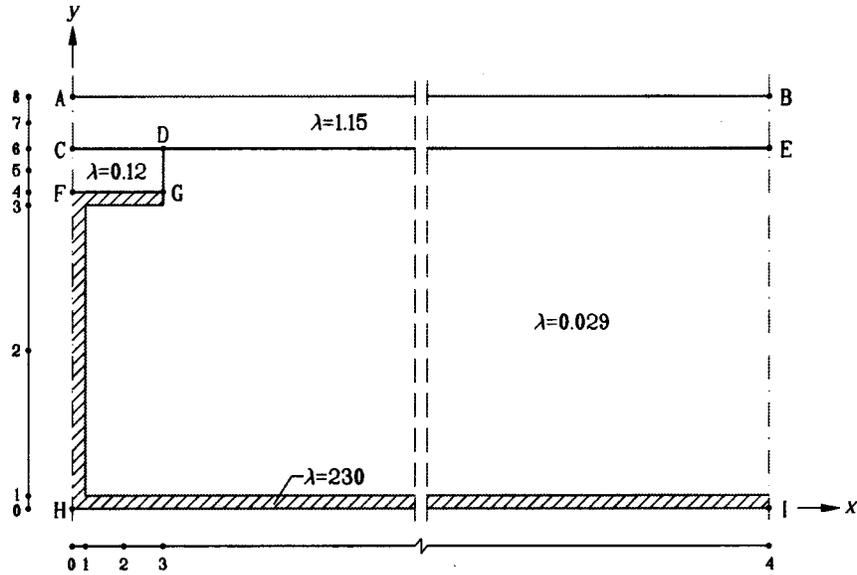


Figure 9.39: An example of example of a suitable input mesh.

9.3.3 Calculations for different meshes

Table 9.1 shows the calculated heat flows through the wall for six different expansive numerical meshes and the required computation time on a Pentium II 450 MHz. The number of cells used in CENEX1.DAT is 1254 (see fifth row). The error ϵ for the heat flow is relative to the last case. It is sufficient with about 1200 cells to have an error of 0.1%. The calculation time in this case is a few seconds (about 5000 iterations). The calculation time in the last case with 56880 cells is relatively long since the smallest cells are about 10^{-5} m.

The last column shows the best choices of the relaxation coefficient ω . The calculation time depends largely on this choice for this example. Consider the case with 1254 cells. Using $\omega=1.995$ gives a calculation time of 2 seconds. Using $\omega=1.95$ (default value) gives a calculation time of about 20 seconds. A good choice of the relaxation coefficient may be made as follows. Look at the difference between the heat flows at the surfaces after a few hundred iterations for some different values of ω . Choose the coefficient that gives the largest difference. Note that the optimum ω normally becomes larger with a denser grid. It tends to be close to 2 for problems with a wide difference in thermal conductivities as this one.

Cells	Q (W/m)	ϵ (%)	CPU-time	Iterations	Optimal ω
32	8.541	11	1 sec.	1900	-
72	9.382	1.2	1 sec.	-	-
374	9.456	0.4	1 sec.	-	-
1254	9.479	0.1	2 sec.	4600	1.995
5016	9.490	0.02	8 sec.	5000	2.0
56880	9.492	-	45 min.	100000	2.0

Table 9.1: Calculated heat flows for different meshes.

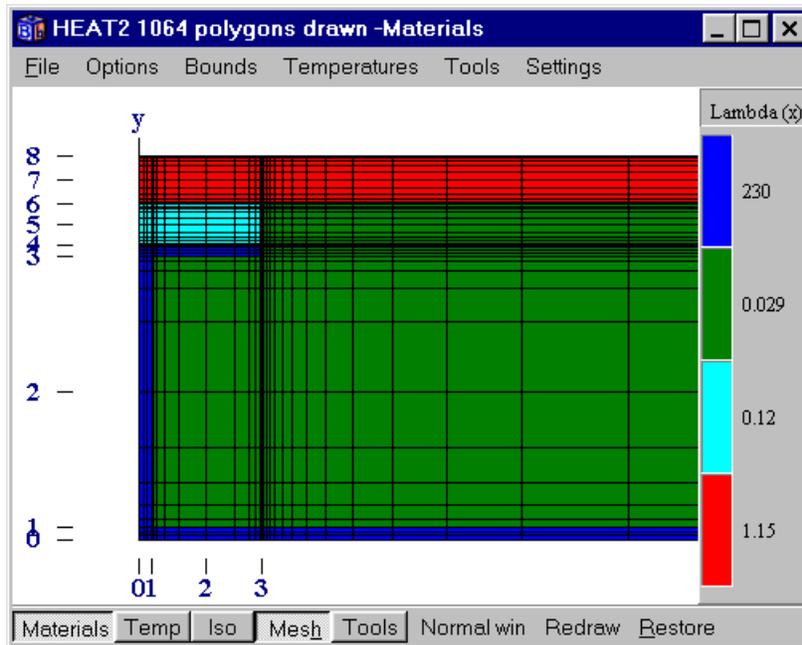


Figure 9.40: Numerical mesh for the left-most part of the problem (1254 cells).

Figure 9.40 shows the numerical mesh for the critical left-hand part of the problem in the case with 1254 computational cells. The remaining 5/6 of the wall contains about 200 of the 1254 cells.

The heat flow through the wall is 9.5 W/m according to (CEN, 1995). The heat flow calculated by the method to be validated, in this case HEAT2, should not differ by more than 0.1 W/m. About 400 numerical cells suffice to fulfill this requirement.

Table 9.2 shows the surface temperatures obtained by HEAT2 for the cases with 1254 and 5016 cells, respectively. According to (CEN, 1995) the difference between the temperatures calculated by the method to be validated and the listed values shown in the last column should not exceed 0.1 °C. All temperatures are the same, except at location *G*, see Figure 9.38. The temperature gradients are large at this location. This makes the interpolation difficult. About 4000 cells are needed to meet the CEN requirement. Figure 9.41 shows the isotherms for the left-hand part of the problem.

Location	(x,y)	<i>N</i> =1254	<i>N</i> =5016	CEN
<i>A</i>	(0, 0.0475)	7.1	7.1	7.1
<i>B</i>	(0.5, 0.0475)	0.8	0.8	0.8
<i>C</i>	(0, 0.0415)	7.9	7.9	7.9
<i>D</i>	(0.015, 0.0415)	6.3	6.3	6.3
<i>E</i>	(0.5, 0.0415)	0.8	0.8	0.8
<i>F</i>	(0, 0.0365)	16.4	16.4	16.4
<i>G</i>	(0.015, 0.0365)	16.1	16.3	16.3
<i>H</i>	(0, 0)	16.8	16.8	16.8
<i>I</i>	(0.5, 0)	18.3	18.3	18.3

Table 9.2: Calculated surface temperatures (°C).

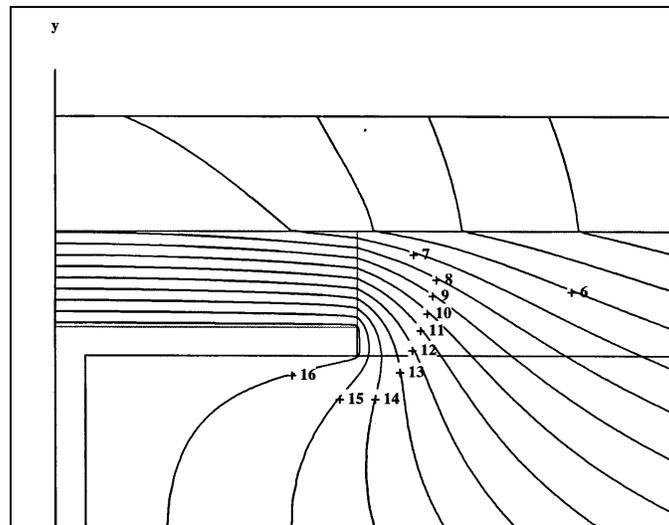
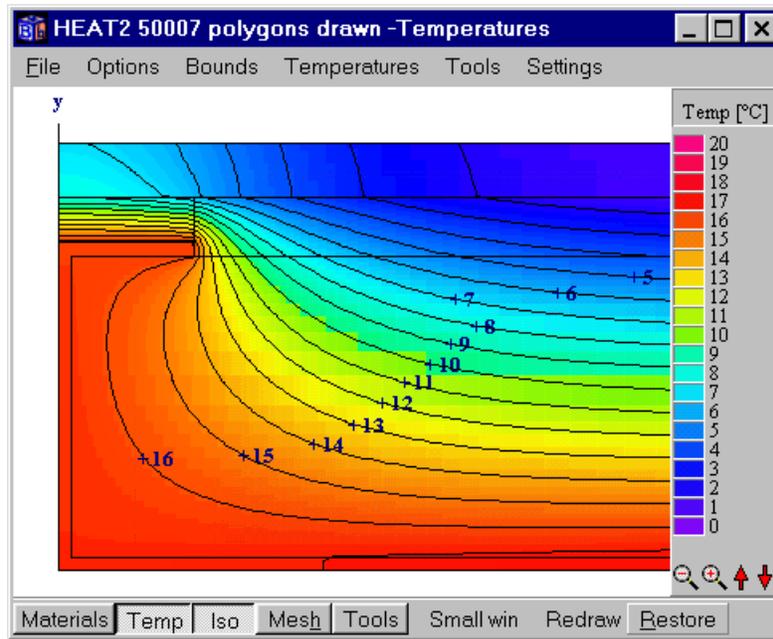


Figure 9.41: Isotherms for the left part of the problem.

9.3.4 Conclusions

This reference case is a relatively difficult numerical problem due to the high ratio of the maximum and minimum thermal conductivity (almost 8000). About 1000 cells are needed to give an error of 0.1% for the heat flows through the wall. This calculation took a couple of seconds on a Pentium II 450 MHz. The rate of convergence for this steady-state problem depends largely of the chosen successive over-relaxation coefficient.

The large temperature gradients, especially at location G, makes the interpolation of temperatures difficult. About 4000 cells is needed using HEAT2 to fulfill the requirement for temperatures in (CEN, 1995). About 300 numerical cells suffice to fulfill the requirement for heat flows.

9.4 Example 4 - Roof section with hollow bricks

9.4.1 Introduction

Radiation inside cavities will here be illustrated by a single application concerning an Italian brick roof. Each large lightweight brick contains 16 cavities in a rectangular pattern. See Figure 9.42. The thermal conductivities of different areas are shown. The steady-state process in the two-dimensional section is to be simulated.

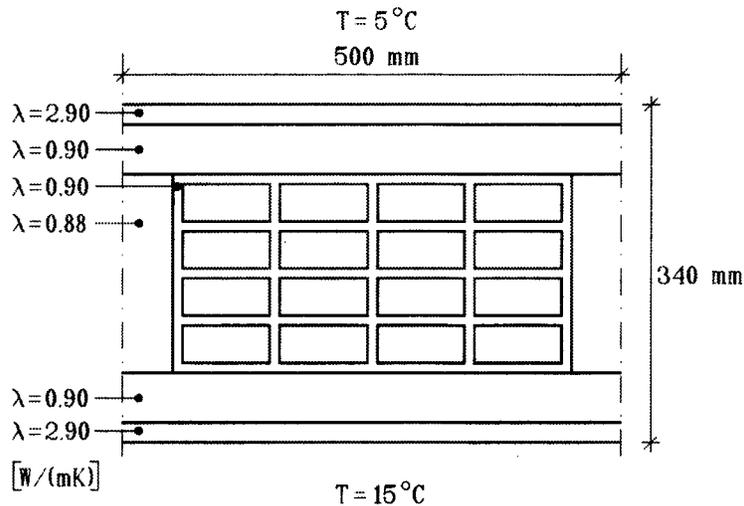


Figure 9.42: Considered section of a brick roof with 16 cavities.

The outside temperature is 5 °C, and the temperature below the roof is 15 °C. The surface resistances inside and outside are both 0.1 m²·K/W. The vertical sides are considered adiabatic.

The cavities are not ventilated ($K_v=0$). The convective heat transfer coefficients α_c are put to 1.0 W/(m²·K) which is the expected order of magnitude from empirical relations (Kreith et al, 1986). The emissivity ε of all surfaces is 0.9. The temperatures are between 5 °C and 15 °C. The temperature level T_{so} is therefore put to 10 °C for all 16 cavities.

The file *radex.dat* shows the input for this case.

9.4.2 Results for different numerical meshes

The process has been simulated for different meshes. Figure 9.43 shows one of the finer meshes. The number of computational cells is here 3860. Each cavity has $N_R=40$ surface elements. Figure 9.44 shows the input data for the cavities.

Table 9.3 shows the calculated heat flow (W/m) through the brick roof section for five computational meshes. A second order of approximation is used, see Section 5.4.1. The results will in these cases be the same if higher orders of approximation are used. The finest mesh with 15440 cells and 80 radiative elements in each cavity is judged to have a very small numerical error. The error relative to the value for this finest mesh is 2.1 % with 1412 cells, and 0.3 % with 3860 cells. Even in the case with 127 cells and 4 radiative elements (smallest mesh possible with one cell in each segment), the error turns out to be relatively small (2.6 %). The last column shows the approximate calculation time on a Pentium II 450 MHz.

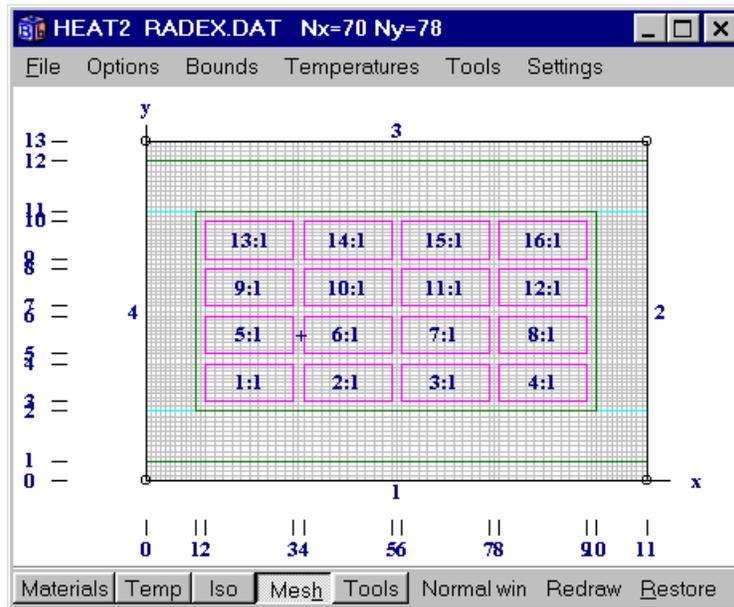


Figure 9.43: The numerical mesh with 3860 computational cells and 16 cavities with radiation ($N_R=40$).

Hole with air - radiation and ventilation		
Gas heat capacity	1.293000	[kJ/(m ³ K)]
Ventilation rate	0.000000	[h ⁻¹]
Ventilation temperature	18.000000	[°C]
Emissivity at lower side	0.900000	[-]
Emissivity at right side	0.900000	[-]
Emissivity at upper side	0.900000	[-]
Emissivity at left side	0.900000	[-]
Conv. coeff. lower side	1.000000	[W/(m ² K)]
Conv. coeff. right side	1.000000	[W/(m ² K)]
Conv. coeff. upper side	1.000000	[W/(m ² K)]
Conv. coeff. left side	1.000000	[W/(m ² K)]
Reference temperature	10.000000	[°C]
Order of approximation (1-5)	2	(this will be used for all areas with radiation)
<input type="checkbox"/> Show matrices in info log		
Close		

Figure 9.44: Input data for the cavities.

Figure 9.45 shows the solve window (left) and boundary heat flows (right). Figure 9.46 shows the calculated isotherms. Figure 9.47 and Figure 9.48 show the air temperatures and heat flows for each of the 16 cavities.

cells	N_R	heat flow (W/m)	Error (%)	Iterations	CPU-time
127	4	5.822	2.6	900	1 sec.
1412	4	5.849	2.1		
1684	8	5.915	1.0		
3860	40	5.960	0.3	1500	23 sec.
15440	80	5.976	-	3300	3.5 min.

Table 9.3: Heat flow, error, iterations and CPU-time for four different meshes.

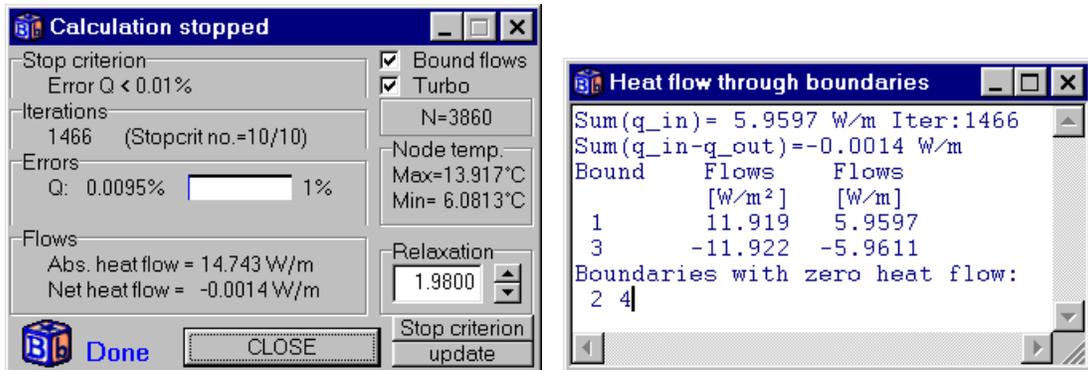


Figure 9.45: The solve window (left) and boundary heat flows (right).

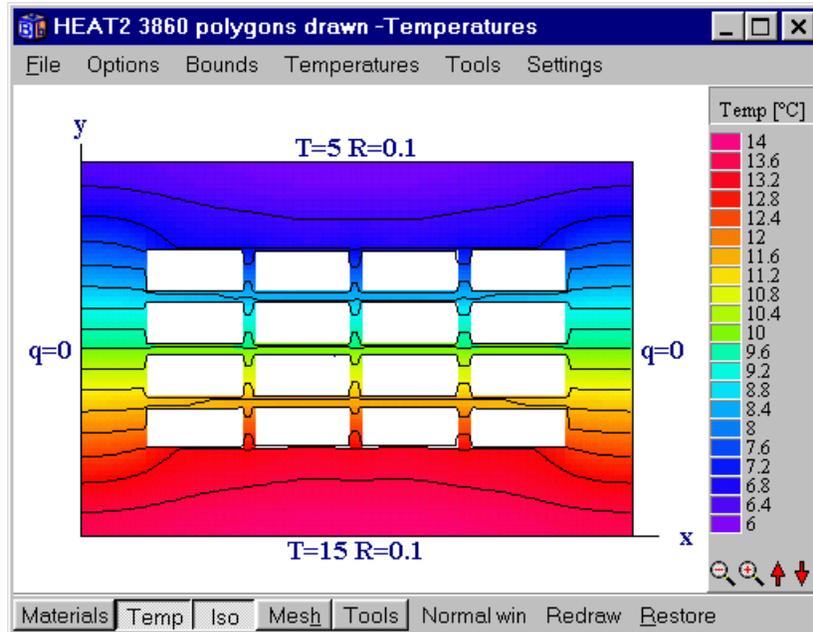


Figure 9.46: Calculated isotherms shown with the interval 0.5 °C.

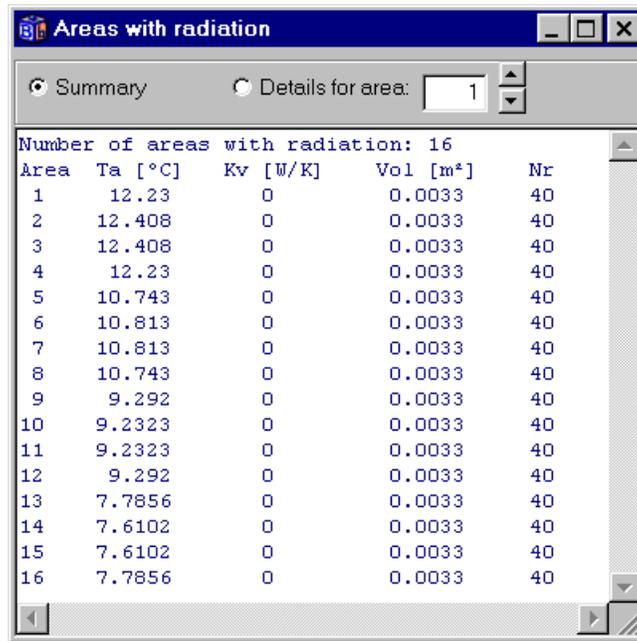


Figure 9.47: Air temperatures in each of the 16 cavities.

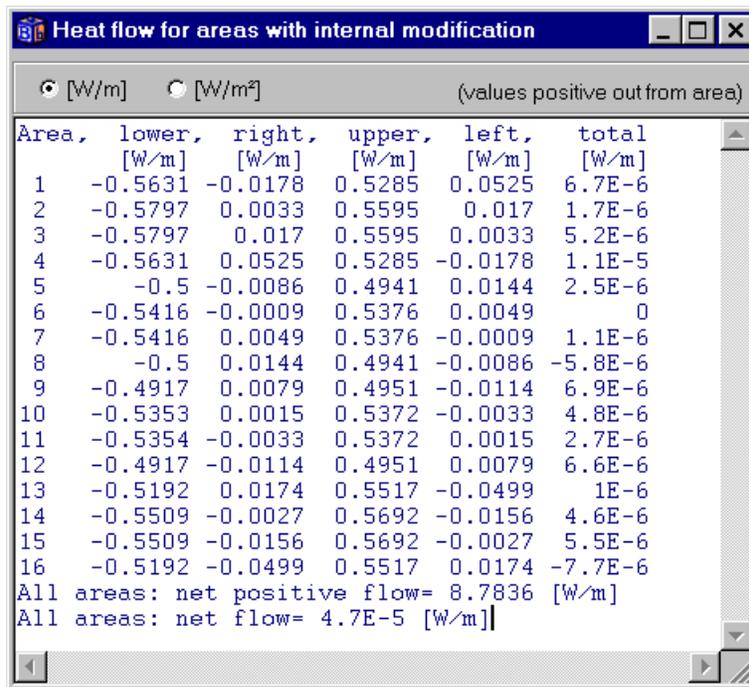


Figure 9.48: Heat flows through the sides of the 16 cavities.

9.4.3 Choice of T_{so} and rate of convergence

The temperature level T_{so} for the linearization procedure may be chosen at will. The choice will influence the rate of convergence in the iteration. Table 9.4 shows the calculated heat flow for different iteration steps for the mesh with 3860 computational cells and $N_R=40$.

	$T_{so} (^\circ\text{C})$					
Iteration step	10	5	15	0	-80	100
1	5.959	5.897	5.898	5.836	5.030	7.192
2	5.960	5.960	5.960	5.960	5.970	5.996
3					5.959	5.961
4					5.960	5.960

Table 9.4: Calculated heat flows (W/m) for the mesh with 3860 computational cells and $N_R=40$.

Table 9.4 shows that the choice of T_{so} is not particularly critical, especially with higher orders of approximation. If T_{so} is put to the same value as one of the prescribed boundary temperatures, 5 °C or 15 °C, the error in the case with linear (first order) approximation turns out to be about one per cent. Even extreme choices such as $T_{so}=-80$ °C or $T_{so}=100$ °C require only three iterations to obtain three digits accuracy.

The calculations take somewhat longer time with higher orders of approximation. For most building physics applications, the linear approximation should be sufficient. Two iterations may be performed to check this in case of doubts.

Note that it would have been sufficient to consider only half the problem since there are symmetry along the *vertical* line in the middle. It is however not symmetry along the *horizontal* line in the middle due to the temperature dependent radiation (higher orders of approximation).

9.4.4 Conclusions

The solution procedure was illustrated by an example concerning a brick roof section with 16 cavities. A calculation with some 1700 nodes for the heat conduction and 8 sub-surfaces in each cavity gave a numerical accuracy on the one per cent level and required only a few seconds to be solved. The linear approximation, or one iteration step only, was shown to be sufficient in the considered case.

The choice of the linearization temperature T_{so} influences the required number of iteration steps. However, it was shown that more than one iteration was necessary only for quite extreme choices.

10. A few tips

10.1 Introduction

There are different kinds of problems involved with the input and simulation of a problem. This chapter gives a few tips that may come in handy.

10.2 Numerical aspects

10.2.1 Numerical mesh

The number of cells required to obtain satisfactory numerical accuracy depends on various parameters such as geometry and materials. The following criterion how to choose a proper mesh is recommended as a European standard (CEN, 1995). The sum of the absolute values of all the heat flows entering through the boundaries is calculated twice, once for n cells and once for $2n$ cells. The relative difference between the flows must be smaller than 2%. Further mesh division is required if the difference is larger.

If a transient calculation takes too long time try to increase the size of the cell that determines the stable time step. The info log (item *Output/Info log*) contains information of input data and generated numerical mesh for the considered problem, see Figure 6.33. The stable time step is shown and the location for the cell that determines it. The stable time step is often caused by small cells, especially those with high thermal conductivities. The worst cases are often problems with large differences in thermal conductivity, such as steel in insulation. It is often sufficient to consider only one cell (or a few cells) for the steel.

10.2.2 Expansive meshes

Expansive meshes may be used to concentrate the cells towards areas with large temperature gradients. Be aware that an expansive mesh may lead to small computational cells causing the time-step to be very short. A badly chosen computational mesh can increase the computational time dramatically, especially for transient analyses. Do not use extreme expansion coefficients with a large number of cells that would cause very small cells in a segment. "Normal" values are in most cases between 0.8-1.25.

10.2.3 Steady-state calculations

The following criterion for when to stop a calculation for steady-state is recommended as a European standard (CEN, 1995). The sum of all heat flows (positive and negative) entering the boundaries, divided by the sum of the absolute values of all these heat flows, must be less than 0.001.

10.2.4 Symmetric cases

Use symmetry whenever possible. A slab on the ground may be one example when it is sufficient to consider half the problem.

10.2.5 Computational area for calculations of heat losses to the ground

The computational area for calculations of heat losses to the ground should involve a part of the ground. A rule of thumb is to use an extension three times the width of the house for a two-dimensional calculation. An adiabatic condition is put on the vertical and horizontal boundaries

at this distance. It may be noted that the horizontal extension from the house is more important than the vertical one. More recipes for slab on the ground calculations are given in (Hagentoft, 1988).

10.2.6 Transient calculations for heat losses to the ground

A transient calculation for heat losses to the ground from a house may require hours or even days to perform. Using less numerical cells gives shorter calculation time but increases the numerical error. It may be a good idea to make steady-state calculations (that probably requires only a few minutes) for different numerical mesh to obtain the numerical errors, see Section 10.2.1. When solving for a transient case using a mesh with relative few cells, one can assume the same numerical error as in the steady-state cases. As an example, if the calculated steady state heat loss for a chosen mesh is 5 % too small compared with a case with more cells, one can add this error to the transient heat losses.

Another problem is the long time scale for the ground. In HEAT2, the initial temperatures will be used when a transient simulation is started. It will probably take 10-15 simulated years until the temperatures reaches semi steady-state (the same temperature variation in the ground year after year). It may be a good idea to first make a steady-state calculation using the mean annual outdoor air temperature, and then start the transient simulation using the calculated temperatures. The process is described in the list below.

1. Give the input data for the problem. Do not forget to give correct heat capacities.
2. Perform a steady-state simulation using the mean annual air temperature as a boundary condition.
3. Change the boundary condition to e.g. a sinusoidal variation of the outdoor temperature.
4. Start the transient calculation. Let the problem be simulated for a few years before using the results (one may check that the problem is semi steady-state by comparing results for different consecutive years).
5. A sudden cold spell (or several ones) may now be simulated by changing the boundary condition for the outdoor temperature (e.g. constant or step-wise constant values).
6. Interesting results may be e.g. maximum heat loss, coldest floor temperature, and estimation of the ground frost depth to avoid frost heave.

10.2.7 Sloping boundaries

Sloping internal and external boundaries may be approximated by steps. It is often relatively easy to do this using the post-processor HEAT2W, see Figure 9.36.

References

Blomberg, T. 1994. *HEAT2R - A PC-program for heat conduction in cylindrical coordinates r and z* . Department of Building Physics, Lund University. P.O. Box 118, S-221 00 Lund, Sweden. CODEN:LUTVDG/(TVBH-7178).

Blomberg, T. 1998. *HEAT3 - A PC-program for transfer in three dimensions. Manual with brief theory and examples. Ver 3.0*. Department of Building Physics, Lund University. P.O. Box 118, S-221 00 Lund, Sweden. CODEN:LUTVDG/(TVBH-7206).

Blomberg, T. 1996. Thesis. *Heat conduction in two and three dimensions. Computer modelling of Building Physics applications*. Department of Building Physics, Lund University. P.O. Box 118, S-221 00 Lund, Sweden. Report TVBH-1008. ISRN LUTVDG/TVBH-96/1008--SE/1-188. ISBN 91-88722-05-8.

Claesson J., C-E Hagentoft. 1983. *Återvinning av värmeförluster från fjärrvärmekulvertar (Regain of heat losses from district heating culverts)*. LUTVDG (TVBH-7075).

CEN. 1995. *Thermal bridges in building construction - Heat flows and surface temperatures - Part1: General calculation methods*. European Committee for Standardization, rue de Stassart 36, B-1050 Brussels, Belgium. Ref. No. EN ISO 10211-1.

Efring, B. 1990. *Numerisk beräkning av temperaturförlopp. (Numerical calculations of thermal processes)*. Byggforskningsrådet. The Swedish Council for Building Research. Report R81:1990.

Hagentoft, C-E. 1988. *Heat loss to the ground from a building*. Department of Building Physics, Lund University. P.O. Box 118, S-221 00 Lund, Sweden. CODEN:LUTVDG/ TVBH-1004.

Hirsch, C. 1992. *Numerical computation of internal and external flows*. John Wiley & Sons Ltd. ISBN 0-471-92385-0.

Holman, J.P. 1986. *Heat transfer*. McGraw-Hill Book Company. New York.

Kreith, F.; M-S. Bohn. *Principles of heat transfer*. Fourth edition, revised edition. Harper & Row, Publishers, New York (1986).

Appendix A. Limits and restrictions of data

HEAT2 has the following limits for the input data:

Number of computational cells	62500 (250-250)
Number of computational cells in the x-direction	250
Number of computational cells in the y-direction	250
Number of input mesh segments in the x-direction	250
Number of input mesh segments in the y-direction	250
Number of areas with different thermal properties	250
Number of areas with different initial temperatures	10
Number of boundaries	250
Number of boundary condition types	20
Number of internal modification types	20
Number of internal modification areas	50
Number of cavities with radiation	20
Number of radiative elements in each cavity	250
Number of functions	3
Number of steps in each step-wise function type	about half a million
Number of resistance lines	50

HEAT2 has the following restrictions for input data (*ims* stands for input mesh segments):

Denotation	Description	Restriction	Unit
I_{\max}	Num. of ims in x-dir	$1 \leq I_{\max} \leq 250$	
J_{\max}	Num. of ims in y-dir	$1 \leq J_{\max} \leq 250$	
ΔX_I	Length of ims in x-dir	>0	m
ΔY_J	Length of ims in y-dir	>0	m
N_I	Number of cells in each ims in x-dir	$1 \leq N_I \leq 250$	
N_J	Number of cells in each ims in y-dir	$1 \leq N_J \leq 250$	
λ_x	Thermal conductivity in x-dir	>0	W/(m·K)
λ_y	Thermal conductivity in y-dir	>0	W/(m·K)
C	Volumetric heat capacity	>0	MJ/(m ³ ·K)
R	Surface resistance	≥ 0	m ² ·K/W
R_x	Internal resistance in x-dir	≥ 0	m ² ·K/W
R_y	Internal resistance in y-dir	≥ 0	m ² ·K/W
T	Temperature	no restriction	°C
q	Heat flow over boundary	no restriction	W/m ²

Appendix B. How to add and edit materials

How to add and edit materials in HEAT2

MTRL30.TXT is a text file used by HEAT2 that contains thermal data for materials, i.e. thermal conductivities and volumetric heat capacities. This file may be modified to your own needs. The file is listed below. Open the file in an editor, e.g. Word or Notepad, to edit it.

There are two groups in *MTRL30.TXT*. The first group contains thermal properties that is taken from *Catalogue of Material Properties, International Energy Agency, report Annex XIV, Volume 3*. Average data is taken for density, specific heat capacity, and thermal conductivity.

The second group has thermal conductivities for window materials from *CEN*. Note that heat capacities are not given here. The data should only be used for steady-state calculations.

Material	C	λ_x	λ_y	(λ_y does not have to be specified if $\lambda_x=\lambda_y$)
IEA Report Annex XIV				
Brick	1.49	.45		
Concrete	1.83	2.7		
Concrete cellular	.53	.52		
Concrete lightweight	1.83	.49		
Cork	.21	.042		
Glass (window)	2.3	1.0		
Glass-wool	.062	.033		
Gypsum board	.88	.1		
Masonry	.88	.1		
Mortar	1.21	.53		
Plywood	1.17	.125		
Polystyrene expanded	.039	.033		
Polystyrene extruded	.059	.024		
Rock-wool	.094	.037		
Steel	3.7	60		
Wood (Oak)	1.7	.19		
Woodwool cement	.64	.071		
CEN (windows, no capacities)				
METALS				
Aluminum alloy	1	160		
Steel	1	50		
Stainless steel	1	17		
WOODS				
Hardwood	1	.2		
Softwood	1	.14		
Plywood	1	.15		
Redwood/cedar	1	.11		
GLAZING MATERIALS				
Float Glass	1	1		
Plexiglass	1	.2		
FRAME MATERIALS				
Glass fibre resin	1	.3		
PVC/Vinyl (rigid)	1	.17		

Wood chip board	1	.13
Porous wood fibre panel	1	.08

SPACER MATERIALS

Butyl rubber (solid)	1	.24
Glass fibre resin	1	.3
Polyester resin	1	.3
Silica gel (desiccant)	1	.13
Silicone Foam	1	.12

THERMAL BREAK MATERIALS

Polyethylene (high dens)	1	.52
Polyvinyl	1	.4
Polyurethane	1	.3
Polyamide	1	.3
Glass fibre	1	.23
Epoxy fibre	1	.23
Acrylic resin	1	.2
PVC	1	.17
FRP Nylon	1	.23

SOLID PLASTICS

Polycarbonate	1	.23
PTFE	1	.24
Polyethylene	1	.4
PVC	1	.2
Polystyrene	1	.17
Acrylic resin	1	.2
Epoxy resin	1	.2

SEALANTS

Silicone	1	.35
Urethane (liquid)	1	.3
Butyl (hot melt)	1	.24
Polyisobutylene	1	.24
Polysulphide	1	.19

WEATHERSTRIPPING

Nylon	1	.25
Neoprene	1	.25
Vinyl (flexible)	1	.12
Foam rubber	1	.06
Mohair	1	.14

How to add and edit materials in HEAT2W

Section 7.4 describes how to edit the materials in HEAT2W. The thermal properties are taken from *Catalogue of Material Properties, International Energy Agency, report Annex XIV, Volume 3* (the same as the ones used in HEAT2).

Appendix C. Input data file example

```
*****
* HEAT2 version 3.01 - Input data file CENEX1.DAT
* Date: 1998-09-17 Time: 14:30:17
* Thomas Blomberg, Building Physics Group, University of Lund
*****
-MEMORY NOTES FOR PROJECT-
Heat flow through a wall with metal studs
```

```
* Following blocks are done:
  Geometrical Meshes
  Thermal Properties
  Functions
  Boundary Conditions
  Internal Modifications
  Internal Resistances
  Initial Temperatures
  Simulation and Printing
```

```
*****
*  GEOMETRICAL MESHES BLOCK  *
*****
```

```
* Number of input mesh intervals in X-direction: 4
* Number of input mesh intervals in Y-direction: 8
* Number of boundaries: 4
* Description of boundaries
bound from I J
  1      :0 :0
  2      :4 :0
  3      :4 :8
  4      :0 :8
```

```
* Lengths, number of computation cells and expanding coefficient
  between the mesh points
* X-direction
      length [m]  comp cells  exp coeff
0 and 1 : 0.0015   : 3       : 1
1 and 2 : 0.0068   : 5       : 2
2 and 3 : 0.0068   : 5       : 0.5
3 and 4 : 0.485    : 20      : 1.4
```

```
* Y-direction
      length [m]  comp cells  exp coeff
0 and 1 : 0.0015   : 3       : 1
1 and 2 : 0.0168   : 5       : 1.6
2 and 3 : 0.0168   : 5       : 0.5
3 and 4 : 0.0015   : 5       : 0.7
4 and 5 : 0.0025   : 5       : 1.4
5 and 6 : 0.0025   : 5       : 0.7
6 and 7 : 0.003      : 5       : 1.3
7 and 8 : 0.003      : 5       : 0.7
```

```
*****
*  THERMAL PROPERTIES BLOCK  *
*****
```

```
* Basic properties for whole area

Lambda(x)      : 230 [W/(m-K)]
Lambda(y)      : 230 [W/(m-K)]
```

Volumetric heat capacity : 1 [MJ/(m³·K)]

none

* Number of partial areas :4

* Area and corresponding thermal properties

* Area lower left upper right Lambda(x) Lambda(y) C

* I J I J [W/(m·K)] [W/(m·K)] [MJ/(m³·K)]

1 :1 :1 :4 :3 :0.029 :0.029 :1 none

2 :3 :3 :4 :6 :0.029 :0.029 :1 none

3 :0 :4 :3 :6 :0.12 :0.12 :1 none

4 :0 :6 :4 :8 :1.15 :1.15 :1 none

* FUNCTIONS BLOCK *

* Combinations

* A Periodic $f(t)=f_1+f_2*\sin(2*\pi*(t-t_0)/t_p)$

* B Stepwise Constant

* C Stepwise Linear

* Number of Functions:3

* Typenumber/Combination

1/ :A f1:1 f2:1 t0:0 tp:86400

2/ :A f1:1 f2:1 t0:0 tp:86400

3/ :A f1:1 f2:1 t0:0 tp:86400

* BOUNDARY CONDITIONS BLOCK *

* Combinations

* A Flow - Constant

* B Temp - Constant

* C Flow - Function

* D Temp - Function

* Number of Types:3

* Typenumber/Combination

1/ :A Flow : 0 [W/m]

2/ :B Temperature : 0 [°C] R : 0.06 [m²·K/W]

3/ :B Temperature : 20 [°C] R : 0.11 [m²·K/W]

* Typenumber/Corresponding Boundaries (list ends with zero)

1/ :2 :4 :0

2/ :3 :0

3/ :1 :0

* INTERNAL MODIFICATIONS BLOCK *

* Combinations

* A Heat source - Constant

* B Heat source - Function

* C Area with given temp - Constant

* D Area with given temp - Function

* E Hole with air (no capacity)

* F Hole with fluid (with capacity)

* G Hole with air (radiation+vent.)

* H Pipe (heat source) - Constant

* I Pipe (heat source) - Function

* J Pipe (given temp) - Constant

* K Pipe (given temp) - Function

* Number of Internal Modifications:1

* Typenumber/Combination

1/ :A Flow : 1 [W/m]

```

* Number of Areas :0

*****
* INTERNAL RESISTANCES BLOCK *
*****

* Number of resistance lines :0

*****
* INITIAL TEMPERATURES BLOCK *
*****

* Basic temperature for whole area : 0 [°C]

* Number of partial areas :0

*****
* SIMULATION/PRINTING BLOCK *
*****

Start time: 0
Stop time: 31536000.000 s <=> 1y
Print start time: 0.000 s
Print Stop time: 31536000.000 s <=> 1y
Time interval: 31536000.000 s <=> 1y
Print Temperatures :Y
Print Boundary Flows :Y
Results To Screen :Y
Results To File :N

& 1.995 (Relaxation)
& 0 (Flow crit)
& 1E-5 (Flow crit.)
& 0.0001 (Temp crit.)
& 100000 (Max iter.)
& 1 (Update)
& 10 (Iter)
& 3 (CPU)
& 3600 (Time)
& 0 (record disabled)
& 4595 (Number of iterations)
& 1 (notify when solved enabled)
& 1 (beep when solved enabled)
& 1 (Order of approximation)

```