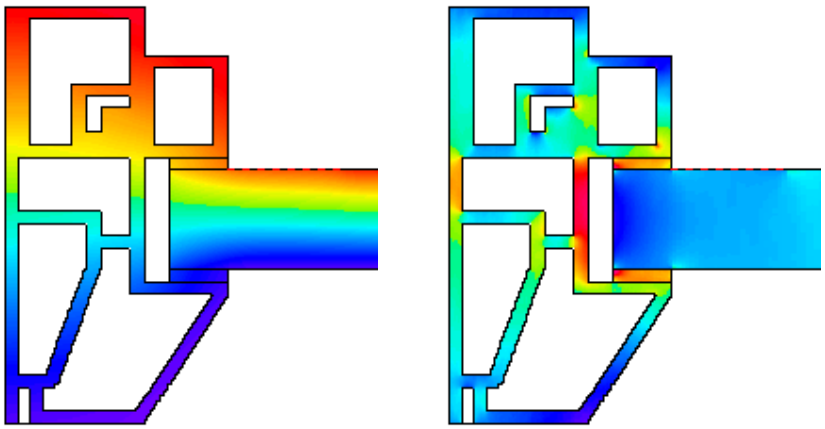
HEAT2

A PC-program for heat transfer in two dimensions.
Manual with brief theory and examples.

Version 5.0



August 1, 2000

Dr. Thomas Blomberg

Lund-Gothenburg Group for Computational Building Physics
Department of Building Physics, Lund University
Building Technology Group, Massachusetts Institute of Technology

Contents

1. Introduction

1.1 How to almost avoid reading this manual

For a quick start read Chapter 4 *Overview of input.* The examples in Chapter 8 would also give a short introduction.

1.2 Introduction to HEAT2

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:

General heat conduction problems

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

Analysis of window frames

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 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:

- An integrated pre-processor facilitates the input procedure, see section 1.2.5.
- Material properties may easily be edited and added. Several material list are available. The default list (Default.mtl) contains about 200 common building materials. The list General.mtl has over 1200 defined materials. Another file with over 200 materials (in German) from the German standard DIN (Deutsches Institut für Normung, DIN V 4108-4) is also available.
- Extensive graphical capabilities: figures showing geometry, materials, numerical mesh, boundary conditions, 2D and 3D temperature and heat flow fields and isotherms may be displayed. Features: zoom, panning, rotation, color/gray-scale, high-resolution printing. Heat flows and temperatures can be recorded and shown during the simulation. Drawing of heat flow magnitude in color or in black and white. Images of heat flows make it simpler to determine thermal bridges and to improve designs by optimizing the insulation at areas with large heat flows. Arrays indicating magnitude and direction of heat flow may be drawn.
- The recorder may save at specified intervals (transient or steady-state) temperatures and heat flows at given points, heat flow through boundary segments, heat flows through internal modification areas, and through internal lines.
- A simple numerical mesh generation option is available. The mesh can easily be changed.

- 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 an air 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). Several other formats with climatic data can be imported.
- 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 makes it e.g. possible to simulate a room or a crawl-space ventilated with an outdoor temperature that varies sinusoidally. The maximum number of areas with internal modifications is 100 (one application is analysis of floor heating with many pipes).
- Conductances and capacities may be written to file.
- Temperature field may be written to file and can easily be imported into other programs such as Matlab.
- Temperatures, heat flows, and numerical cell indices may be presented at arbitrarily chosen coordinates (x, y).
- Extensive window frame analysis has been implemented according to the proposed standard prEN ISO 10077-2:2.2000 draft, **Thermal performance of windows, doors and shutters – Calculation of thermal transmittance – Part 2: Numerical method for frames**.

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 or as the window frame in Figure 1.1, have to be modeled in HEAT2 by step-wise pieces. It is however quite easy to cut and paste multiple step-wise pieces in the pre-processor.

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 10 000 computational nodes is usually less than a minute on a Pentium.

1.2.5 Pre-processor

The integrated pre-processor 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. The geometry and mesh are automatically generated. Boundary conditions are specified before solving the problem. Figure 1.1 shows a construction (window frame section) drawn. Figure 1.2 and Figure 1.3 show calculated temperatures and heat flows, respectively.

The following editing features are available: zoom, fit in window, select, select all of a current material, move, bring to front, send to back, import image file (EMF, WMF, BMP, ICO), import objects from other data files, add reference points.

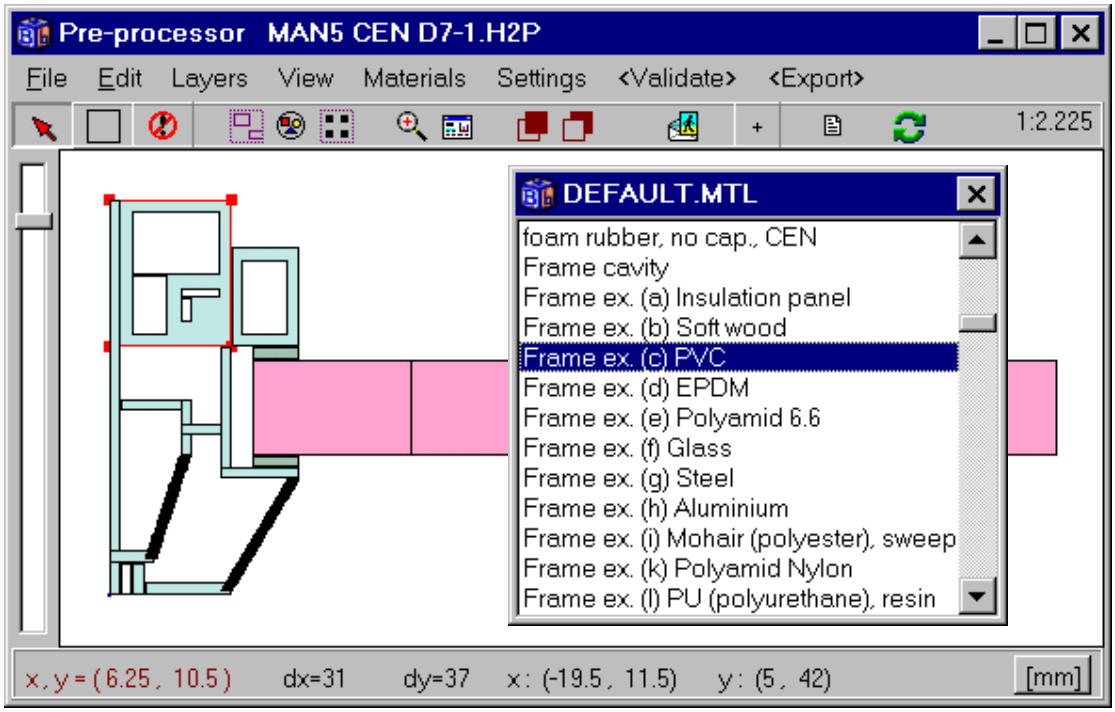


Figure 1.1: A window frame section drawn using the pre-processor.

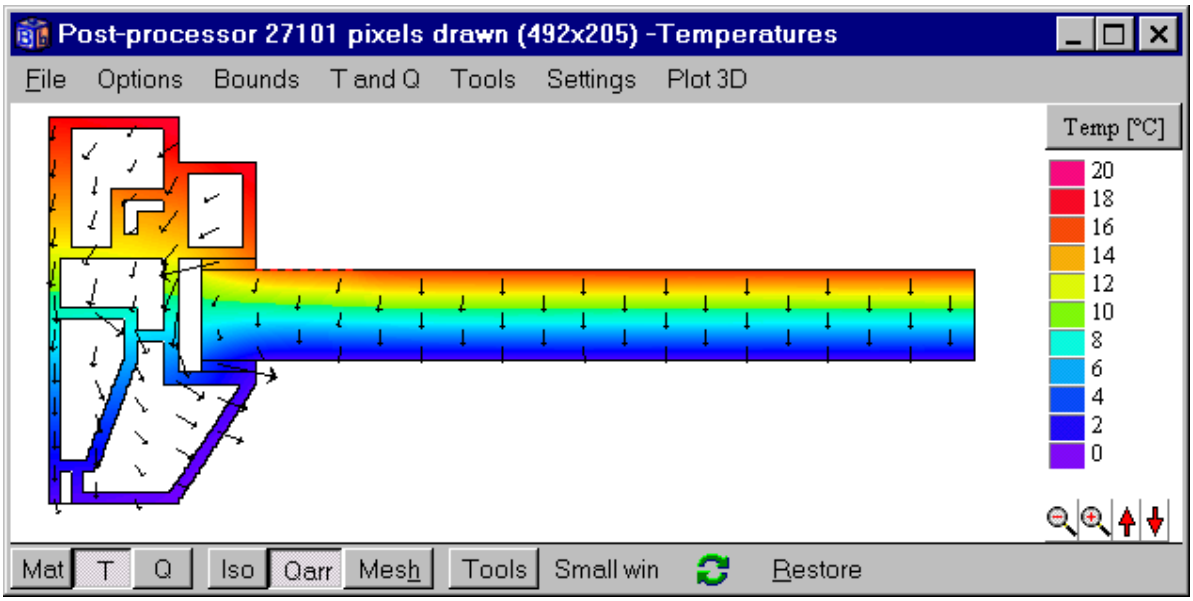


Figure 1.2: Calculated temperature field and heat flow arrays.

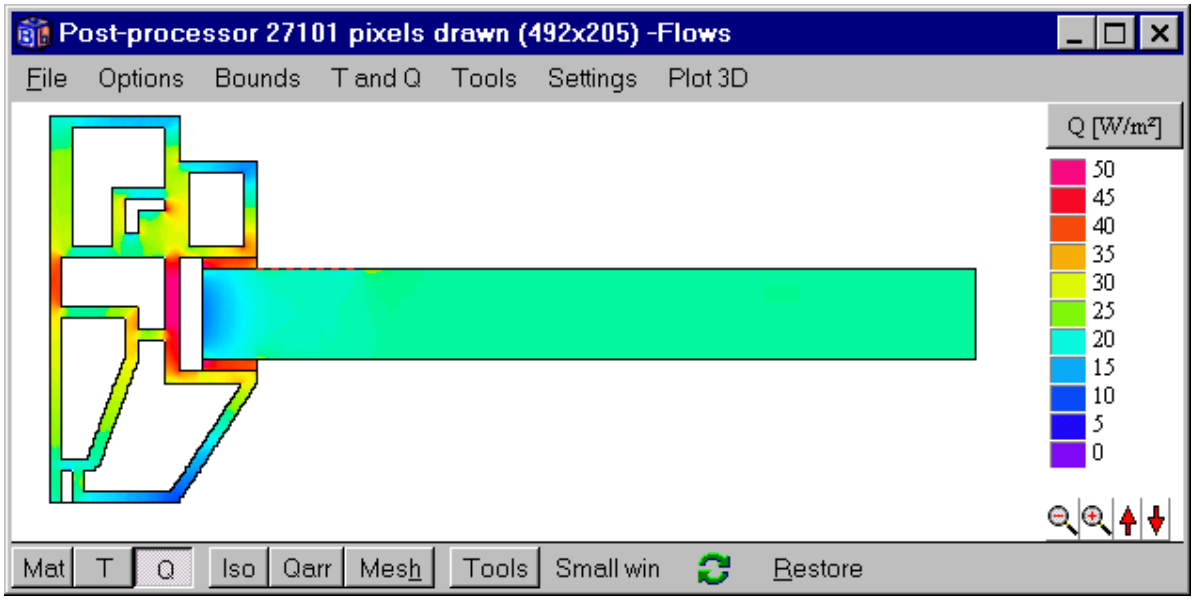


Figure 1.3: Calculated heat flow intensities.

1.3 Update info for HEAT2 version 5.0

HEAT2 version 5.0 reads input files from former versions, but files saved with version 5 should not be used by earlier version. The following list shows the most important new features compared with HEAT2 version 4.

1.3.1 Integrated pre-processor

A new better CAD-like integrated pre-processor has been developed. See Section 4.1.

1.3.2 Material properties

Material properties may easily be edited and added. Over 1200 materials are available. See Section 5.2.

1.3.3 Import of climate files

Various formats with climatic data may be imported, see Section 5.9.3.

1.3.4 Surface plot in 3D

Surface plots of temperatures and heat flows may be drawn, see Section 5.18.9.

1.4 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.

1.5 System requirements

Windows 95/98/NT/2000 is required. A PC with a Pentium processor and 16 MB RAM is recommended.

1.6 Installation

Before using the installation diskette, it is recommended to make a backup copy. To install HEAT2 run the SETUP.EXE program from Windows. HEAT2 version 5 will be installed in directory *HEAT2_50* by default. The following steps describe this process in detail:

1. Start the setup program (SETUP.EXE).
2. Follow the instructions on the screen.

The following files will be installed:

- HEAT2.EXE is the main program.
- Files with extension MTL are files with materials.
- MTRL50.TXT is a file with thermal properties that may be edited.
- Files with extension DAT are HEAT2 input data files.
- Files with extension H2P are files created by the pre-processor.

1.7 Technical support

News is 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

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

Here *I*, (W/m³), is the rate of internal heat generation. The thermal conductivities in the *x*,*y*-directions are denoted by λ_x and λ_y, (W/(m·K)), respectively. The volumetric heat capacity is denoted by *C*, (J/(m³·K)), which is the density ρ, (kg/m³), times the specific heat capacity c_p, (J/(kg·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*, (m²·K/W):

$$T_{b.s.}(t)-T\big|_{surf}=R\cdot(-\lambda)\frac{\partial T}{\partial n}\bigg|_{surf} \qquad (°C) \qquad (2.2)$$

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

$$(-\lambda)\frac{\partial T}{\partial n}\bigg|_{b.s.}=f(t) \qquad (W/m^2) \qquad (2.3)$$

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 $\hat{n}$. The heat flow across the boundary is continuous. The condition of continuous heat flow perpendicular to the boundary is

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

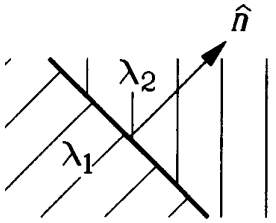


Figure 2.1: The normal $\hat{n}$ at an internal boundary.

There may be a contact resistance R_{ins}, (m²·K/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:

$$\lambda_2 \frac{\partial T}{\partial n}\bigg|_2 = \frac{T\big|_2 - T\big|_1}{R_{ins}} = \lambda_1 \frac{\partial T}{\partial n}\bigg|_1 \tag{2.5}$$

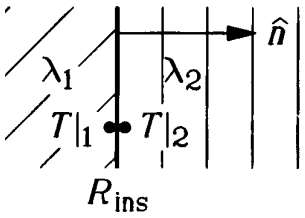


Figure 2.2: Case involving internal insulation.

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}, (J/(m³·K)). The inward normal for the region is denoted by $\hat{n}$, and the volume of the fluid by V_{fluid}, (m³). 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:

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

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

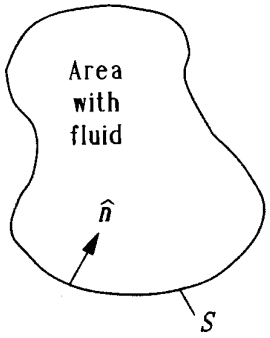


Figure 2.3: A fluid region enclosed by the boundary surface *S*.

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 non-linearity due to the forth-power radiation law, are formulated to suit computer modeling.

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 $T_{i,j}$.

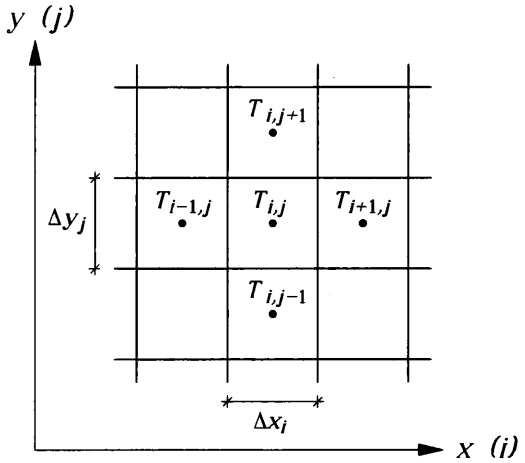


Figure 3.1: Choice of indices for the cells in the computational mesh.

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.

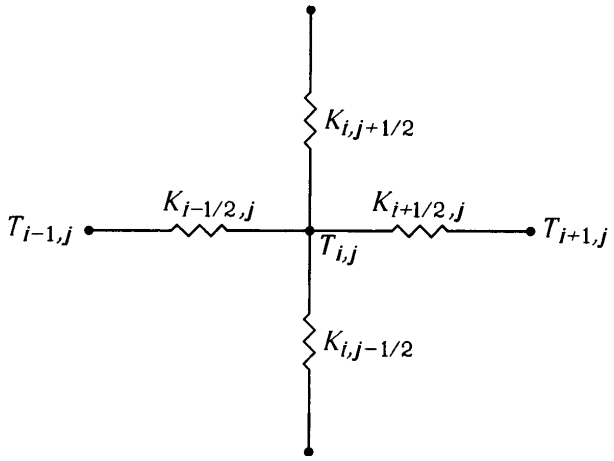


Figure 3.2: The thermal conductances between cell (*i,j*) and neighboring cells.

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

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

Here $\lambda_{xi,j}$ 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 $R_{i-1/2,j}$, (m²·K/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

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

Here $R_{1/2,j}$, (m²·K/W), is the boundary surface resistance.

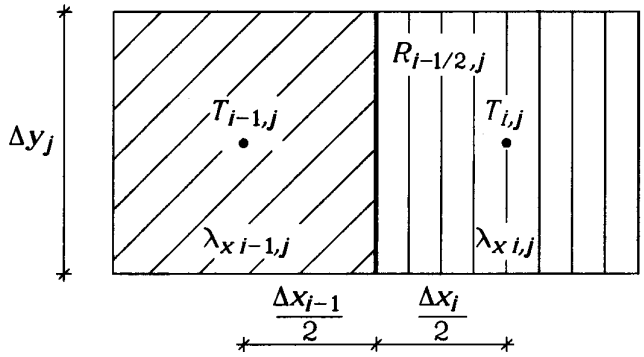


Figure 3.3: Thermal resistance between cell (*i-1,j*) and cell (*i,j*).

3.4 Heat flows

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

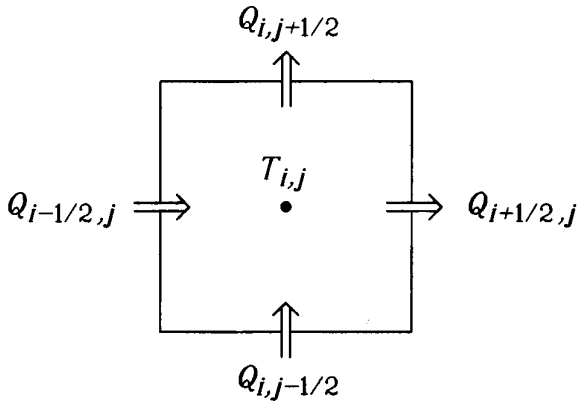


Figure 3.4: The four heat flows of cell (*i*,*j*).

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

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

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

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

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

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

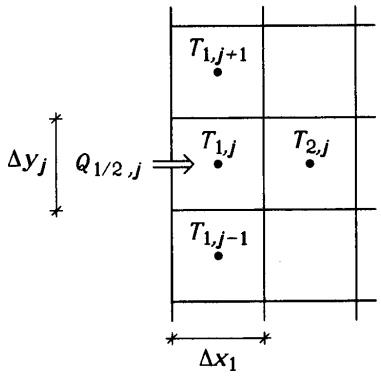


Figure 3.5: Heat flow $Q_{1/2,j}$ through the outer boundary of boundary cell (*1*,*j*).

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

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

The accumulated heat flow during a time-step Δt for the left boundary segment to which cell (*1,j*) belongs is

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

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.

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

The final equation for the new temperature becomes

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

3.6 Choice of time-step,

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

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

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 (Eftring, 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 (ω=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

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

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

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

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 modeling 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:

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

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:

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

4. Overview of input

An overview of how to input and solve a simple problem is shown below.

4.1 Pre-processor

Figure 4.1 shows two rectangles drawn in the pre-processor. The bottom row shows the mouse pointer coordinates (*x*,*y*) in red. The width (dx=0.8) and the height (dy=0.3) are shown for the selected object, in this case the upper right blue rectangle. The extensions in both directions are also shown (the rectangle starts at *x*=0.5 and ends at *x*=1.3). In this case a *reference point* is placed at the lower left side of the left-most (red) rectangle.

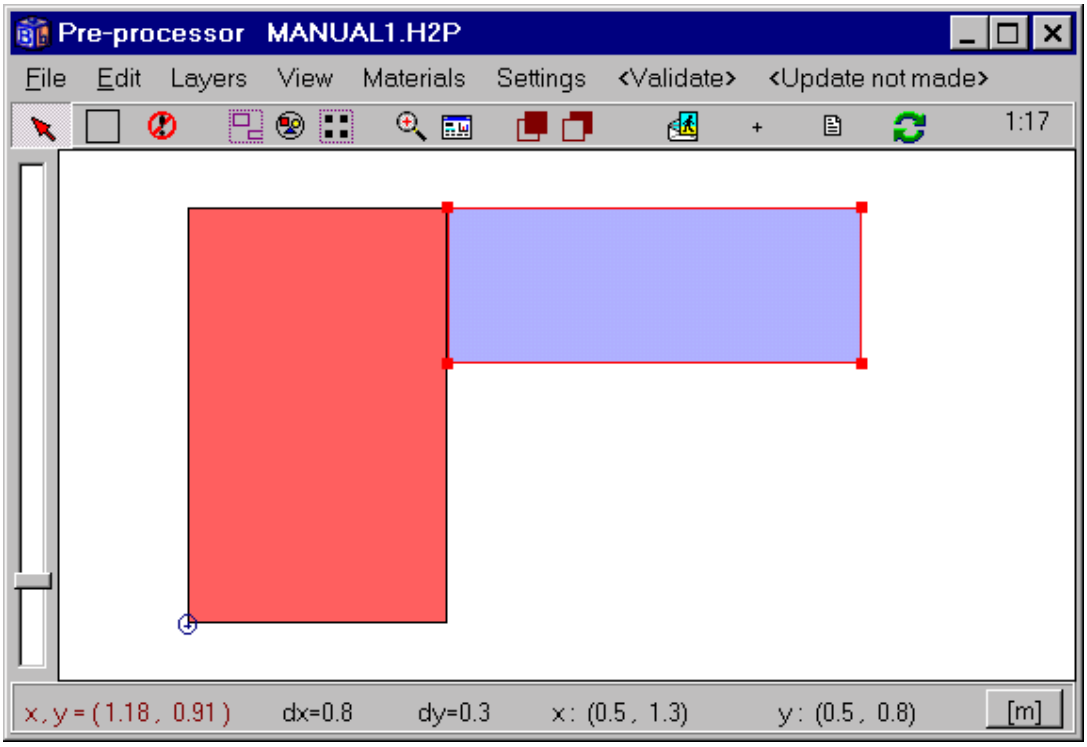


Figure 4.1: Two material rectangles drawn in the pre-processor.

The material is chosen from the material pick list, see Figure 4.2 (left). Double-clicking a rectangle brings up a window with material name and properties, see Figure 4.2 (right).

After the complete geometry has been described in the pre-processor it has to be updated for the HEAT2 solver. This is made by pressing the upper right menu item <*Update not made*>. The complete *input mesh* will be automatically generated and the external boundaries will be found. The post-processor, see Figure 4.3, shows the generated input mesh along with boundary numbers and lengths.

An equidistant numerical mesh has also been generated (in this case 50*50 as specified in the pre-processor, see menu item *Settings/Numerical mesh*) . The numerical mesh can be changed later on.

Before the simulation can be done, boundary conditions are assigned to the boundary numbers.

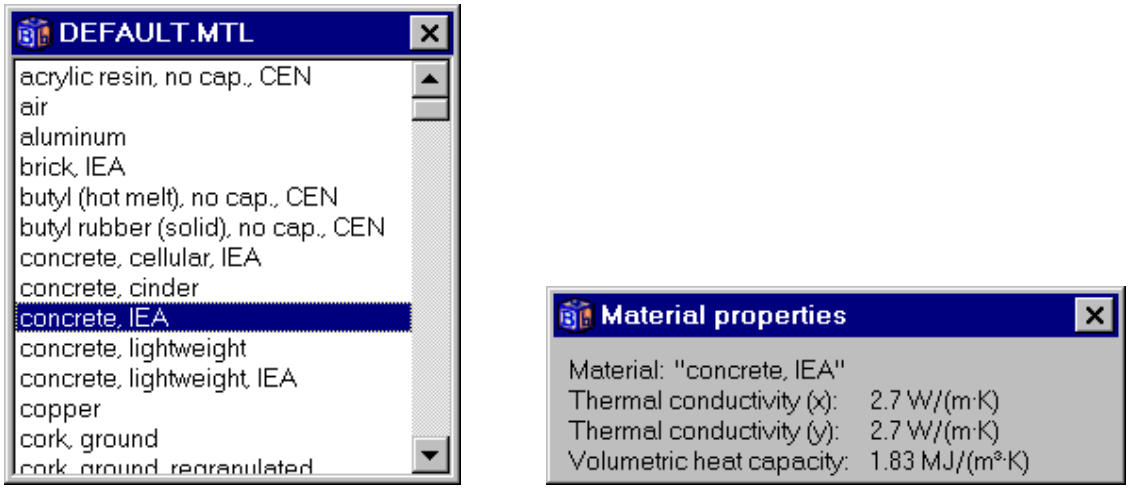


Figure 4.2: Material pick list (left). Double-clicking a rectangle brings up a window with material name and properties (right).

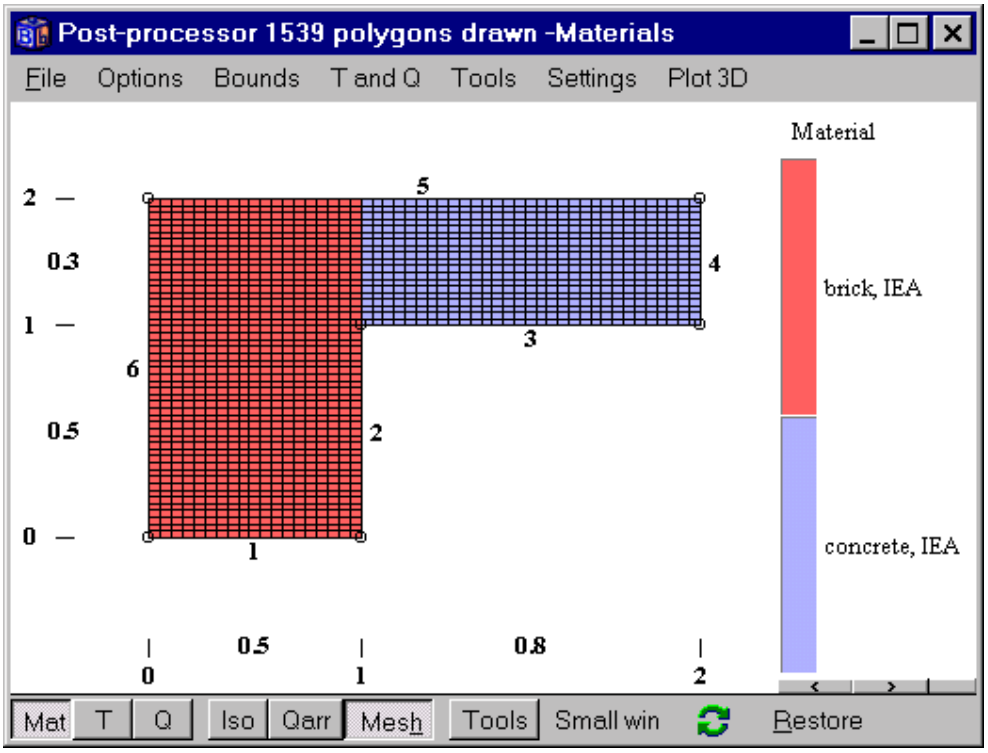


Figure 4.3: The post-processor showing the generated input mesh along with boundary numbers and lengths.

4.2 Boundary conditions

Different *types* of boundary conditions may be defined. A boundary condition type may either be an air temperature T, (°C), with a surface resistance R, (m²·K/W), between the air and the surface, or a given heat flow q, (W/m²). 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 5.9.

The user defines the boundary conditions in a list and assigns them to the boundary segments. Consider Figure 4.4. There are in this case three boundary conditions with list numbers n=1, 2, and 3 in the first

column. The third column shows what boundary segments that are assigned to a certain type (the boundary segment numbers 1 to 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, the latter will prevail. The boundary conditions may also be shown graphically, see Figure 4.5.

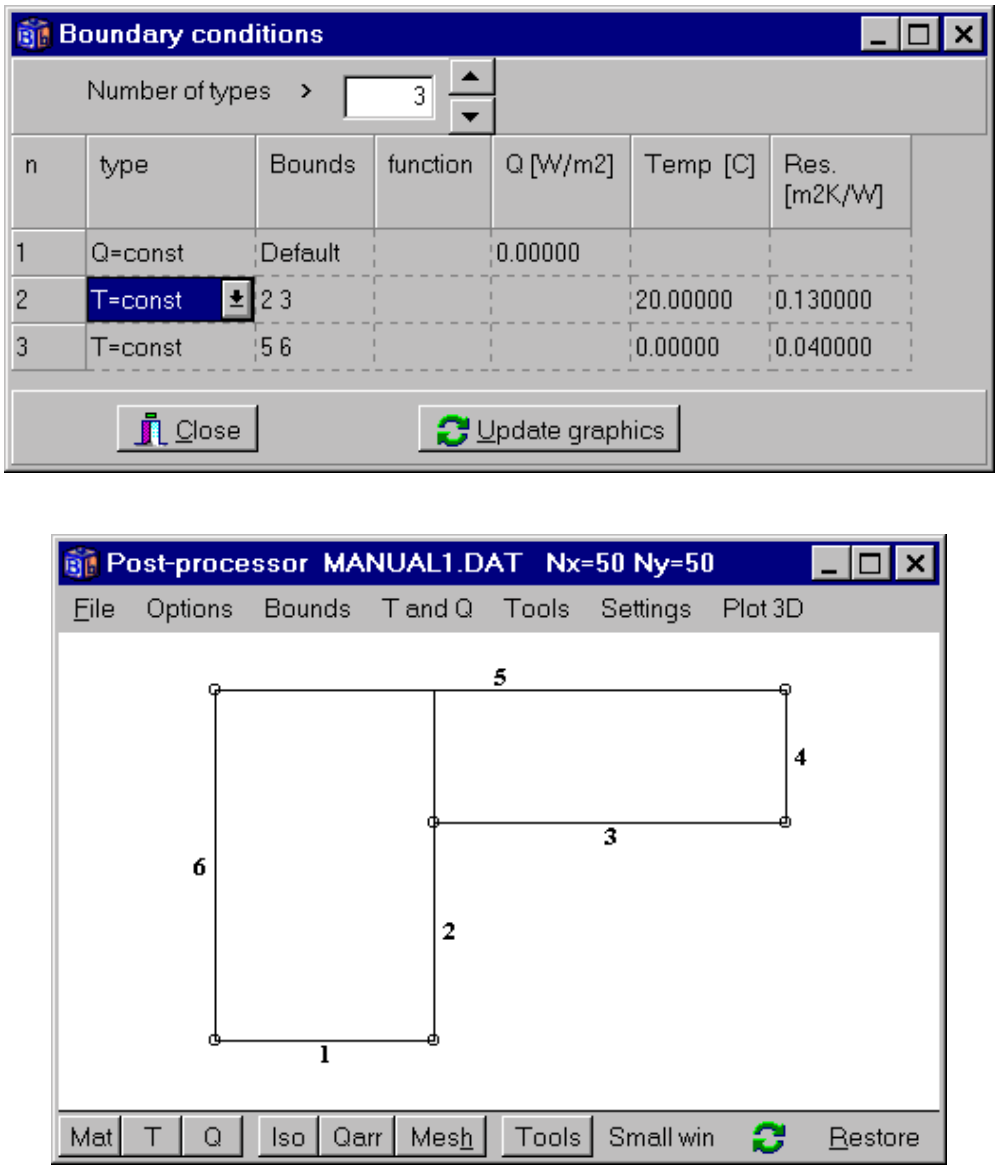


Figure 4.4: The boundary conditions are assigned to the corresponding boundary segments numbers 1 to 6.

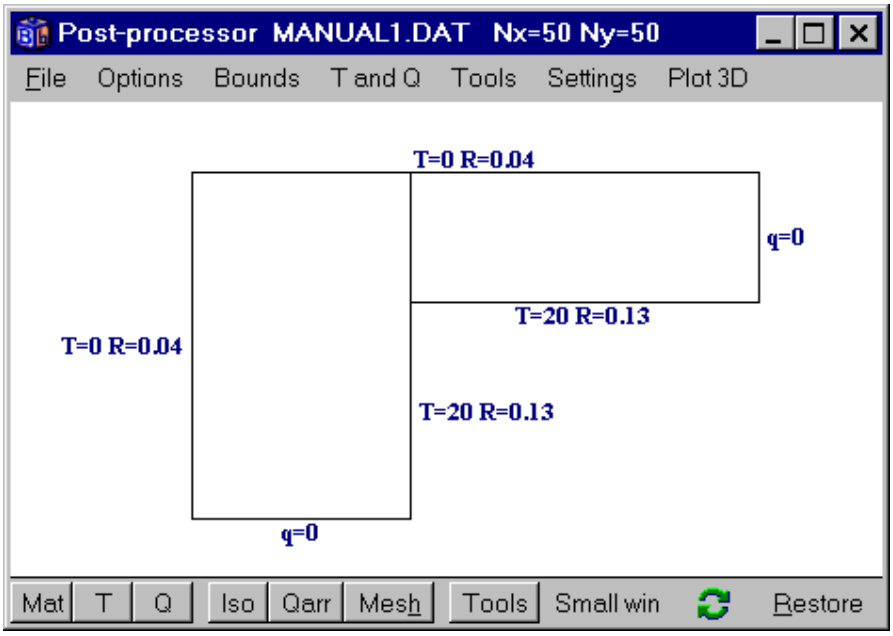


Figure 4.5: The boundary conditions may also be shown graphically by enabling item *Bounds/BC type values*.

4.3 Solving the steady-state problem

To start a steady-state simulation select *Solve/Start steady-state calculation*. A window, see Figure 4.6, will be displayed showing information such as the chosen stop criterion, number of iterations, and errors for temperature and flow, respectively. Figure 4.7 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 4.8 (*Output/Boundary flows*).

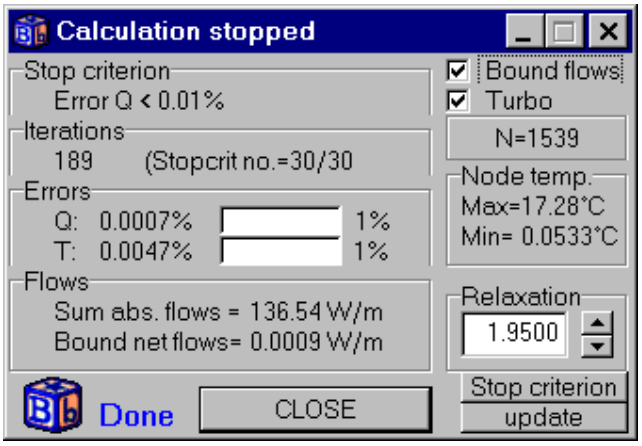


Figure 4.6: Window shown during steady-state calculation.

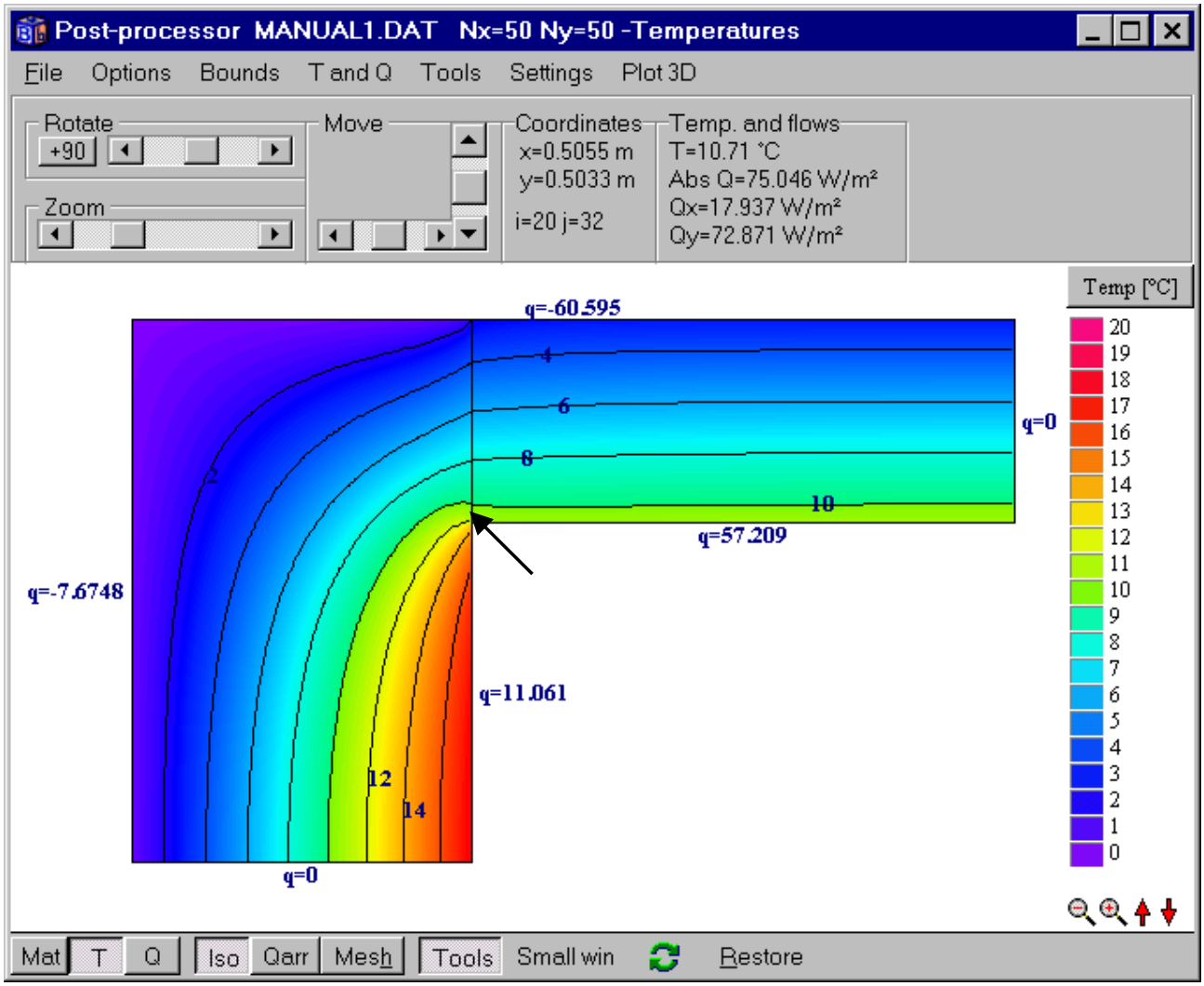


Figure 4.7: Calculated temperature field and isotherms. The upper toolbar (menu item *Tools/Upper toolbar visible*) shows options to rotate, zoom, and move the image, and coordinates with temperature and heat flows for the cursor location.

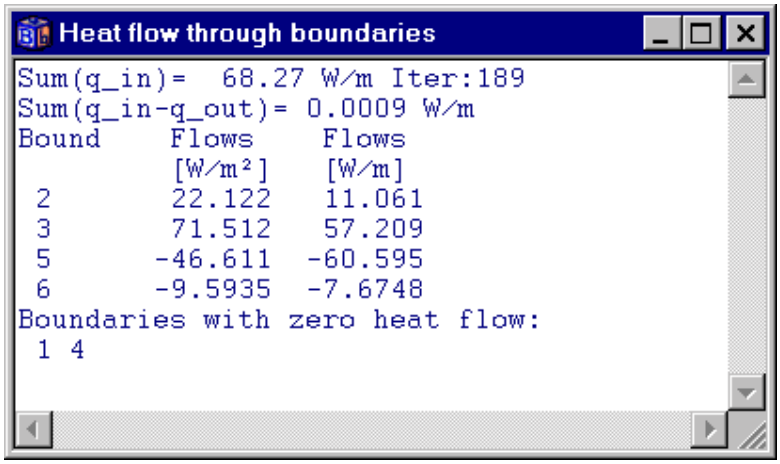


Figure 4.8: Heat flows through the boundaries.

4.4 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 for the above described corner problem. First, define a sinus function (item *Functions/Function 1*) with a daily variation between 0 and 20 °C with maximum at noon, see Figure 4.9.

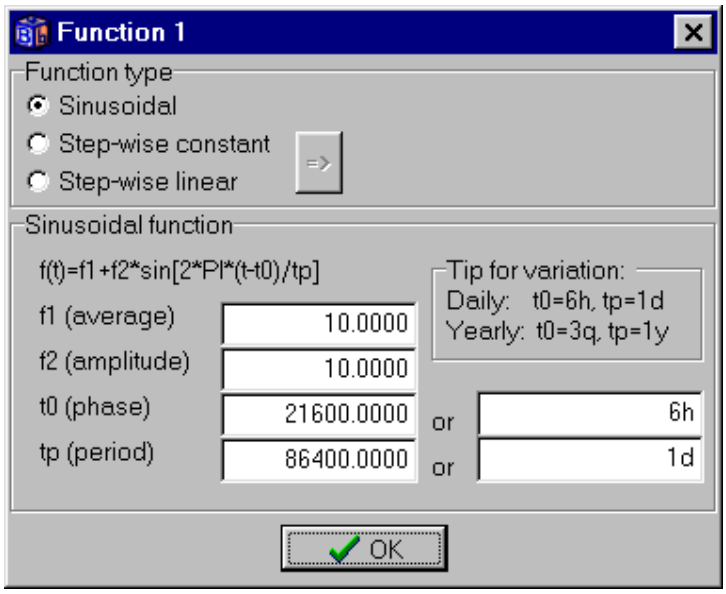


Figure 4.9: A sinus function with a daily variation between 0 and 20 °C with maximum occurring at noon. The time for the phase and period may be given either in seconds or a time string.

Next step is to couple the function to a boundary condition type (*Input/Boundary conditions*), see Figure 4.10. Finally, the stop time 4 days (Solve/Options for transient) is given, see Figure 4.11.

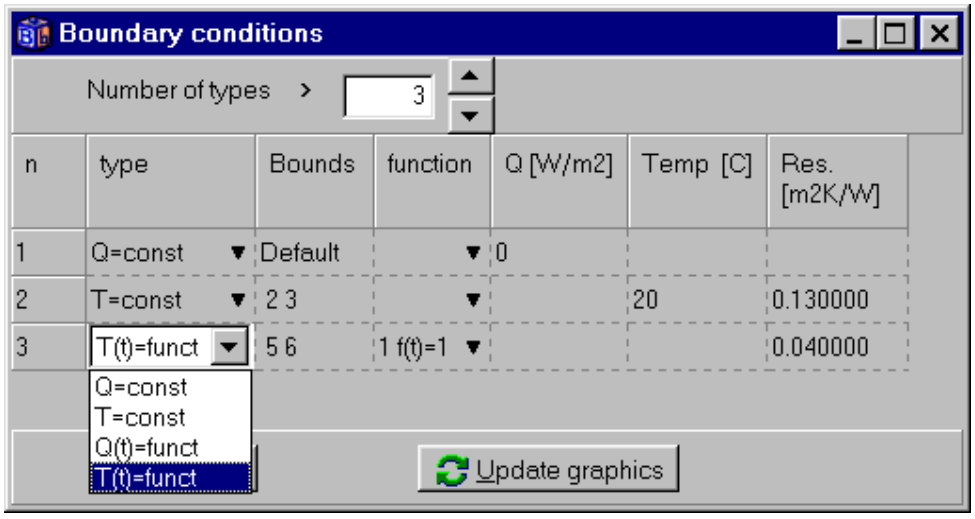


Figure 4.10: A temperature function is chosen in the pick list for the boundary condition type. The function number is selected in the function pick list (number 1 is default).

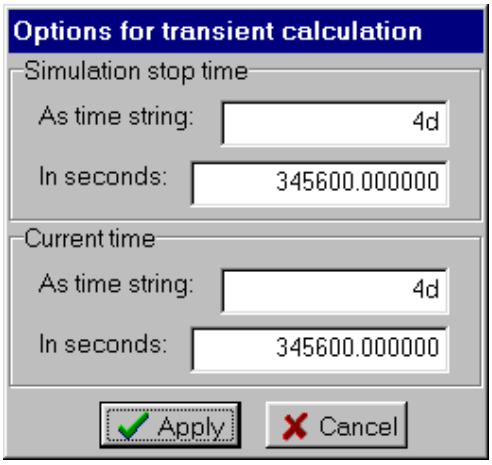


Figure 4.11: The stop time is 4 days.

In this example we will also look at the flows through the internal boundaries during each hour. Enable record (the check box in item *Output/Recorder*) and pick the second item in the list as in Figure 4.12 (top). Click in the data column and give the numbers 2 and 3 for the internal boundaries as in Figure 4.12 (bottom). This will record the sum of the two boundary heat flows at each screen update (*Solve/Update*) as given in Figure 4.13, 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 are yet recorded).

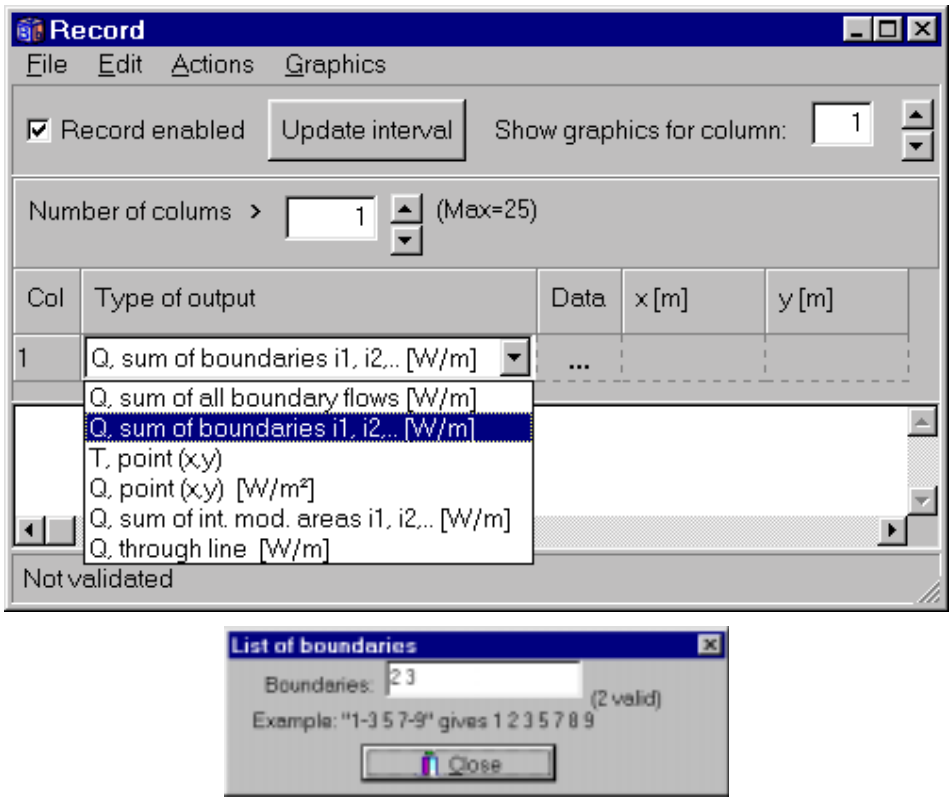


Figure 4.12: Flows and temperatures may be saved to a list during the simulation. In this case the sum of the heat flows through boundaries 2 and 3 will be recorded.

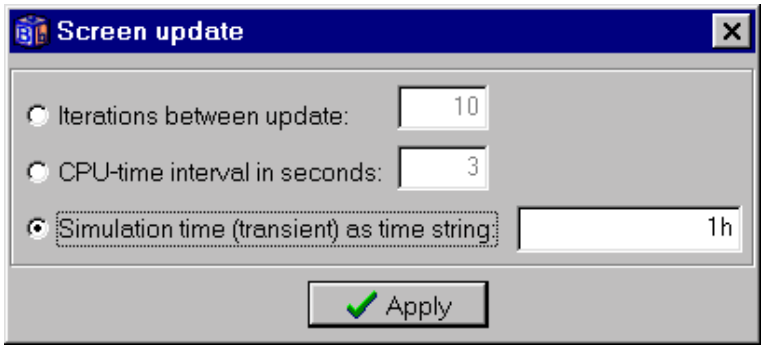
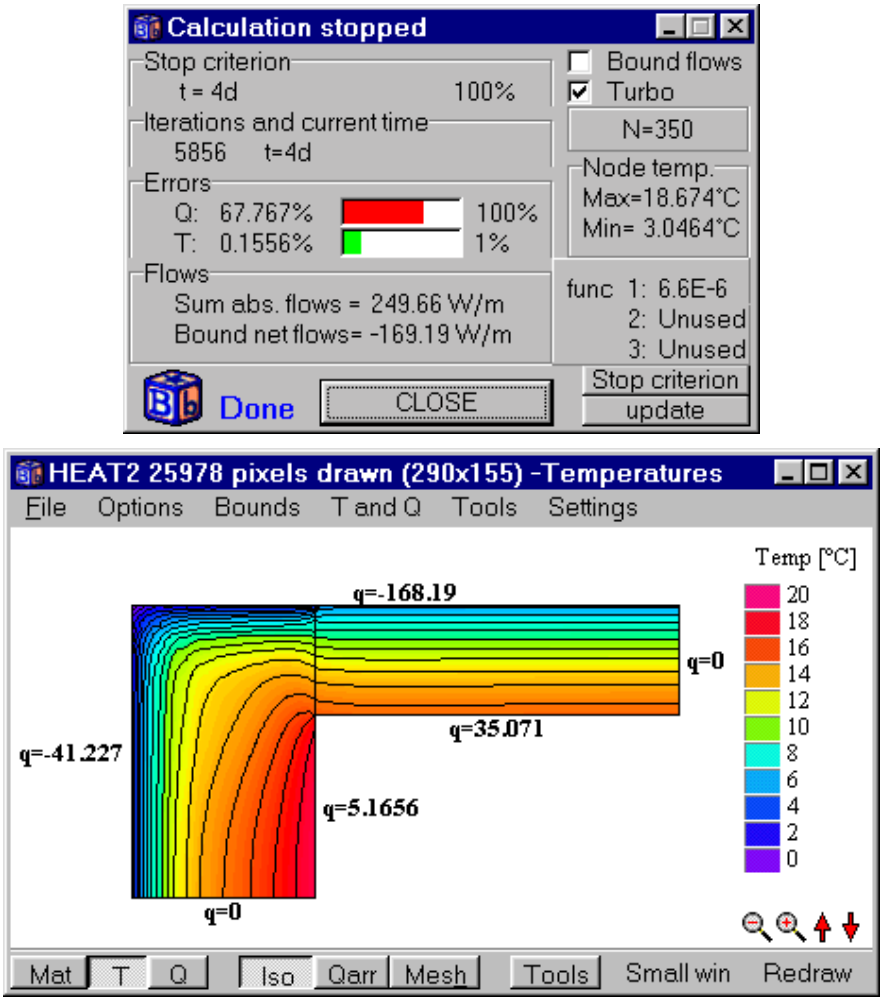


Figure 4.13: The time interval for recorded data is given in item *Solve/Update*.

Start the transient simulation by selecting *Solve/Start transient calculation*. Consider Figure 4.14. A window (top figure) 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.



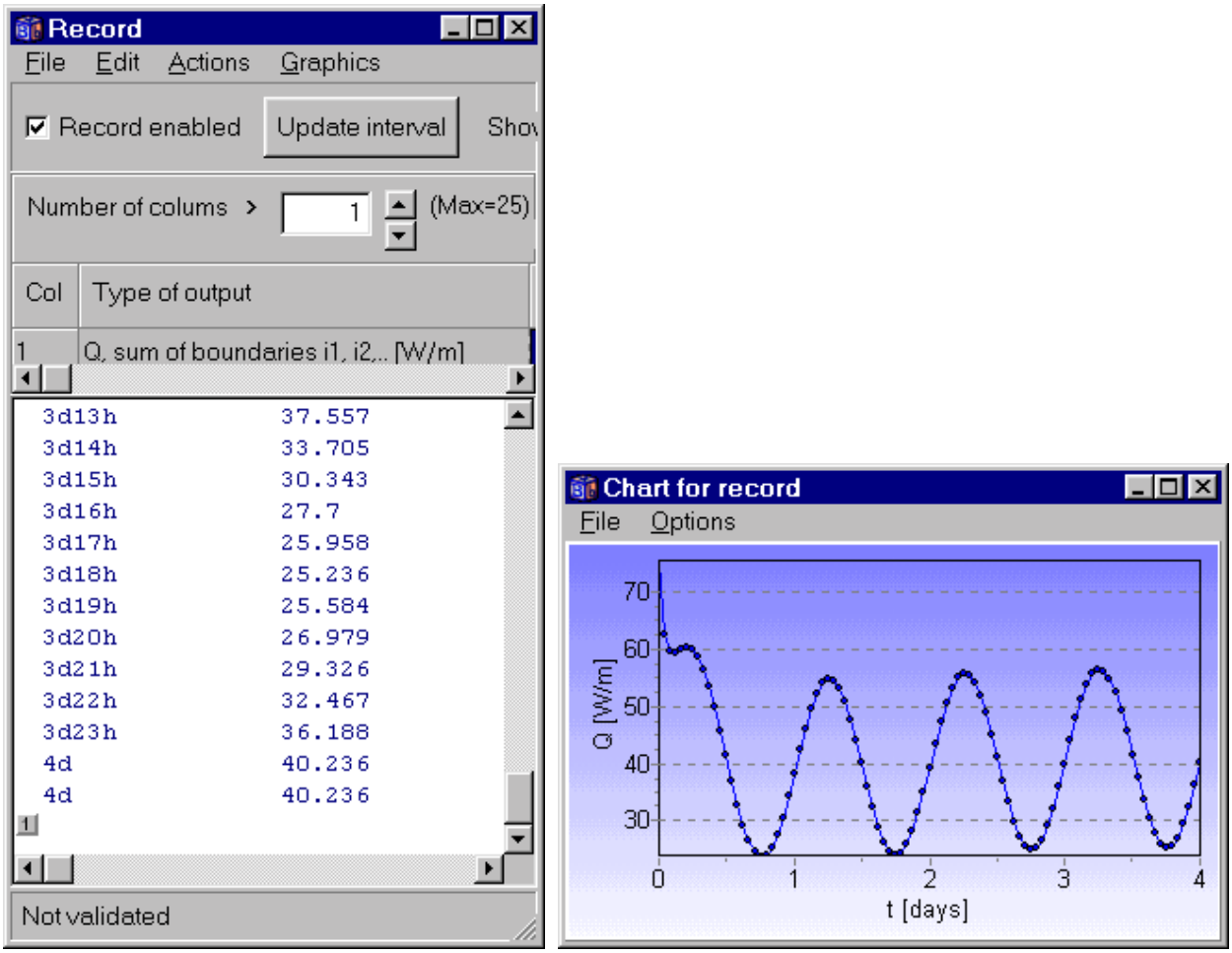


Figure 4.14: The solve window (top) shows chosen stop criterion, number of iterations, and error for temperature and flow. The record window (bottom left) and the chart window (bottom right) show the internal heat flows during each hour.

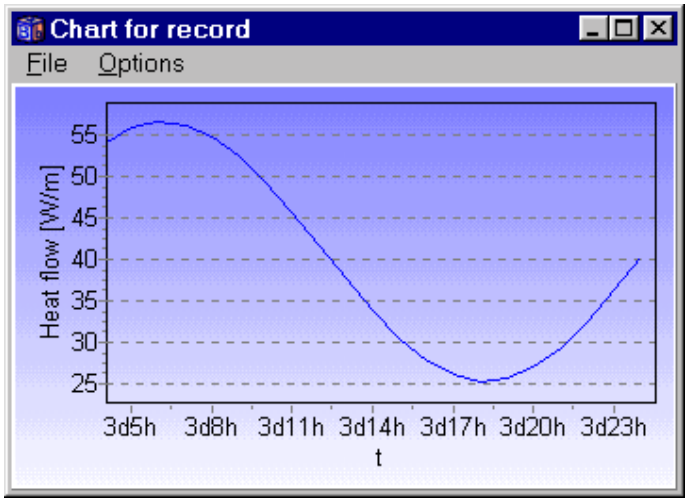


Figure 4.15: The maximum heat flow through the internal boundaries into the wall occurs at six-o-clock in the morning and the minimum at six-o-clock in the afternoon.

4.5 The input mesh

The *input mesh* facilitates the description of the geometry, the numerical mesh, and the boundary conditions. A suitable input mesh will automatically be generated when the pre-processor is used. It is also possible to give the input for a problem numerically without using the pre-processor. Below is a description of the input mesh. It is also shown how the input is made for the geometry in Figure 4.1

Consider as a rather simple example the L-shaped region in Figure 4.16. The required input mesh is shown. Here, *I,J* are the *input mesh coordinates*. The *mesh segment I* between the mesh coordinates *I-1* and *I* has the length ΔX_I. 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 4.16 there are two in both directions.

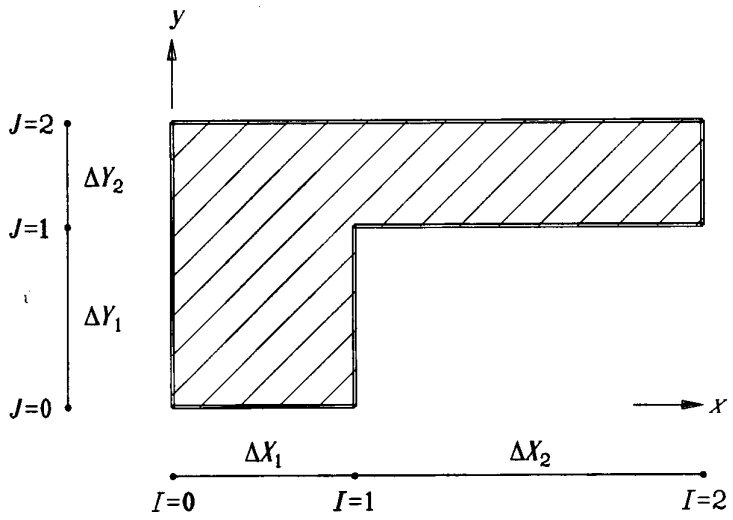


Figure 4.16: An simple example with a suitable input mesh.

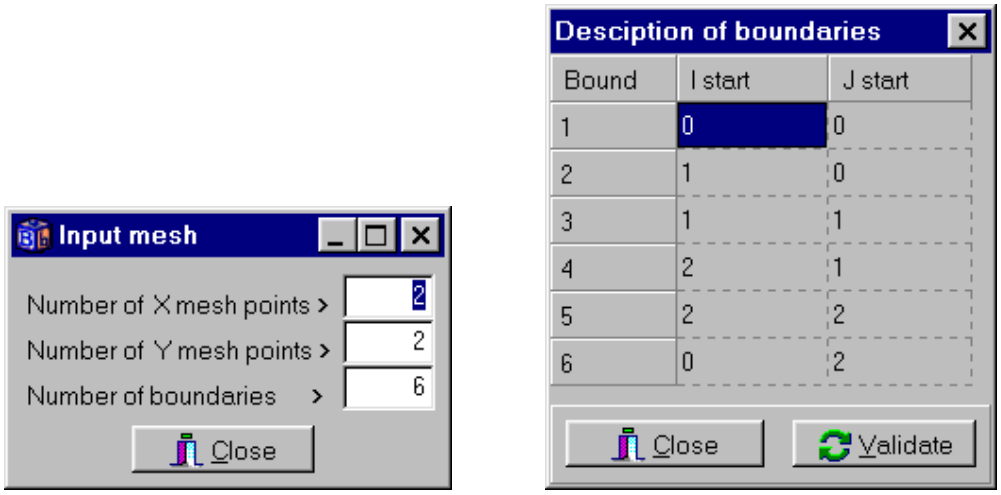


Figure 4.17: The number of mesh coordinates in each direction and the number of boundary segment have to be given first (left figure). The computational area is then defined by enumerating the boundary segments consecutively counterclockwise using the input mesh coordinates (right figure).

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 4.17. Figure 4.18 shows the boundary segments 1 to 6.

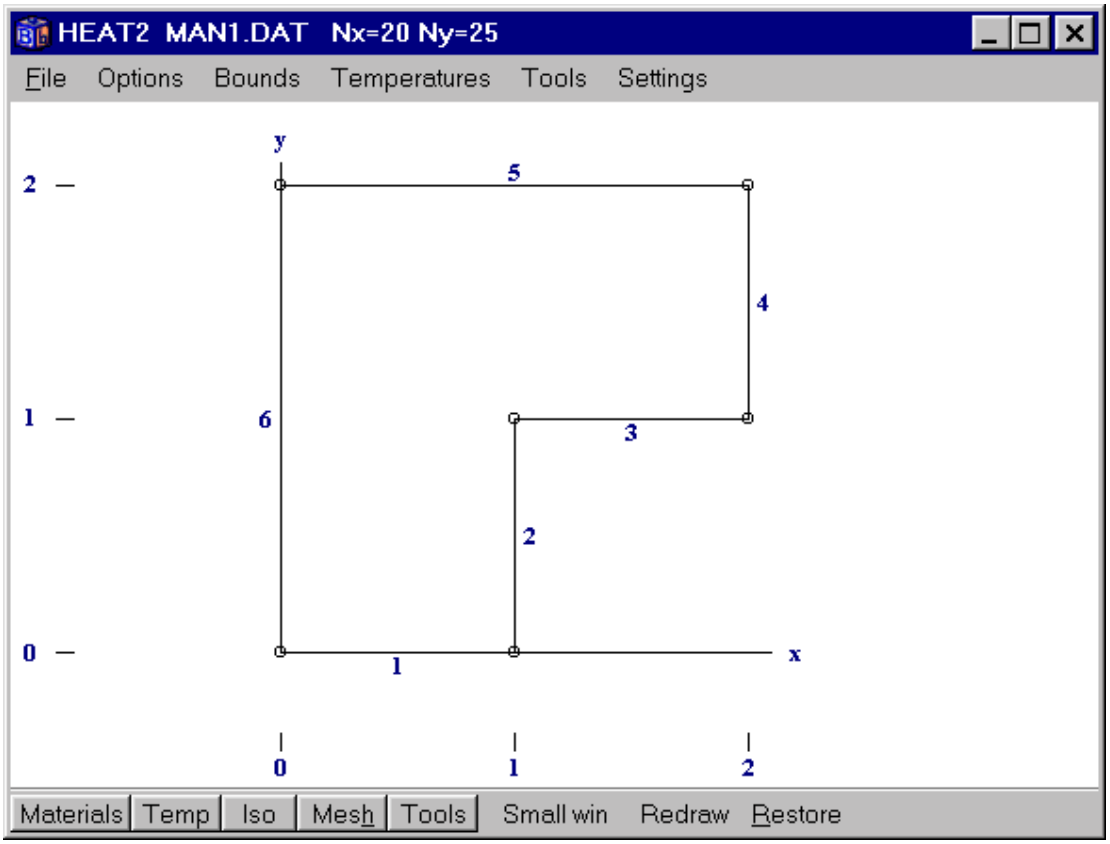


Figure 4.18: The six boundary segments defines the computational area.

4.6 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 4.19. The cells are normally placed in an equidistant mesh. The computational cells in each segment have the same size, see Figure 4.20. In general, capital letters refer to the input mesh (ΔX_I), while small letters refer to the numerical mesh (Δx_i). There must be at least one numerical cell in each input mesh segment. In Figure 4.20 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.

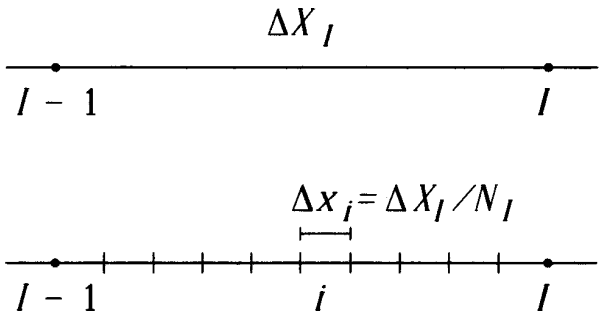


Figure 4.19: The length Δx_i in the computational mesh.

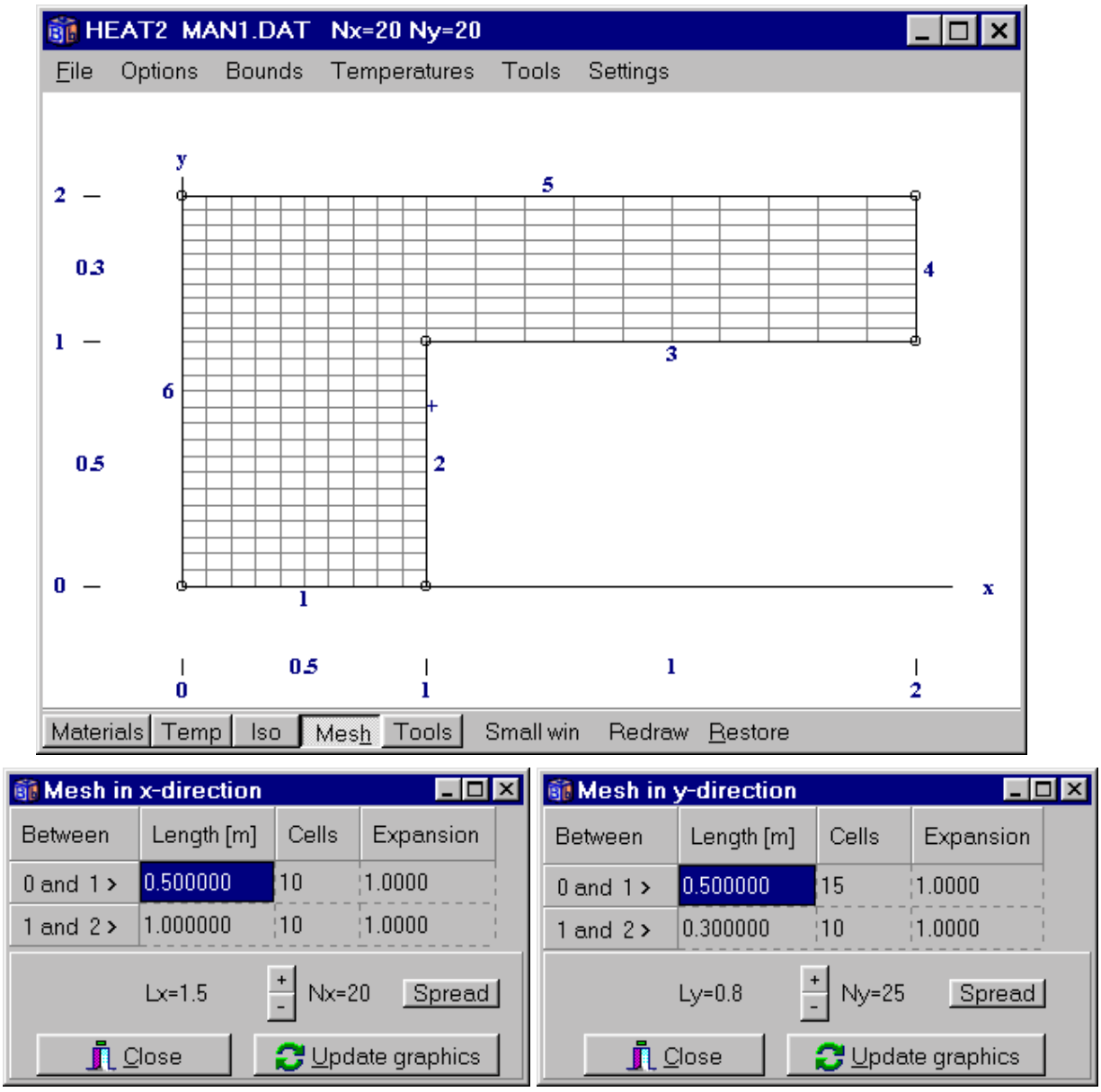


Figure 4.20: A numerical mesh is shown here with 20·25 cells (equidistant in each mesh segment).

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 and 1.25.

Figure 4.21 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.

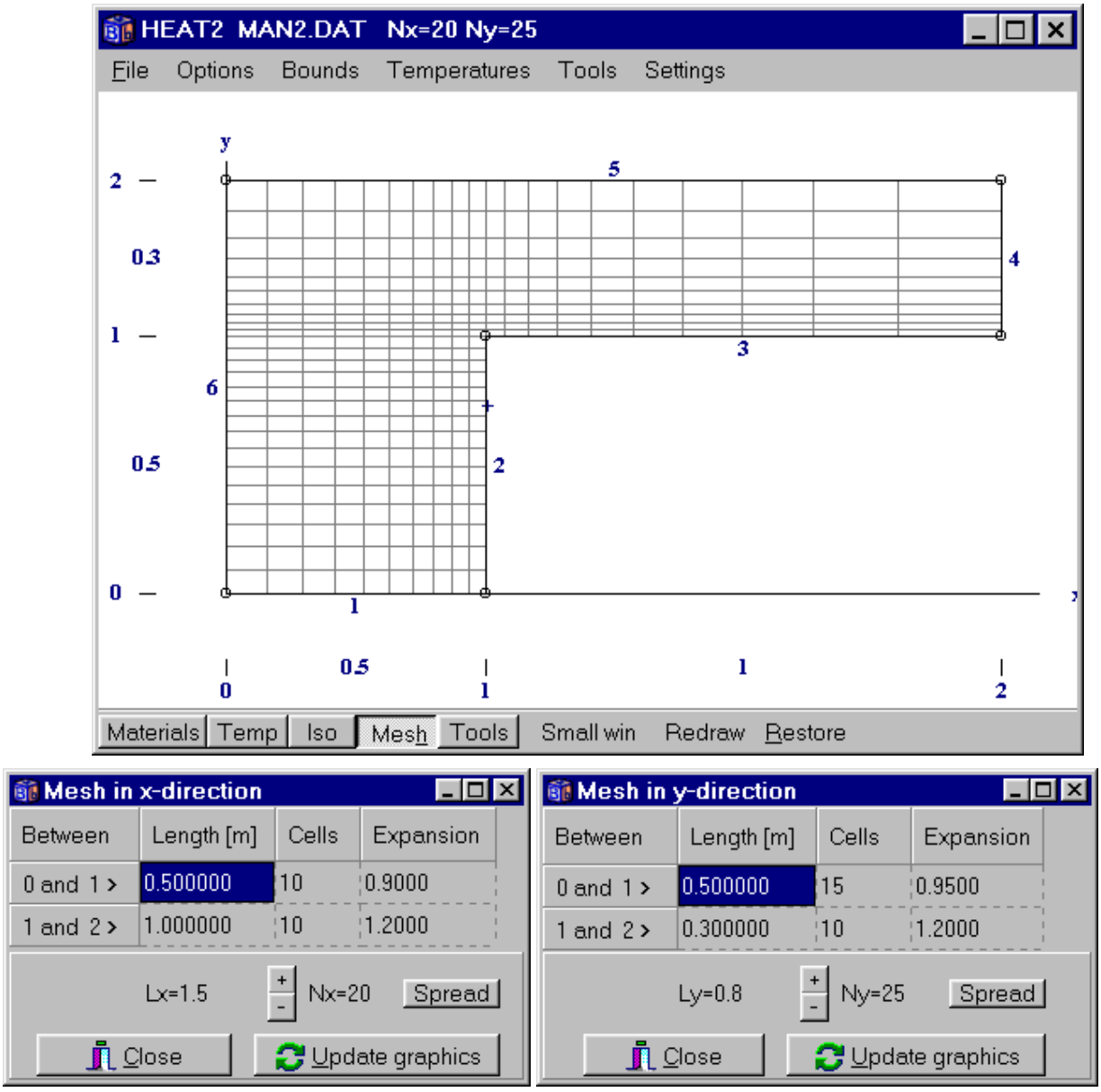


Figure 4.21: An expansive mesh with smaller cells near the inner corner (the temperature gradient is higher in the corner requiring smaller cells to decrease the numerical error).

4.6.1 Simple mesh generation

Figure 4.22 (left) shows the dialogue box for the mesh in *x*-direction. The buttons with + and – signs will double or half the number of cells in all mesh segments. The **Spread** button will distribute the cells evenly as far as possible in an equidistant numerical mesh. The total length is also shown (L_x=0.7).

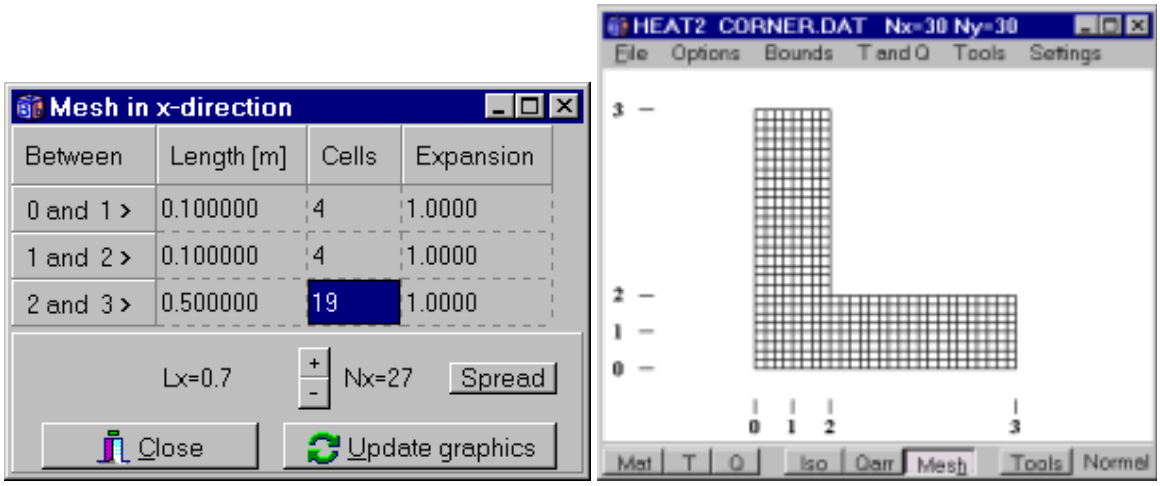


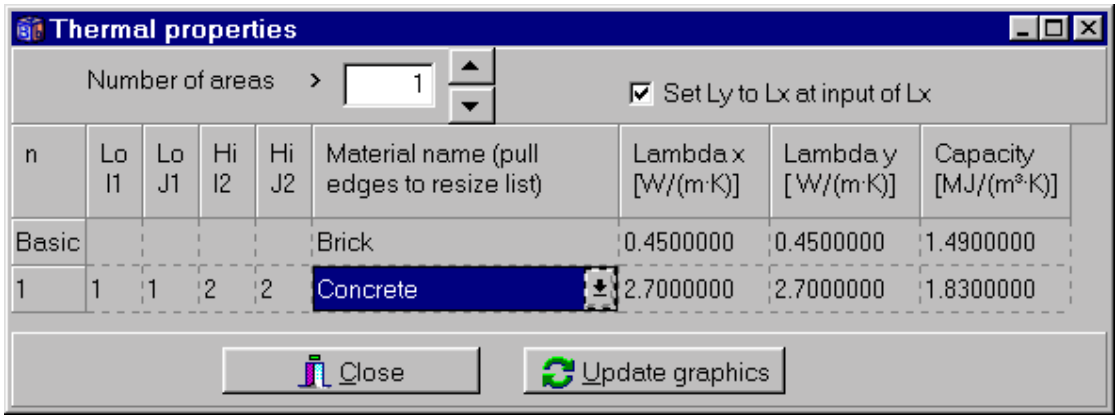
Figure 4.22: A simple equidistant mesh.

4.7 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 normally specified for each of the material in the pick list used for the pre-processor. It is however also possible to give properties numerically in the case when the pre-processor is not used as follows. 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 (the data for the overlapping rectangle will prevail).

Consider the problem in Figure 4.23 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).



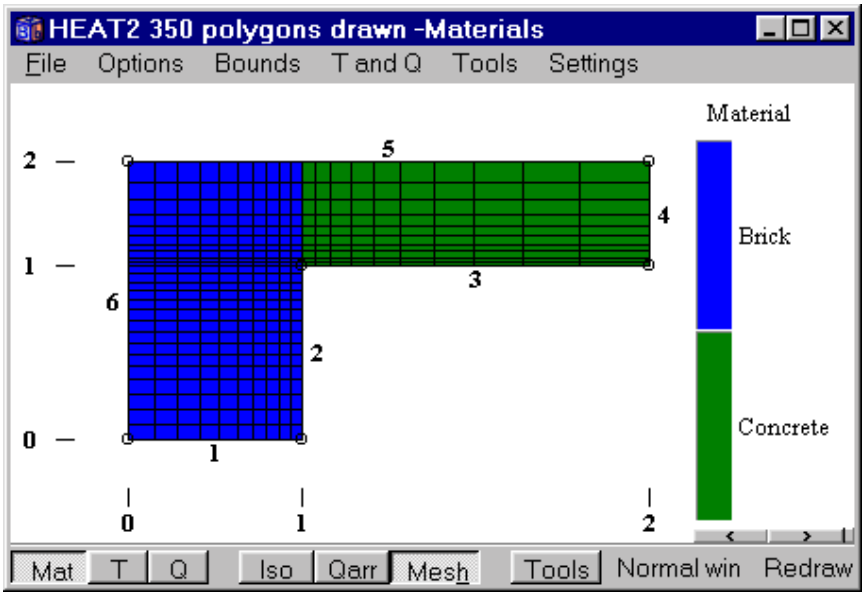


Figure 4.23: The materials are defined by giving default thermal properties that are valid for the whole computational area, and then using overlapping rectangles of different properties. The green (upper right) area shows a rectangle with lower left coordinate (1,1) and upper right coordinate (2,2).

4.8 Initial temperatures

Areas with different initial temperature areas are given by using the pick list item "Initial temperature area". Figure 4.24 shows two given areas that is outlined by black rectangles (the current selected object has a red rectangle with marked corners).

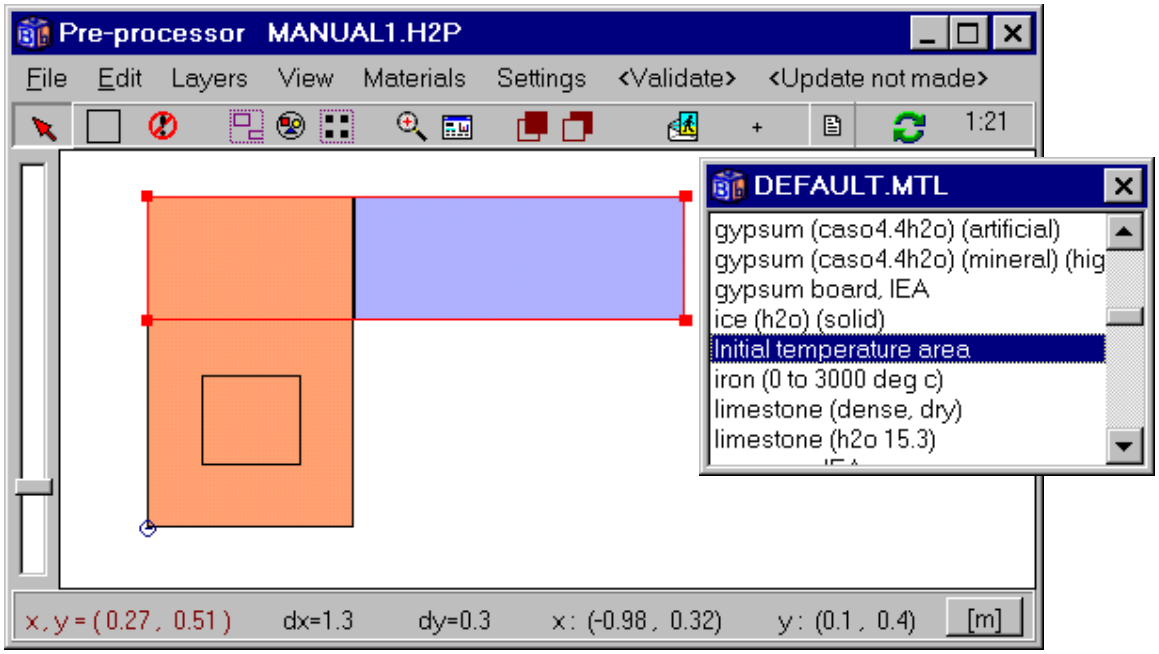


Figure 4.24: The drawn rectangle specifies an area with a certain initial temperature.

The pre-processor will generate a proper input mesh and the initial temperature list shown in menu item *Input/Initial temperature*", see Figure 4.25. It is up to the user to give the temperature values. In this case,

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. The third area has a temperature of 5 °C.

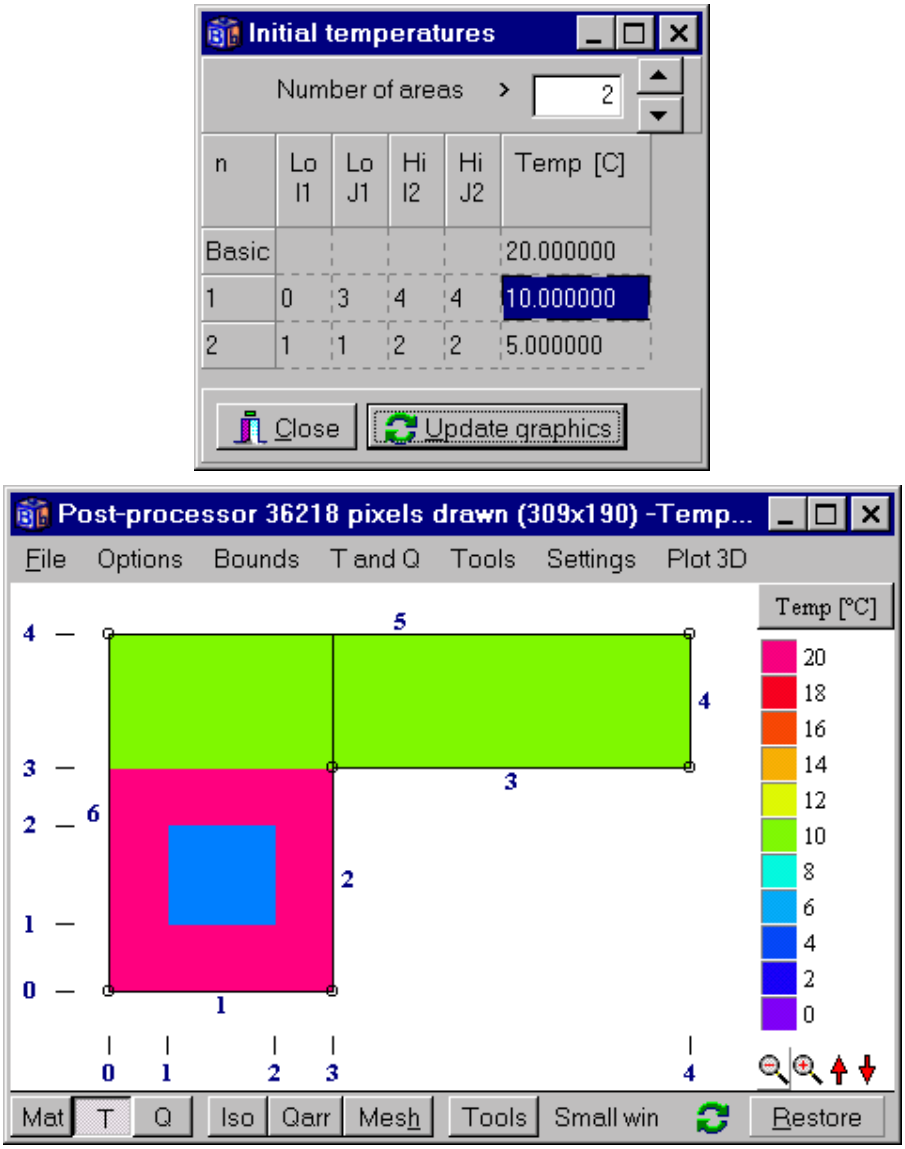


Figure 4.25: Initial temperatures are defined by giving a default (basic) temperature that is valid for the whole computational area, and then using overlapping rectangles assigning different temperatures. In this case, three different temperatures are used.

4.9 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 9). Approximated circular pipes with a given heat source or temperature (constant or time-dependent) may be used, see Chapter 6.

Internal modifications may be drawn directly, Figure 4.26. The type is chosen in the pick list. In this case eight rectangles specifying pipes with a given temperatures are defined. The relating data for each type are given in menu item *Input/Modification – Types*.

Figure 4.27 shows the temperature field when the pipes have a temperature of 30 °C. The external temperature is -15 °C. The color for T=0 is disabled here, see scale to the right.

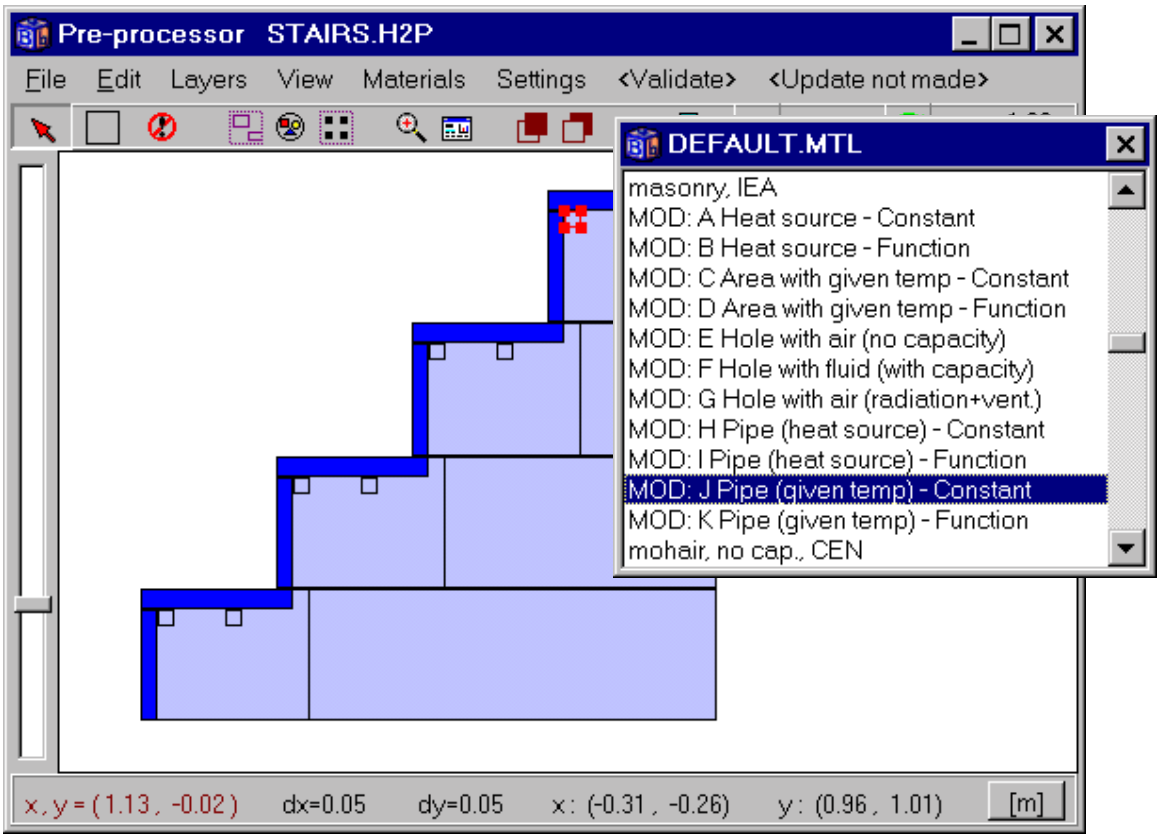


Figure 4.26: Internal modifications are drawn directly. In this case eight rectangles specifying pipes with given constant temperatures are defined.

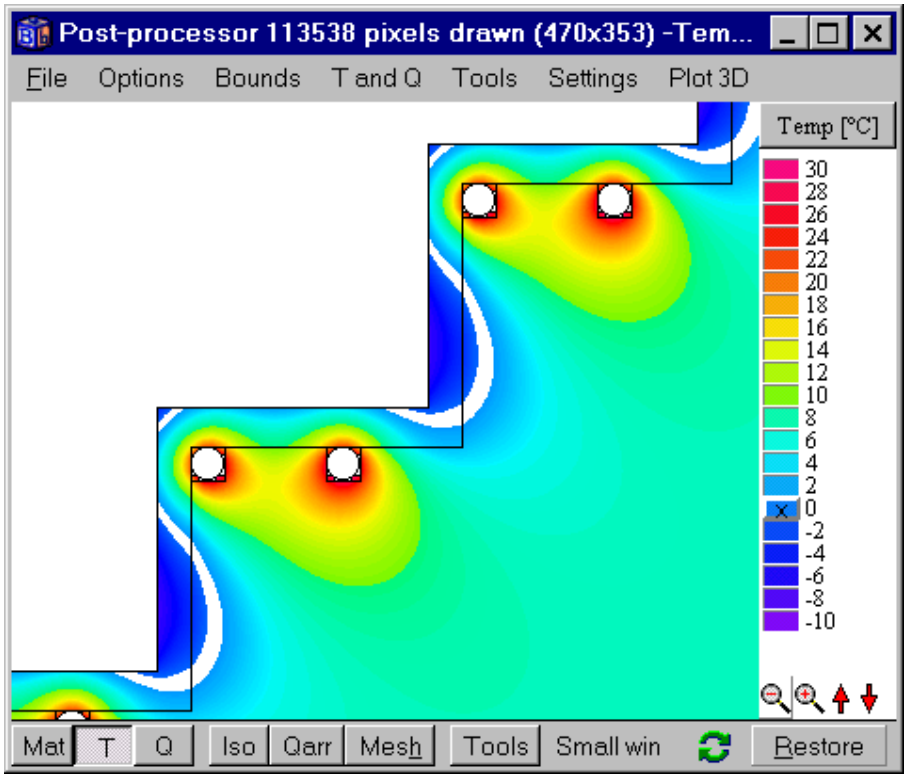


Figure 4.27[col]: Stairs heated by pipes with a temperature of 30 °C. The external temperature is -15 °C. The color for T=0 is disabled, see scale to the right, which here means that temperatures between –1 and +1 are not drawn, see the white band near the edge. The edge of each step is below zero and can be slippery.

The internal modifications must be placed according to some specific rules, see Figure 4.28. There should also 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.

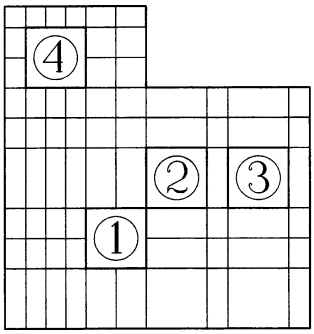


Figure 4.28: An example of how four internal modifications may be placed.

[col] See Appendix D for color image

4.10 Internal resistances

Internal resistances R, (m²·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 4.29. The vertical line located between the input mesh points *(1,1)* and *(1,4)* implies a resistance R_x in the x-direction.

4.11 Summary

A problem's geometry may be drawn using the pre-processor. An input mesh will automatically be generated together with the boundary segments.

The geometry may also be described solely using text input. It is however often more convenient to use the pre-processor, especially for more complex cases. Using text input, 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 4.29. 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*.

It makes no difference which one of the boundaries that is given first, but the enumeration must be done counterclockwise. Consider Figure 4.29. 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*

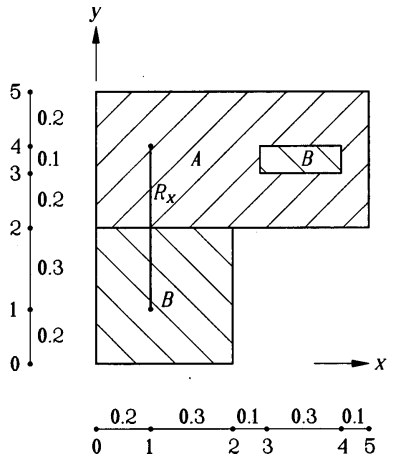


Figure 4.29: A suitable input mesh has to be defined taking the geometry, materials, initial temperatures, boundary segments with different boundary conditions, internal modifications, internal resistances, and numerical mesh into account. As an example, the mesh point 4 in the y-direction is used to describe the end-point for the resistance R_x, and the upper boundary for the smaller rectangle with material *B*.

5. Working with HEAT2

5.1 The pre-processor

5.1.1 Introduction

The pre-processor makes it relatively simply to describe the geometry and materials for a large range of heat transfer problems. The program works as common CAD-program for creating drawings. The construction is drawn using overlapping rectangles of different materials, see the over-view in Section 4.1.

There is a tool bar shown above the drawing area, see Figure 4.1. The tools are chosen by clicking on them with the mouse or picking the item in menu. To draw a rectangle, choose the 'Rectangle' (second icon from the left). Put the cursor somewhere in the drawing area and press the left mouse button. Then, move the mouse and press the left button again. There will now be a rectangle in the drawing area.

Figure 4.1 shows two rectangles drawn in the pre-processor. The bottom row shows the mouse pointer coordinates (x,y) in red. The width (dx=0.8) and the height (dy=0.3) are shown for the selected object, in this case the upper right blue rectangle. The extensions in both directions are also shown (the rectangle starts at x=0.5 and ends at x=1.3). In this case a *reference point* is placed at the lower left side of the left-most (red) rectangle marking the origin.

The material is chosen from the material pick list, see Figure 4.2 (left). Double-clicking a rectangle brings up a window with material name and properties, see Figure 4.2 (right).

After the complete geometry has been described in the pre-processor it has to be updated for the HEAT2 solver. This is made by pressing the upper right menu item <*Update not made*>. The complete *input mesh* will be automatically generated and the external boundaries will be found. The post-processor, see Figure 4.3, shows the generated input mesh along with boundary numbers and lengths.

An equidistant numerical mesh has also been generated (in this case 50*50 as specified in the pre-processor, see menu item *Settings/Numerical mesh*) . The numerical mesh can be changed later on.

Before the simulation can be done, boundary conditions are assigned to the boundary numbers.

5.1.2 The tool bar

The following tools are available:

Select: To select objects. The width and height may be changed for a selected object by stretching any of its sides or corners. Several objects can be selected by pressing the Shift key when clicking. Selected objects can be moved together. The material name can easily be picked in the material list for all currently selected objects. To un-select any object, move the cursor to an empty area and press the left button, or press the right button (this will make the **Rectangle** tool active).

Rectangle: Tool to draw rectangles. Press right button to make the **Select** tool active.

Undo/redo: Undo/redo function applies to deleted, moved, or resized objects.

Select multiple: All objects within the outlined rectangle will be selected. Additional objects may be selected by pressing the Shift key when clicking on them.

Select all: All objects will be selected.

Select all of one kind: Selects all objects of the same material (or of the same internal modification). A powerful feature is to copy the selected objects using "Ctrl-V" or the "Insert"

button. The selected objects could easily be changed to another material (or internal modification).

Zoom: Enlarges the outlined area.

Fit in window: The zoom scale will be chosen to fit all objects in the drawing window.

Bring to front: Rectangles that are hidden under other rectangles may be put on the top.

Send to back: Rectangles that are on top on other rectangles may be sent to the back.

Import image: Click in window to position an imported picture (BMP, EMF, WMF, ICO)

Reference point: Place a reference point that will define origo. Reference points may be moved (indicating a new origo). Several points may be placed. The last placed (or moved) point will define the current origo.

Material pick list: Brings up the material pick list window (same as TAB key).

Redraw: Invalidates the drawing area.

5.1.3 Copying objects

To copy and paste objects do as follows. Select the objects and press “Ctrl-V” or the “Insert” button to copy them at the cursor position, see Figure 5.1.

5.1.4 Picking material

The material of the rectangle may be changed by picking the desired item in the material list (TAB key).

5.1.5 Snap to grid

By default there is a grid of 10 pixels that the drawn rectangles will snap to. It is possible to disable this feature. In this case the rectangles may be placed at any pixel. It can however be difficult to place them correctly. Try to change the scale factor instead of disabling *Snap to grid*.

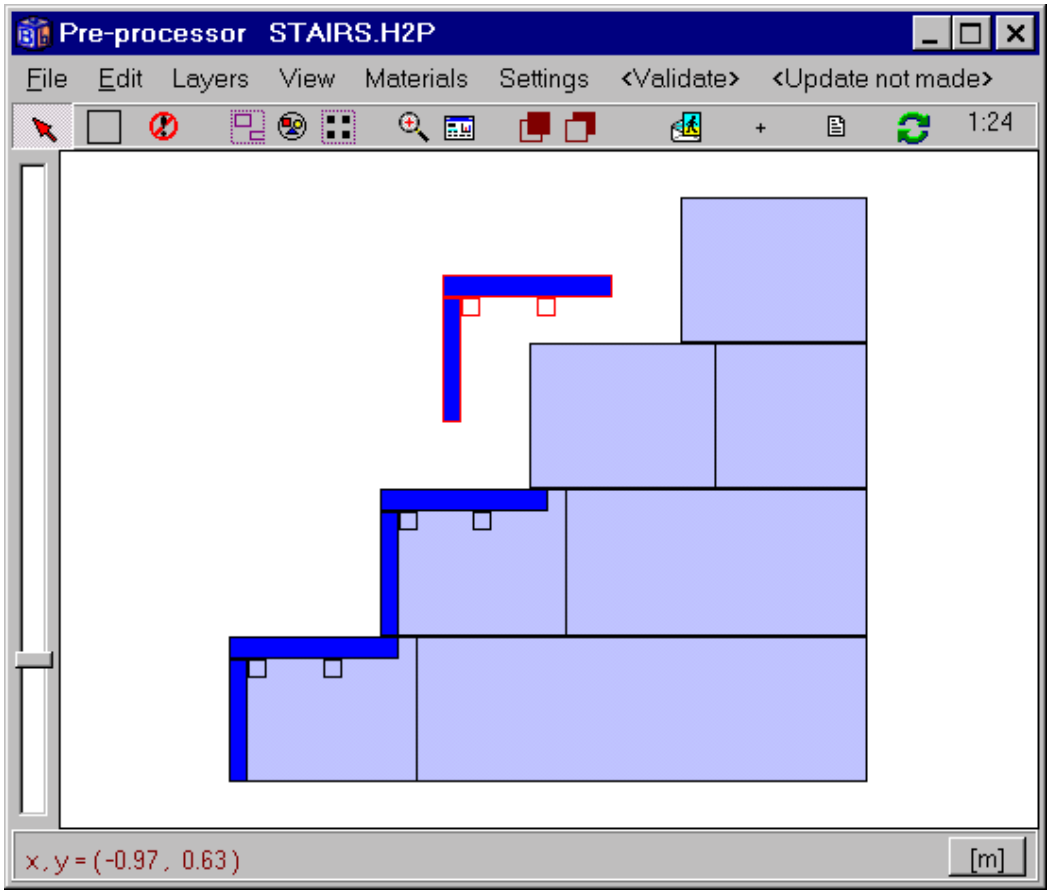


Figure 5.1: Multiple objects may be selected and copied. In this case, a stairway heated by pipes is built by copying one stair.

5.1.6 Zoom scale

The zoom scale in the drawing window is shown at the right on the tool bar. Use the slide bar left of the drawing window to change the zoom scale.

5.1.7 Scale factor

The scale factor has a default value of 1:10. This means that 1 cm in the drawing area is 10 cm in the real world. Choosing 1:0.01 means that 1 cm is 0.1 mm, and 1:1 000 means that 1 cm in the drawing area is 10 m in the real world.

There is a limitation of 3200 for the ratio of the smallest and largest rectangle that can be drawn using the pre-processor. You can for example draw a wall that is 1.6 m thick with a thermal bridge effect due to a 0.5 mm thick steel bar across the wall. Another example is calculation of heat losses from a house to the ground where the computational area includes soil within 32 m downwards/outwards from the house. So, the smallest possible thickness of the slab or any other detail will in this case be 1 cm. **Note however that you can change all lengths more or less arbitrarily in menu items *Input/Mesh in x-dir* and *Input/Mesh in y-dir* after the geometry has been updated.**

The scale factor should normally be set according to the smallest width that is to be drawn, see Table 5:1. The sizes that can be drawn are multiples of this smallest size. As an example, if a steel stud of width 0.5 mm is to be placed in insulation, a proper scale factor would probably be 0.5. This means that all lengths 0.5 mm, 1 mm, 1.5 mm, and so on up to 1.6 m can be given. If the steel stud had a width of 0.7 mm it would probably be best to use 0.1 that would give a resolution of 0.1 mm. By default, the scale factor is 10.

Smallest size to be drawn	Largest size that can be drawn	Use factor
0.5 mm	1.6 m	0.5
1 mm	3.2 m	1
1 cm	32 m	10
1 m	3200 m	1000

Table 5:1: The scale factor is chosen according to the smallest size that will be drawn.

5.1.8 Drawing in metric or English units

By default lengths are drawn using metric units. It is also possible to change the unit to inch, foot or yard. To do this click on the button located in the lower right corner. The lengths will automatically be calculated to the metric unit when updating occurs. Figure 5.2 shows a drawn rectangle with a width of 100 inches and a height of 20 inches (top figure). The metric lengths are calculated automatically (bottom figure). Note that the drawn rectangles still are scaled according to the scale factor. Note also that all results are presented in metric units.

Choosing the units *m*, *cm* and *mm* will not affect any scaling. Only the displayed lengths at the bottom will be shown accordingly.

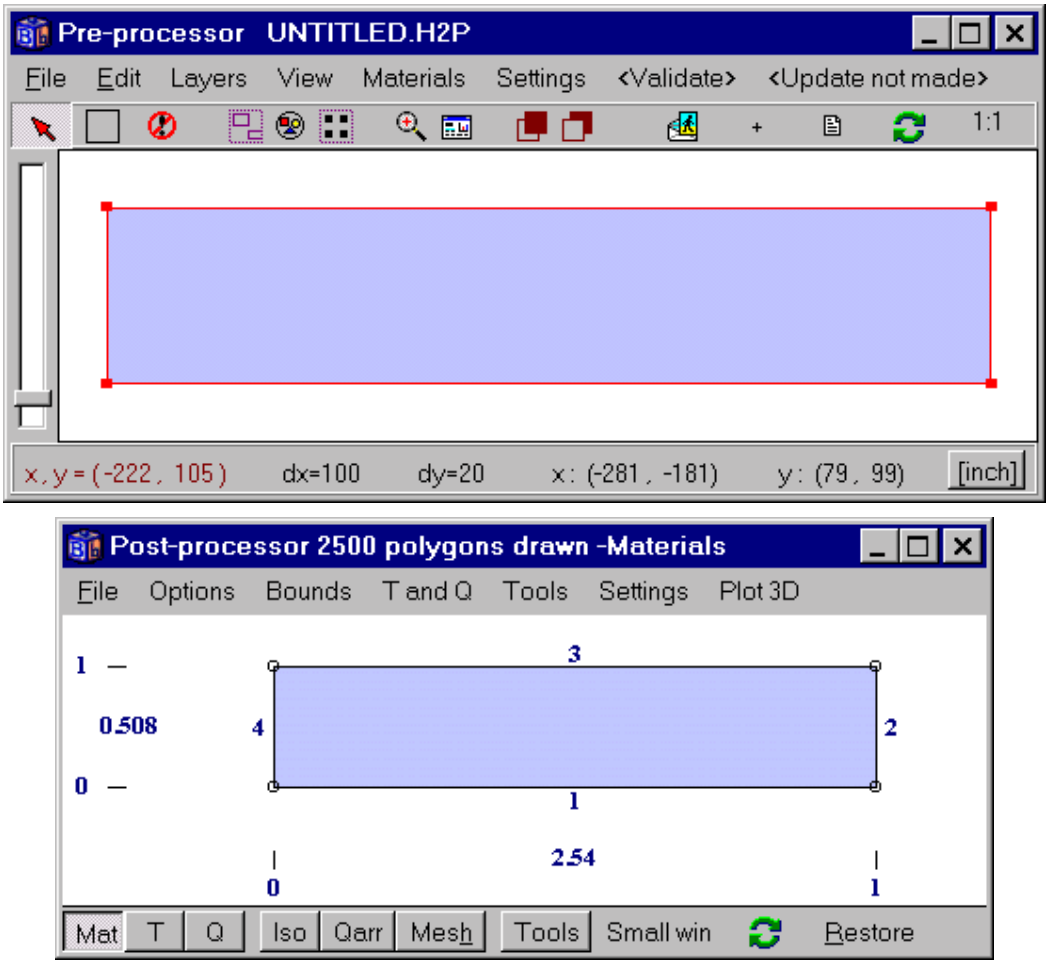


Figure 5.2: The drawn rectangle has a width of 100 inches and a height of 20 inches (top). The metric lengths are calculated automatically and here shown in meters (bottom).

5.1.9 Updating the geometry

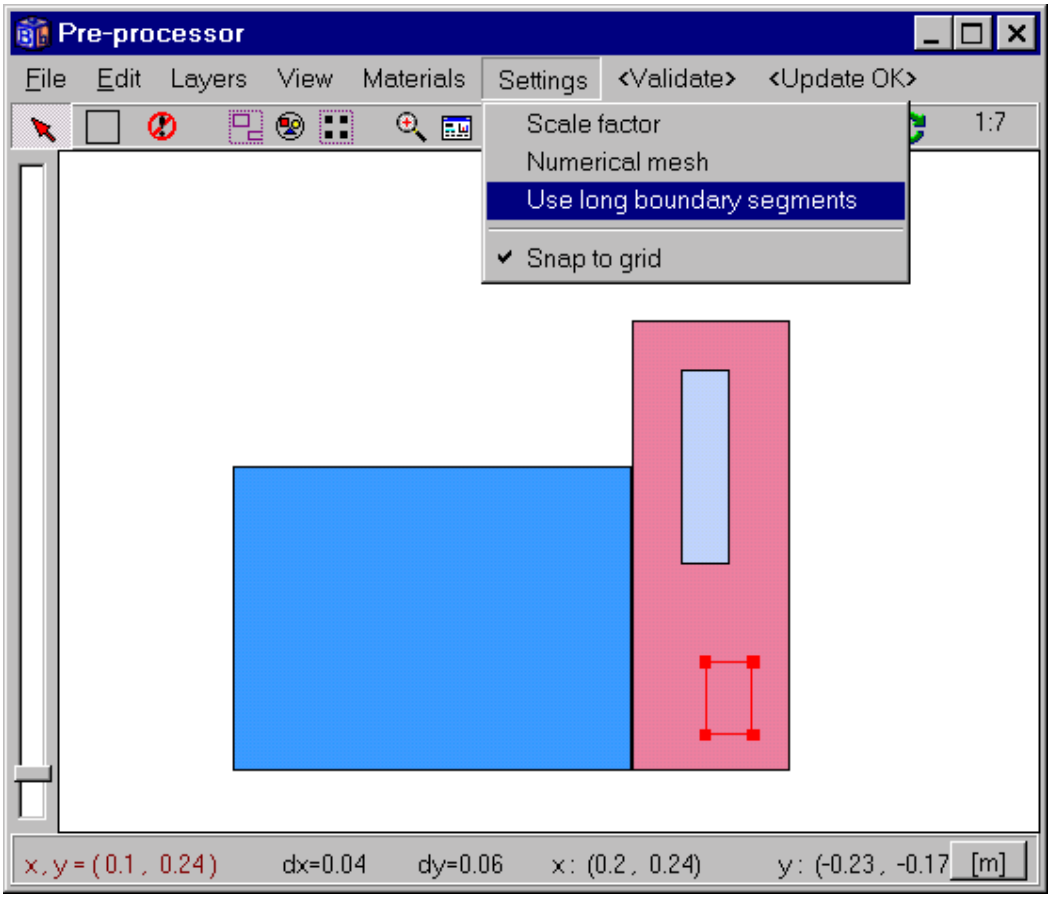
The drawn geometry in the pre-processor is not valid until the menu item <*Update*> is pressed. The following forms under the *Input* menu are affected by updating the pre-processor:

Input mesh, *Description of boundaries*, *Mesh in x-dir*, *Mesh in x-dir*, *Thermal properties*, *Initial temperatures* (if an initial temperature area is drawn), *Modifications types and areas* (if modifications are drawn).

Several checks are made to ensure that the drawing is valid. The area must be continuous, to mention one of the restrictions. If there are holes (empty areas enclosed by rectangles) in the drawing, these will be treated as frame cavities, see chapter 10. The menu item <*Validate*> will check that the drawing is ok and display number of used rectangles and boundaries, but not update the geometry.

5.1.10 Long boundary segments

The boundary segments enclosing the computational are automatically generated. By default long boundary segments are used. Disabling menu item *Settings/Use long boundary segments* gives a boundary segment break at each input coordinate. Figure 5.3 (top) shows four drawn rectangles. One rectangle overlaps another with the same material (selected in the figure). The middle figure shows the boundary segment numbers when long boundary segments is enabled and the bottom figure shows the boundaries when the item is disabled. In the latter case different boundary conditions may now be assigned to each of the boundary segments 2-12 (see bottom figure).



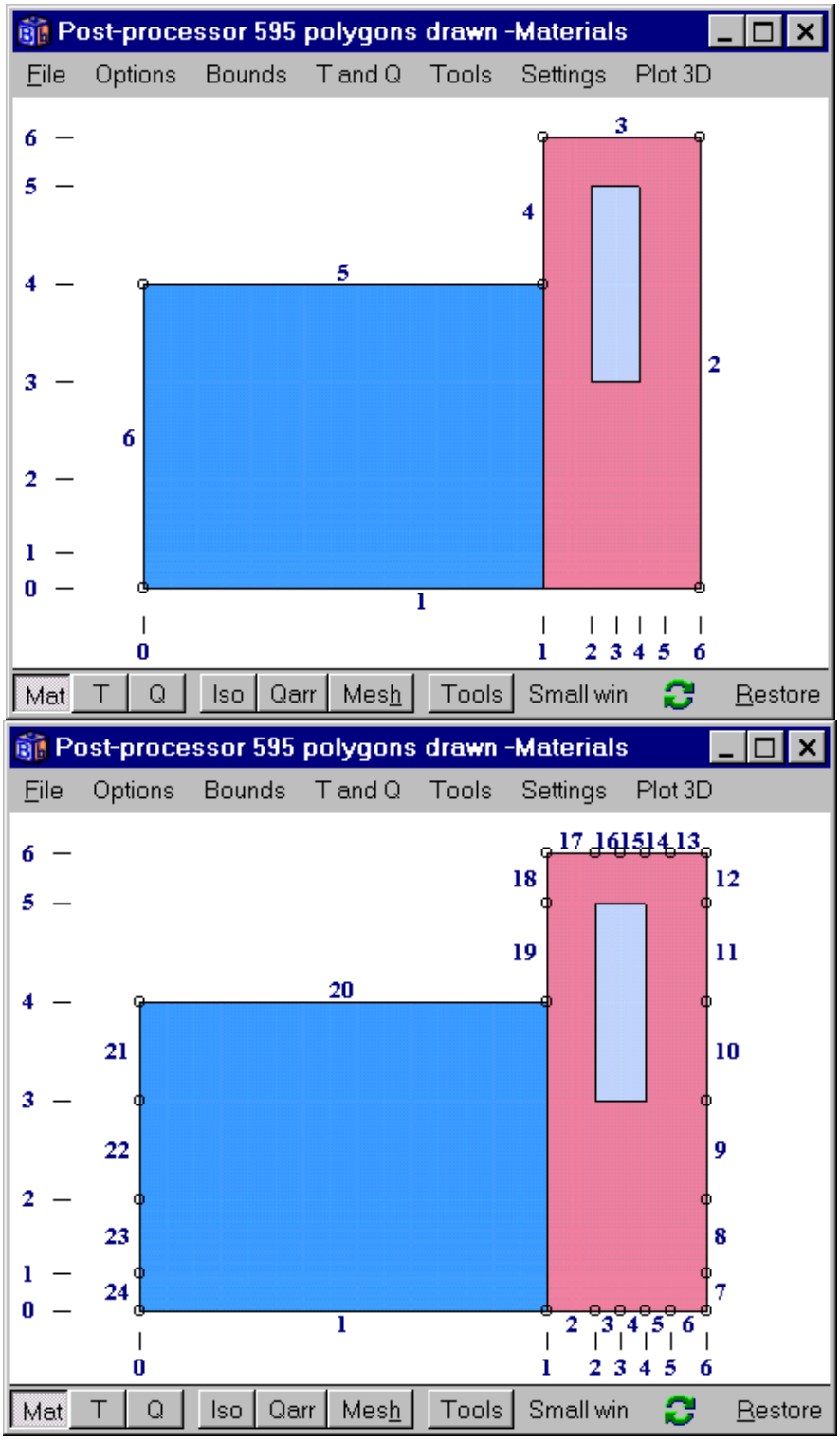


Figure 5.3: Geometry drawn in the pre-processor (top figure). Long boundary segments enabled (middle figure) and disabled (bottom figure).

5.1.11 Numerical mesh

The number of computational cells is specified in each direction, see Figure 5.4. The cells will be spread out in an equidistant mesh. The default numbers are 50*50 cells. Note that there always has to be a

minimum number of cells for the computational region. However, this is automatically taken care for. As an option it is also possible not to over-write the current specified mesh.

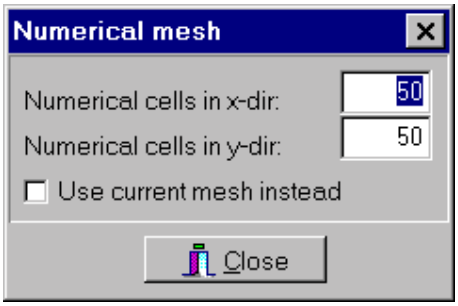


Figure 5.4: The number of numerical cells are given.

5.1.12 Importing pictures

Pictures in format BMP, EMF, WMF, ICO can be imported. One example is to import a scanned building design drawing. The actual heat transfer problem can be described by drawing rectangles ”on” the imported design drawing.

5.1.13 Pre-processor data files H2P

The drawn geometry is saved to a file with the extension *.H2P in the same directory folder as the ordinary input file *.DAT (or *.PSE). So, when a DAT (or PSE) file is saved, the pre-processor file will also be saved with the same filename but with extension H2P.

A geometry from another H2P file may be imported, see menu item *File/Import H2P*. The geometry will be added to the current drawn objects if menu item *File/Import H2P and add* is chosen. As an example a window frame may be added from a library of window frames to the current drawn wall section.

It is also possible to explicitly export the geometry to a new H2P-file, see menu item *File/Export H2P*.

The H2P file contains the geometry and the material name for each drawn object. This means if the thermal properties is changed in the material editor for a certain material name, the new values will be applied when updating in the pre-processor occurs.

5.1.14 Drawing areas with internal modifications

Do as follows to draw areas with internal modifications. Choose the modification type in the material pick list, see Figure 5.5, 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 boundaries, see Figure 4.28). In this case, three pipes of a given heat effect are placed. This effect is by default 1 W/m and can be changed in menu item *Input/Types*. Note the generated mesh that will approximate the circular pipes. Figure 5.6 shows an example of isotherms for three pipes with given heat sources. The thermal conductivity and the heat capacity for the area with an internal modification will be the same as the material where it is drawn upon.

The internal modifications are shown in white or just outlined (transparent) depending on the modification type.

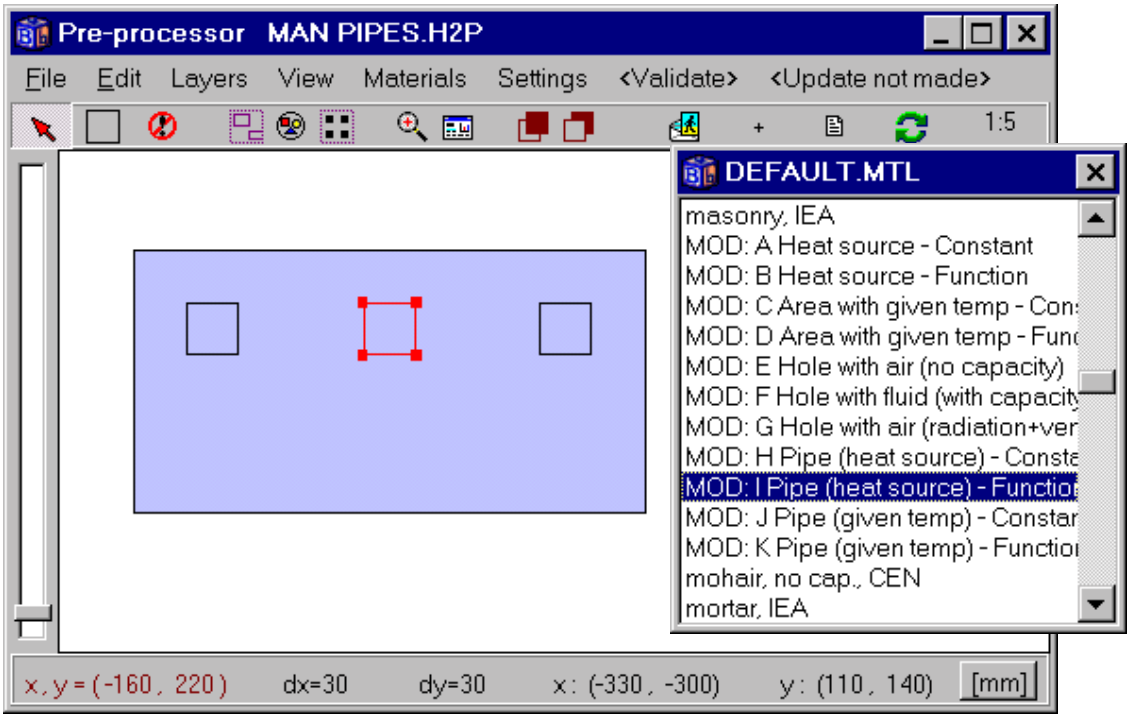
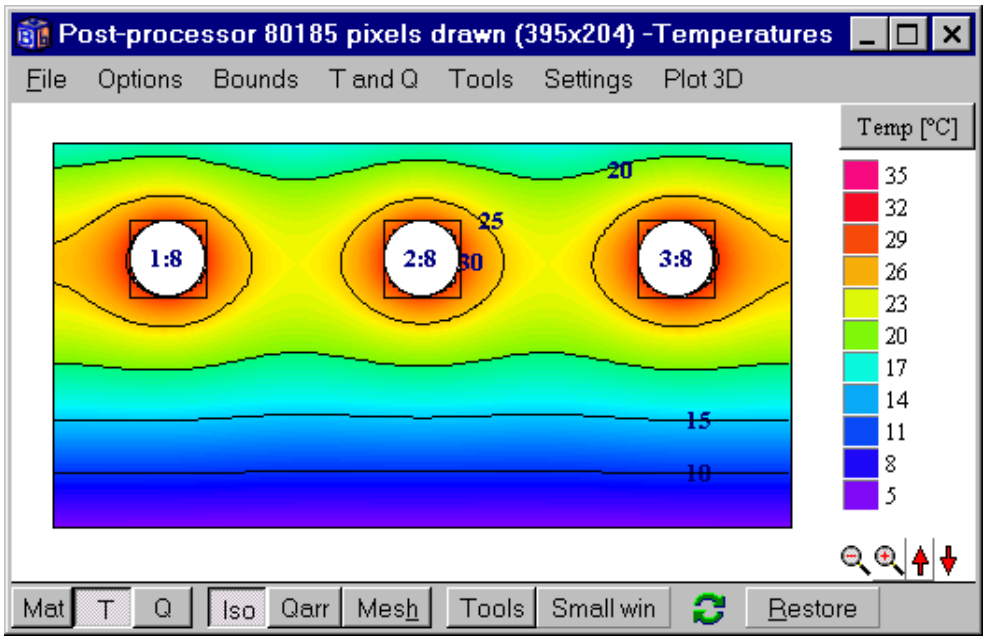


Figure 5.5: Internal modifications is drawn by picking any of the MOD types in the material pick list.



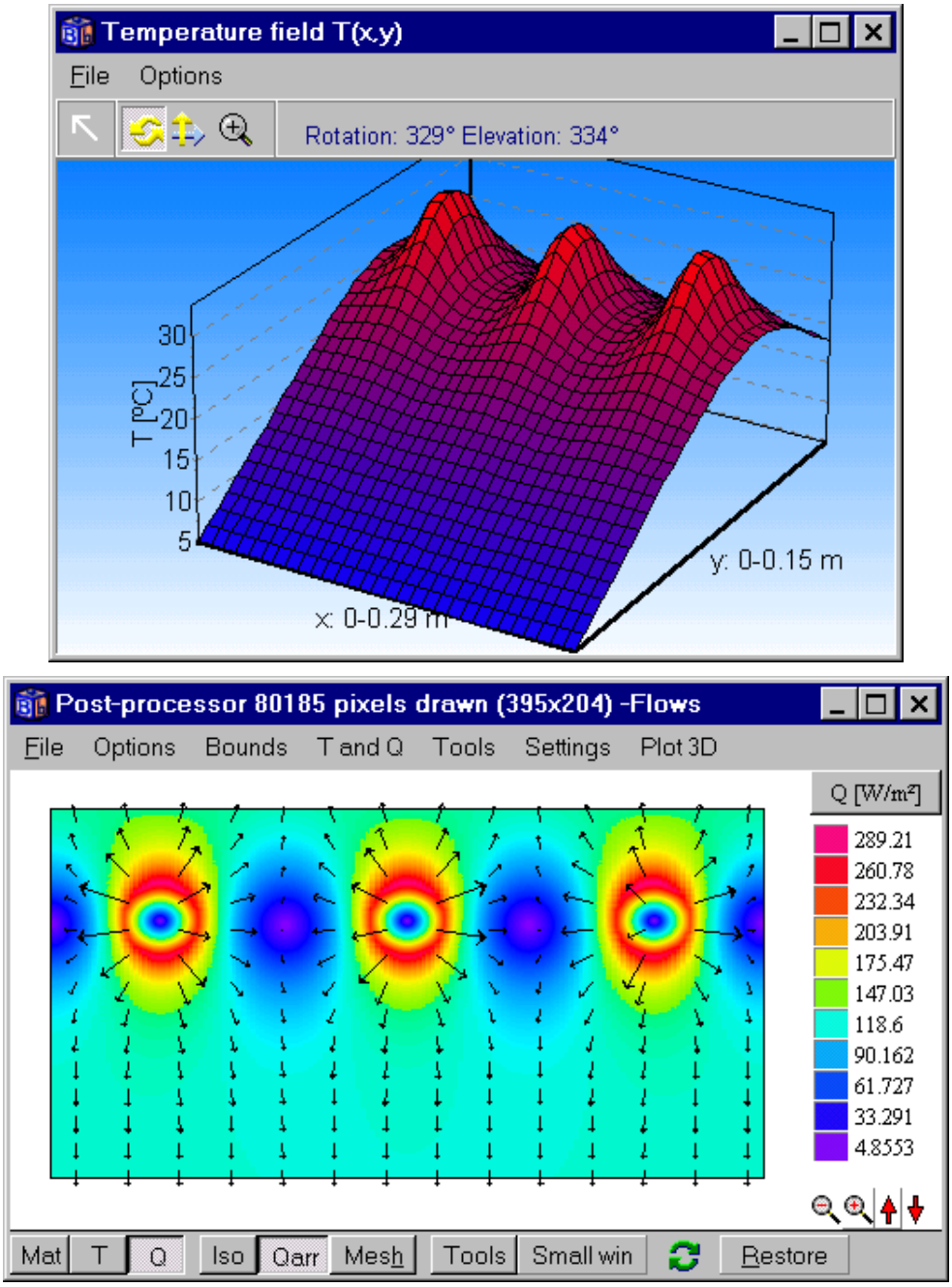


Figure 5.6[col]: The top and middle figures show temperatures, bottom figure shows heat flows.

5.2 The material editor

5.2.1 Introduction

Material properties may easily be edited and added. Several material list are available. The default list (Default.mtl) contains about 200 common building materials. The list General.mtl has over 1200 defined materials. Another file with over 200 materials (in German) from the German standard DIN (Deutsches

[col] See Appendix D for color image

Institut für Normung, DIN V 4108-4) is available. **See www.blocon.se or www.blocon.com for material file updates.** The material properties are taken from the following sources:

1. A.L. Edwards, A Compilation of Thermal Property Data for Computer Heat Conduction Calculations, UCRL-50589, February 24, 1969.
2. CEN (European Committee for Standardization). These materials are marked with "CEN".
3. 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. These materials are marked with "IEA".
4. German standard DIN (Deutsches Institut für Normung, DIN V 4108-4). No heat capacities (use only for steady-state calculations).

In some cases heat capacities are not given (marked with "No Cap."), and the heat capacity is set to 1. These materials should only be used in steady-state calculations.

Figure 5.7 shows the material editor (menu item *Material/Edit materials* in the pre-processor). New materials can be inserted into the list. Existing properties may be edited. Materials can also be deleted from the list.

Name, thermal properties, color, and brush may be edited for each material. Note that the color and brush does not have to be unique. This means that it is possible to add new materials that will have the same color and brush as existing ones. It may be good idea to give a color for a material that are similar to other colors of materials of the same kind. E.g. concrete, lightweight concrete and cellular concrete can be defined with a bluish nuance.

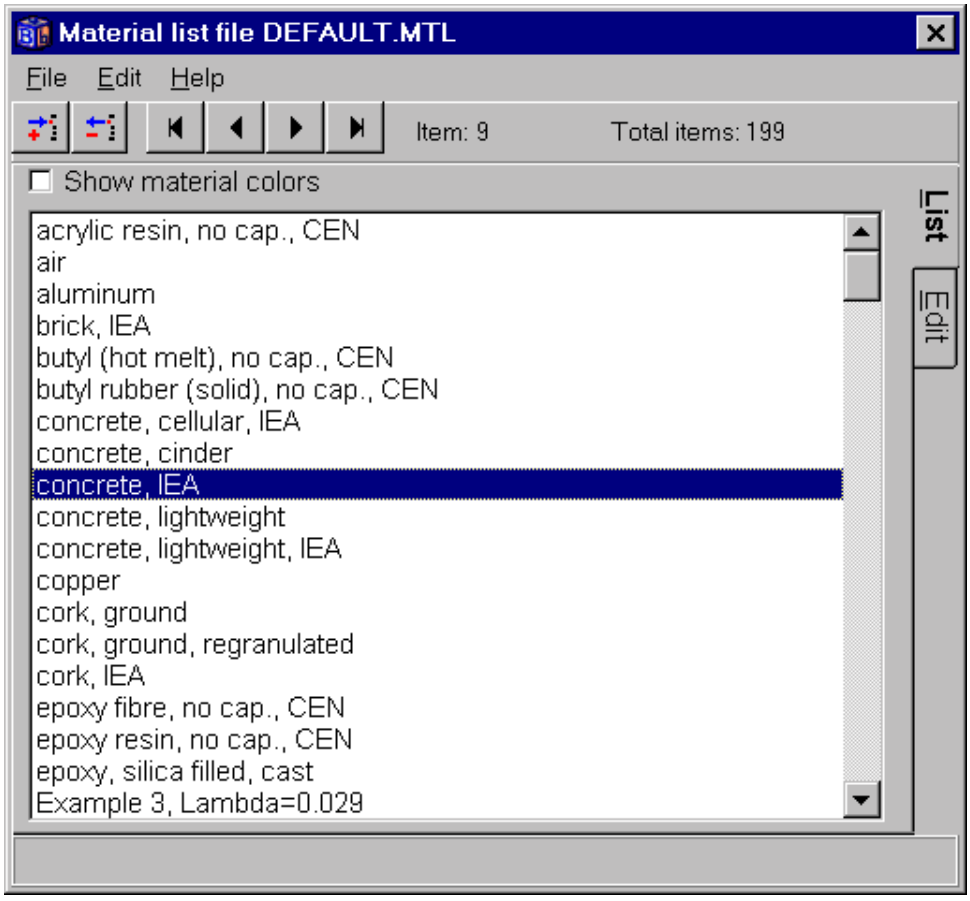


Figure 5.7: The material editor.

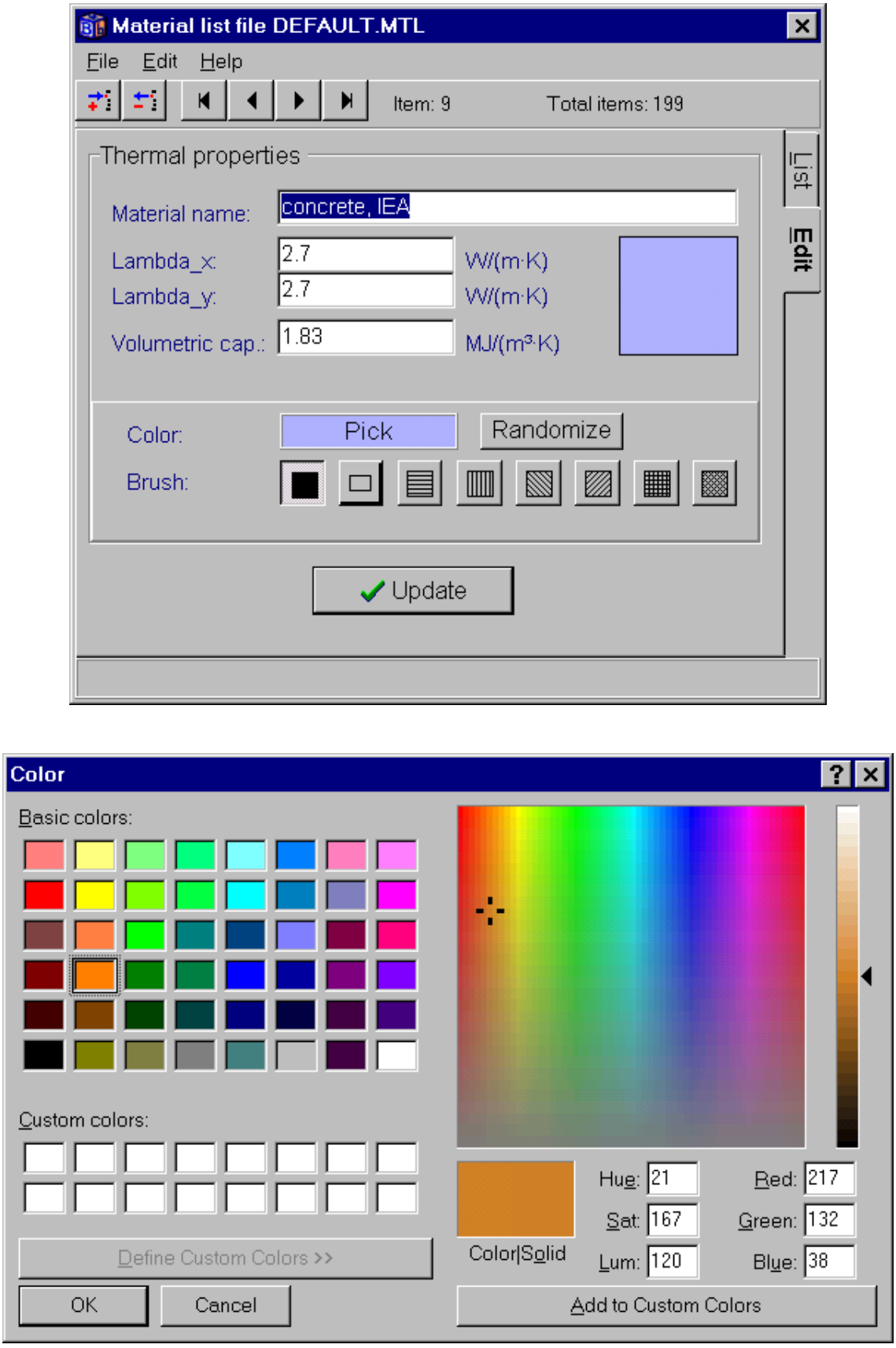


Figure 5.8: Name, thermal properties, color, and brush may be edited for each material.

5.2.2 Saving material files

Usually the user will add new materials (or edit existing ones) to DEFAULT.MTL. This material file will be loaded whenever HEAT2 is started. It is of course also possible to save or open other material files. Any new file should be saved (menu *File/Save* or *File/Save As*) in the same directory folder as HEAT2.EXE is located in.

The input data file (DAT or PSE) will contain the material file name if it is not the default one (DEFAULT.MTL). In this case, the material file will be loaded when the input data file is loaded.

If material properties are changed in the editor but not saved to file, the changes will only affect the current session of HEAT2.

Every time updating occurs (<*Update*> in the pre-processor), the material names of each drawn rectangle will match its name in the material list and take the current material properties. This means that the current material properties will be used, even if it is an old input data file. If the list does not contain the material name (e.g. if a rectangle is drawn and then the material name is deleted from the list), a warning will be issued and the drawn rectangles without a valid reference will be marked in black

5.2.3 Importing and merging other material files

Instead of adding materials one by one in the material editor, different file formats may be loaded and merged into the current material list. Material properties will not be over-written if the material name already exists in the list. Materials can be loaded from another MTL-file and added to the current MTL-file (menu item *File/Merge with material file*). There are also two text formats that can be imported as follows;

Type 1 (Menu item *File/Merge with text file format 1)***:*

The material name should be defined in the first 40 columns, then comes volumetric heat capacity C (J/(m³·K)), and the thermal conductivities λ_x, λ_y, (W/m·K). If λ_y is not given, λ_y will be equal to λ_x.

Example:

My material name one 1 0.1

My material name two 2 0.1 0.2

Type 2 (Menu item *File/Merge with text file format 2)***:*

Material name defined in the first 40 columns, then density (kg/m³), the specific heat capacity c_p (J/(kg·K)), and the thermal conductivity λ_x (=λ_y), (W/(m·K))

Example:

My material name one 1 1 0.1

My material name two 2 2 0.2

5.3 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 restart and continue 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.

5.4 Inserting mesh coordinates

Mesh coordinates are usually generated by the pre-processor. It is however possible to insert new mesh coordinates using text input. 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 (menu item *Input/Insert mesh point*, see Figure 5.9). 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). Mesh points can not be removed once they have been inserted.

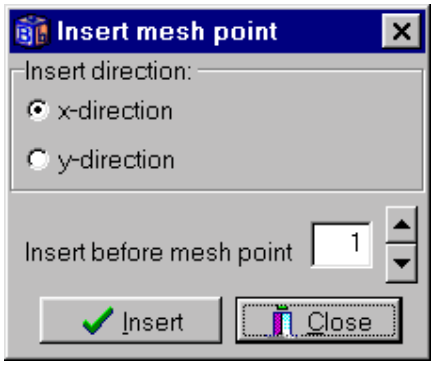
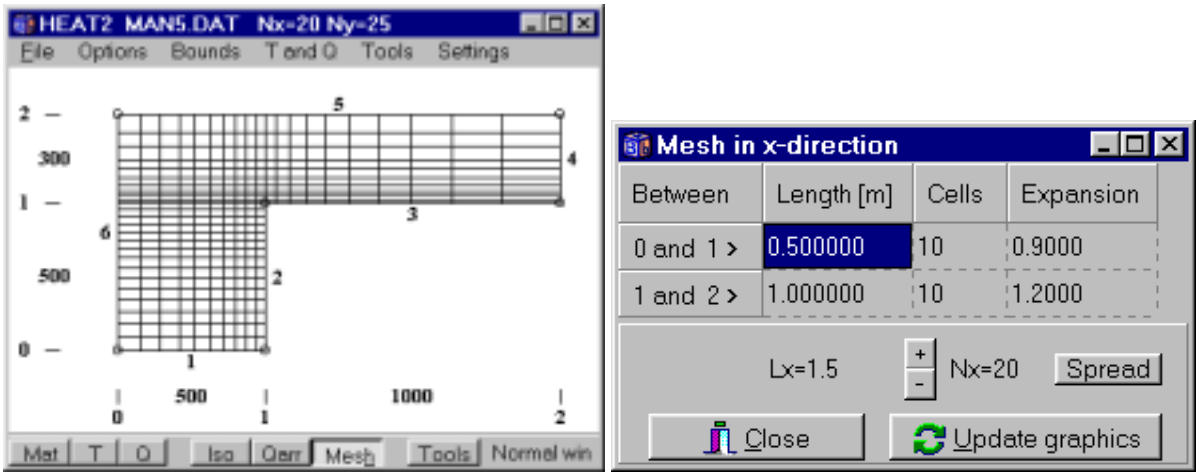


Figure 5.9: Menu for inserting mesh coordinates.

Consider Figure 5.10. 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.



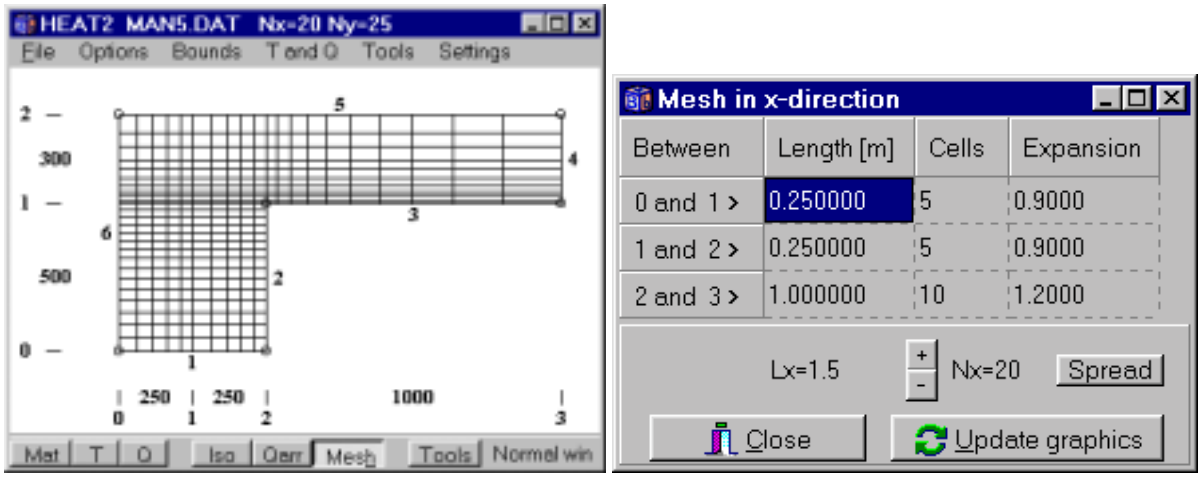


Figure 5.10: A mesh point has been inserted in the bottom figures in the *x*-direction before mesh point 1.

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 5.11.

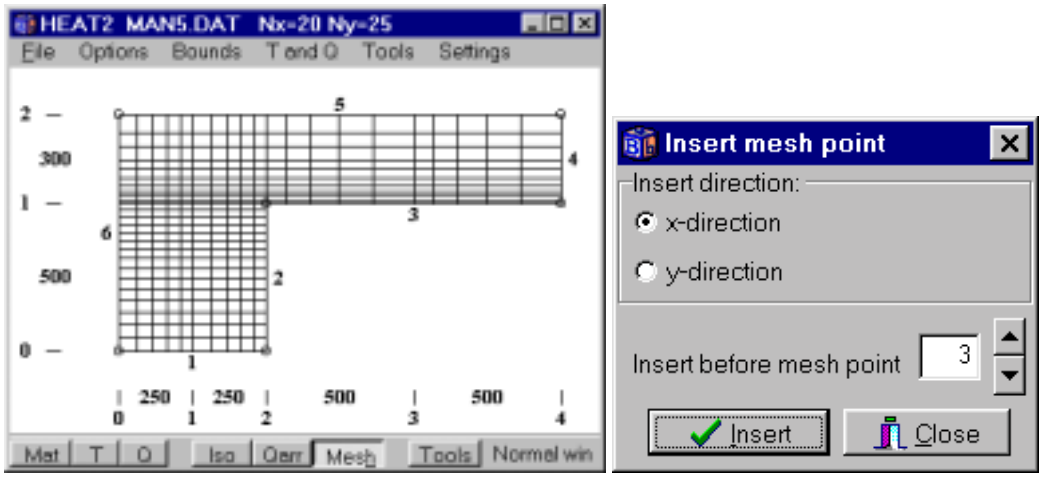
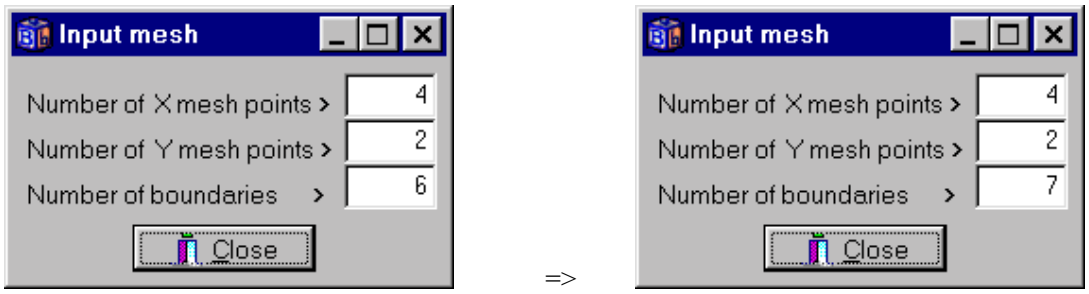


Figure 5.11: New mesh point inserted in the *x*-direction before mesh point 3.

Change the number of boundaries (*Input/Input mesh*) from 6 to 7 as in Figure 5.12 (top). Change the description of boundaries for boundaries 6 and 7 as in Figure 5.12 (bottom). There are now four segments in the *x*-direction, and seven boundary segments (delimited by small circles), see Figure 5.13. Segments 5 and 6 can now be assigned different boundary conditions.



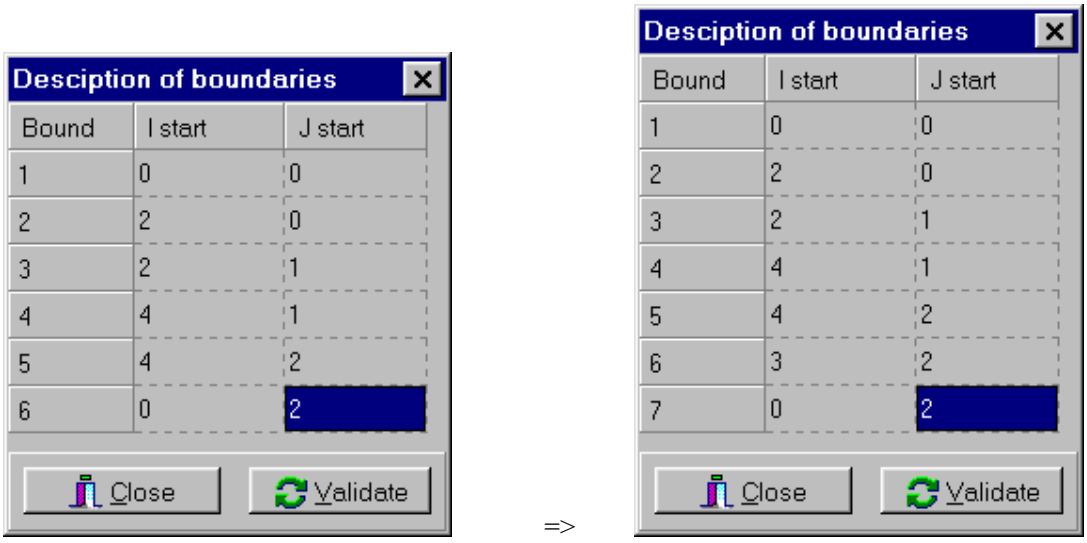


Figure 5.12: The number of boundaries is changed from 6 to 7. New descriptions of boundaries 6 and 7 are made.

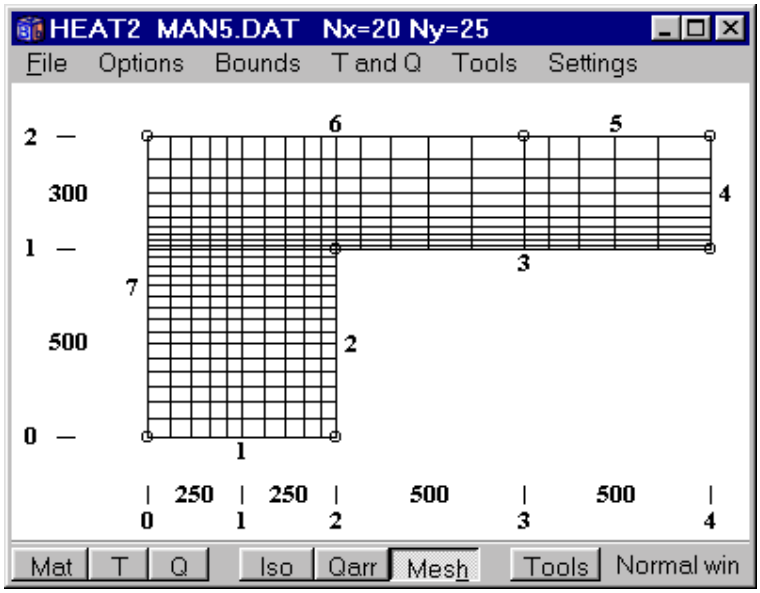


Figure 5.13: Segments 5 and 6 can now be assigned different boundary conditions.

5.5 Thermal properties

Thermal properties are usually generated by the pre-processor, but it can also be edited by text input. Figure 5.14 shows the window for thermal properties. See Section 4.7 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 (the data for the overlapping rectangle will prevail).

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 (MTRL50.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, (J/(m³·K)), which is the density ρ, (kg/m³), times the specific heat capacity c_p, (J/(kg)), i.e. $C=\rho \cdot c_p$. Note that C is given in the unit (MJ/(m³·K)), that is $1.0 \cdot 10^6$ J/(m³·K). The default value of C is 1.0 MJ/(m³·K). For the steady-state simulation the value of C is neglected.

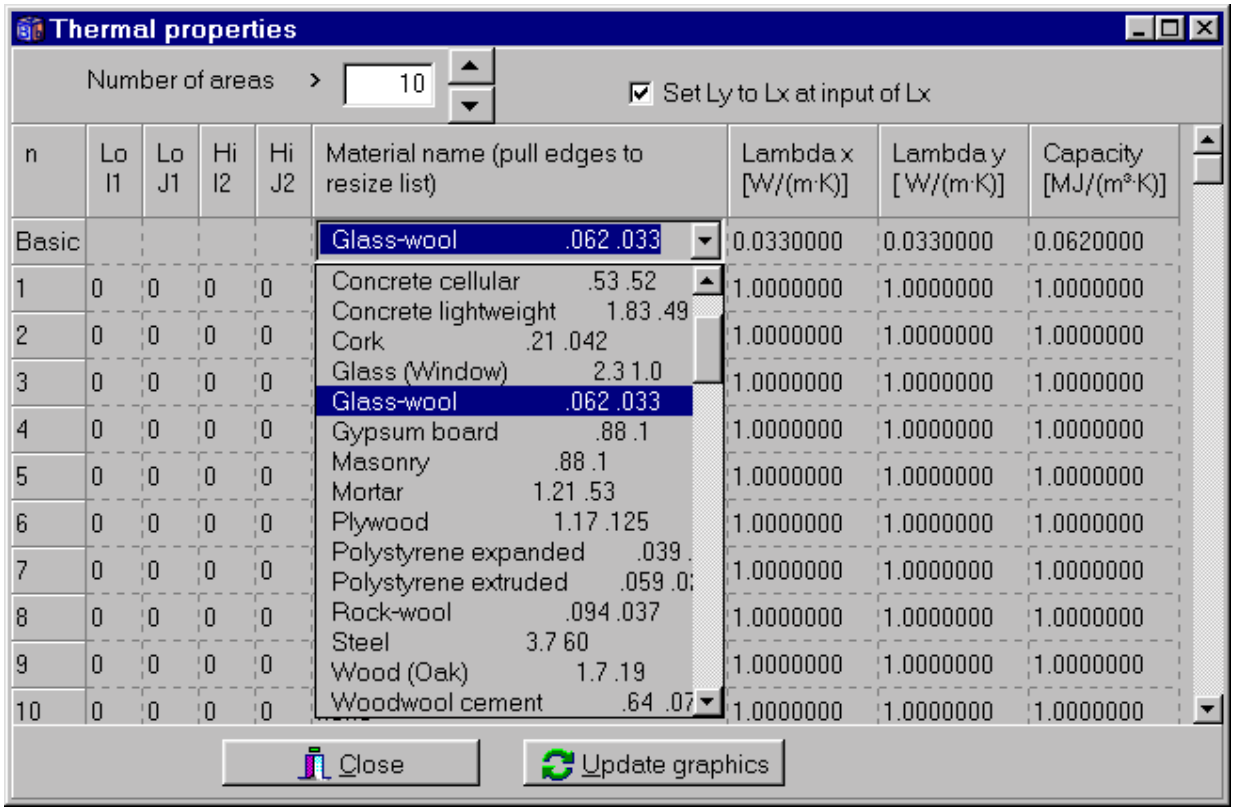


Figure 5.14: Data entry for thermal properties.

The thermal conductivity in the y-direction will be set equal to the one in the x-direction when the value in the x-direction is given if the upper right check box in Figure 5.15 is checked (this is default). Most materials are isotropic. Wood may be one exception.

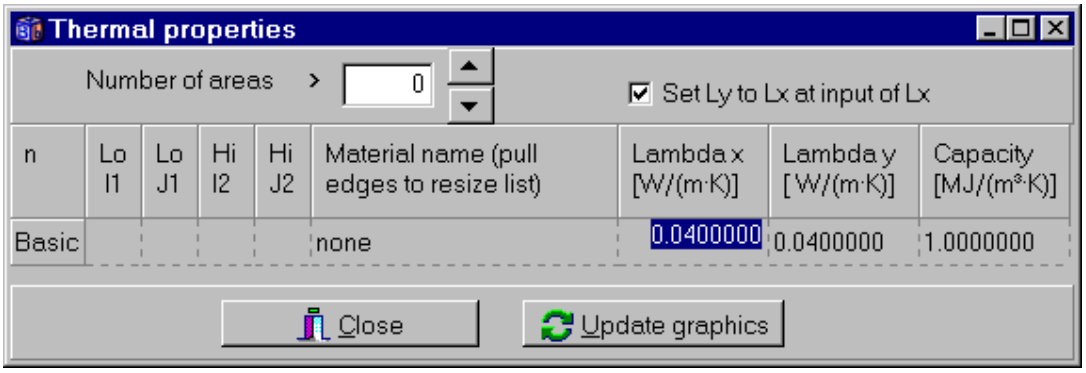


Figure 5.15: The thermal conductivity in the y-direction does not have to be explicitly specified when it is the same as in the x-direction if the upper right check box is checked.

5.6 Boundary conditions

A boundary condition type may either be a temperature T, (°C), with a surface resistance R, (m²·K/W), or a given heat flow Q, (W/m²). 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 5.9.

The user defines the boundary conditions in a list and assigns them to the boundary segments. Consider Figure 5.16. 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).

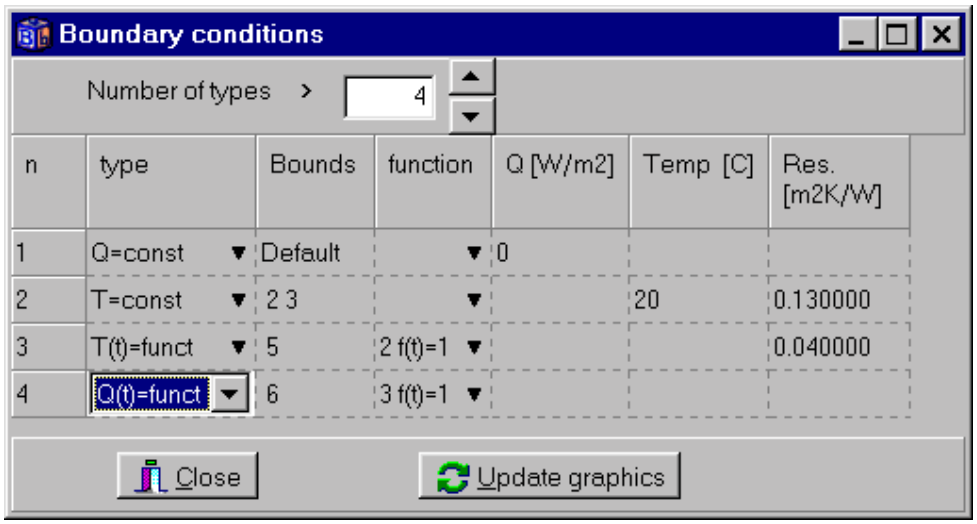


Figure 5.16: The boundary types are linked to each boundary segment.

A shorter input format for lists with consecutively enumerated items is available. Figure 5.17 shows the boundary conditions dialogue for the slab problem (see input file slab1.dat). The four boundaries in the second row may be given as the string “6-9” instead of the string of “6 7 8 9”.

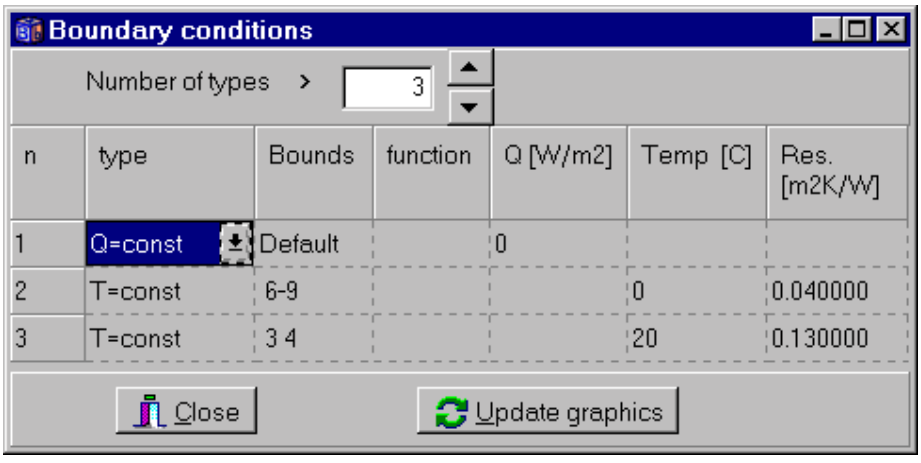


Figure 5.17: The consecutive boundaries in the second row may be given as the string “6-9” instead of the string of “6 7 8 9”.

5.7 Initial temperatures

See Section 4.8. In order to initialize the temperatures use *Solve/Reset Temperatures* or click on the *update graphics* button, see Figure 5.18.

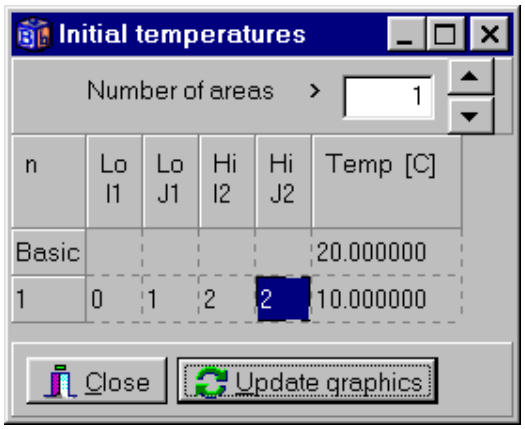


Figure 5.18: Initial temperatures.

5.8 Internal modifications

5.8.1 Introduction

Internal modifications may be used for simulating problems with e.g. floor heating, cavities with fluids, radiation inside cavities, etc. Figure 5.19 shows an example of a quadrate with 85 heat sources of 1 W/m placed in a checkerboard pattern. Each square is 4·4 mm².

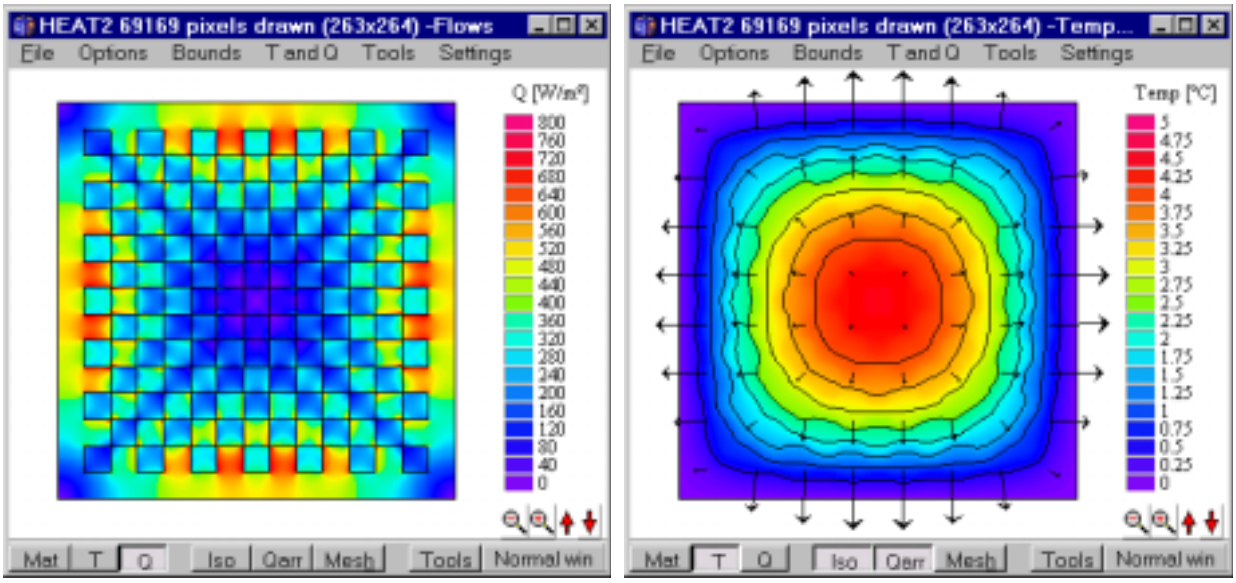


Figure 5.19[col]: Quadrate with 85 heat sources of 1 W/m placed in a checkerboard pattern. The boundary surface temperature is T=0. The left figure shows heat flow intensity. The heat flow is zero in the middle. The right figure shows temperatures, isotherms and heat flow arrays.

[col] See Appendix D for color image

5.8.2 Defining internal modifications types

Figure 5.20 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 5:2 shows the input data for the available modifications.

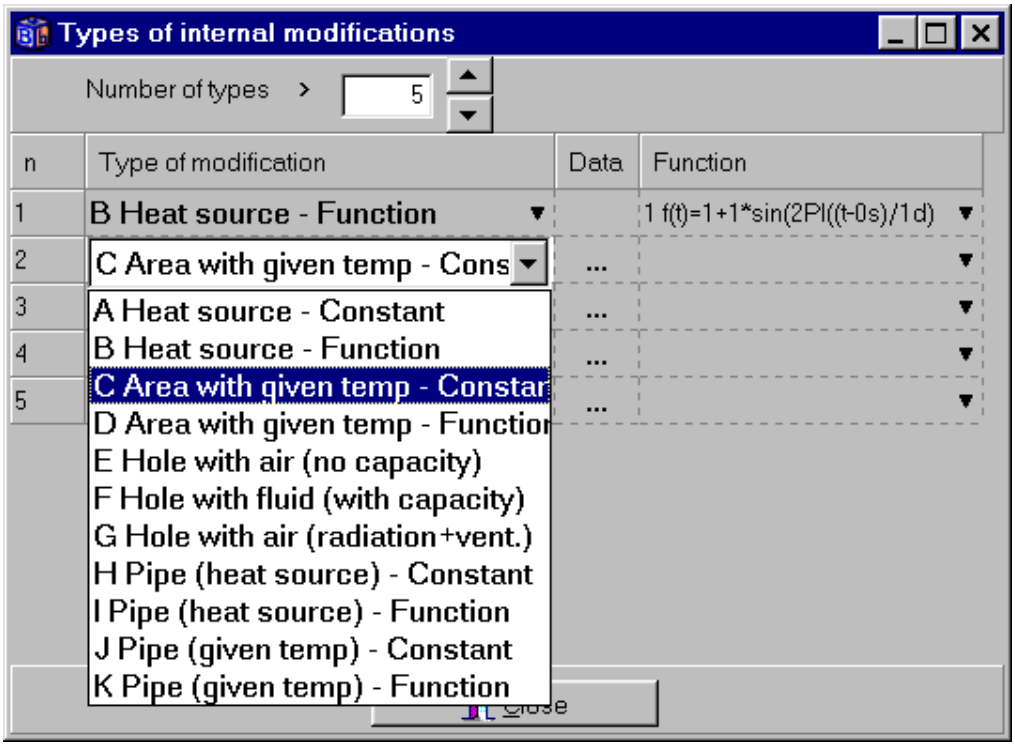
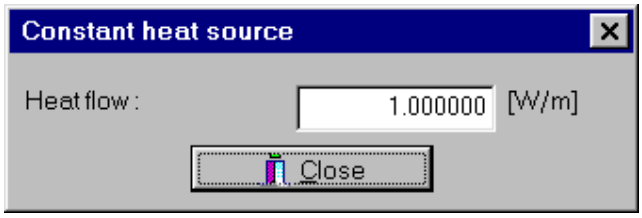


Figure 5.20: Internal modifications menu.

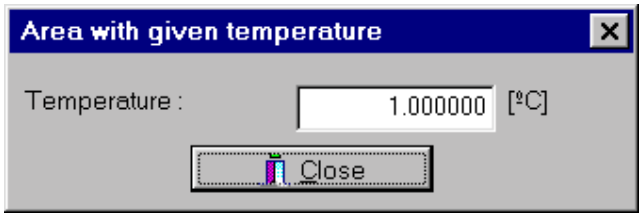
Type	Description
A	Constant heat source, (W/m)
B	Heat source that varies in time (a link is made to a pre-defined function)
C	Constant temperature
D	Temperature that varies in time (a link is made to a pre-defined function)
E	Air hole, see Sections 2.4 and 3.9. Input is the surface resistance, R, (m²·K/W).
F	Hole with a fluid with capacity, input is the surface heat transfer coefficient α, (W/(m²·K), the fluid's volumetric capacity C, (MJ/(m³·K)), and the initial temperature of the fluid, T_0, (°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 $C=\rho \cdot c_p$, (kJ/(m³·K)). The default value of C=1.239 kJ/(m³·K) is for air in room temperature. The temperatures are linearized around a suitable temperature level $\widetilde{T}_{so}$, see Chapter 9.
H	Approximated circle with constant heat source (pipe), see Chapter 6.
I	Approximated circle with non-constant heat source (pipe), see Chapter 6.
J	Approximated circle with constant temperature (pipe), see Chapter 6.
K	Approximated circle with non-constant temperature (pipe), see Chapter 6.

Table 5:2: Available internal modifications.

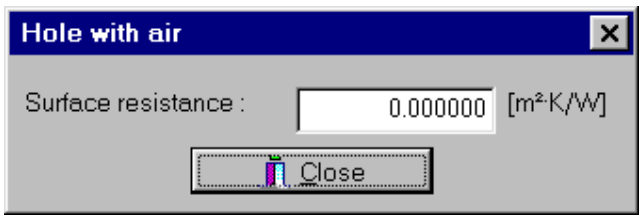
The data entry for all constant modifications is shown below. In the non-constant cases, a link to a function is made.



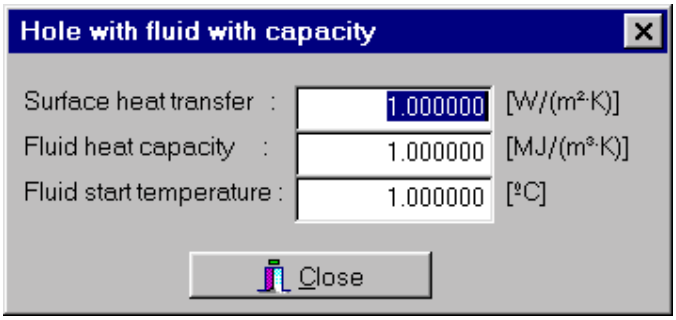
A: Heat source constant



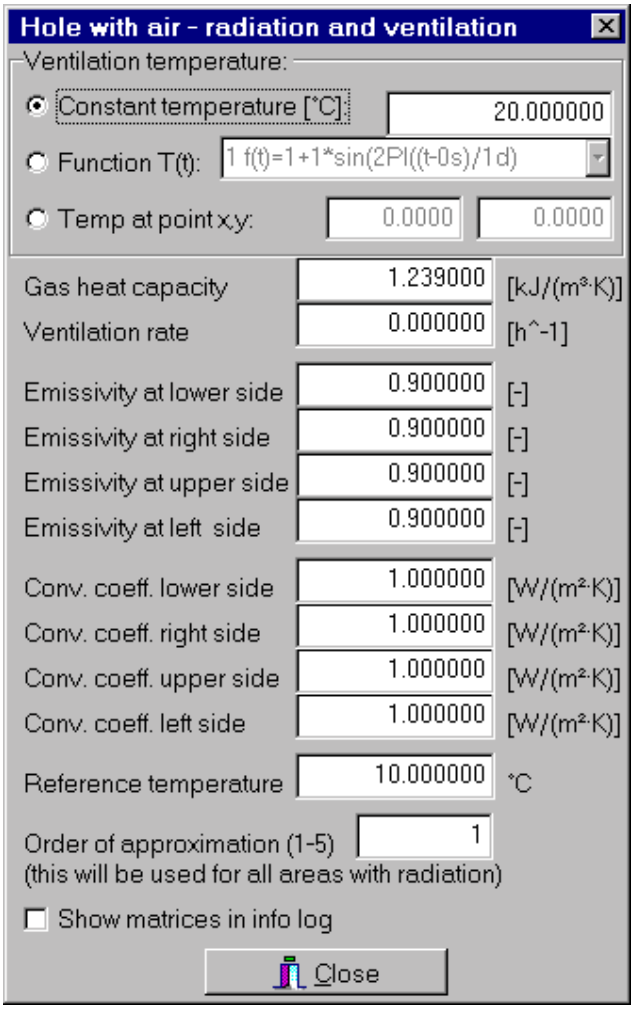
C: Temperature constant



E: Air hole



F: Hole with fluid with capacity



G: Hole with air, radiation and ventilation

Note on hole with air, radiation and ventilation (type G)

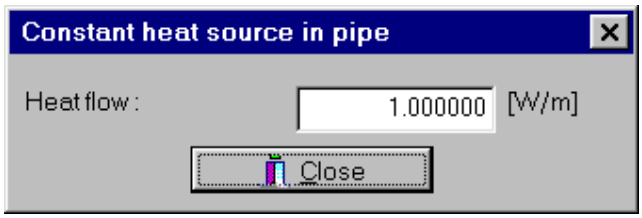
The internal modification with radiation in a cavity may be ventilated at a given rate with a constant temperature.

The ventilation temperature may also be a function chosen from the function list. An example of this may be a room ventilated with an outdoor temperature that varies sinusoidally.

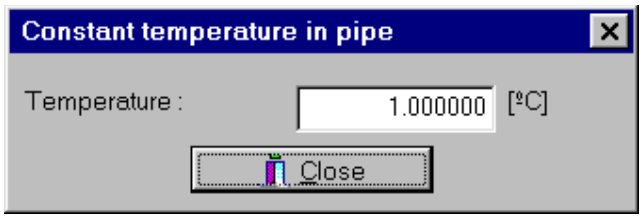
The cavity may also be ventilated with a current calculated temperature at an arbitrary point (x,y). The point must be located within the computational area. It could e.g. be inside another cavity.

Note that heat is not extracted/inserted at the specified point. It is only a prescribed ventilation temperature for the ventilated cavity.

The order of approximation for the radiation calculation may be given here, see Chapter 9. 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).



H: Pipe heat source



J: Pipe given temp

5.8.3 Defining areas

Areas with internal modifications are normally given using the pre-processor, but it can also be edited by text input.

Figure 5.21 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.

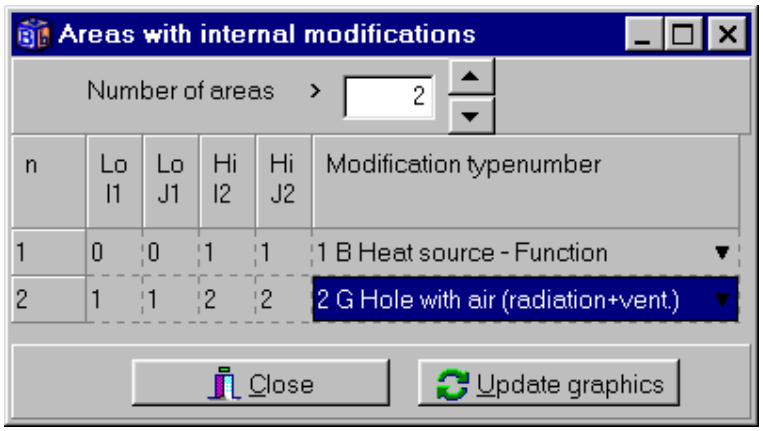


Figure 5.21: Each area is defined by the lower left and upper right input mesh coordinates. A link is made to one of the earlier defined modifications available in a pick list.

5.9 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:

- $f(t) = f_1 + f_2 \cdot \sin\left(\frac{2\pi(t - t_0)}{t_p}\right) \quad t_p > 0$ *(sinusoidal)*

- $f(t) = \begin{cases} 0 & t < t_1 \\ f_1 & t_1 \le t < t_2 \\ f_2 & t_2 \le t < t_3 \\ \vdots & \\ f_n & t_n \le t \end{cases}$ *(step-wise constant)*

- $f(t) = \begin{cases} f_1 \cdot t_1 & t < t_1 \\ f_1 + (f_2 - f_1) \cdot \frac{(t - t_1)}{(t_2 - t_1)} & t_1 \le t < t_2 \\ f_2 + (f_3 - f_2) \cdot \frac{(t - t_2)}{(t_3 - t_2)} & t_2 \le t < t_3 \\ \vdots & \\ f_n & t_n \le t \end{cases}$ *(step-wise linear)*

5.9.1 Sinusoidal function

Figure 5.22 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.

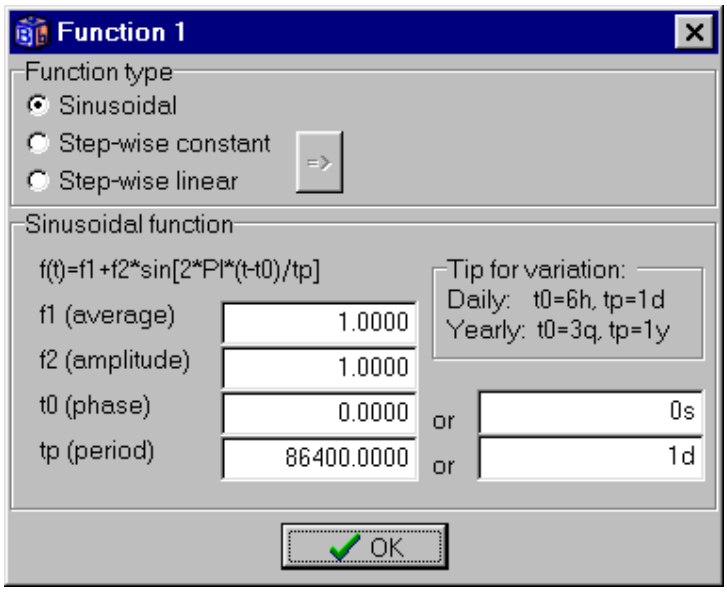


Figure 5.22: Functions are given in item *Solve/Function*.

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

y	year (365 days, 31536000 seconds)
q	month (year/12, 30.417 days, 2628000 seconds)
d	day (86400 seconds)
h	hour (3600 seconds)
m	minute (60 seconds)
s	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

5.9.2 Step-wise constant and step-wise linear function

The step-wise constant and step-wise linear values are given in an editor, see Figure 5.23. 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 5.23 (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 5.28 and Figure 5.29. 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.

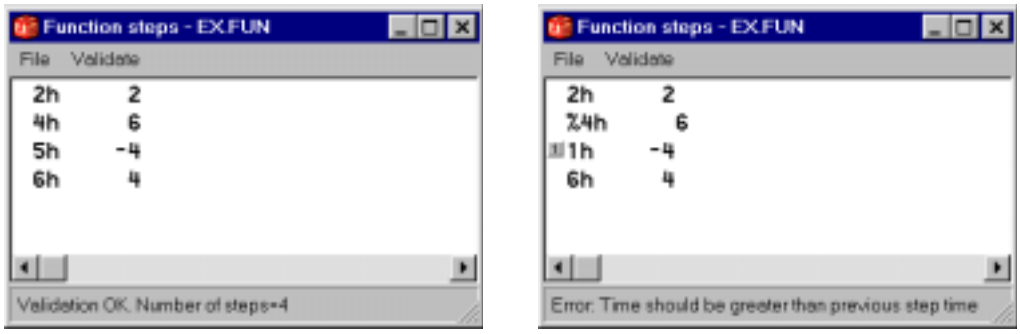


Figure 5.23: Data for function steps are given in an editor. The input is validated and a marker will be shown if there is an error.

The function values may be generated in other programs and pasted into the function editor. Figure 5.24 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 5.15.

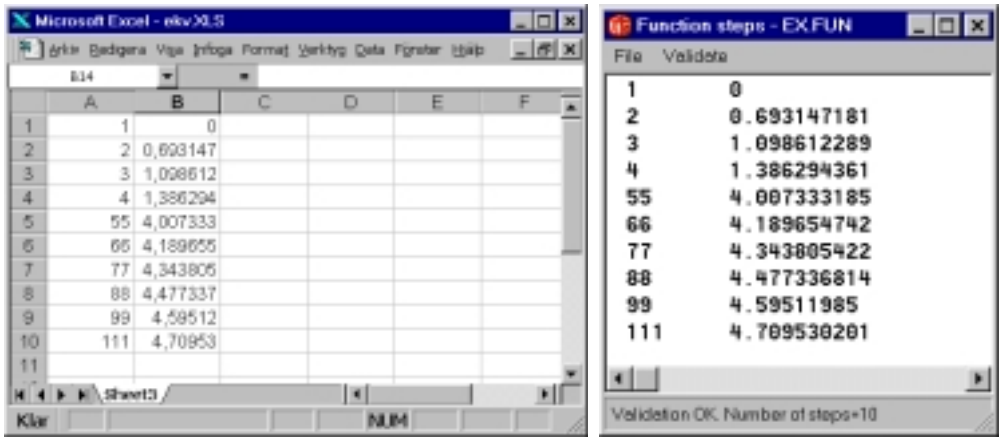


Figure 5.24: Arbitrary function step values may be pasted from other programs. Here is the function *ln(x)* created in Microsoft Excel for a few values. They have been pasted into the function editor (right).

5.9.3 Import from other file formats

Various formats with climatic data may be imported form menu item *Import climate file formats* in the *Function 1 step* window. Figure 5.25 shows these formats. The file may then be saved as an ordinary FUN-file.

These files normally has one year of hourly data. Do as follow to simulate several year using the same yearly data. Simulate the first year, then reset the calculation (menu item *Solve/reset calculation* or set the current time to zero in menu item Solve/Options for transient). Continue to simulate the second year, reset the time, and so on.

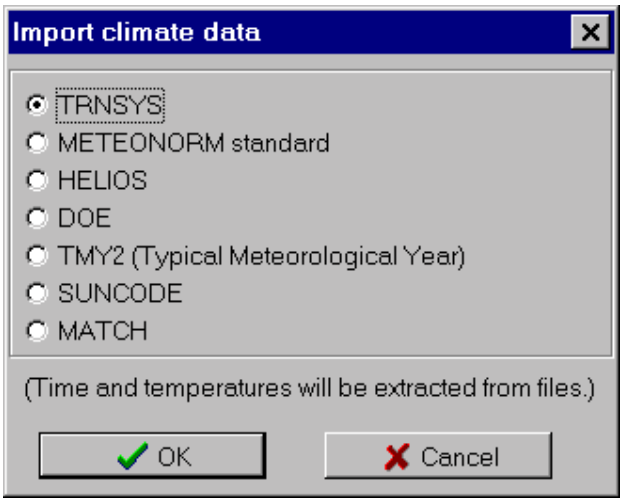


Figure 5.25: Time and temperatures can be extracted from several climatic files.

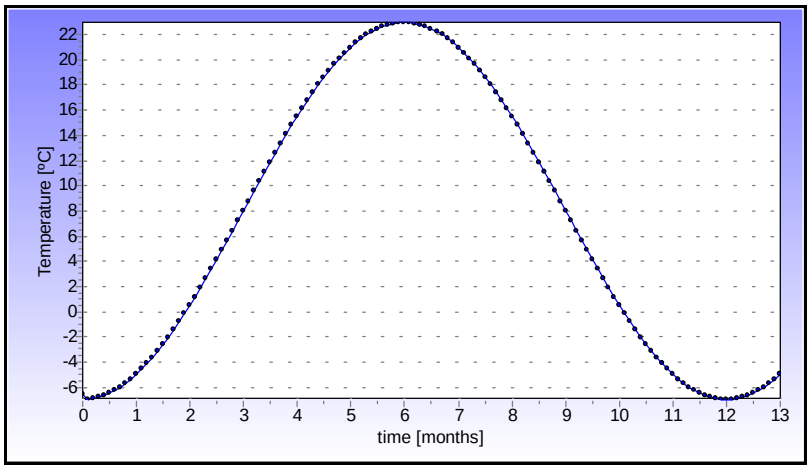
5.9.4 Example 1 - sinusoidal function with a time period of one year

Figure 5.26 shows a sinusoidal function with the average value 8 °C, the amplitude 15 °C, the time phase three months (-3q), and the time period one year (1*y*). With this time phase (-3*q* or +9*q*), the coolest temperature (-7 °C) will occur after 0, 12, …, 12*n* $(0 \le n \le \infty)$ months, i.e. on the 1:st of January every

year. The warmest temperature (23 °C) will occur after 6, 18, …, months, i.e. on the 1:st of July every year[1].

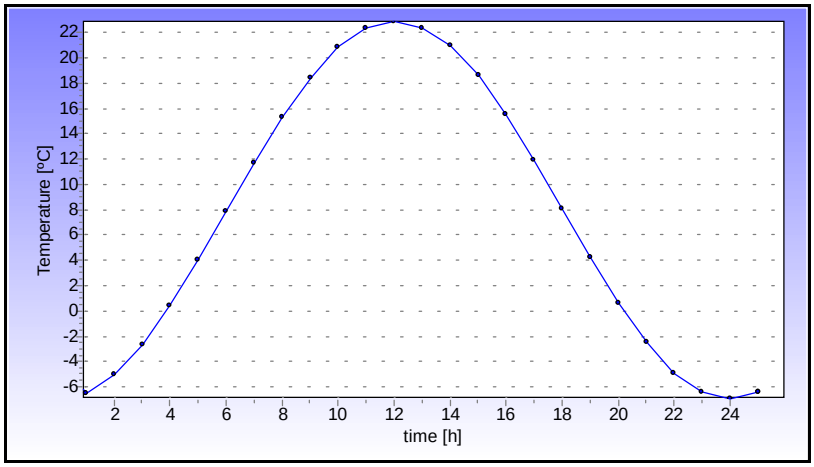
5.9.5 Example 2 - sinusoidal function with a time period of one day

Figure 5.27 shows a sinusoidal function with the average value 8 °C, the amplitude 15 °C, the time phase six hours (-6*h*), and the time period one day (1*d*). With this time phase (-6*h* or +18*h*), 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 5.26: Sinusoidal function with f_1=8, f_2=15, t_o=3*q*, and t_p=1*y*.



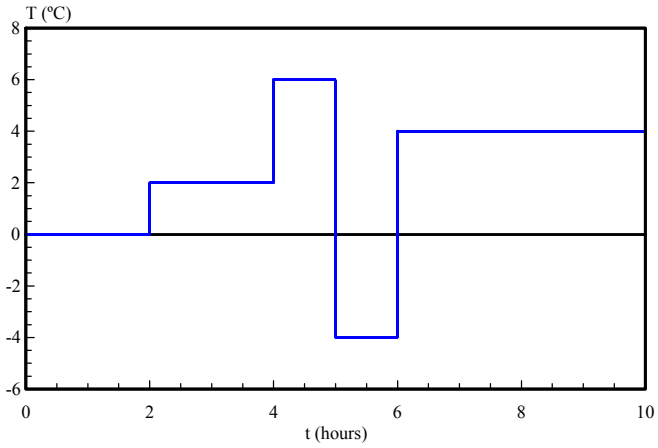
$$f(t) = 8 + 15 \cdot \sin\left(\frac{2\pi(t - 6h)}{1d}\right)$$

Figure 5.27: Sinusoidal function with f_1=8, f_2=15, t_o=6*h*, and t_p=1*d*.

[1] Using time phase –4*q* instead will give min and max at Feb 1 and Aug 1, respectively.

5.9.6 Example 3 - step-wise constant function

Figure 5.28 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=2*h*, 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.

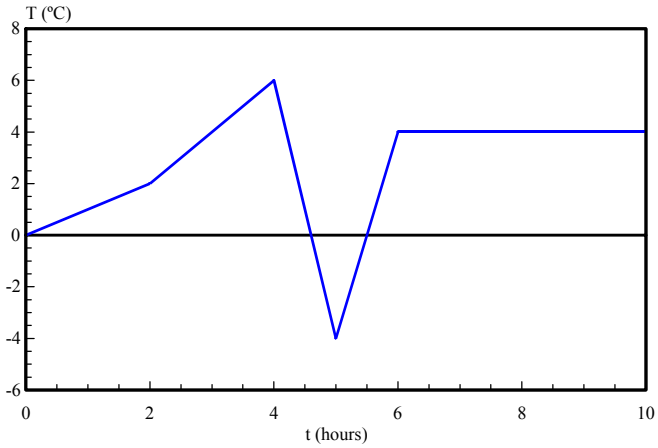


t_1 = 2*h*	f_1 = 2 °C
t_2 = 4*h*	f_2 = 6 °C
t_3= 5*h*	f_3 = -4 °C
t_4 = 6*h*	f_4 = 4 °C

Figure 5.28: Step-wise constant function for the four steps given to the right.

5.9.7 Example 4 - step-wise linear function

Figure 5.29 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.



t_1 = 2*h*	f_1 = 2 °C
t_2 = 4*h*	f_2 = 6 °C
t_3= 5*h*	f_3 = -4 °C
t_4 = 6*h*	f_4 = 4 °C

Figure 5.29: Step-wise linear function for the four steps given to the right.

5.10 Internal resistances

Figure 5.30 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.

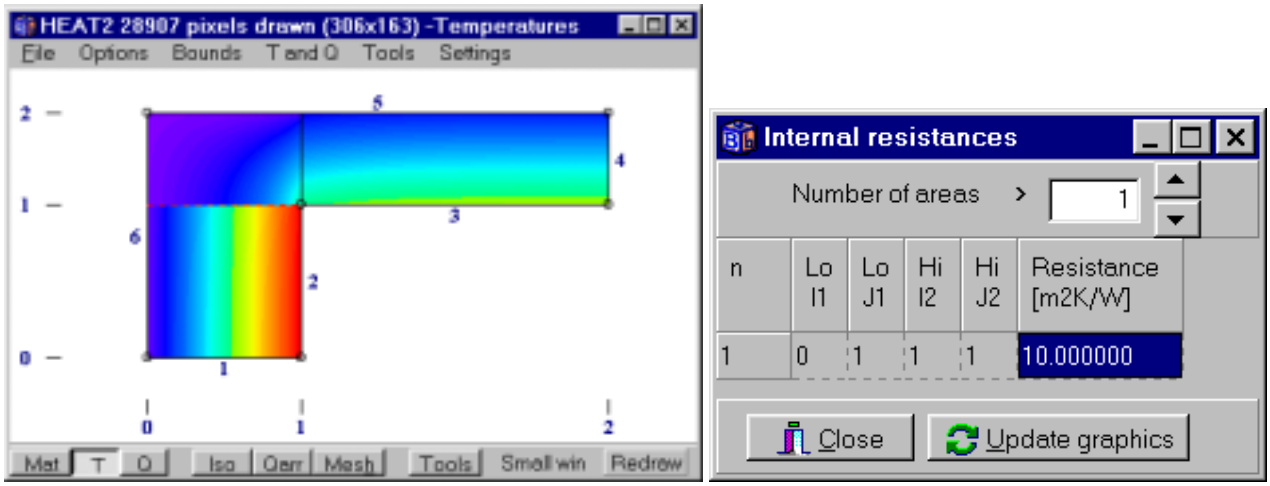


Figure 5.30: A vertical internal resistance of 10 m²·K/W has been inserted at the horizontal line defined by lower left (0,1) and upper right (1,1). The calculated temperature field is shown to the left.

5.11 Solving the problem

5.11.1 Steady-state stop criteria

There are three ways to give a stop criterion for a steady-state simulation, see Figure 5.31. 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. (5.1). 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.

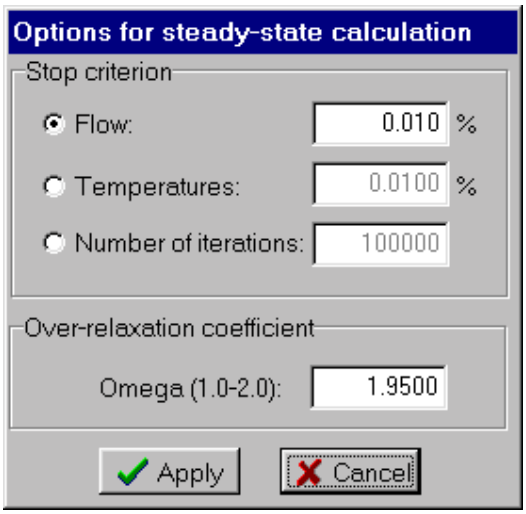


Figure 5.31: Steady-state stop criteria.

$$\frac{\sum Q_b}{\sum |Q_b|} \leq F \qquad (5.1)$$

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:

$$\left| 1 - \frac{T_{i,j,k}}{T_{i,j,k}^{new}} \right| \leq R \qquad (5.2)$$

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 *30 consecutive*[2] time-steps before the simulation stops.

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

5.11.2 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 ω=1.95.

5.11.3 Simulation time for transient calculation

Simulation stop time is given in *Solve/Options for transient calculation*, see Figure 5.32. 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 5.9.1. The current simulation time will be put to zero when a reset is made, see Section 5.11.6.

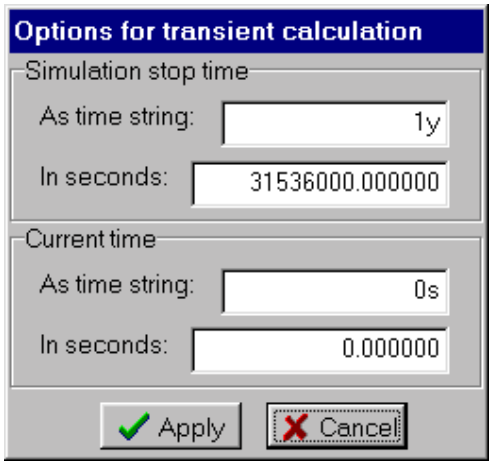


Figure 5.32: Simulation time and current time.

5.11.4 Simulation window

A window with information about the calculation is shown during the simulation. Figure 5.33 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 *5.12.1*, *Stop criterion,* see Sections 5.11.1 and 5.11.3, and to *Screen update,* see Section 5.11.5.

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

[2] It was sufficient if the criterion was fulfilled ten times in version 3.0 of HEAT2.

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.

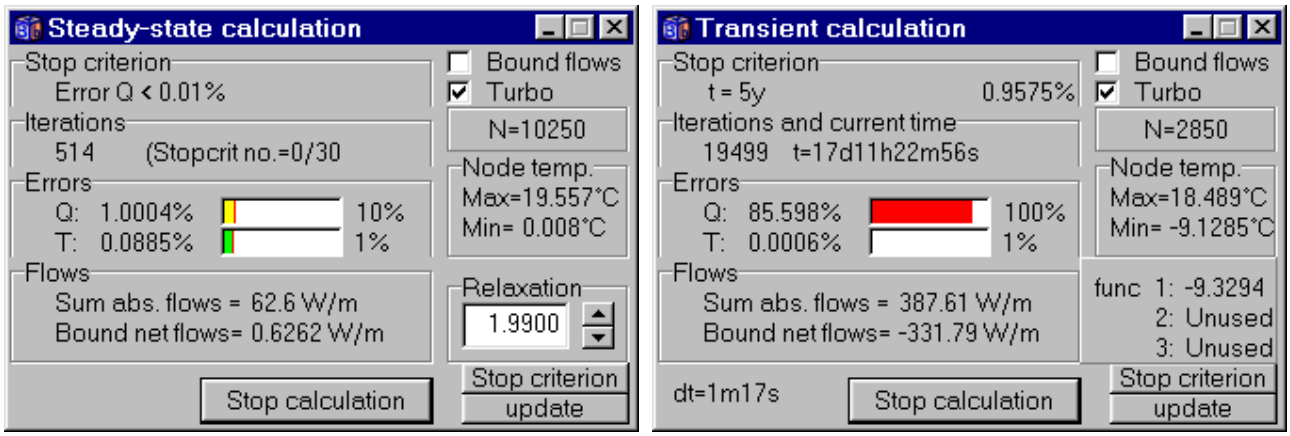


Figure 5.33: Windows shown during steady-state and transient simulations.

5.11.5 Screen update

Data shown during the calculation are updated as defined in item *Solve/Screen update*, see Figure 5.34. 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 5.15.

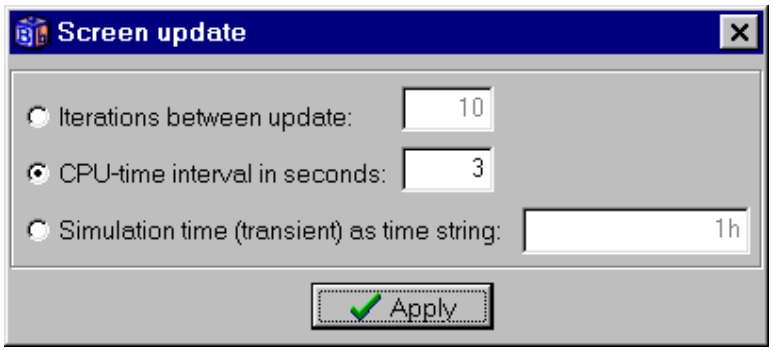


Figure 5.34: Data shown during the calculation are updated as defined here.

5.11.6 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.

5.12 Output

5.12.1 Boundary flows

Figure 5.35 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.

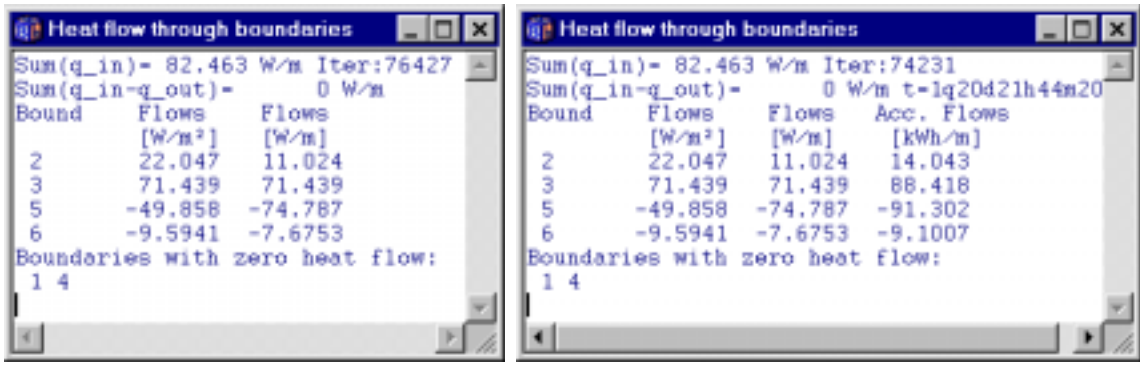


Figure 5.35: Windows showing boundary flows for steady-state (left) and transient problems (right). Data may be cut and pasted by right clicking in the window.

5.12.2 Heat flows for internal modifications

5.12.2.1 Example 1

Figure 5.36 shows the heat flows for a problem with one heat source (cavity 1) and an empty cavity with radiation (cavity 2). The boundary conditions are T=0 on the left surface, T=1 on the right surface and q=0 on the top and the bottom. The heat flow $\sum |Q_b|$, shown as **Sum.abs. flows** in the solve window, include the flows through the cavity boundaries (1.6613 W/m) and the external boundaries (0.679+1.679 W/m) and becomes 4.0193 W/m.

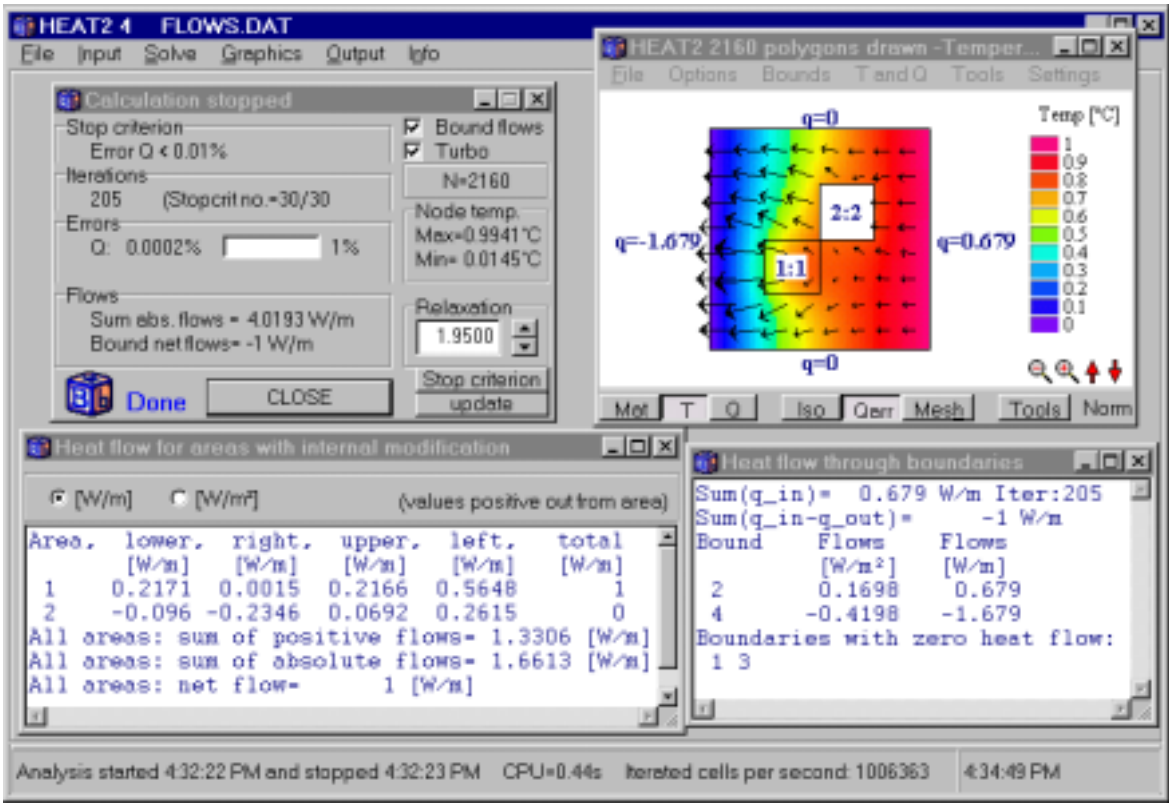


Figure 5.36: Calculated heat flows for a problem with one heat source (cavity 1) and an empty cavity with radiation (cavity 2).

5.12.2.2 Example 2

Consider Figure 5.37 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. Figure 5.38 shows the heat flow intensities.

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.

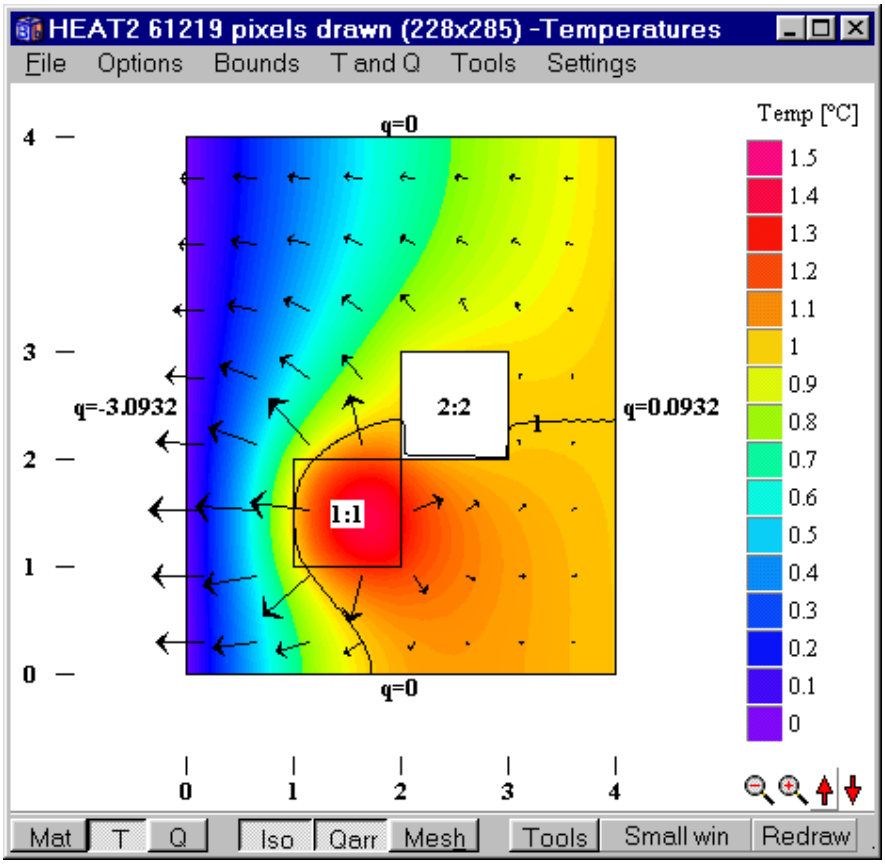


Figure 5.37[col]: Temperature field for a problem with two internal modifications. The first one is a heat source of 3 W/m, the second one is a cavity with radiation (the white field). The boundary condition on the left and right sides are T=0 with R=0.04 and T=1 with R=0.13, respectively.

[col] See Appendix D for color image

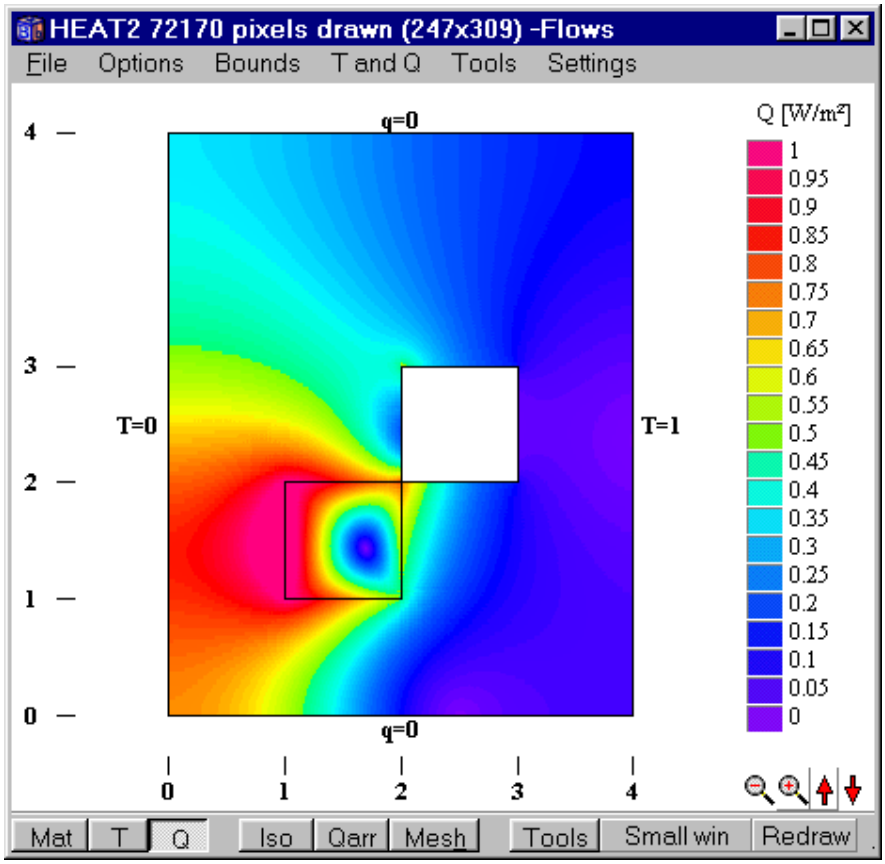


Figure 5.38[col]: Heat flow intensities.

Figure 5.39 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 5.40 shows the boundary flows. Note the absolute heat flow in the simulation window, see Figure 5.41. The sum of absolute heat flows (q=6.8402) involves the internal modifications (q=3.6539) and the boundary flows (q=0.0932+3.0932). The boundary net flows at the bottom in Figure 5.41 relate to external boundaries, see Figure 5.37, (i.e. q=0.0932-3.0932=-3 W/m).

[col] See Appendix D for color image

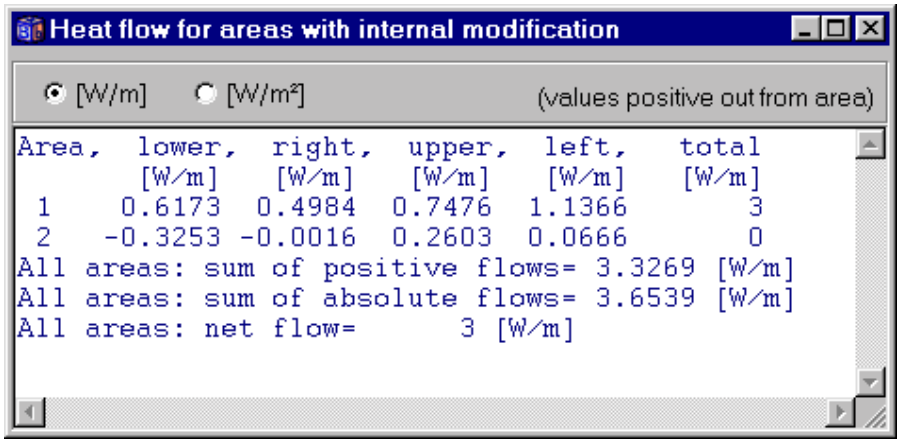


Figure 5.39: Heat flows for internal modifications.

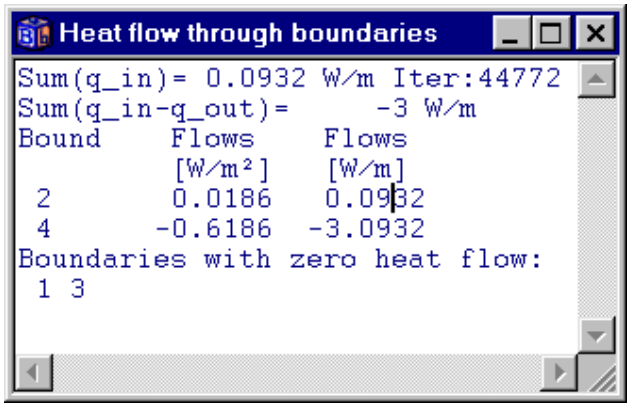


Figure 5.40: Boundary flows.

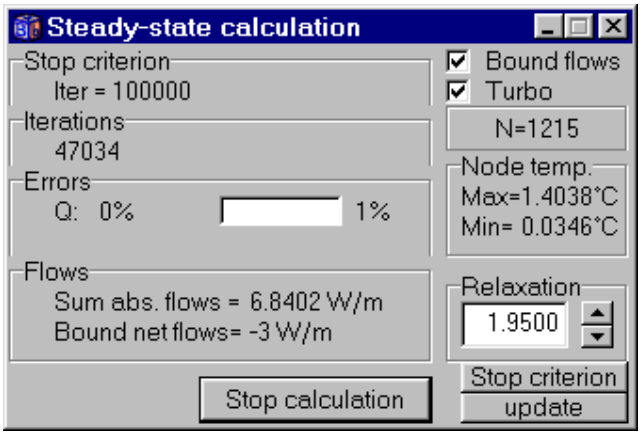


Figure 5.41: Heat flows in the simulation window.

5.12.3 Results for radiation

Figure 5.42 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 9.3), 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).

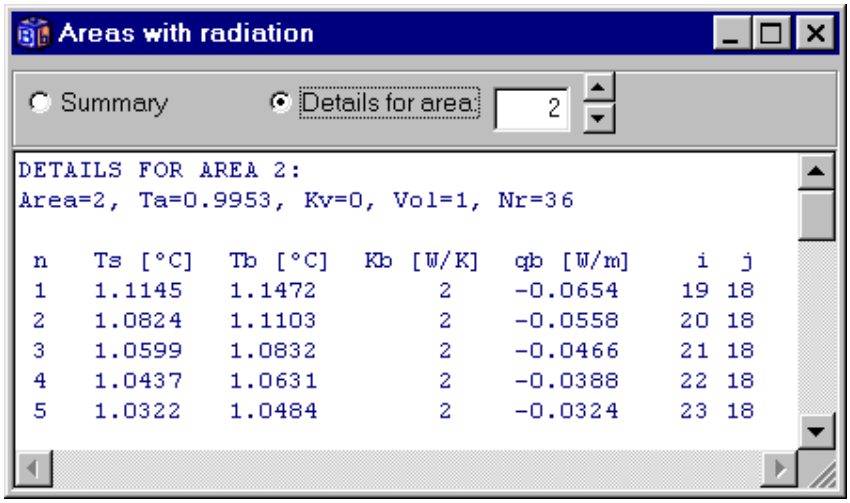


Figure 5.42: Results for areas with radiation.

5.12.4 Temperatures and heat flows at a point

Menu item **Output/temp at point** shows the temperature, heat flow, and the numerical cell indices at arbitrarily chosen points (x, y), see Figure 5.43.

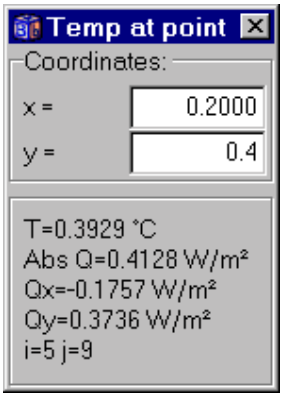


Figure 5.43: **Temp at point** shows the temperature, heat flow, and the numerical cell indices at arbitrarily chosen points (x, y).

The temperature T, (°C), and the heat flow components Q_x, Q_y, (W/m²), may also be obtained for any arbitrarily chosen point (x,y) by positioning the cursor at the desired location when the *Tools* button is selected, see Figure 5.44. 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.

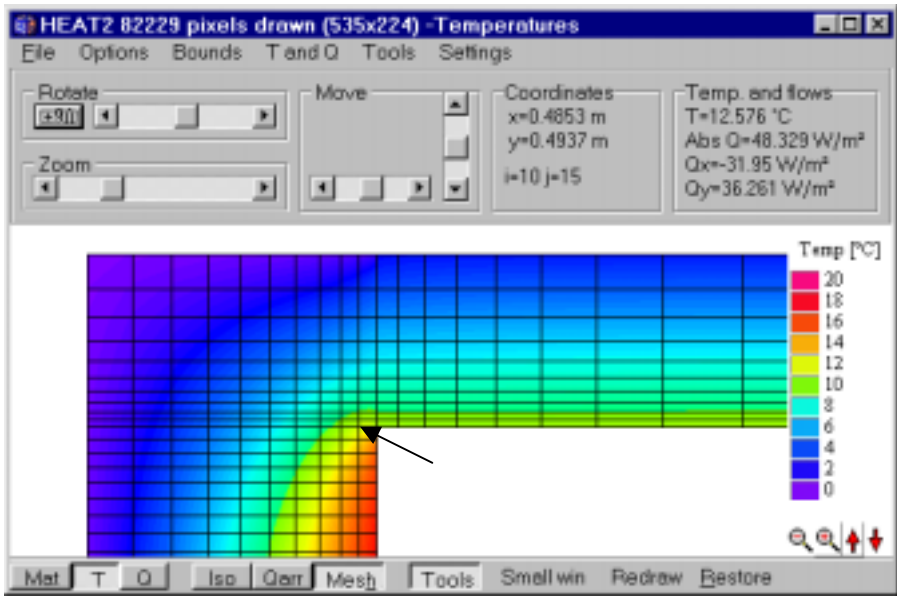


Figure 5.44: Temperatures and flows may be obtained for a point (*x*,*y*).

5.13 Conductances and capacities to file

Conductances and capacities may be written to a text file, see menu item **Output/Conductances and capacities to file**. The format is shown below. The first line contains a headline. The following rows show the index, four conductances, and heat capacity for each cell:

$$i,j \qquad K_{i-1/2,j} \quad K_{i+1/2,j} \quad K_{i,j-1/2} \quad K_{i,j+1/2} \quad C_{i,j}$$

5.14 Temp (MATLAB) to file

The temperatures are written to a file that may be read by MATLAB version 4.0 and forward. Three files will be produced with the extensions *Filename.X*, *Filename.Y*, and *Filename.Z*. The *Z* file contains the cell temperatures (a matrix with $m \cdot n$ values). The files *X* and *Y* hold the coordinates to the middle of the cells (array with *m* and *n* values, respectively).

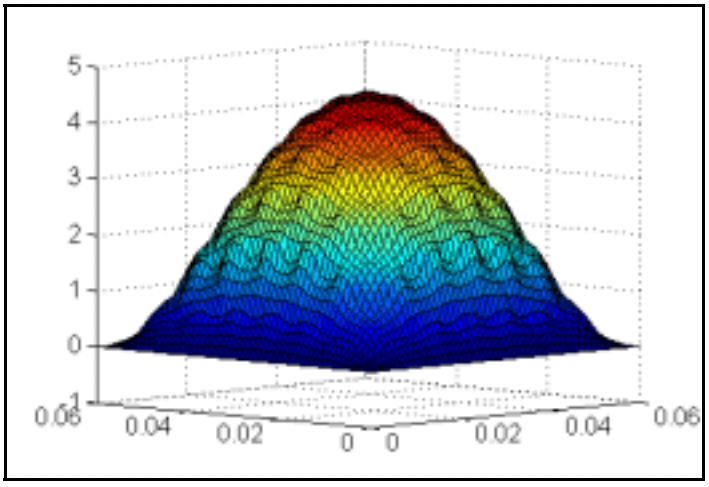


Figure 5.45: Mesh image by Matlab showing temperatures for the problem with 85 heat sources of 1 W/m, see Figure 5.19.

HEAT2 will also make a MATLAB script file with the name *filename.M*. This script can be executed directly in MATLAB which will produce a plot, see Figure 5.45. MATLAB may produce different kind of graphs, such as contour and surface plots. See the MATLAB manuals for further information.

5.15 The recorder

Output values may be saved to a list during the simulation. The recorder can save up to 25 different values as a function of time or number of iterations. What to be recorded is chosen from the pick list in Figure 5.46. It is possible to combine different types, e.g. temperatures or heat flows at 20 locations, and 5 columns that record the heat flow through 5 different boundary segments. A recording will only be made during the simulation if **Record enabled** is checked.

The values will be recorded at each screen update (*Solve/Update* or short cut button **Update interval**) as shown in Figure 5.47, in this case every simulated hour. Each row in the list will contain the time (in time string format or in seconds) or the number of iterations and the actual recorded value(s). 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.

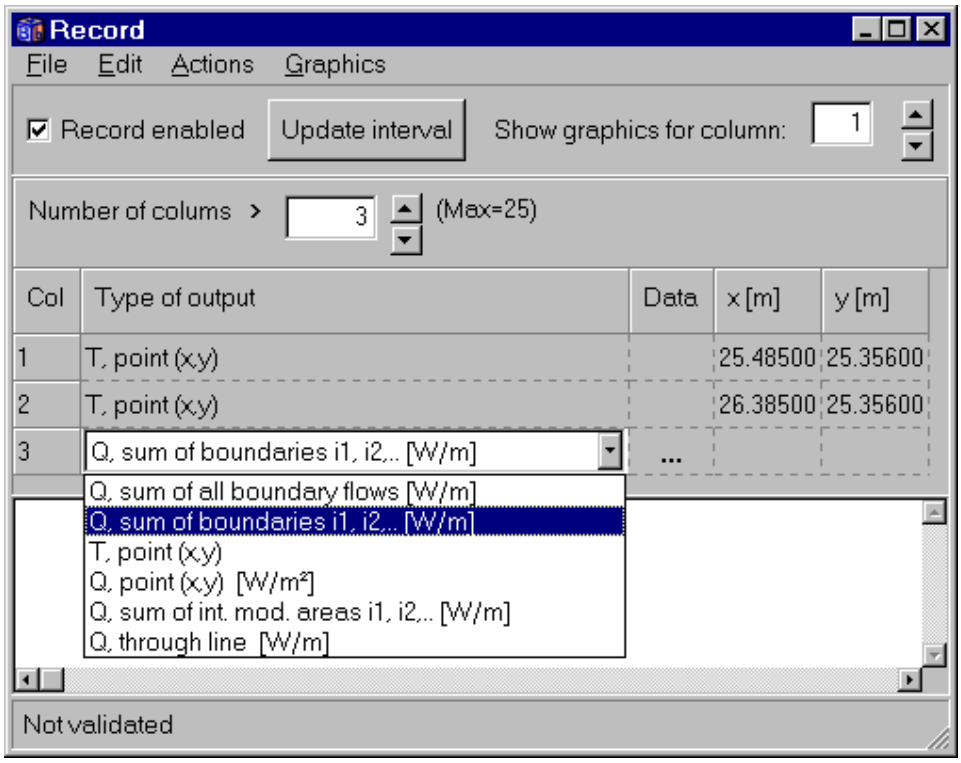


Figure 5.46: Data are recorded in columns as a function of time (transient) or iterations (steady-state).

Figure 5.46 shows the recorder with three columns defined. The two first columns will record the temperatures at (x, y)=(25.485, 25.356) and (26.385, 25.356) for the slab problem (slab1.dat). The points are located 1 dm down in the floor and 1dm and 1 m from the internal side of the wall, respectively. The third column records the heat flow through the floor area (W/m) which is boundary segment 3 (given in the **Data** entry). Figure 5.48 shows the output with an update interval of 1 hour. Figure 5.49 and Figure 5.50 show the recorded columns (see menu item **Graphics** and menu box **Show graphics for column** in Figure 5.46).

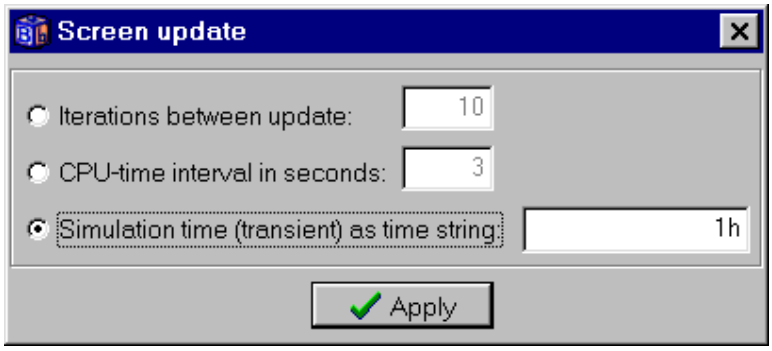


Figure 5.47: The time interval for recorded data is given in item *Solve/Update*.

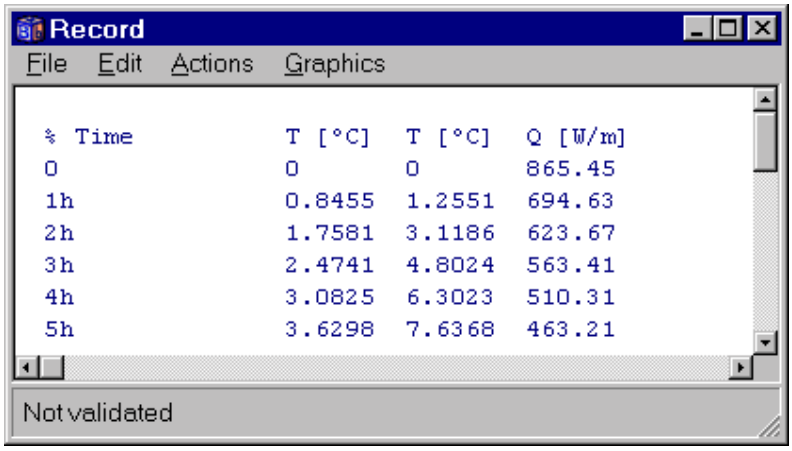


Figure 5.48: Three columns (two temperatures and one heat flow) have been recorded with a update interval of 1 hour.

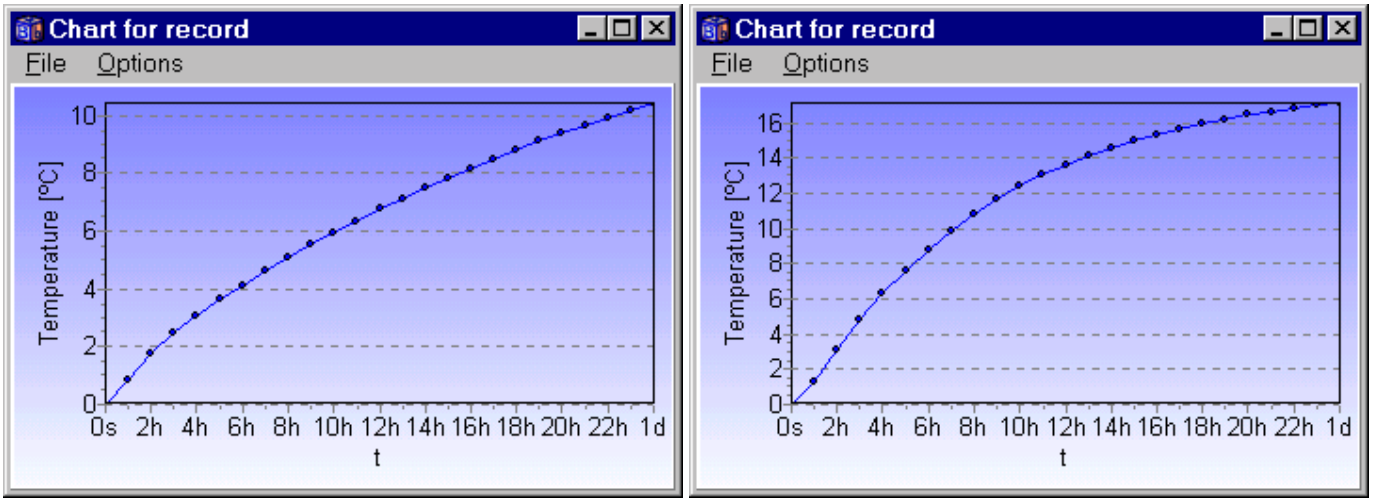


Figure 5.49: Left: Temperature 1 dm down in the slab and 1 dm from the wall. Right: Temperature 1 dm down in the slab and 1 m from the wall.

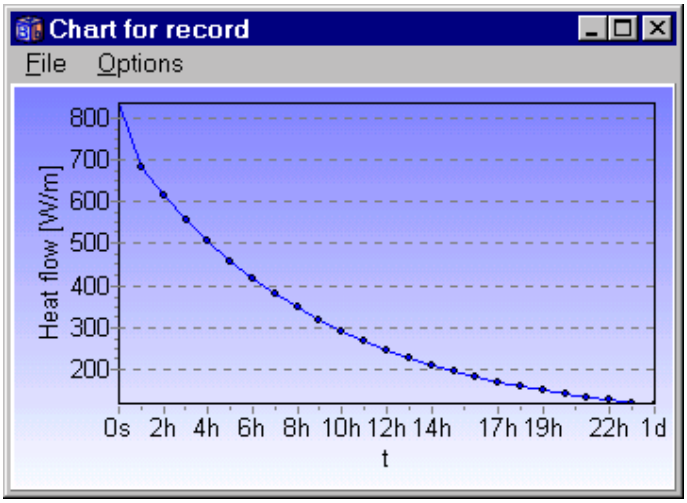


Figure 5.50: Heat flow (W/m) through the floor (boundary segment 3).

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 5.51 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 (ω=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*).

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.

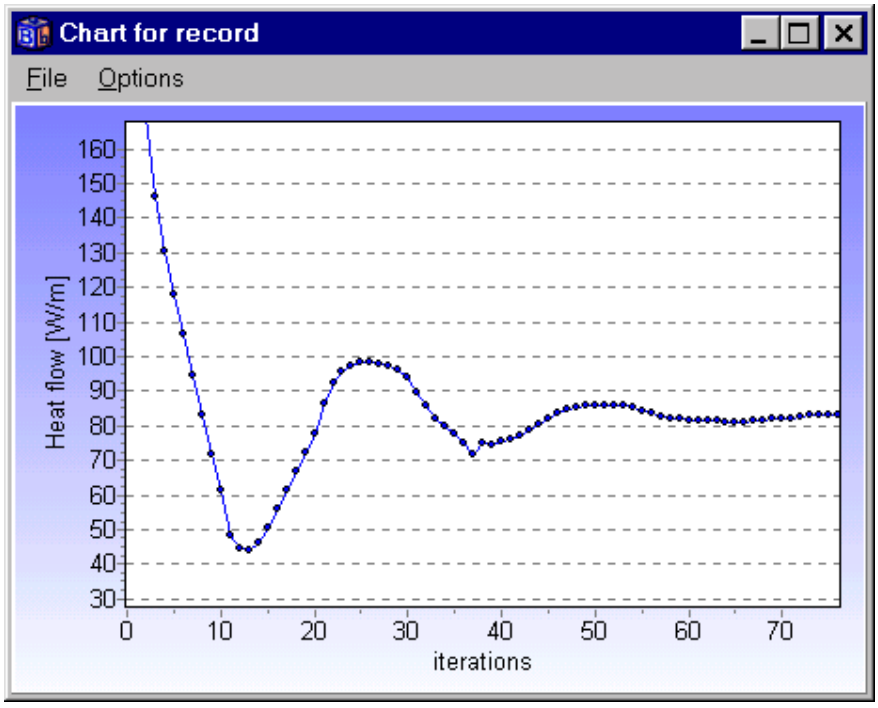


Figure 5.51: Chart showing recorded values.

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, see further section 5.16.

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

5.15.1 What can be recorded?

The pick list in Figure 5.46 shows the six options that may be recorded:

1. Q, sum of all boundary flows [W/m]
2. Q, sum of boundaries i1, i2,.. [W/m]
3. T, point (x, y)
4. Q, point (x, y) [W/m²]
5. Q, sum of int. mod. areas i1, i2,.. [W/m]
6. Q, through line [W/m]

The fifth type is a heat flow through internal modification areas. One or more of the four boundaries of each area may be excluded. Figure 5.52 shows that the added heat flow through the lower boundary will be recorded for four areas 1, 2, 3 and 5.

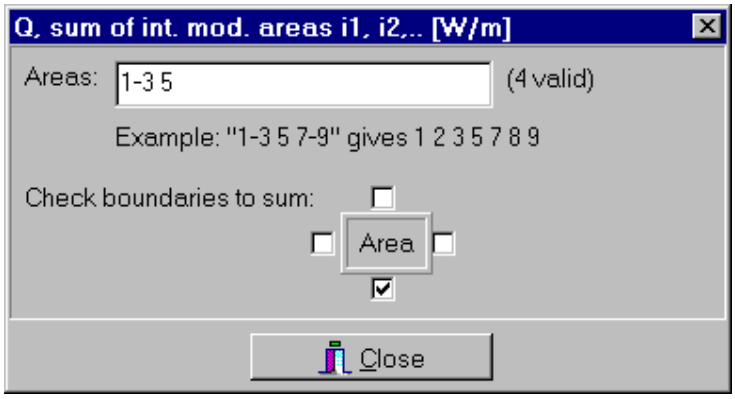


Figure 5.52: The added heat flow through the lower boundary will be recorded for four areas 1, 2, 3 and 5.

The sixth type is a heat flow through an internal line. This allows for internal heat flows at arbitrarily chosen locations to be recorded. Lower left and upper right coordinates are to be specified. The line has to be vertical or horizontal. If there is no mesh points at the desired location, use menu item **Input/insert mesh point** to allocate new mesh points in order to fit the line. The line may be placed along a boundary segment as well as being inside the computational volume.

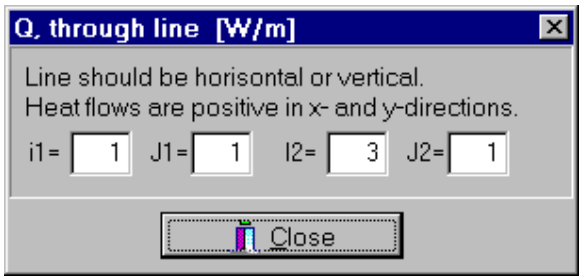


Figure 5.53: Internal heat flows may be recorded for a vertical or horizontal line.

5.16 Changing the record chart

5.16.1 Introduction

The menu box **Show graphics for column** in Figure 5.46 brings up a chart over one of the recorded data columns. The properties of this chart may be changed by the chart editor (**Options/Edit chart** in the **Chart for record** window), see Figure 5.54.

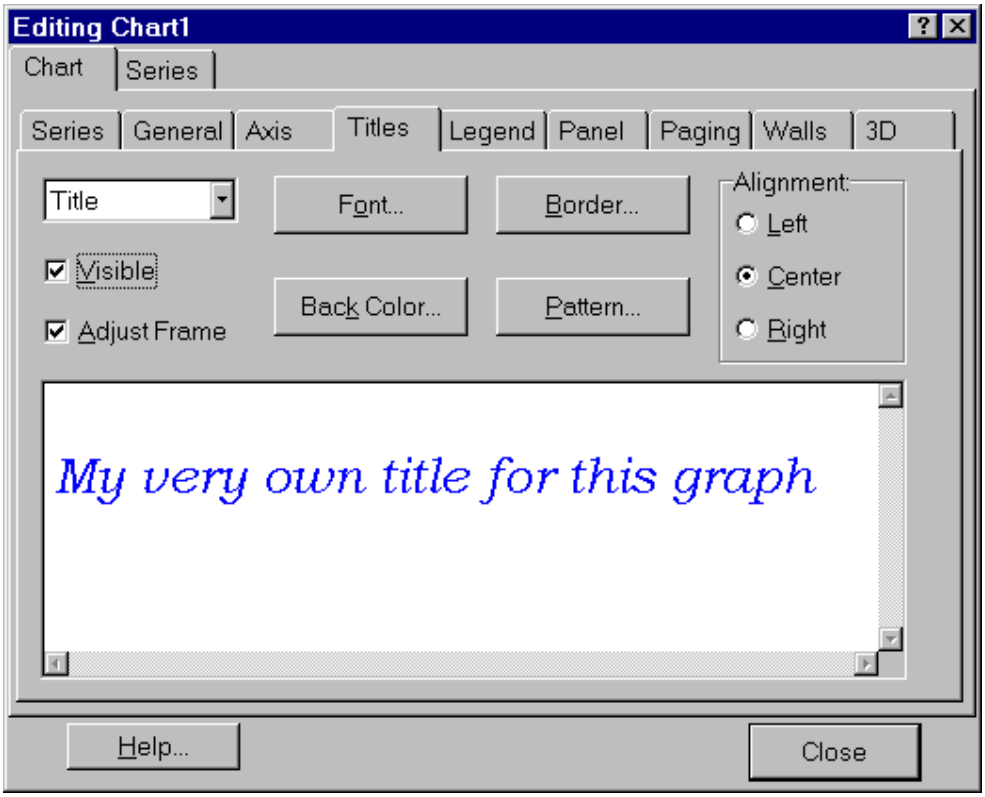


Figure 5.54: Chart settings may be changed in menu item **Options/Edit chart**.

5.16.2 Help

To get help on any topic in the Chart Editor, select the help button (question mark) at the top right hand side of the Editor window and drag it onto the Topic in question.

5.16.3 Editor design

There are two principal sections to the Chart editor, Chart parameters and the Series parameters, which are separated as two tabs of the Chart Editor.

Chart pages

You may define overall chart display parameters as follows:

Series page - You can change a series type to line, bar, area, point, etc. Select the series type of choice from the gallery.

General Page - Chart rectangle dimensions, margins, zoom and scroll, print preview and export

Axis Page - All axes definitions. Some parameters depend upon the series associated with the axis.

Titles Page - Title and Footer

Legend Page - Legend display. Formatted displays work in conjunction with the chart series. See also the 'General' page of the Series.

Panel Page - Chart Panel display properties. Colors, bevels, back images, color gradient and border.

Paging Page - Definition of number of points per chart page

Walls Page - Left, bottom and back wall size and color definitions

3D – 3D perspective options.

Series Pages

The series pages contain parameters dependant on the series type concerned. The most important options are as follows:

Format Page - Contains Series type specific parameters

Point – Visible points, margins

General Page - Series value format, axis association

Marks Page - Series mark format, text, frame and back color and positioning

5.17 Info log

The info log (item *Output/Info log*) contains information of input data and generated numerical mesh for the considered problem, see Figure 5.55. 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.

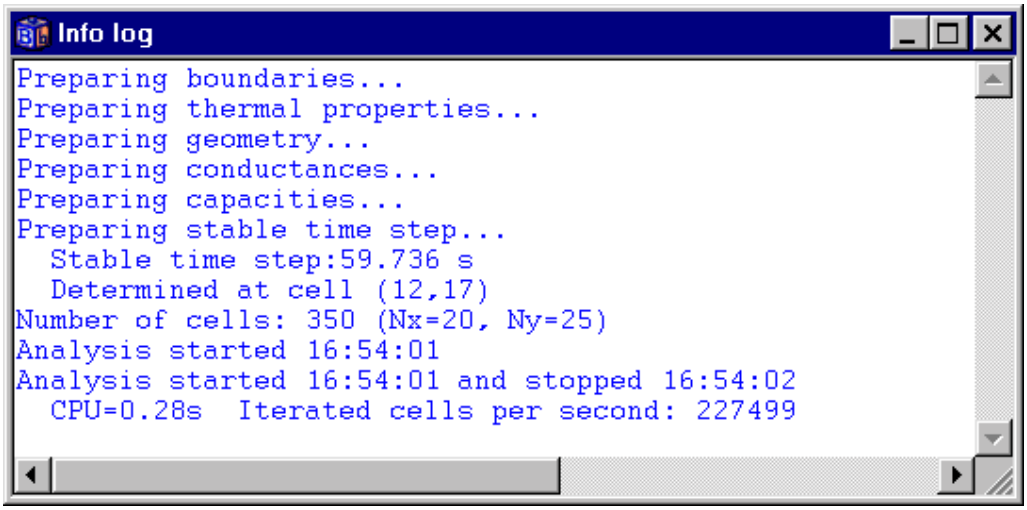


Figure 5.55: The info log.

5.18 Graphics

5.18.1 Introduction

Figure 5.56 shows the post-processor 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*.

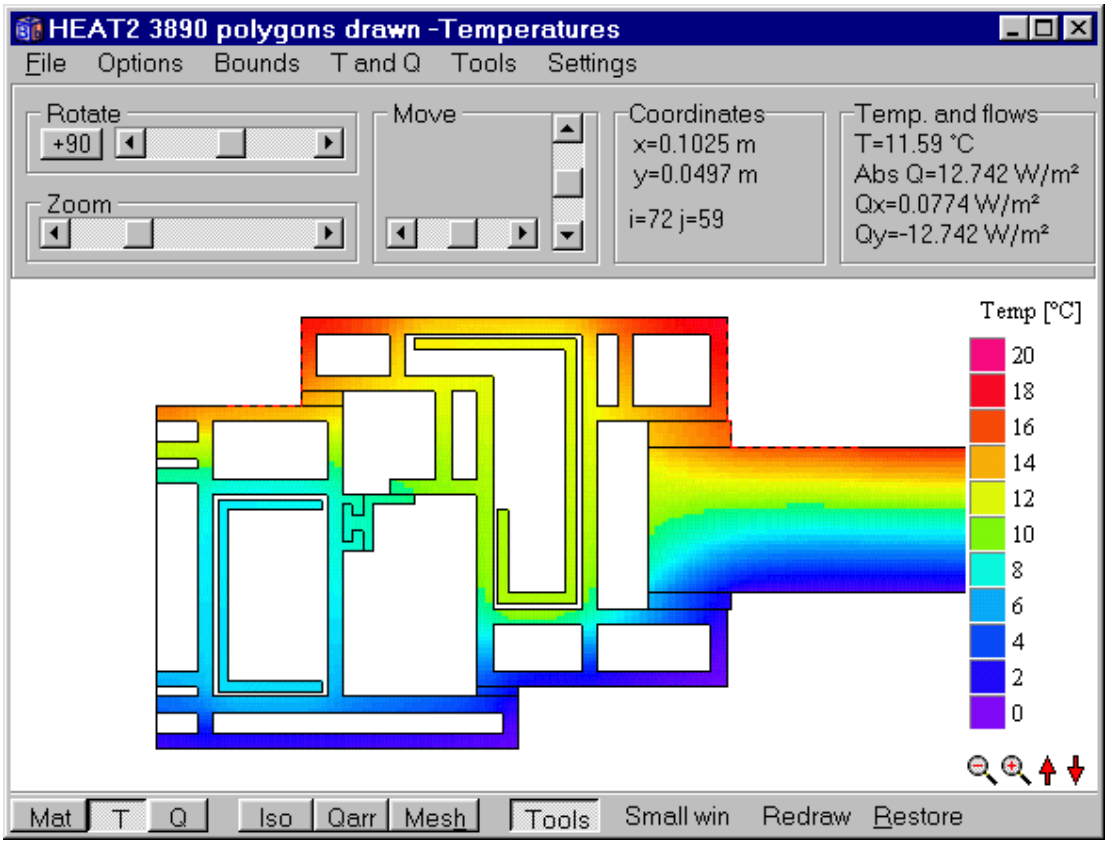


Figure 5.56: The graphic window.

5.18.2 Basic commands

The picture may be rotated, moved, or zoomed using the top tool-bar, the mouse, or directly from the keyboard, see Figure 5.57 for mouse and keyboard commands. The list is also displayed in item *Tools/Help shortcuts*.

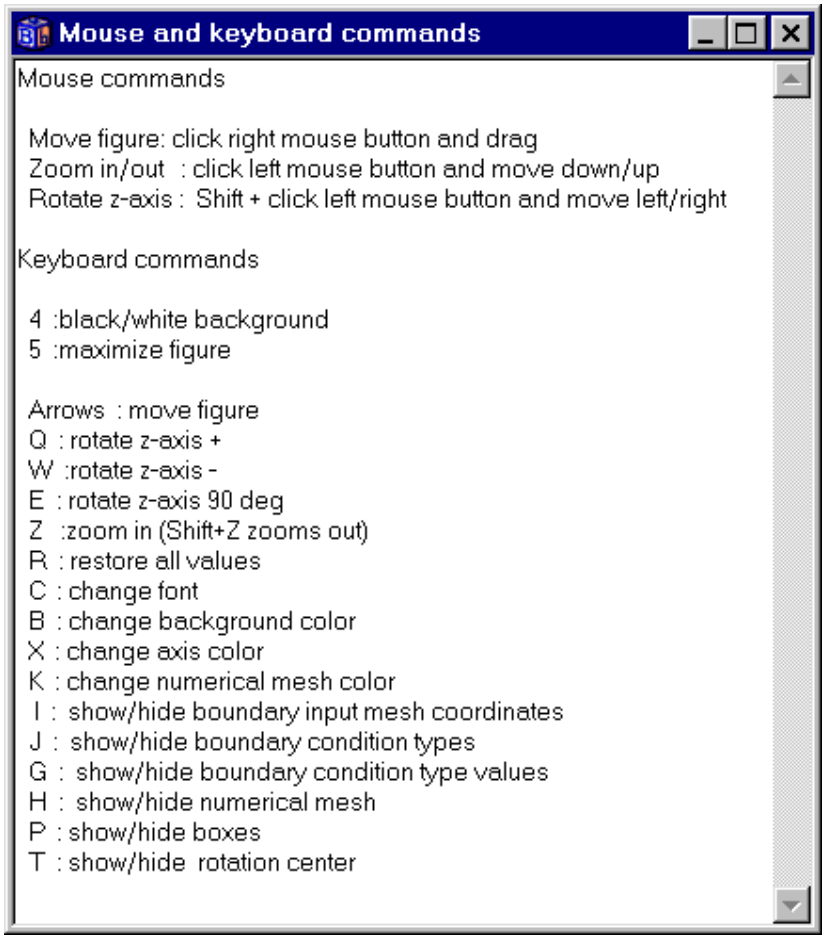


Figure 5.57: Shortcuts for basic commands.

5.18.3 Menu item *File*

5.18.3.1 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 it is in this manual). Using the cut option in *File/Cut image to clipboard* will give an image without the top window frame, see Figure 5.58 (top). Using the two keys **Alt-Print Screen** will capture the current active window (the **Print Screen** key will capture the whole screen). 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 5.61.

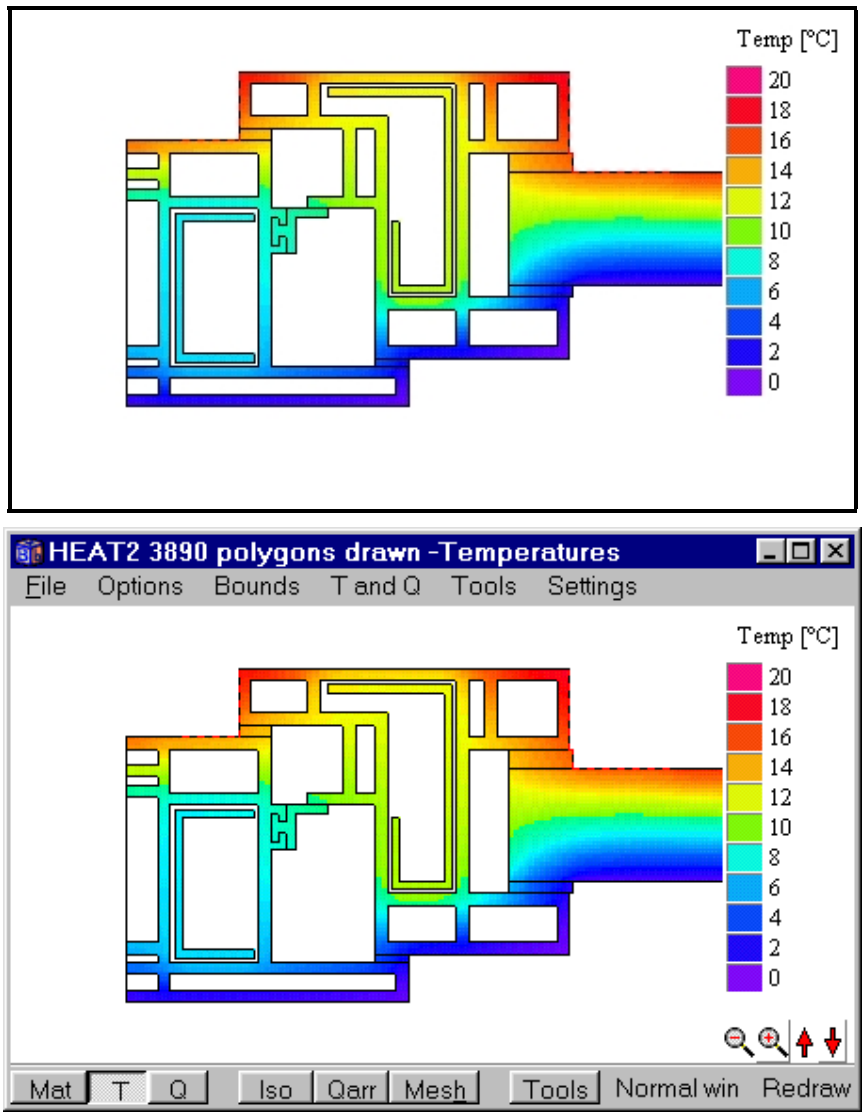


Figure 5.58: Use the cut option in *File/Cut image to clipboard* to get an image without the top window frame (top). Press **Alt-Print Screen** to capture the current active window (bottom). The active window has often a blue top menu row.

5.18.3.2 Printing images from HEAT2

Figure 5.59 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 5.60. 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 5.61. It is also possible to print a screen dump (with poor quality).

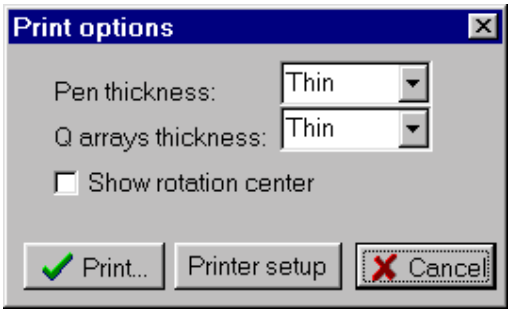


Figure 5.59: Print options.

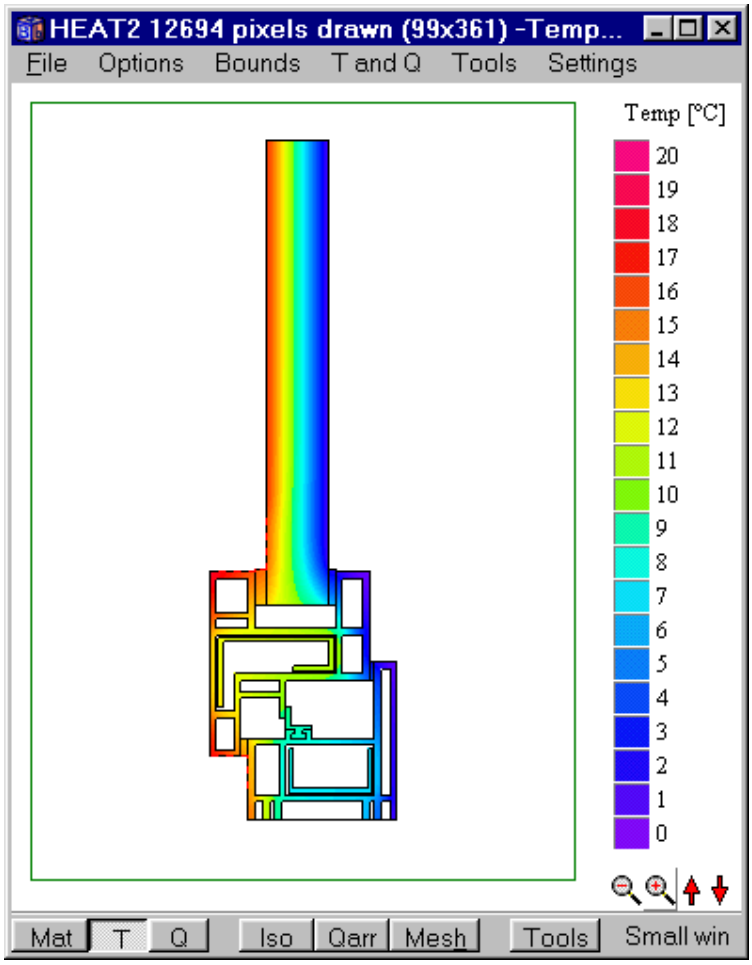


Figure 5.60: A simple preview with a green frame that shows the paper borders.

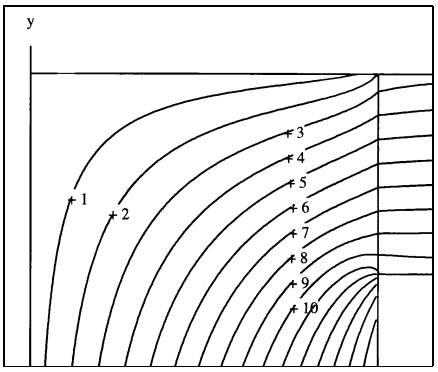


Figure 5.61: Isotherms printed directly from HEAT2. The printer's resolution will be used, in this case 600 DPI. This picture has then been scanned into this document.

5.18.4 Menu item *Options*

Figure 5.62 shows the menu item *Options*.

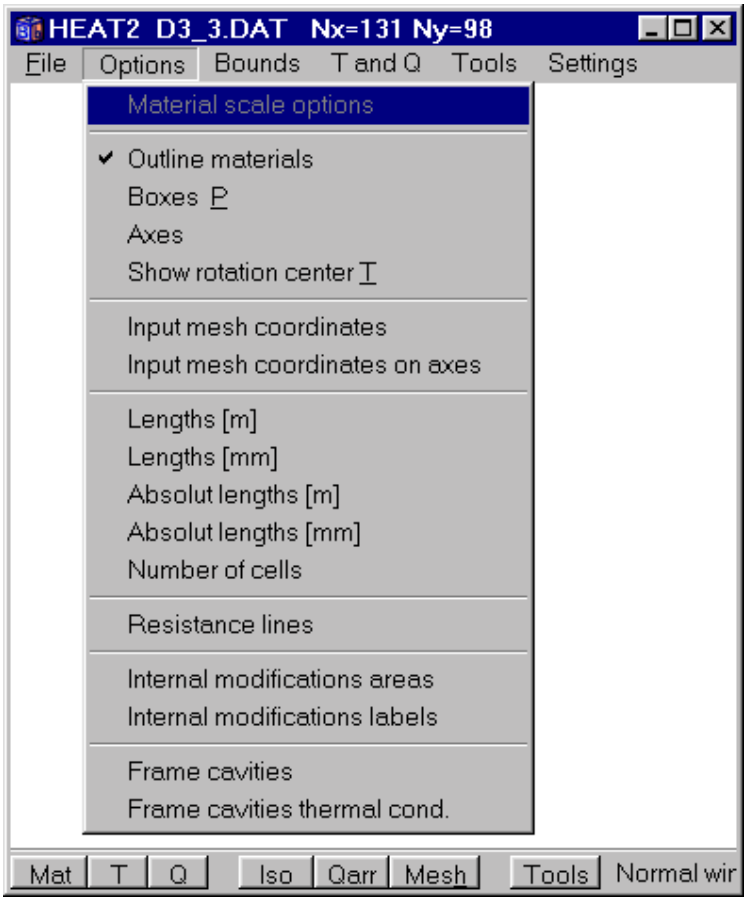


Figure 5.62: Menu item *Options*.

5.18.4.1 Viewing thermal conductivities and heat capacities

Click "**Materials**" on the bottom tool bar (or check item *boxes* in menu item *Options/material scale options*) to view thermal conductivities, see Figure 5.62. 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 5.63. Figure 5.64 (left) shows colors sorted by material names (two material may have the same thermal properties but different names). Figure 5.64 (right) shows sorting by defined areas (white indicates default value), see Section 4.7.

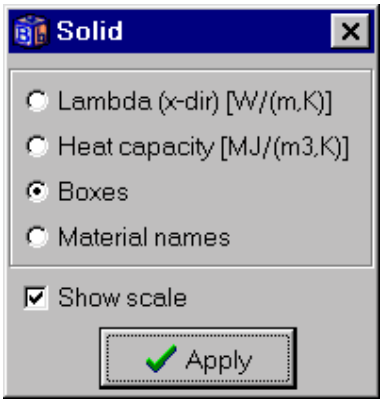
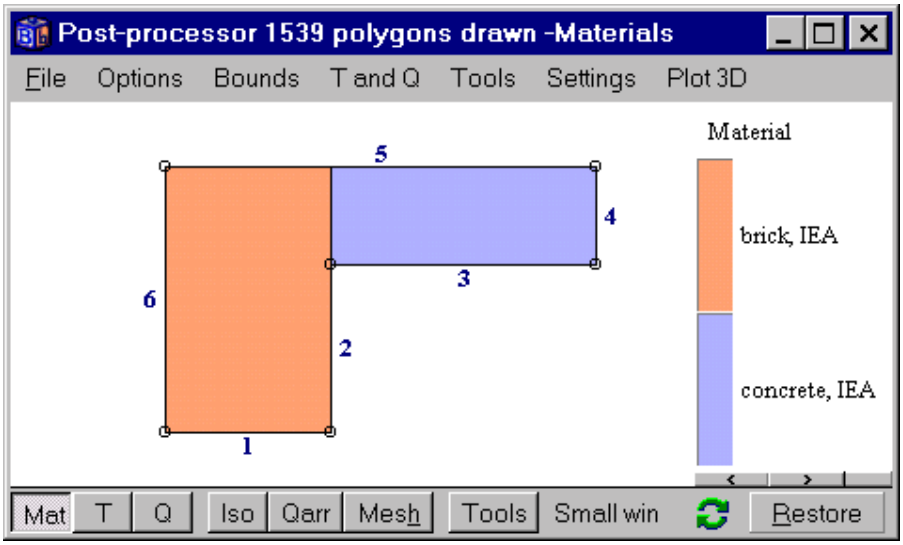


Figure 5.63: Click the scale bar (or *Options/Material scale options*) for options.



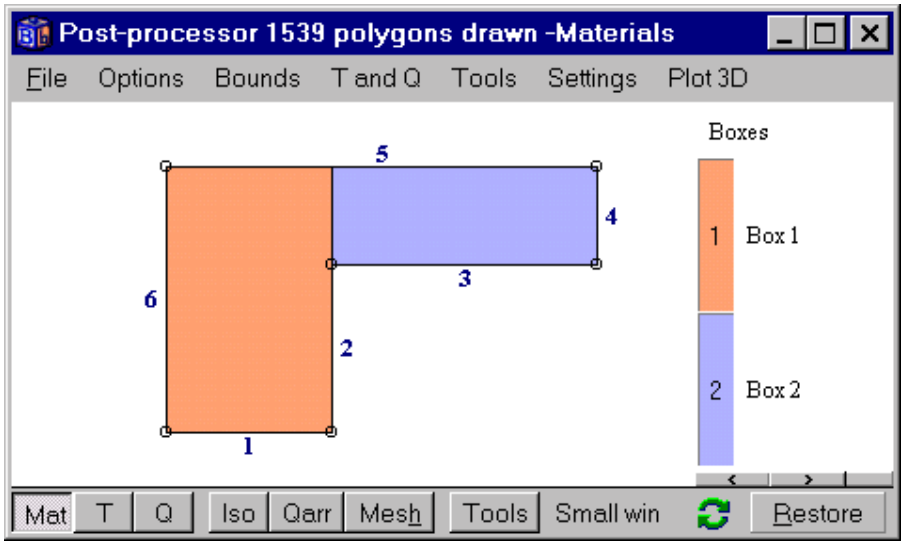


Figure 5.64: Colors indicate material names (top) and boxes (bottom).

5.18.4.2 Outlined materials

Figure 5.65 shows the input boxes as given in the thermal properties list (option *Options/boxes).* Figure 5.66 shows the interface between the materials with a black line (see menu item *Options/Outline materials* in the graphics menu).

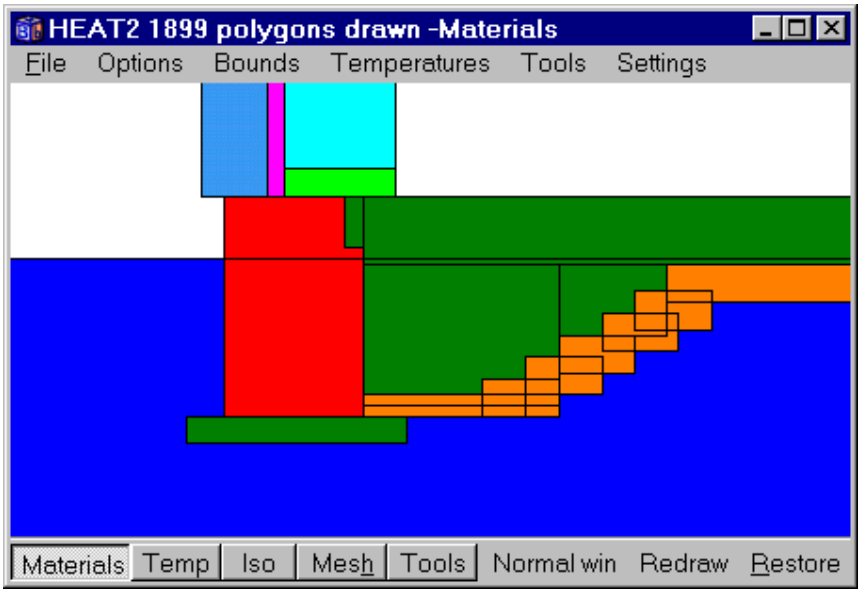


Figure 5.65: Material boxes as by *Options/boxes*.

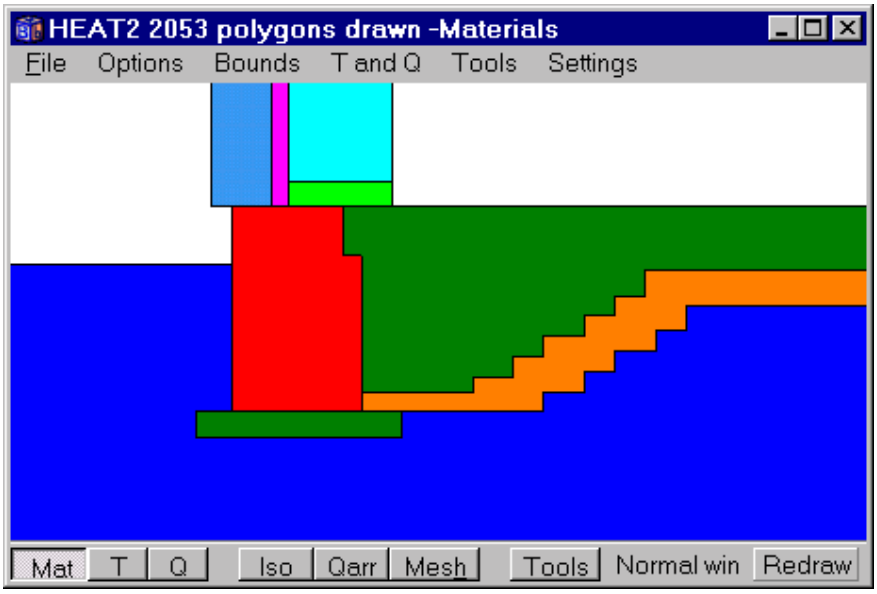


Figure 5.66: Materials outlined as shown by *Options/Outline materials*.

5.18.4.3 Absolute lengths

Figure 5.67 shows the lengths between each mesh segment (left). It is also possible to view the *x*- and *y*-coordinates to each input mesh coordinate (menu item *Options/Absolute lengths*), see Figure 5.67 (right).

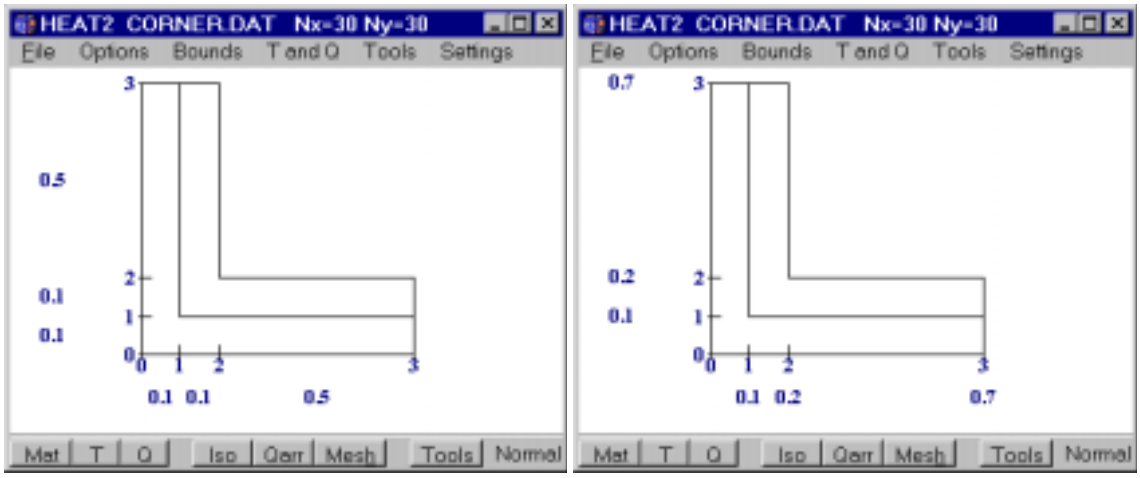


Figure 5.67: Lengths between mesh coordinates shown at left and absolute lengths to each mesh coordinate shown at right.

5.18.4.4 Resistance lines

Resistance lines may be shown by item *Options/Resistance lines* in the graphics menu. Figure 5.68 shows a resistance line between input mesh coordinates (0,1) and (1,1) as a dotted red line.

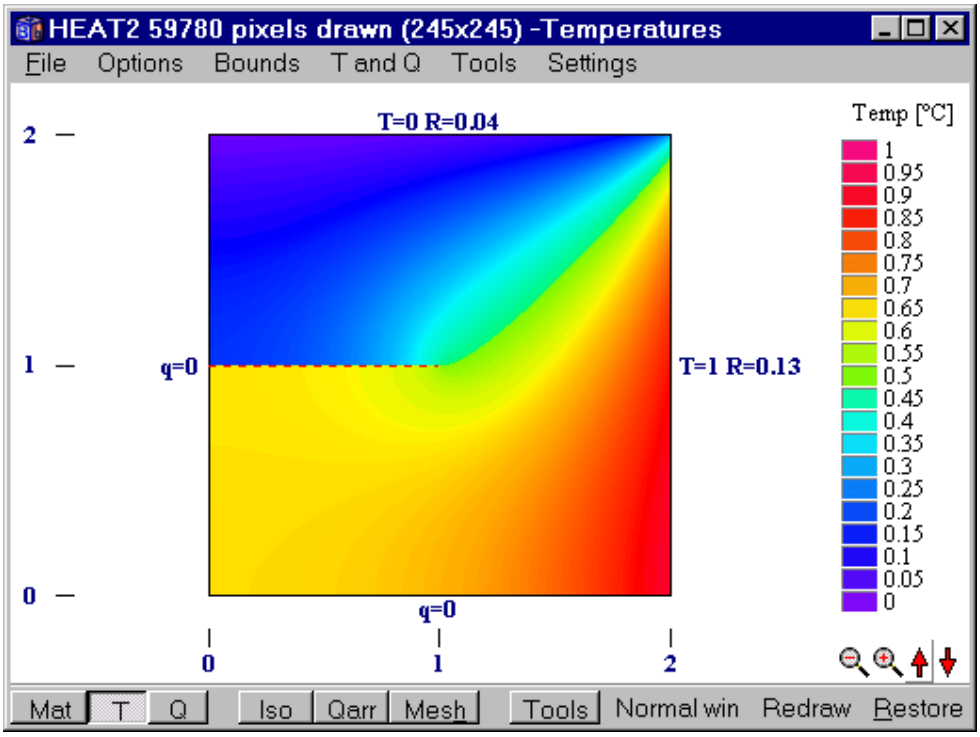


Figure 5.68: The resistance line between input mesh coordinates (0,1) and (1,1) is shown by a dotted red line.

5.18.5 Menu item *Bounds*

Figure 5.69 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).

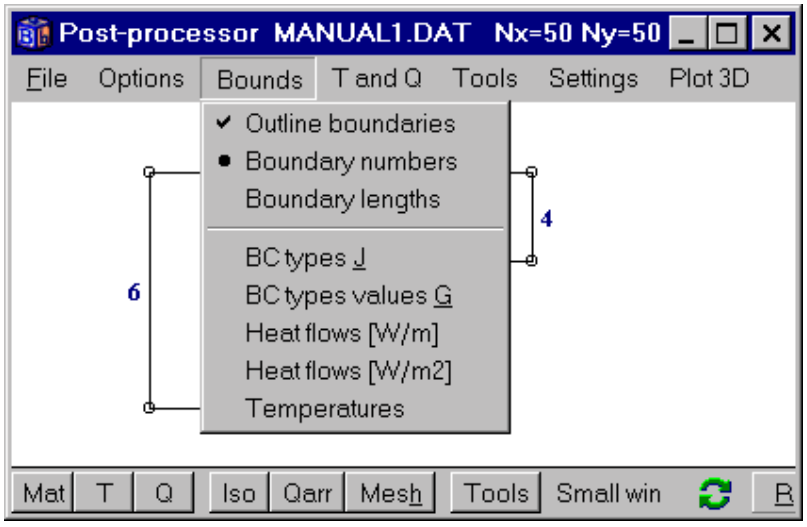


Figure 5.69: Menu item *Bounds*.

5.18.6 Menu item *T and Q*

Figure 5.70 shows the *T and Q* menu. The picture is updated (unless *Temperatures/Update figure on solve* is disabled) during the simulation at time interval given in *Solve/Screen update*.

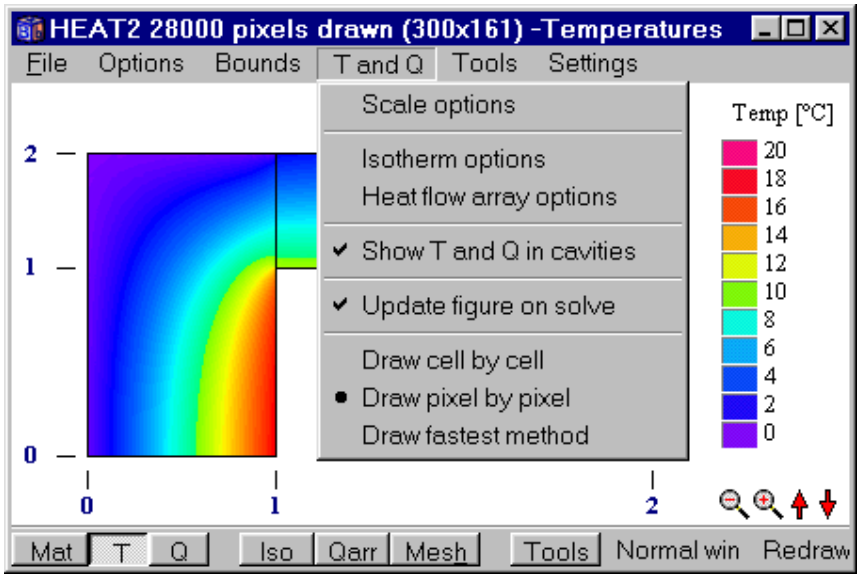


Figure 5.70: Temperatures. Note that the number of bars in the scale will be reduced if the window is resized and too small to hold all bars (in this case the number of bars is reduced automatically from 21 to 11).

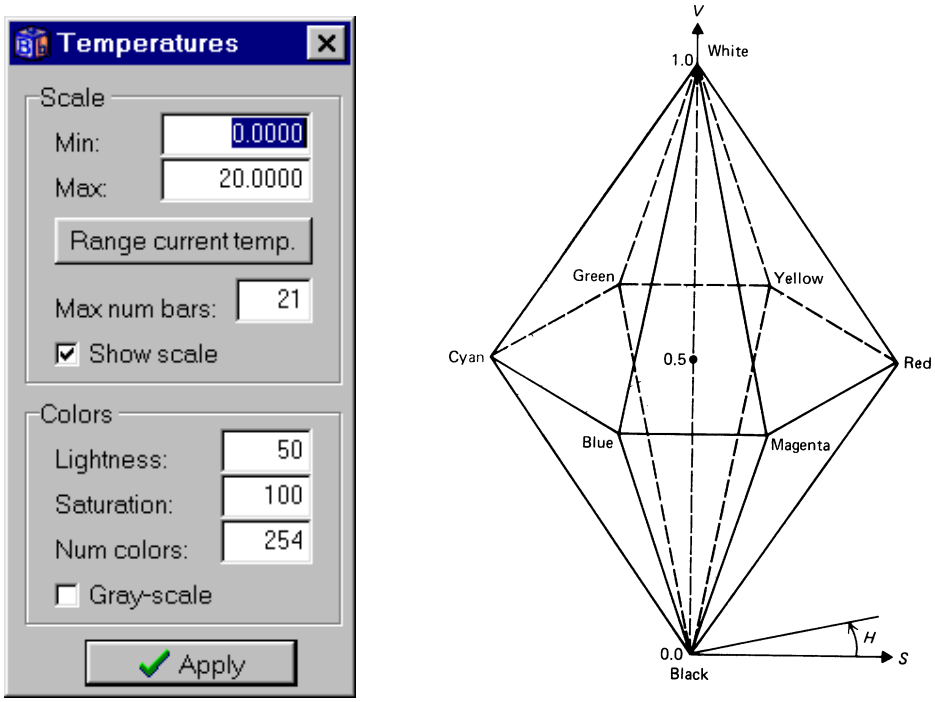


Figure 5.71: Options for temperature scale (left). The hue-lightness-saturation color model (right).

Click the scale bar (or *Temperatures/options for scale*) for options, see Figure 5.71 (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 5.71 (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 5.72 shows two disabled temperatures, T=11 and T=16 ºC. 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 on the bars will enable/disable all intervals.

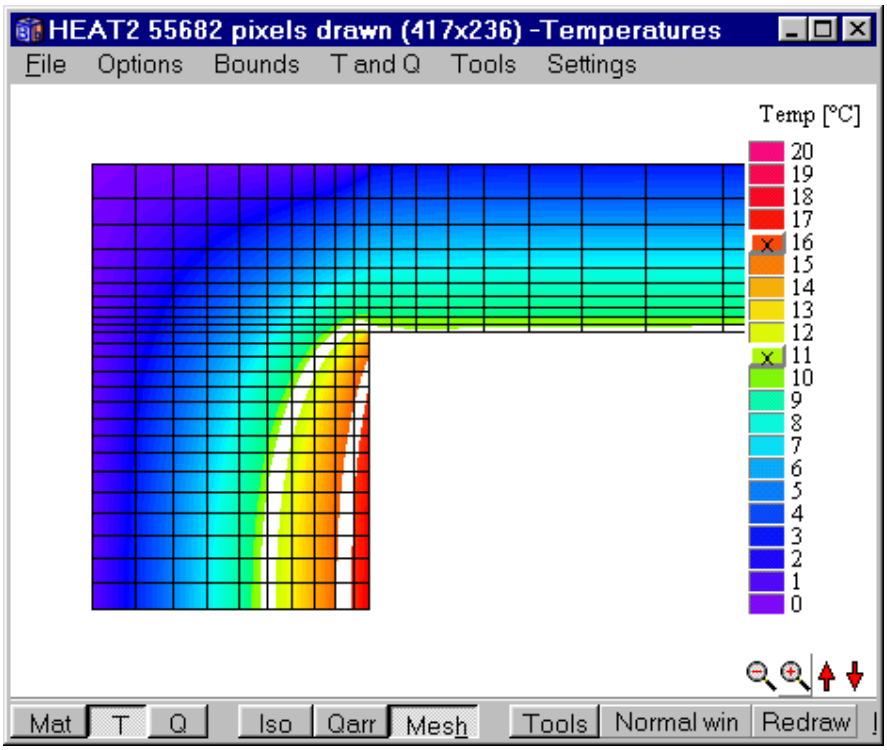


Figure 5.72: Temperature intervals may be hidden by pressing the scale color, in this case T=16 and T=11 ºC.

5.18.6.1 Better picture quality with pixel by pixel drawing options

There are two ways to draw images representing temperatures and heat flows. The first way is to use the option “cell by cell”, see Figure 5.73, where each numerical cell is filled with the color given by the cell temperature or in the heat flow case as an average of the heat flows through the four boundaries of the cells. The second way is to draw each screen pixel individually using interpolation, see Figure 5.74.

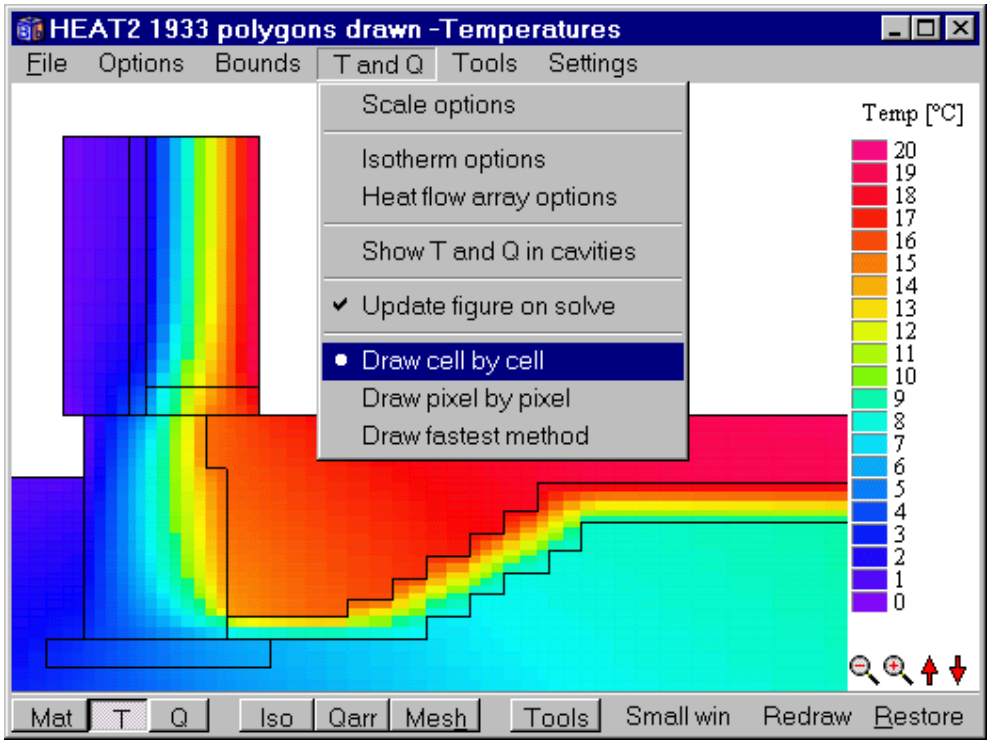


Figure 5.73 Temperatures drawn cell by cell. This method is often quicker but may have a visible cell grid.

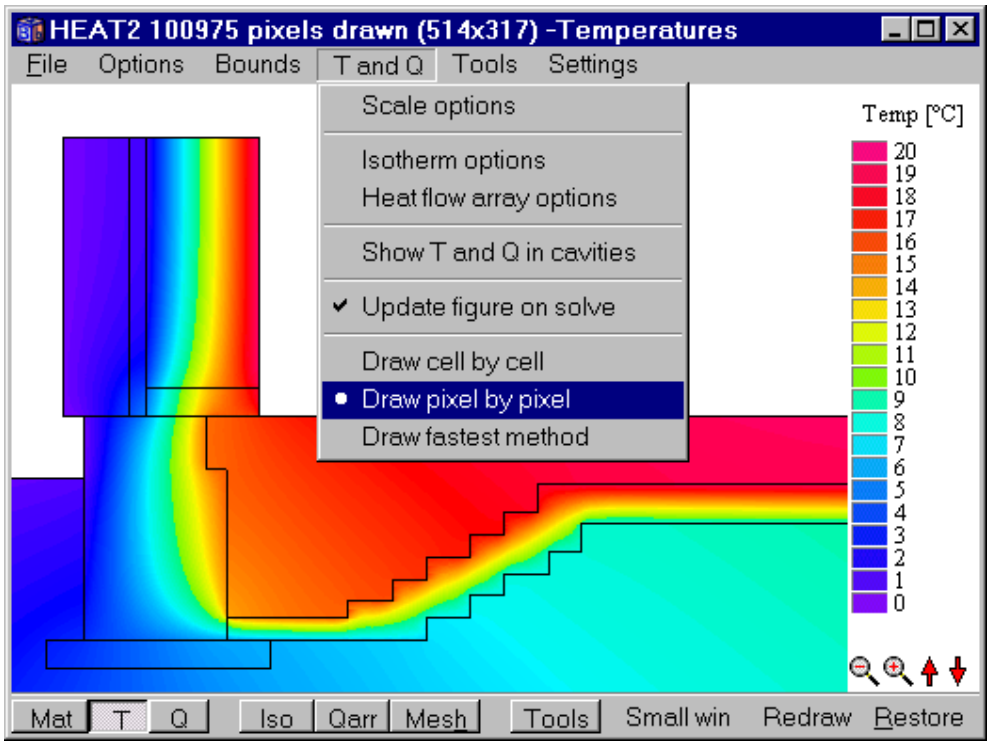


Figure 5.74: Temperatures drawn pixel by pixel. The image is smoother but may take some time to draw.

The drawing time for the two different options depends on the number of cells and how big the image is. The option “cell by cell” is normally faster when the number of pixels is greater than the number of cells.

For a small image (typically with more numerical cells than pixels), the option “pixel by pixel” is normally fastest. HEAT2 draws by default the fastest method (the last menu option). The option “pixel by pixel” gives the best quality for presentation.

5.18.6.2 Heat flow magnitude

The heat flow magnitude (W/m²) may be drawn by clicking the **Q** button on the bottom tool bar. The scale options are the same for temperatures, such as choosing color or black and white presentation. Figure 5.75 shows the heat flow intensity field (W/m²) drawn cell by cell in color. This plot of the heat flow magnitude makes it visually simple to determine thermal bridges and to improve designs by optimizing the insulation at areas with large heat flows. Figure 5.76 shows the heat flows drawn pixel by pixel.

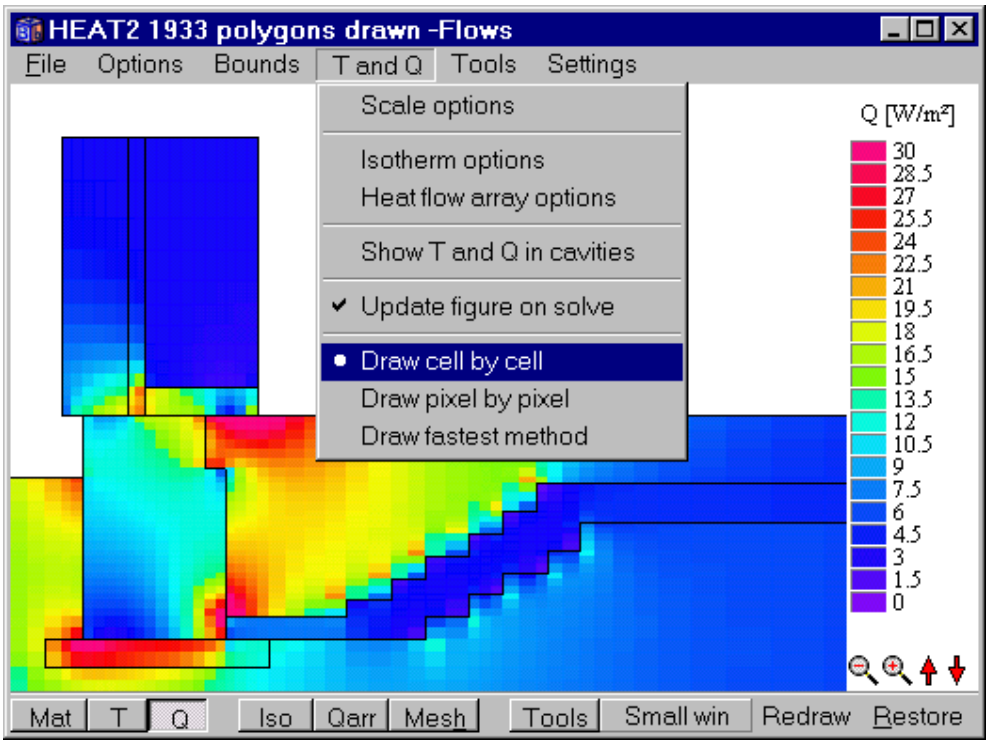


Figure 5.75: The heat flow intensity field (W/m²) drawn cell by cell.

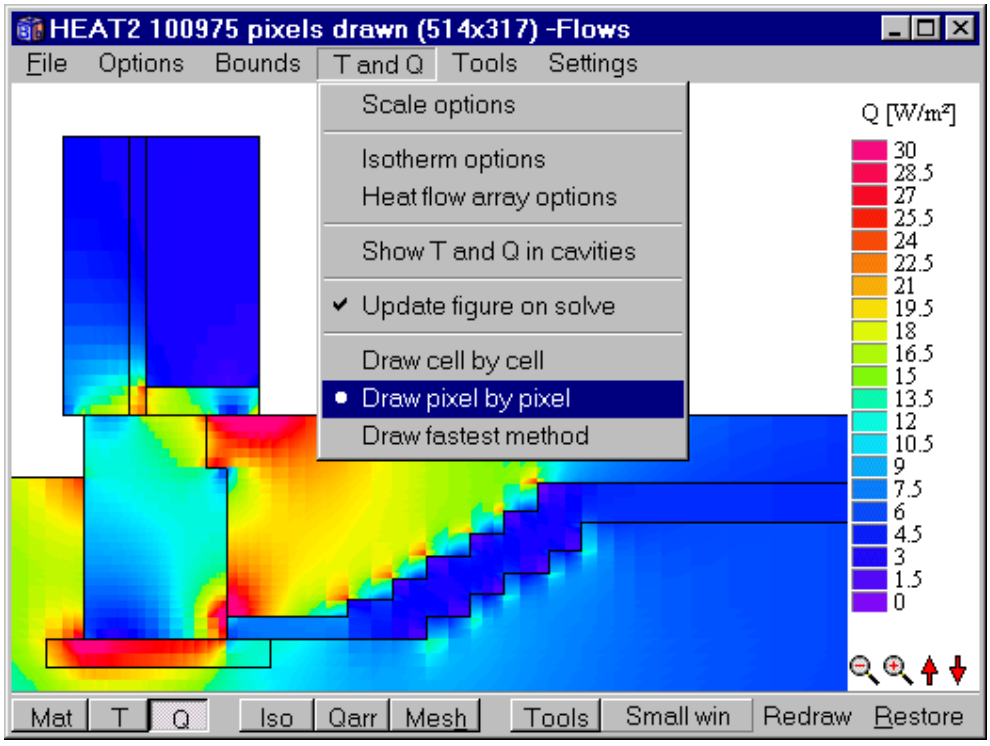


Figure 5.76: The heat flow intensity field (W/m²) drawn pixel by pixel.

5.18.6.3 Heat flow arrays

5.18.6.3.1 Options for heat flow arrays

Arrays indicating magnitude and direction of heat flow may be drawn by pressing the bottom tool bar button **Qarr**. The arrows may be shown separately or together with an image of materials or temperatures or heat flows, along with isotherms. Figure 5.77 shows black arrows drawn together with a color representation of heat flows.

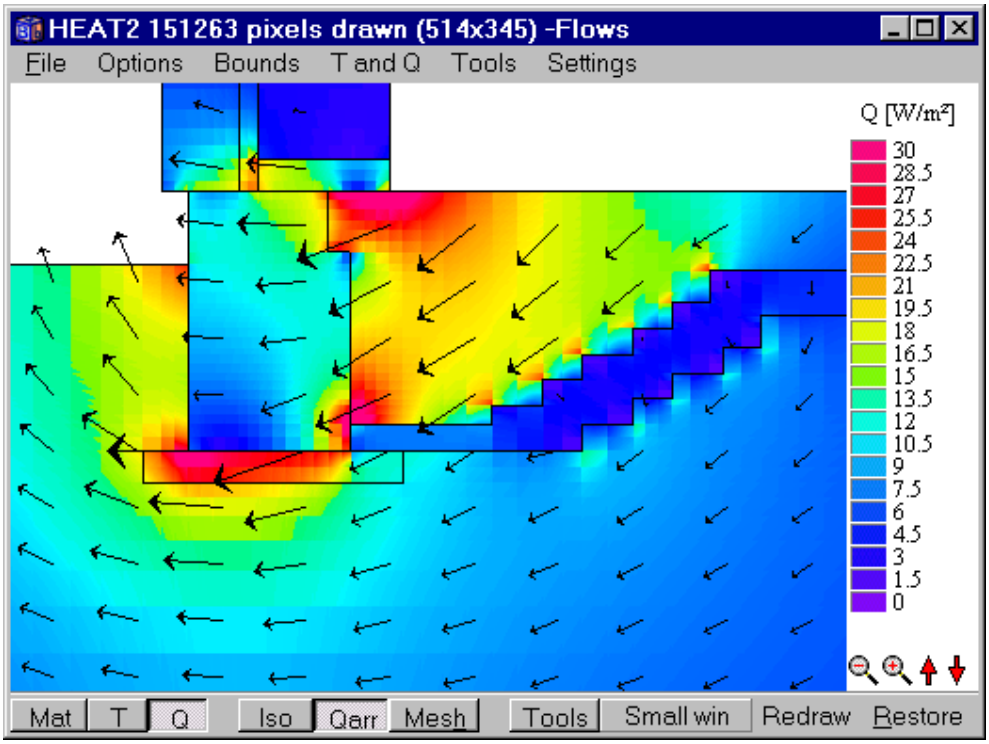


Figure 5.77[col]: Heat flow arrows drawn together with a color representation of heat flow.

Figure 5.78 (left) shows the options for heat flows arrays (right-click bottom tool bar button **Qarr** to show the menu). The lengths of the arrays indicate the magnitude of the heat flow intensity. The lengths are normalized so that the maximum heat flow is shown with an about one centimeter long array. The lengths may be adjusted in option **Length**.

[col] See Appendix D for color image

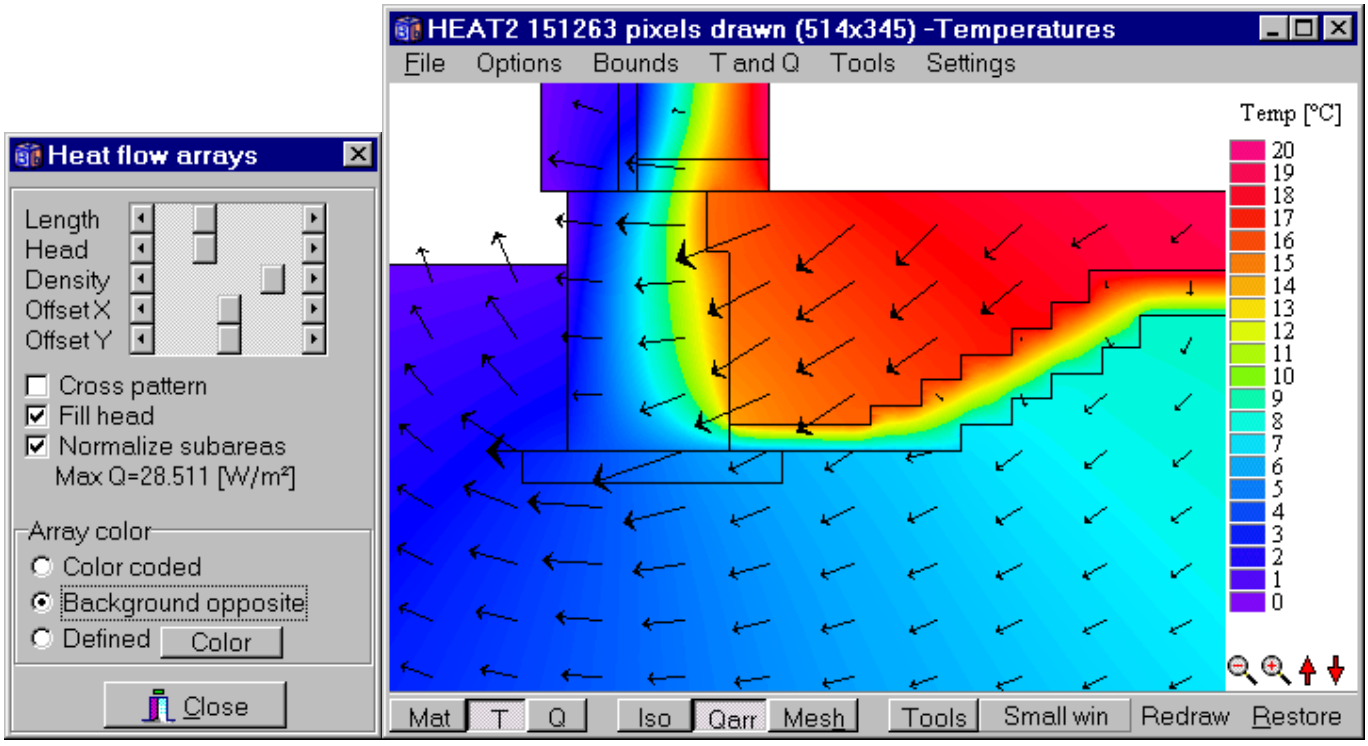


Figure 5.78[col]: Options for heat flows arrays (right-click bottom tool bar button **Qarr**).

The base of each array starts in a grid point. The grid size can be changed by option **Density** and the grid may be moved horizontally and vertically by options **Offset X** and **Offset Y**, respectively. The grid is normally regular but may be changed to a **cross pattern** where arrays are drawn when $i+j$ is even, see Figure 5.79. The size of the array head is given by option **Head** and filled if option **Fill head** is checked, see Figure 5.80.

The option **Normalize sub-areas** means that the lengths of the arrows visible in the current window are relative to each other, where the longest arrow is about 1 cm (this can be scaled by option **length** as mentioned before). In this case the maximum heat flow in the current window is displayed (**Max Q**), see Figure 5.81. When **Normalize sub-areas** is unchecked all arrow lengths are relative to the maximum heat flow in the whole computational area, see Figure 5.82.

[col] See Appendix D for color image

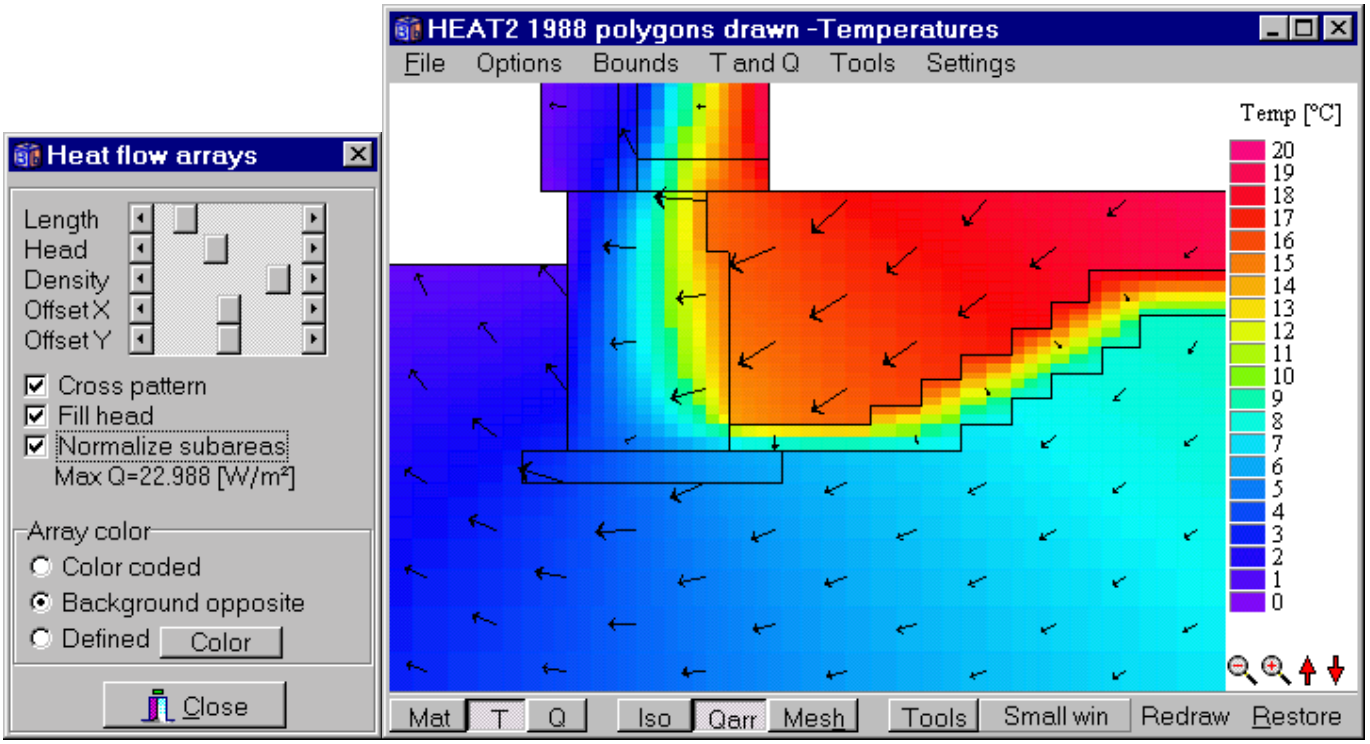


Figure 5.79: Arrays drawn in a cross pattern.

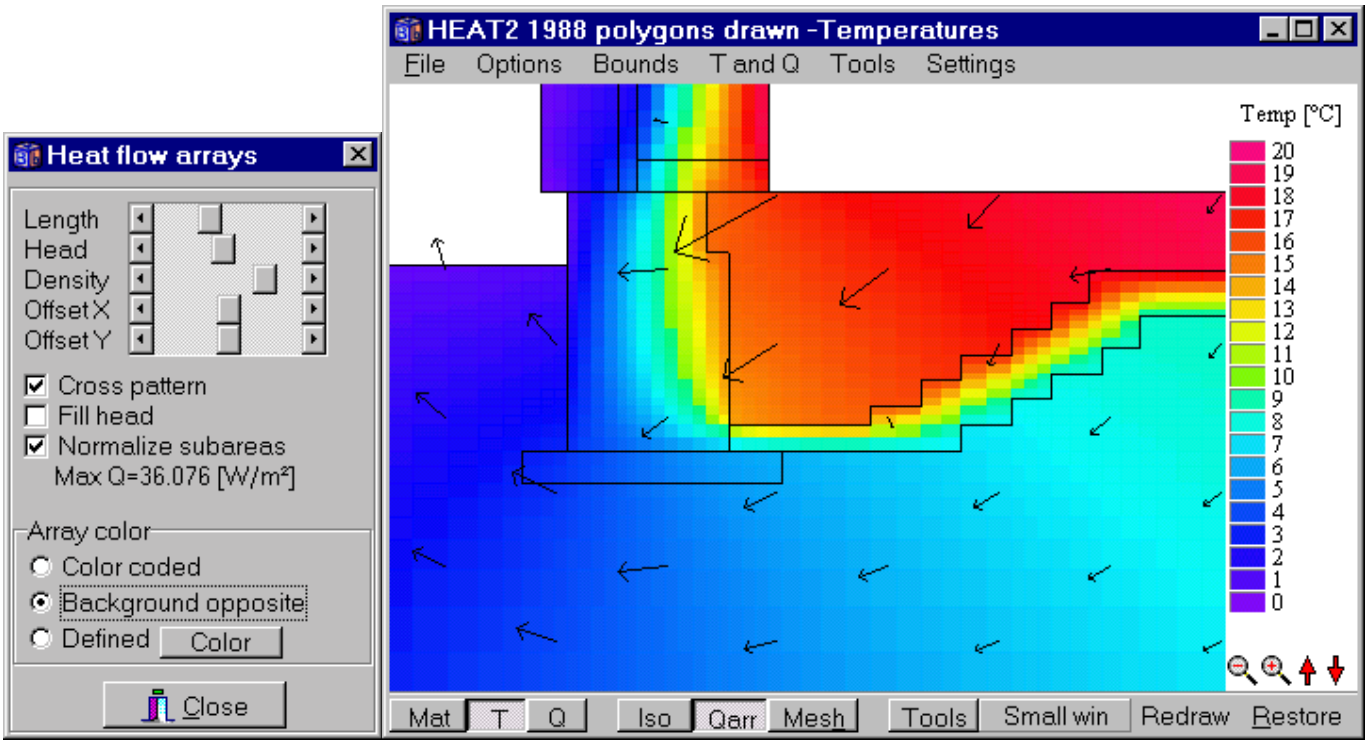


Figure 5.80: Arrays in a lower density cross pattern without filled heads.

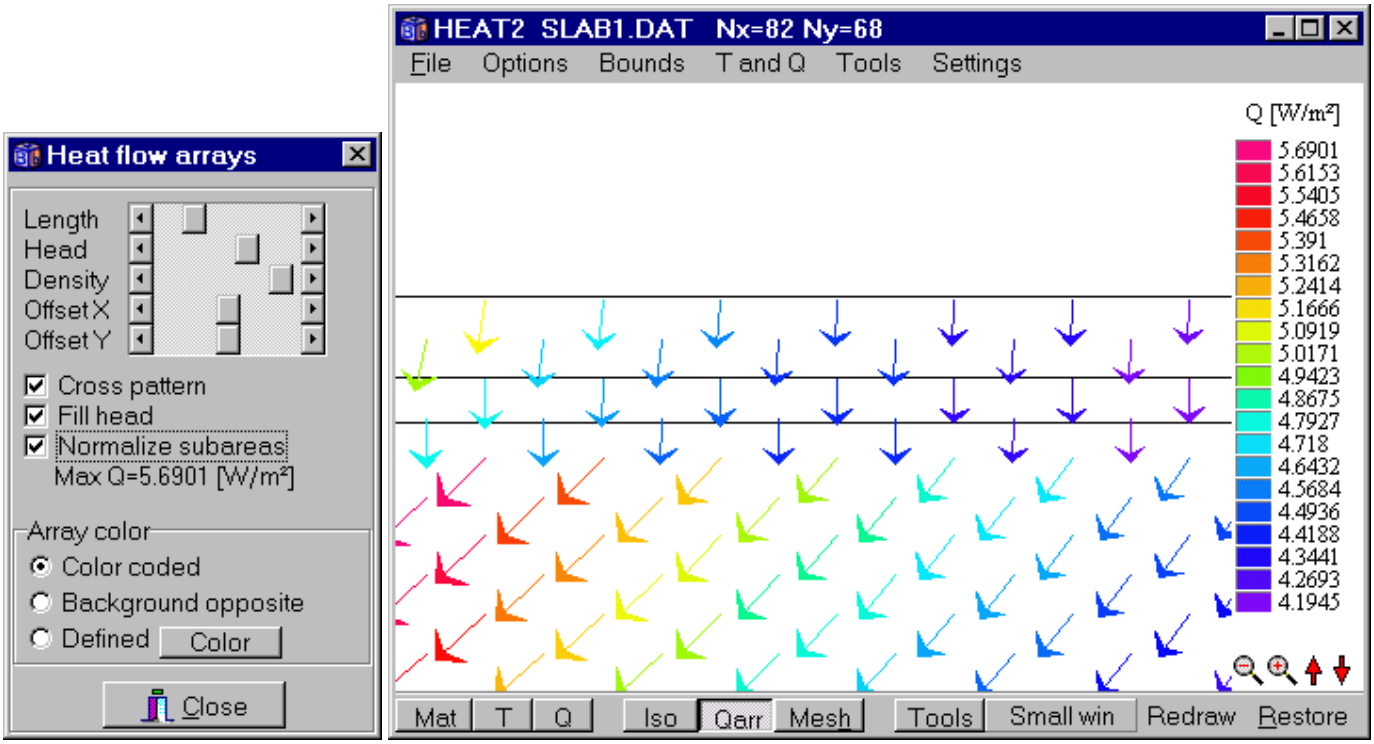


Figure 5.81: The option **Normalize subareas** means that the lengths of the arrows visible in the current window are relative to each other, where the longest arrow (here about 6 W/m² in the middle of the floor) is about 1 cm (can be scaled by option **length**).

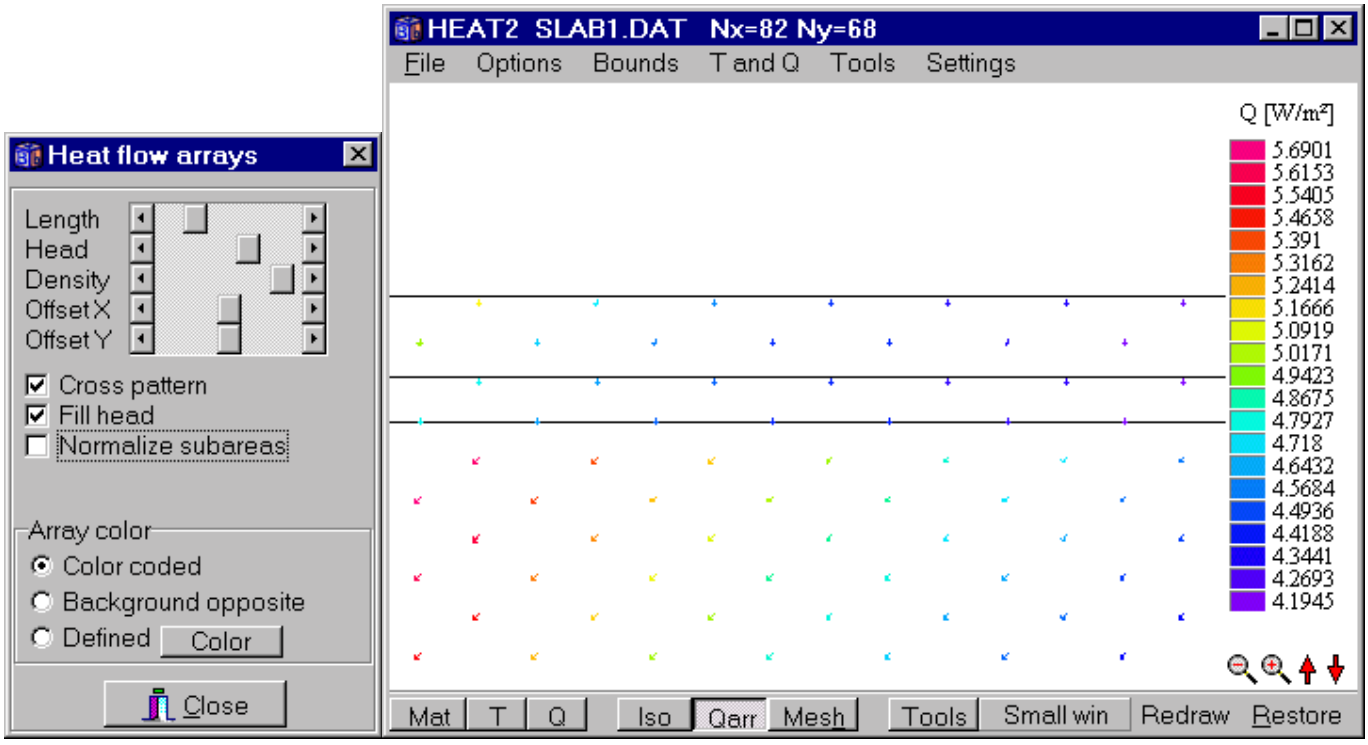


Figure 5.82: The option "Normalize subareas" is unchecked here meaning that the lengths of the arrows are relative to the maximum heat flow in the whole computational area (which is

about 30 W/m²) with an arrow of about 1 cm. The arrows are small here since the largest heat flow in the window is just about 6 W/m². The lengths can be scaled by option "length".

The arrays may be **color coded** according to the heat flow color, see Figure 5.83. Note that if color-coded heat flows arrays are shown together with a color representation of temperatures, only the temperature scale will be shown. The array color may also be **Background opposite** using black arrays when the background is white and vice versa. The color may also be **Defined** arbitrarily. Color-coded arrays allow certain intervals to be masked by clicking the scale bars. Figure 5.84 shows heat flow arrays for two intervals between 4.5-5.5 and 9.5-10.5, respectively.

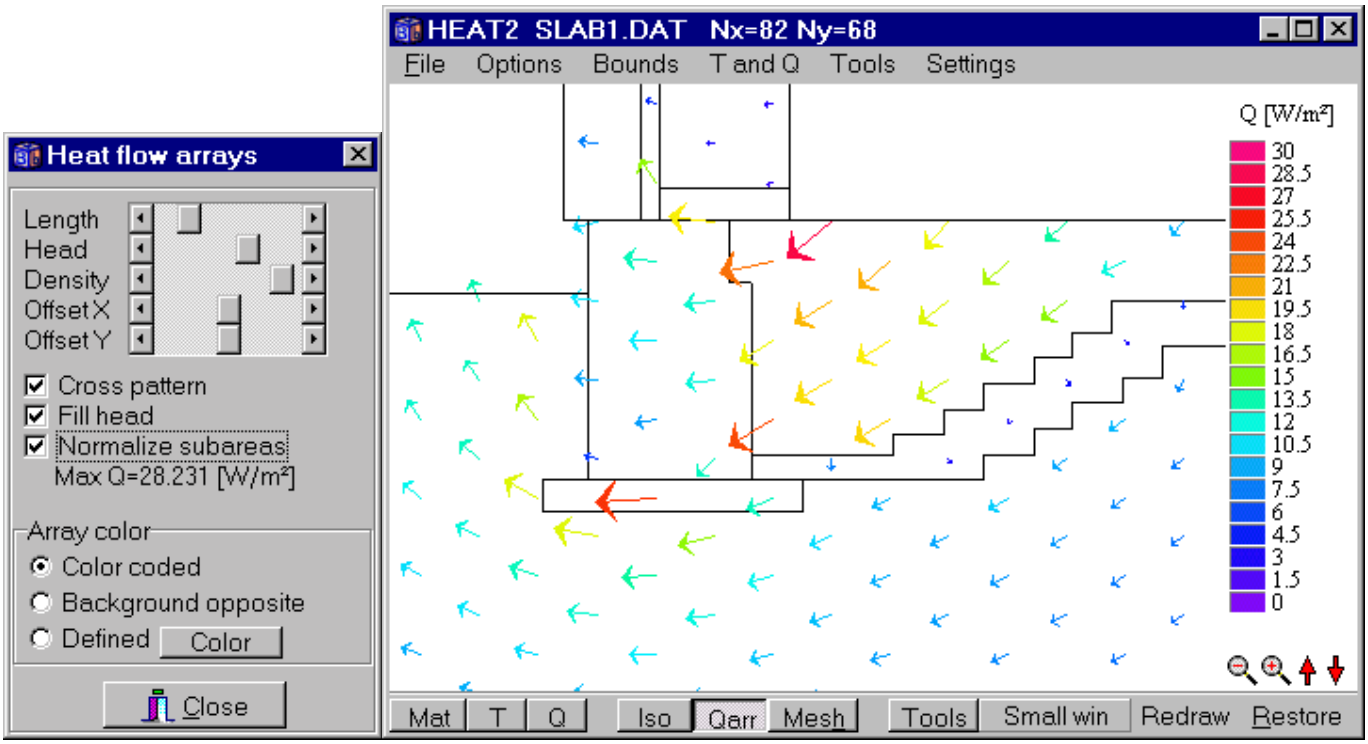


Figure 5.83: Color-coded arrays. The scale range may be changed by clicking in the white area beside the scale bar (or in menu item **T & Q/Scale options**, or by right-clicking the bottom tool bar button **Q**).

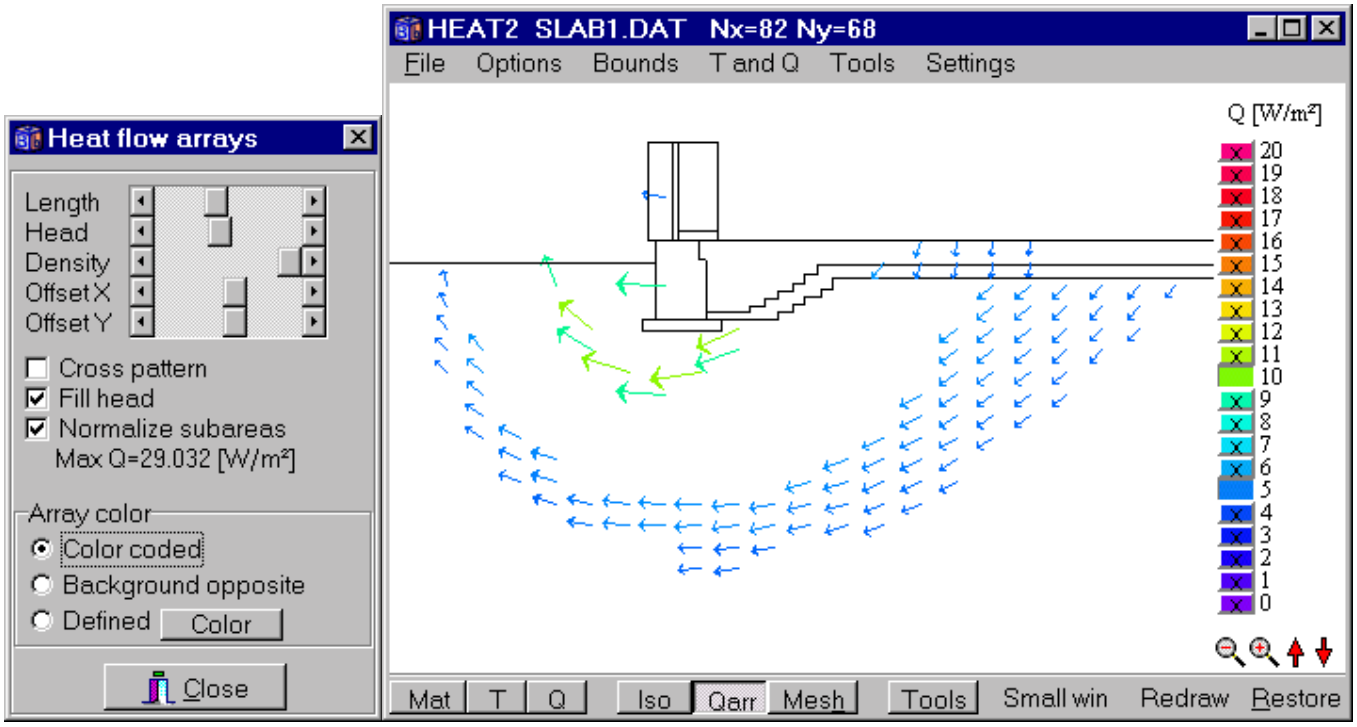


Figure 5.84: Color-coded arrays allow certain intervals to be masked by clicking the scale bars. Heat flow arrays for two intervals between 4.5-5.5 and 9.5-10.5 are shown here.

5.18.6.3.2 Printing heat flow arrays

Printing of heat flow arrays (**File/Print)** will use the printer's resolution (e.g. 600 DPI). The thickness of the heat flow arrays may be chosen in the print dialogue, see Figure 5.85.

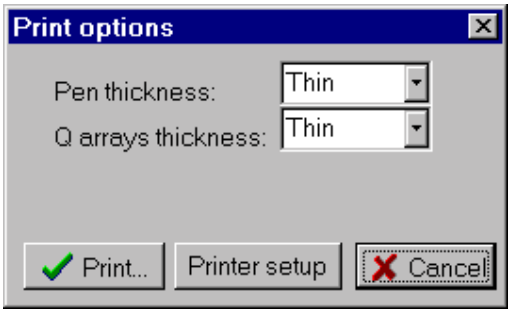


Figure 5.85: The thickness of the heat flow arrays may be chosen in the print dialogue.

5.18.6.4 Viewing isotherms

Figure 5.86 shows options for isotherms (*Temperatures/Options for isotherms* or short-cut by right-clicking on the menu bar item **Iso**). One example is shown in Figure 5.87.

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.

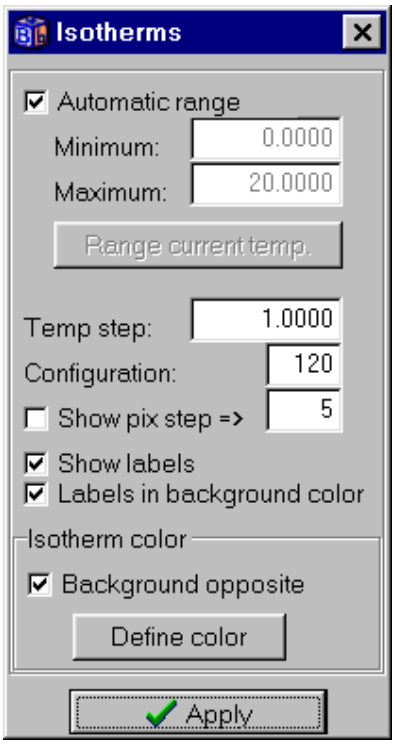


Figure 5.86: Isotherm settings.

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*.

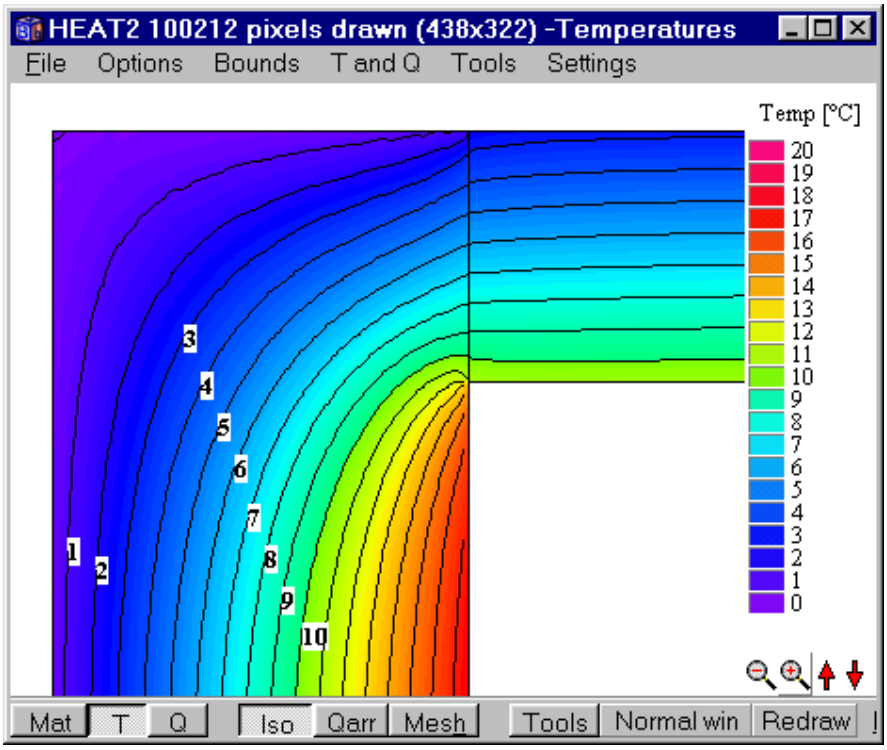


Figure 5.87: Isotherms with interval of 1 °C labeled in background color.

5.18.6.5 Inverted gray-scale

Inverted gray-scale, see Figure 5.88, has been added in the **T and Q/Scale options** menu (also right-clicking on the **T** and **Q** buttons on the bottom tool bar.

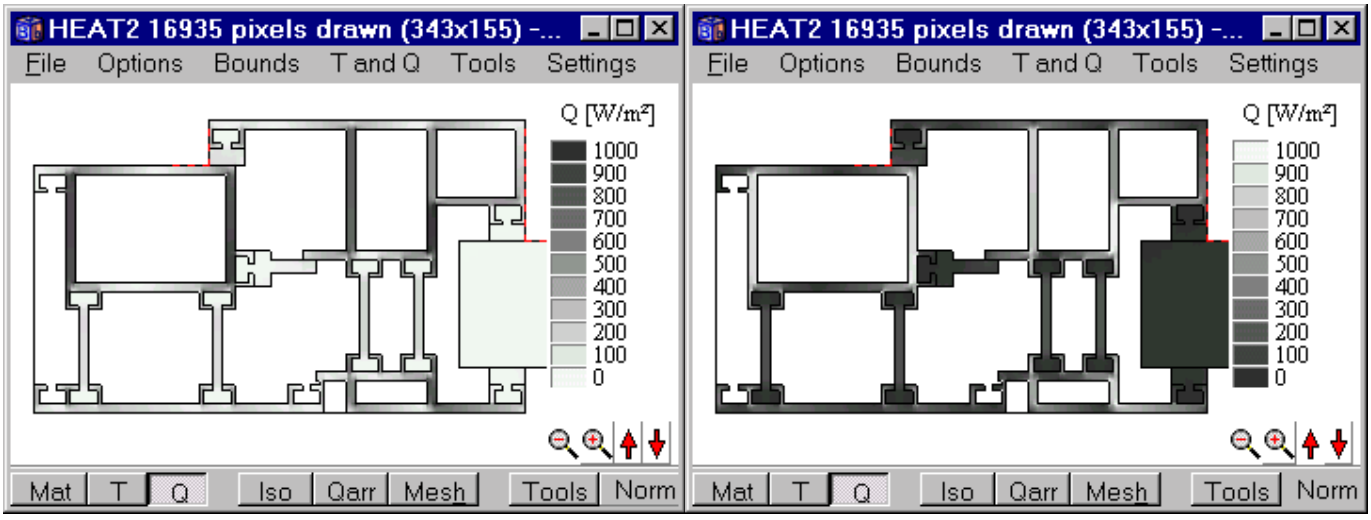


Figure 5.88: Normal gray-scale (left) and inverted (right) of the heat flow magnitude.

5.18.6.6 Showing temperatures and heat flows in cavities

Temperatures and heat flows in cavities may be shown graphically by enabling menu item **T and Q/Show T and Q in cavities** in the **Graphics** menu, see Figure 5.89. This applies to the following types of cavities:

- Air hole (internal modification type E)

- Hole with a fluid with capacity (internal modification type F)
- Radiation and ventilation inside cavities (internal modification type G)
- Window frame cavities

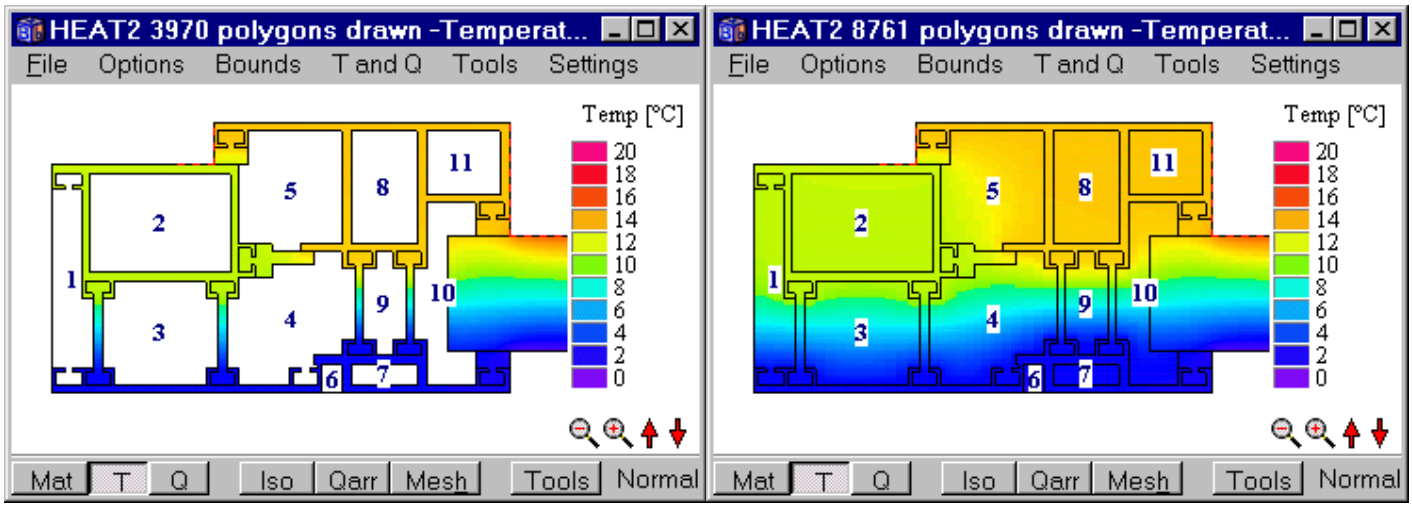


Figure 5.89: Temperatures shown in cavities (right figure).

5.18.7 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 5.90. 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 8.3 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.

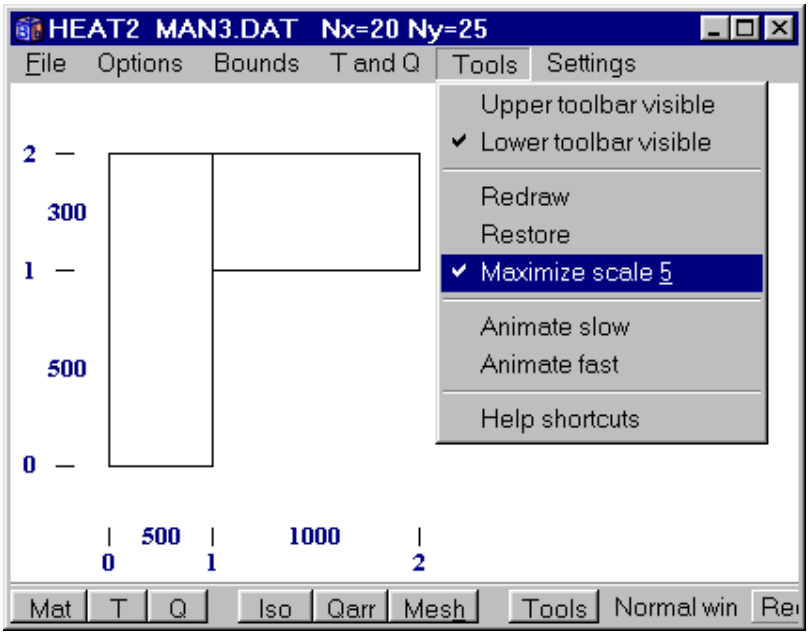


Figure 5.90: The scale in each of the two directions is adjusted to the maximum length in its own direction ('maximized area').

5.18.8 Menu item *Settings*

Fonts and colors may be changed in item *Settings*, see Figure 5.91.

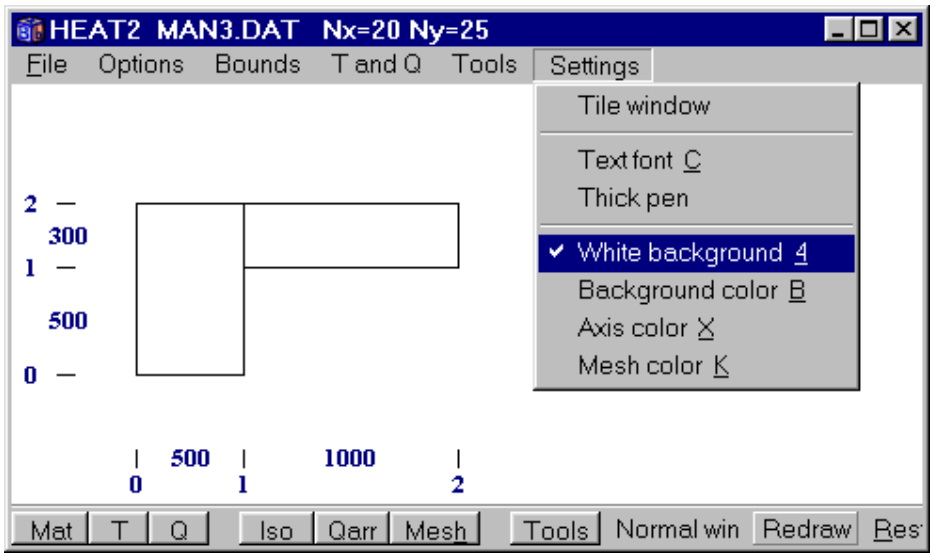


Figure 5.91: Fonts and colors may be changed in item *Settings*.

5.18.9 Menu item *Plot3D*

Surface plots of temperatures and heat flows may be drawn, see Figure 5.92. The picture can be moved, rotated and zoomed.

Figure 5.93 shows the *Options* menu shown and a wire frame surface plot of the temperatures.

If *Automatic scale range* is checked, the temperature (or heat flow) scale will be fitted to min and max values. If *Use defined scale range* is checked, the min and max value are the same as the ones specified in the post-processor (2D).

The surface plot is generated for a specified equidistant grid (not the same as the numerical mesh), e.g. 50*50 points. If Automatic matrix is checked, the number of grid points will be the same as the number of cells in the current numerical mesh, but at most 30*30. A grid with a larger number of points may be slow to draw.

The image may be copied to the Clipboard or saved as an EMF (enhanced meta file).

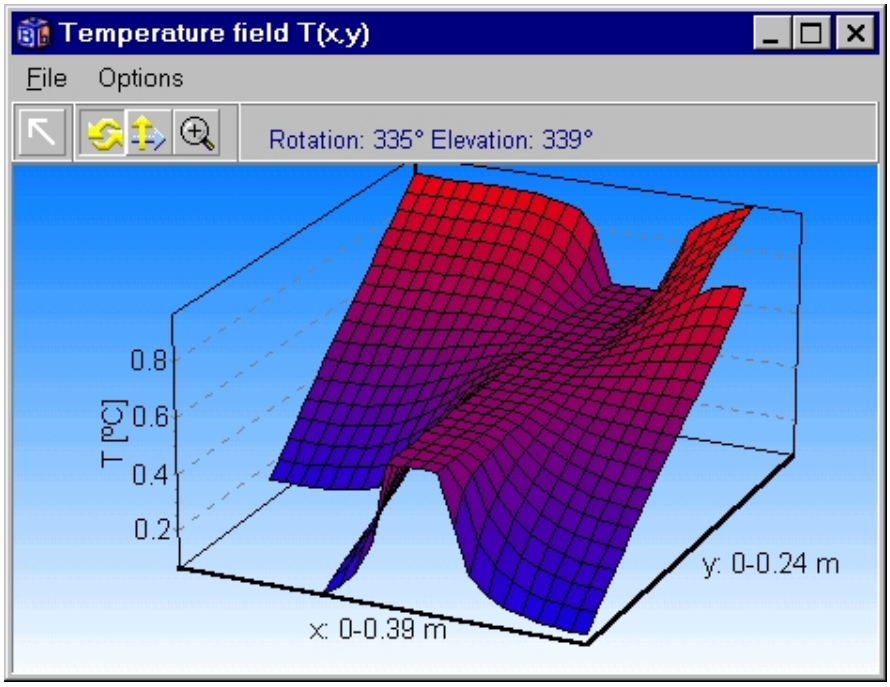


Figure 5.92: A solid surface plot of the temperatures.

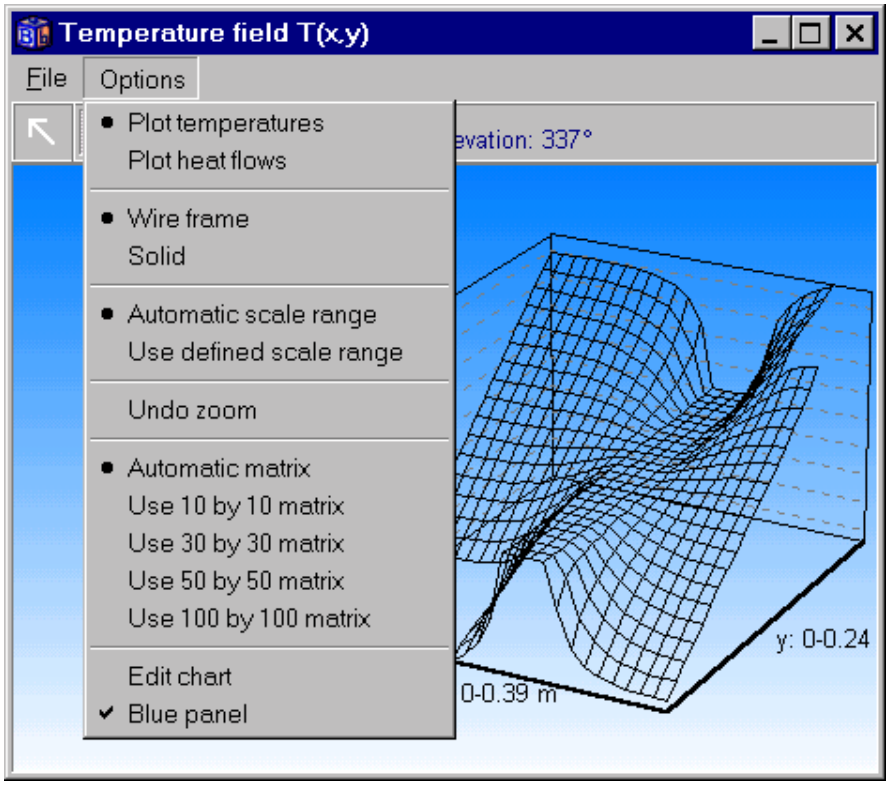


Figure 5.93: Options shown with a wire frame surface plot of the temperatures.

6. Calculations using pipes

6.1 Pipe with a given heat flow

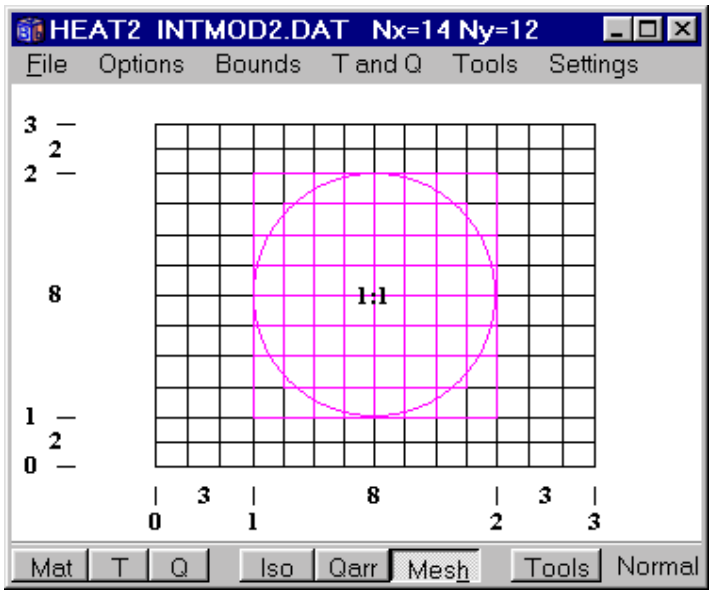
6.1.1 Pipe approximated with a square

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 5.8.

6.1.2 Pipe approximated with enclosing steps

The modification *pipes*, see Section 5.8, may be used which will approximate the boundary of the pipe by small steps. Consider Figure 6.1. 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 white space in Figure 6.1 (right). The number of cells is dependent on the defined numerical mesh. The number of specified cells for the mesh segment is 8·8=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 drawback is that many cells are needed for the calculations. The square approximation is sufficient in most cases.



6.2 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).

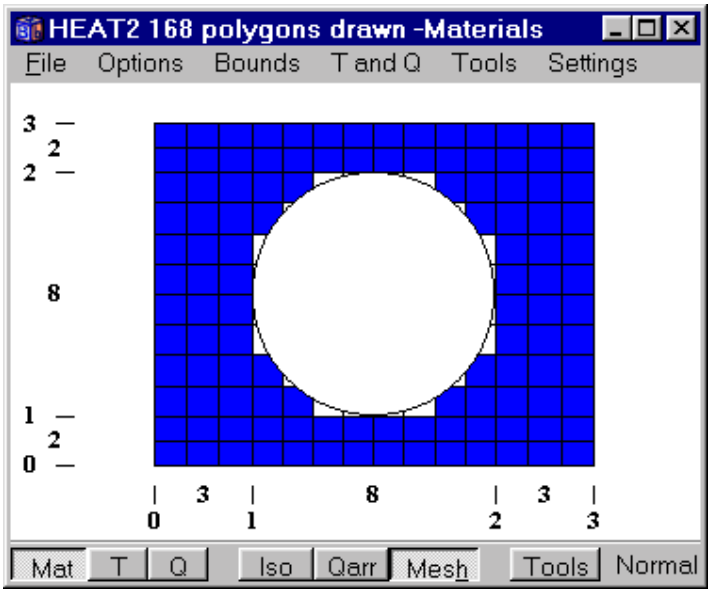


Figure 6.1: The modification *pipes* may be used to approximate the boundary of the pipe by small steps depending on the numerical mesh.

6.2.1 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 6.2.

$$D = \sqrt{\pi} \cdot r \approx 1.77 \cdot r \tag{8.1}$$

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 5.10.

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

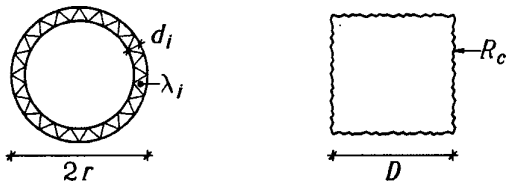


Figure 6.2: An insulated pipe may be approximated by a square and a resistance.

6.2.2 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 6.1.2. The temperature will then be given in all the cells that have 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.

7. Numerical performance

7.1 Benchmarks

HEAT2 version 4 and 5 has increased code optimization giving higher calculation speed. The improvement is greater for Pentium II and III than for ordinary Pentium processors. Table 7:1 and Table 7:2 show benchmarks for the slab problem (slab1.dat) and the corner problem (corner.dat) using Pentium MMX, II and III.

HEAT2 version 4 and 5 is for these two problems 11-23 % faster for steady-state calculations and 14-36% faster for transient analysis compared with HEAT2 version 3. Note that the improvement is greater for the Pentium II and III, especially for transient analyses.

The Pentium III 450 is about 3-4 times faster than the MMX 266 but only 10-30% faster than the Pentium II 400 for these benchmarks.

Table 7:1: Benchmarks for the slab problem (slab1.dat).

Slab1.dat, *N*=4555	**HEAT2 3.0 CPU-time (sec.)**	**HEAT2 4.0 & 5.0 CPU-time (sec.)**	**Improvement 3.0 => 4.0 & 5.0**
Steady-state, 50 000 iterations:			
Pentium III 450 MHz	75	61	23% faster
Pentium II 400 MHz	85	69	23% faster
Pentium MMX 266 MHz	208	188	11% faster
Transient, 1 month (97 032 iter.):			
Pentium III 450 MHz	103	76	36% faster
Pentium II 400 MHz	124	91	36% faster
Pentium MMX 266 MHz	272	236	15% faster

Table 7:2: Benchmarks for the corner problem (corner.dat).

Corner.dat, *N*=675	**HEAT2 3.0 CPU-time (sec.)**	**HEAT2 4.0 CPU-time (sec.)**	**Improvement 3.0 => 4.0 & 5.0**
Steady-state, 200 000 iterations:			
Pentium III 450 MHz	47	40	18% faster
Pentium II 400 MHz	55	46	20% faster
Pentium MMX 266 MHz	136	120	13% faster
Transient, 3 months (291 095 iter.):			
Pentium III 450 MHz	46	36	28% faster
Pentium II 400 MHz	59	46	28% faster
Pentium MMX 266 MHz	168	148	14% faster

8. Examples

8.1 Example 1 - A wall with a joist

8.1.1 Introduction

Figure 8.1 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.

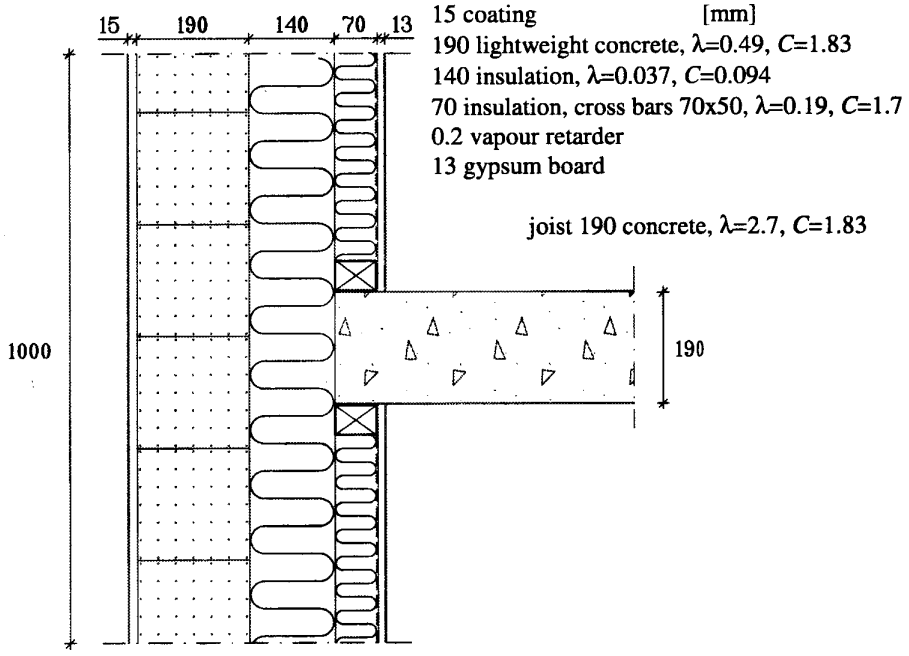


Figure 8.1: A non-supportive wall with a thermal bridge.

8.1.2 Input using the pre-processor

The instructions below show how input can be made using the pre-processor, see Figure 8.2. If you have any problem or do not get the same answer, look at the input file *Example 1.dat*. In this example we neglect the small thermal resistance of coating and gypsum boards.

1. Pick 'Example 1, concrete lightweight' In the material list. Draw a rectangle somewhere in the drawing area.
2. Choose the *Select* tool (the leftmost icon on the tool-bar). Click on the rectangle (which now will be marked on the corners). The rectangle may now be dragged and resized. The width and height is displayed on the bottom row. It should be dx=0.19 m and dy=1 m.
3. Press the *Draw* tool on the tool-bar (right-clicking will toggle between the select and draw tool) . Pick material "Example 1, insulation". Draw another rectangle next to the first one and resize it to dx=0.14 and dy=1. Draw the next insulation strip (dx=0.07 and dy=1).
4. Draw and give the dimensions for the concrete (pick "Example 1, concrete" in the material list). Place the concrete rectangle as in the figure. The vertical placement does not have to be exact since the horizontal heat flow through the wall is one-dimensional a couple of decimeter from the concrete slab.

5. Draw one cross bar (material ‘Example 1, cross bars”). Give the dimensions (0.07*0.05). Copy the selected cross bar by pressing Ctrl-V (or the insert button). Place the wooden cross bars at the right locations.
6. Press menu item <*Update not made*>.

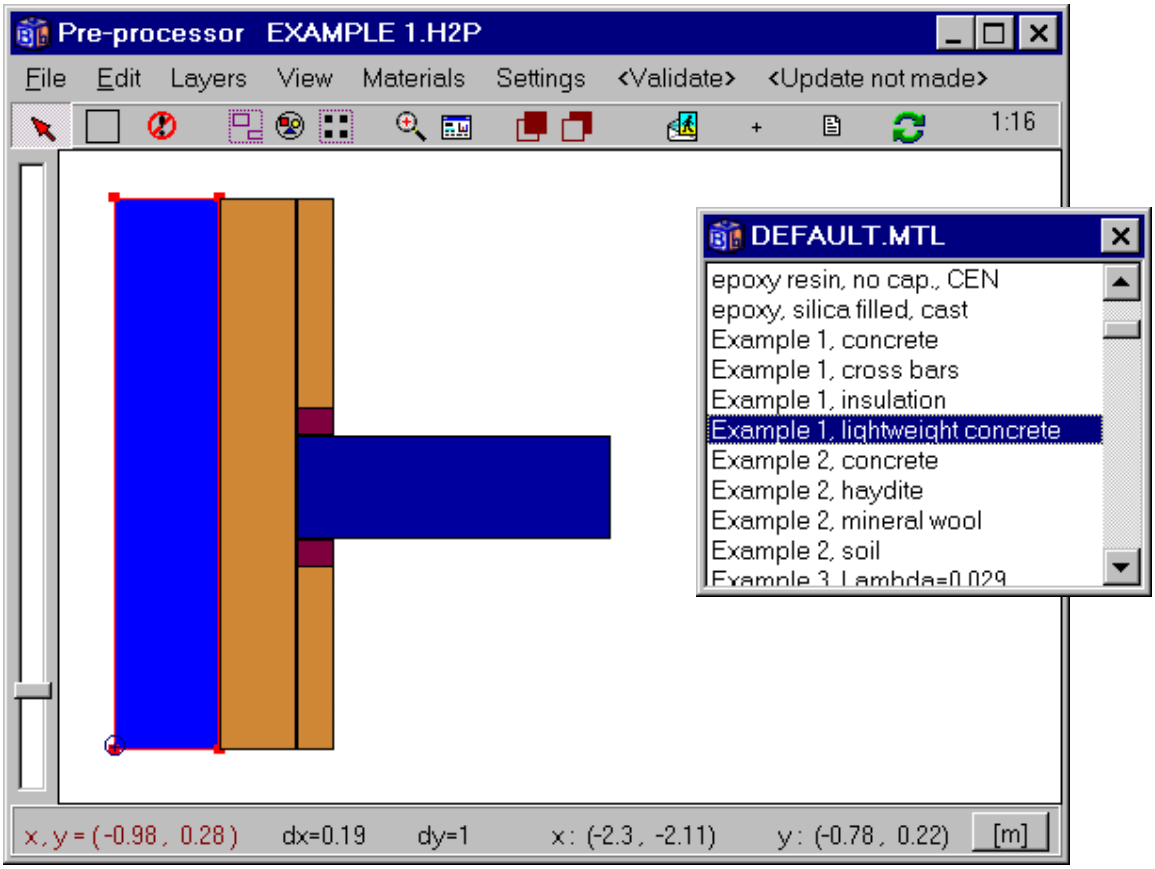


Figure 8.2: The construction drawn in the pre-processor.

8.1.3 Boundary conditions

1. Figure 8.3 shows input mesh (check *Options/Input mesh coordinates*), lengths (*Options/Lengths mm*), boundary segment numbers, and material names. Click **Mesh** on the bottom tool bar to display the generated numerical mesh.
2. The boundary conditions are now to be specified. Figure 8.4 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 8.3. The third type is linked to boundary 8. The other boundaries (1, 4 and 7) will have the default (first) type which is adiabatic.

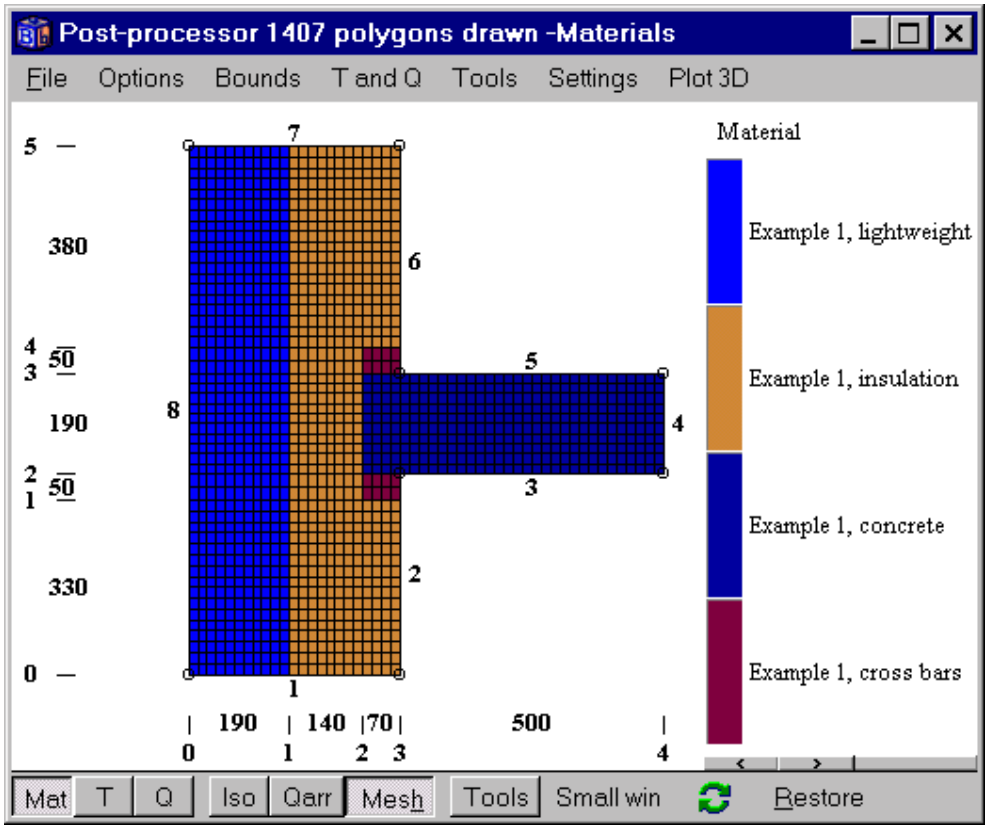


Figure 8.3: Materials, input mesh, lengths, numerical mesh and boundary segment numbers.

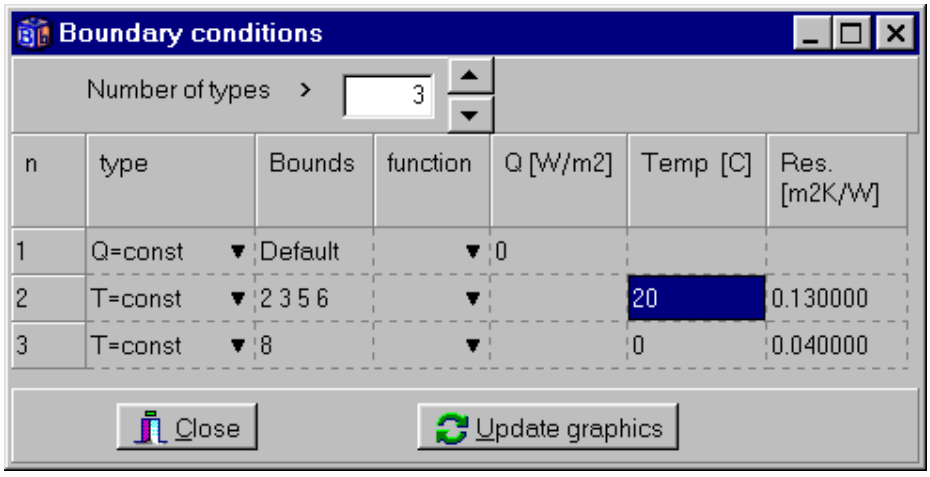


Figure 8.4: Three boundary conditions are defined here. The second type is linked to boundaries 2, 3, 5 and 6, see Figure 8.3. The third type is linked to boundary 8. The other boundaries (1, 4 and 7) will have the default (first) type which is adiabatic.

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 8.5. This will take a few seconds on a Pentium.

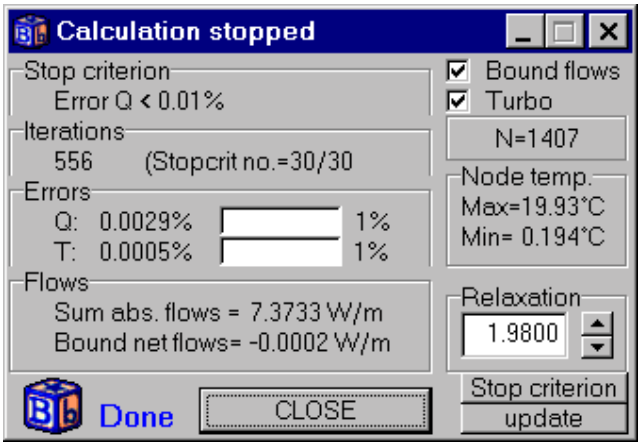


Figure 8.5: Steady-state is reached within a few seconds.

Go to *Output-Boundary Flows*. A list shows the heat flow through the 8 boundary segments, see Figure 8.6. The heat flow for boundary 8 is Q=-3.69 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 1407. Using about 200 cells instead will give a heat flow of 3.67 W/m, i.e. a relative error of less than 1%, (3.69-3.67)/3.69=0.005.

Figure 8.7 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.

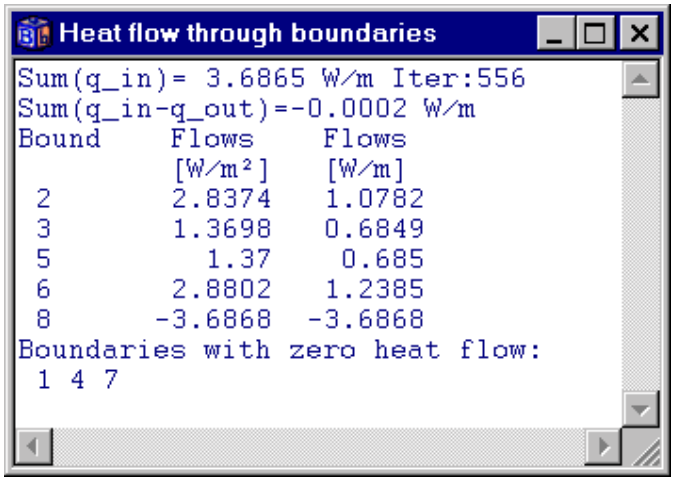


Figure 8.6: Heat flows through the boundaries.

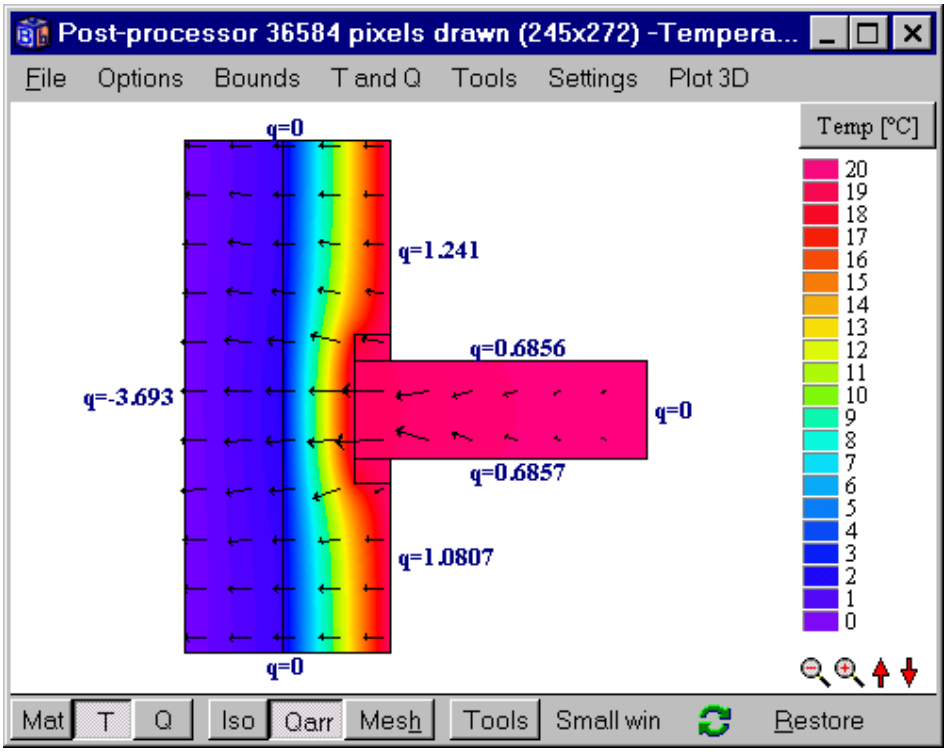


Figure 8.7: Temperature field.

8.1.4 Input without using the pre-processor

Note that all the input could have been made by text only. Figure 8.3 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 8.8 (left).
2. Go to *Description of boundaries*. Give the coordinates for the boundary segments, see Figure 8.8 (right).
3. Go to *Mesh in x-direction*. Give the data as in Figure 8.9 (left).
4. Go to *Mesh in y-direction*. Give the data as in Figure 8.9 (right).
5. Go to *Thermal properties*. Give the data as in Figure 8.10.
6. Continue the input as in Section 8.1.3.

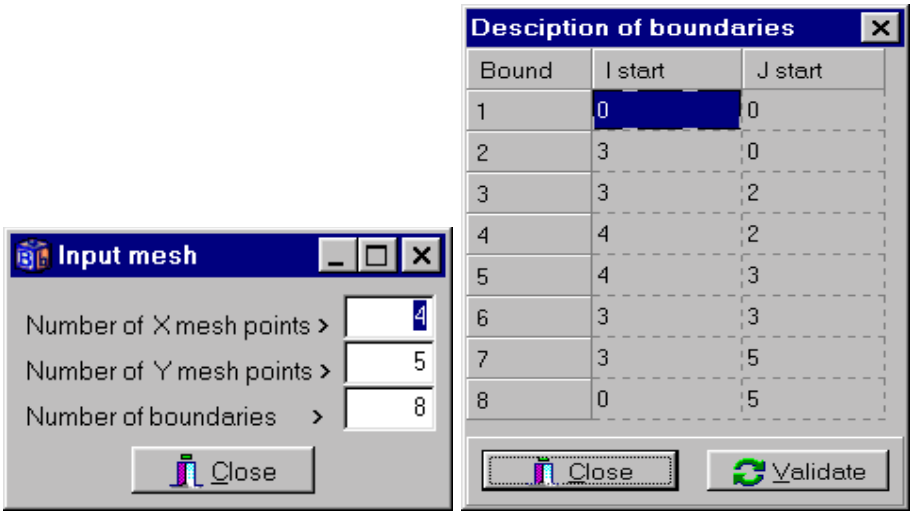


Figure 8.8: Data for input mesh and boundaries.

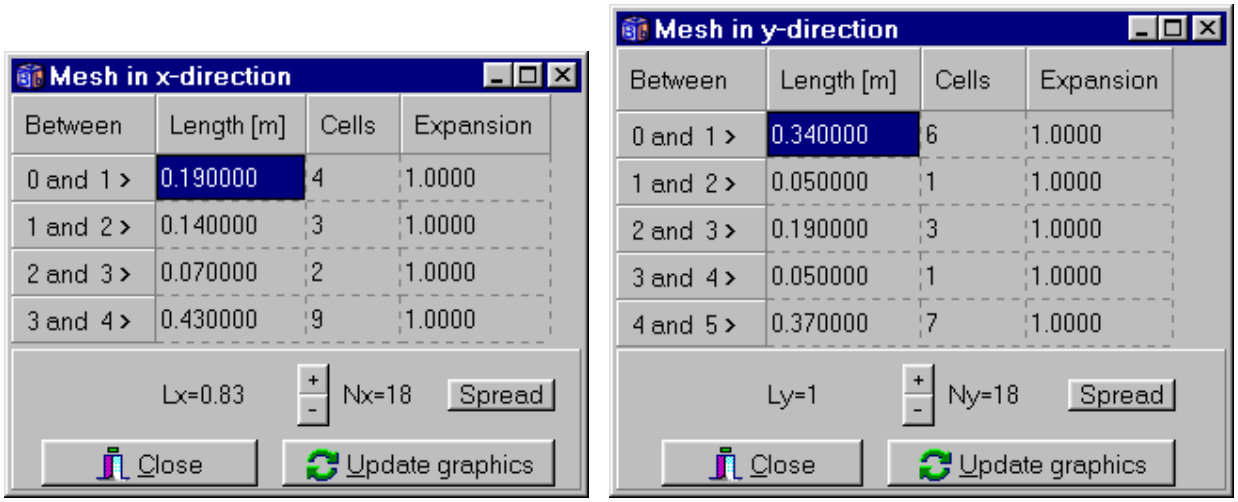


Figure 8.9: Data for lengths and numerical mesh in both directions.

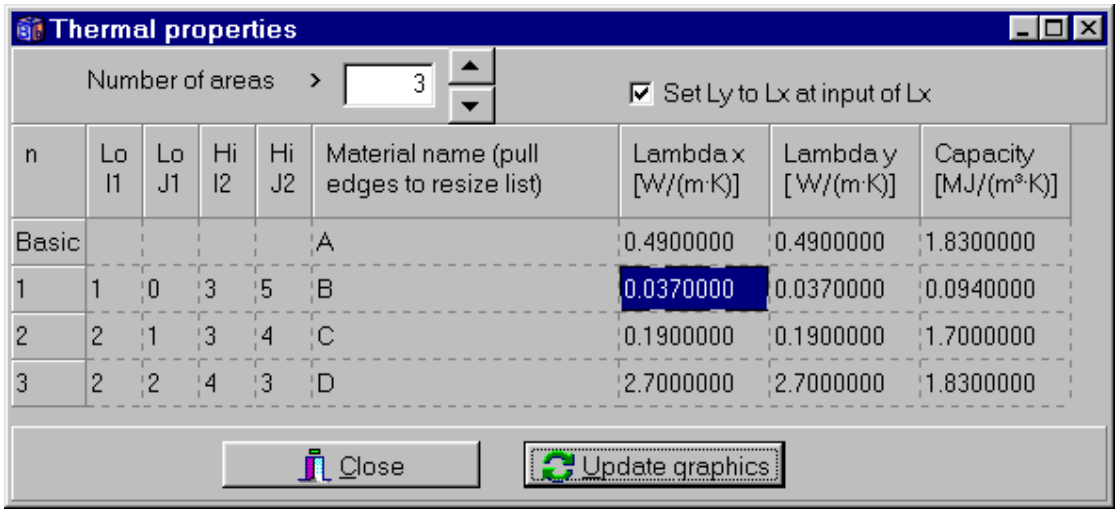


Figure 8.10: Data for thermal properties.

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

8.2.1 Introduction

Figure 8.11 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.

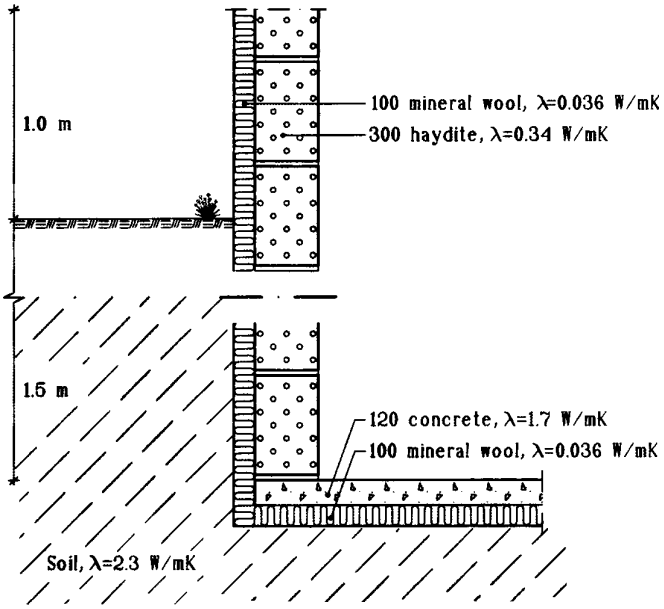


Figure 8.11: The cellar construction.

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 first made without using the pre-processor. See input file *cellar.dat*.

8.2.2 The input mesh

The axes of a chosen input mesh for the example are shown in Figure 8.12. We have a symmetrical case, thus only half of the house is considered.

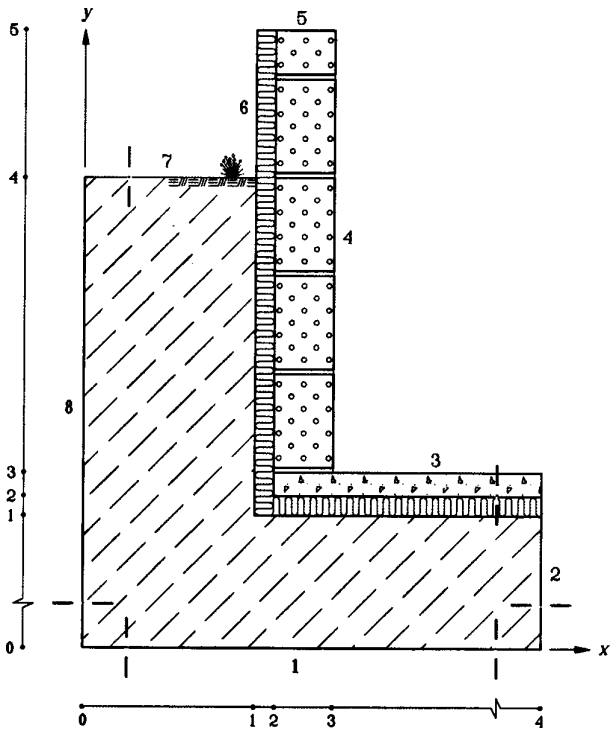


Figure 8.12: An input mesh with enumerated boundary segments.

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 8.13 (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 8.13 (right).
3. The next step is to specify the dimensions, see Figure 8.14. 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. To few cells may lead to inaccurate results. One the other hand, using to 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 8.15 shows the numerical mesh and lengths of boundaries (check *Bounds/Bound length* in the graphics window). Figure 8.16 shows the expansive numerical mesh around the cellar.

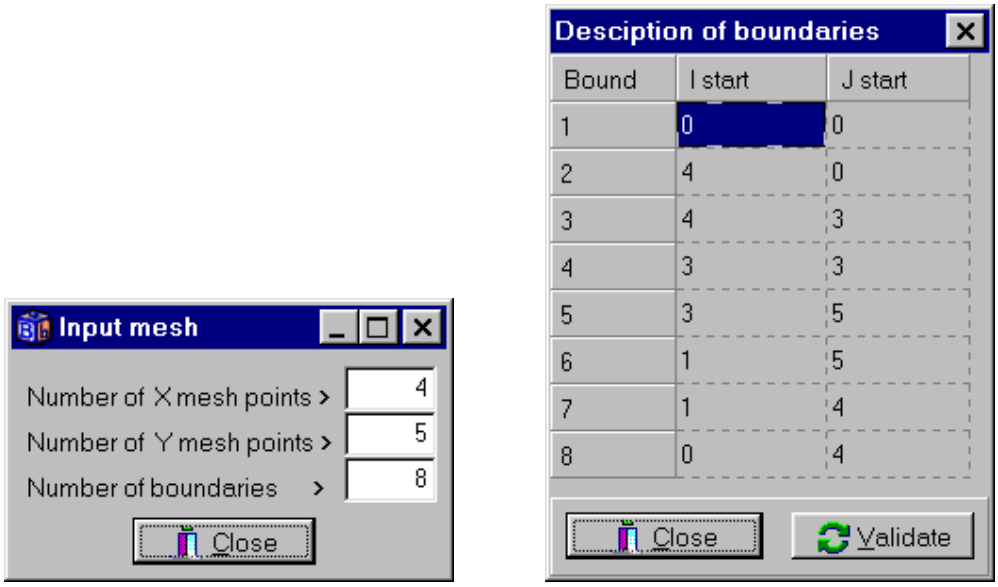


Figure 8.13: Data for input mesh and boundaries.

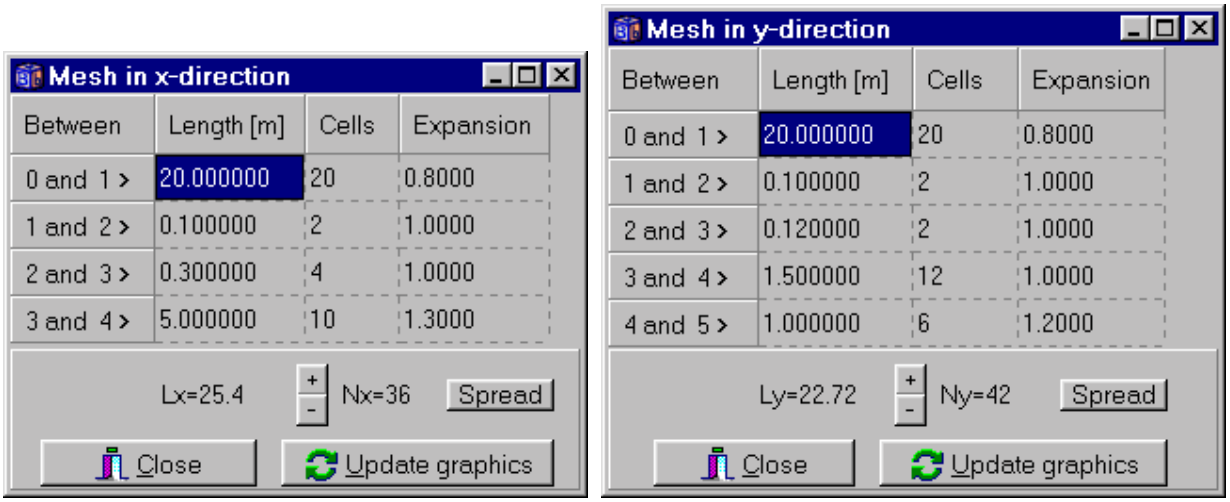


Figure 8.14: Data for lengths and numerical mesh in both directions.

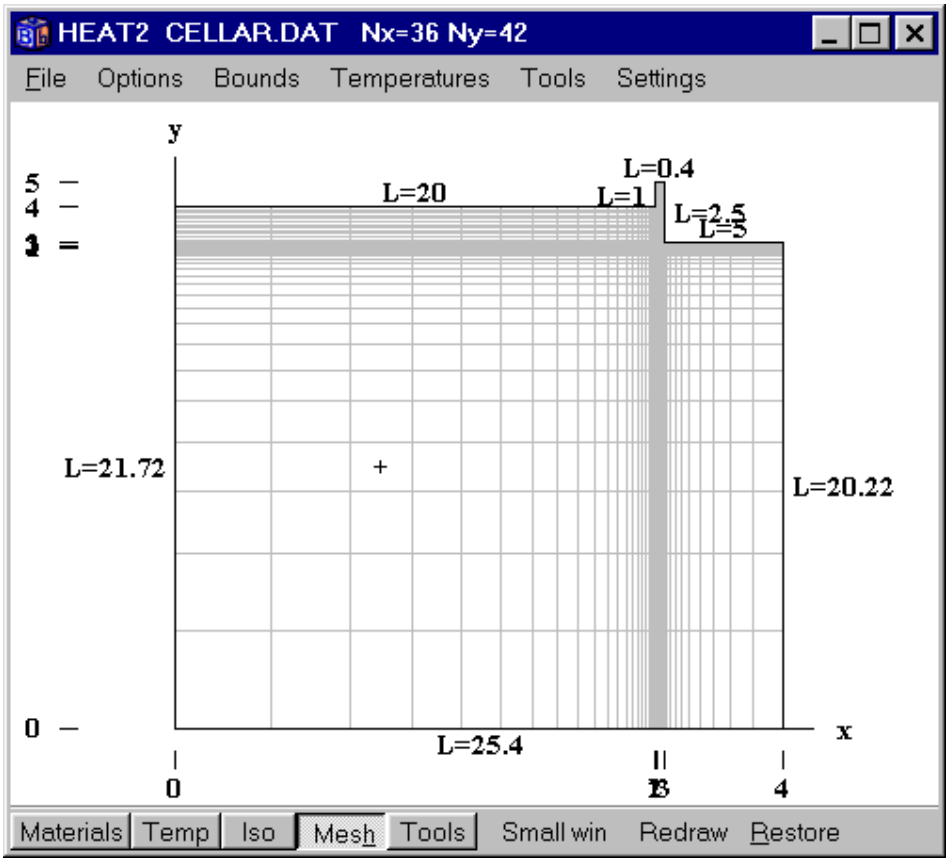


Figure 8.15: Numerical mesh and lengths of boundaries.

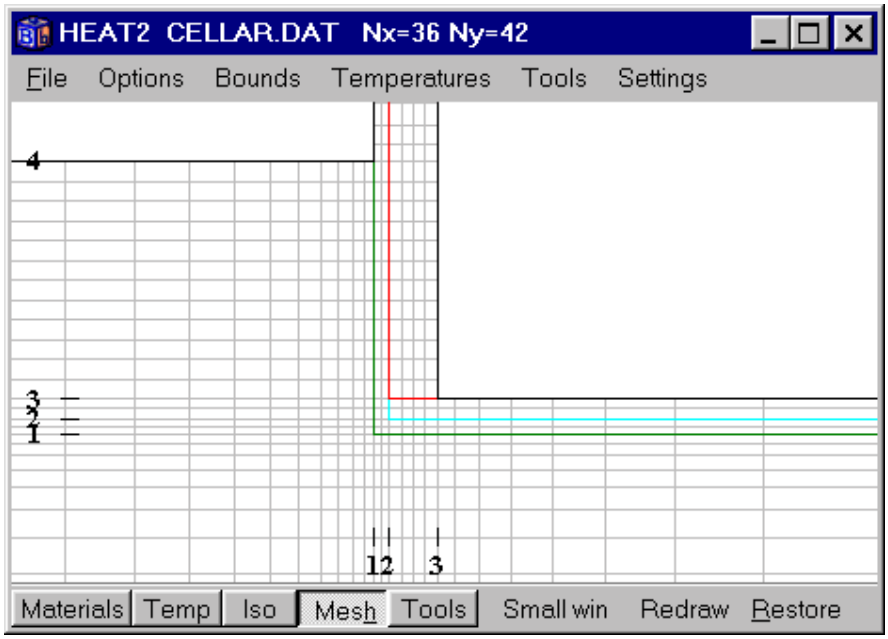
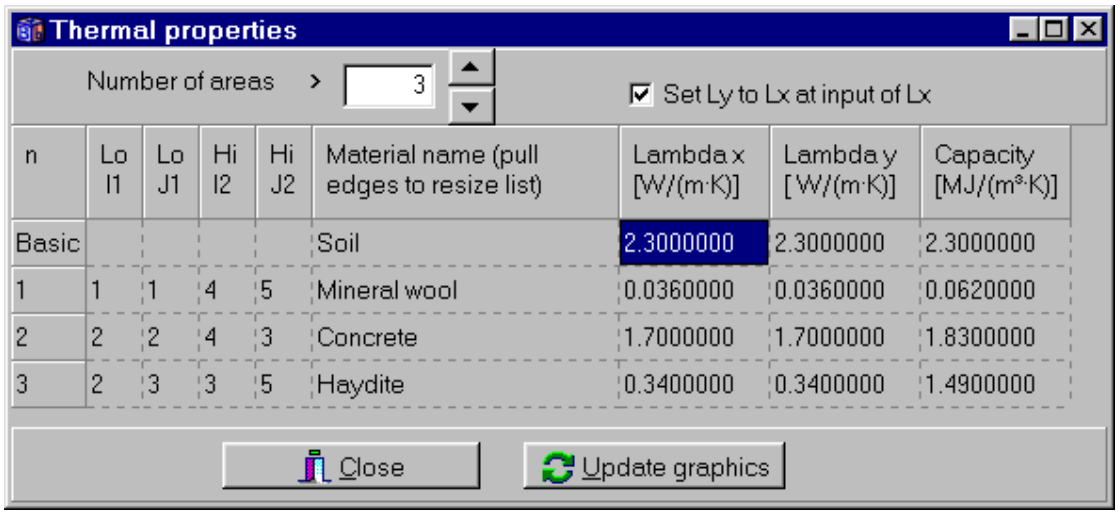


Figure 8.16: Numerical mesh around the cellar.

8.2.3 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 has 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.



n	Lo I1	Lo J1	Hi I2	Hi J2	Material name (pull edges to resize list)	Lambda x [W/(m·K)]	Lambda y [W/(m·K)]	Capacity [MJ/(m³·K)]
Basic					Soil	2.3000000	2.3000000	2.3000000
1	1	1	4	5	Mineral wool	0.0360000	0.0360000	0.0620000
2	2	2	4	3	Concrete	1.7000000	1.7000000	1.8300000
3	2	3	3	5	Haydite	0.3400000	0.3400000	1.4900000

Figure 8.17: Data for thermal properties.

8.2.4 Boundary conditions

There are three types of boundary conditions:

- Adiabatic surface on boundaries 1, 2, 5 and 8, see Figure 8.12
- T=20 °C with a thermal surface resistance 0.13 m²·K/W on boundary 3 and 4
- T=0 °C with a thermal surface resistance 0.04 m²·K/W on boundary 6 and 7

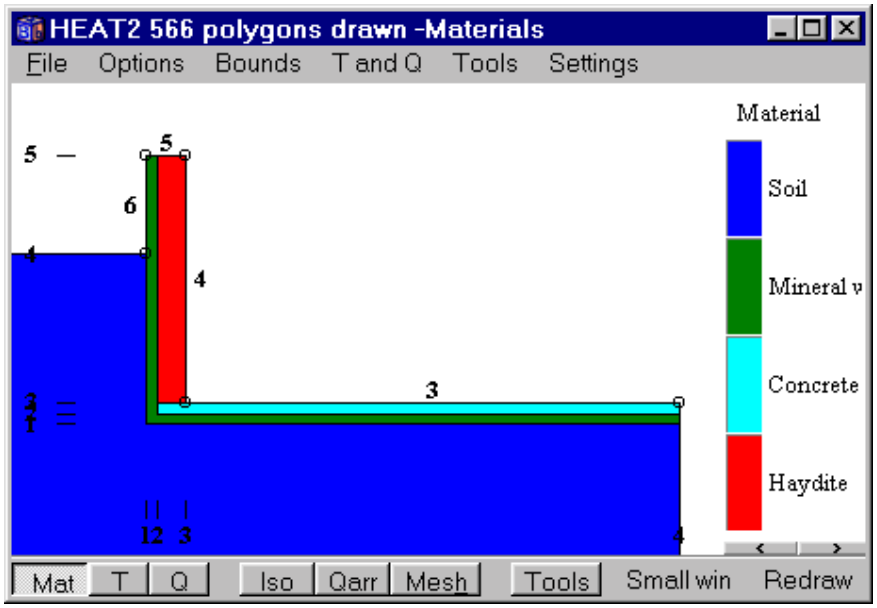


Figure 8.18: Material names.

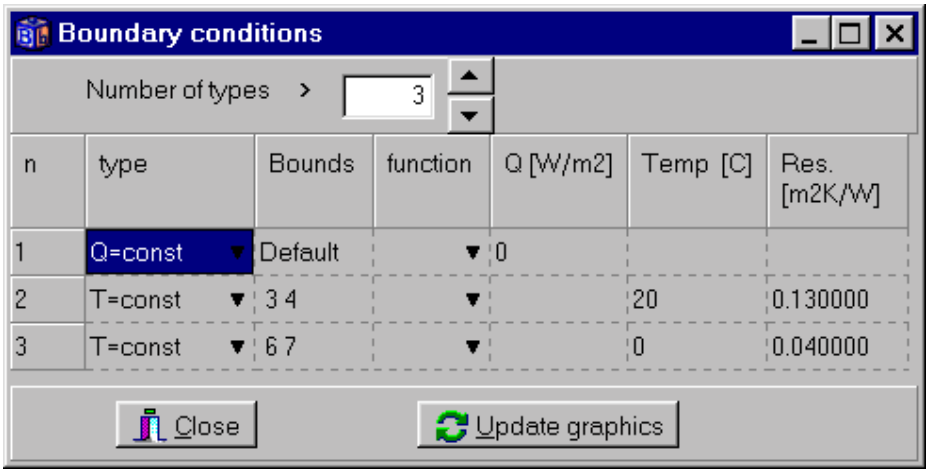


Figure 8.19: Three boundary conditions are defined. The second type is linked to boundaries 3, and 4, see Figure 8.20. The third type is linked to boundary 6 and 7. The other boundaries (1, 2, 5 and 8) will have the default (first) type which is adiabatic.

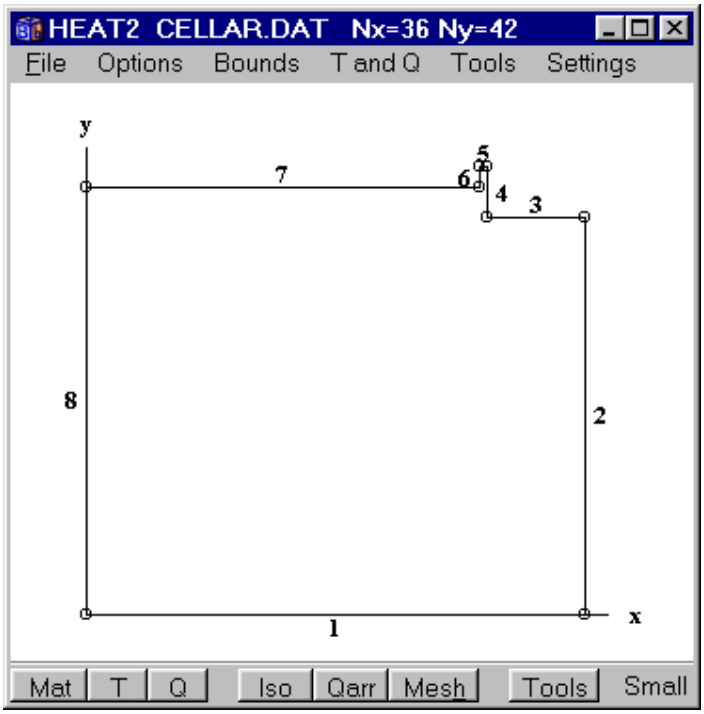


Figure 8.20: The boundary numbers 1..8 are linked to their boundary condition.

8.2.5 The simulation

The steady-state simulation can now be started by pressing F9, see Section 5.11. 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 8.22. The temperature field, isotherms, and boundary heat flows are shown in Figure 8.23.

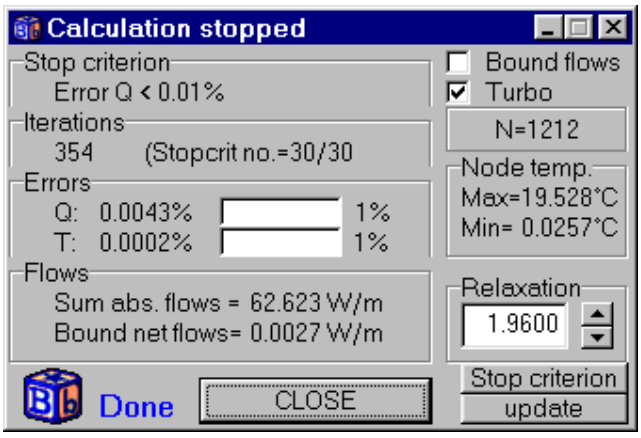


Figure 8.21: The simulation takes a few seconds on a Pentium.

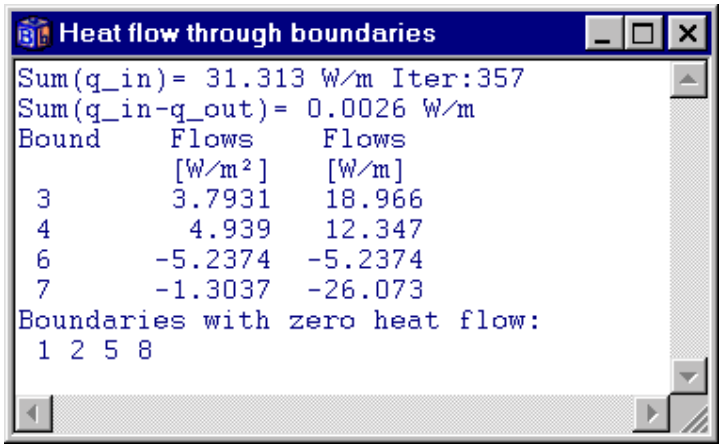


Figure 8.22: Heat flow through boundary segments.

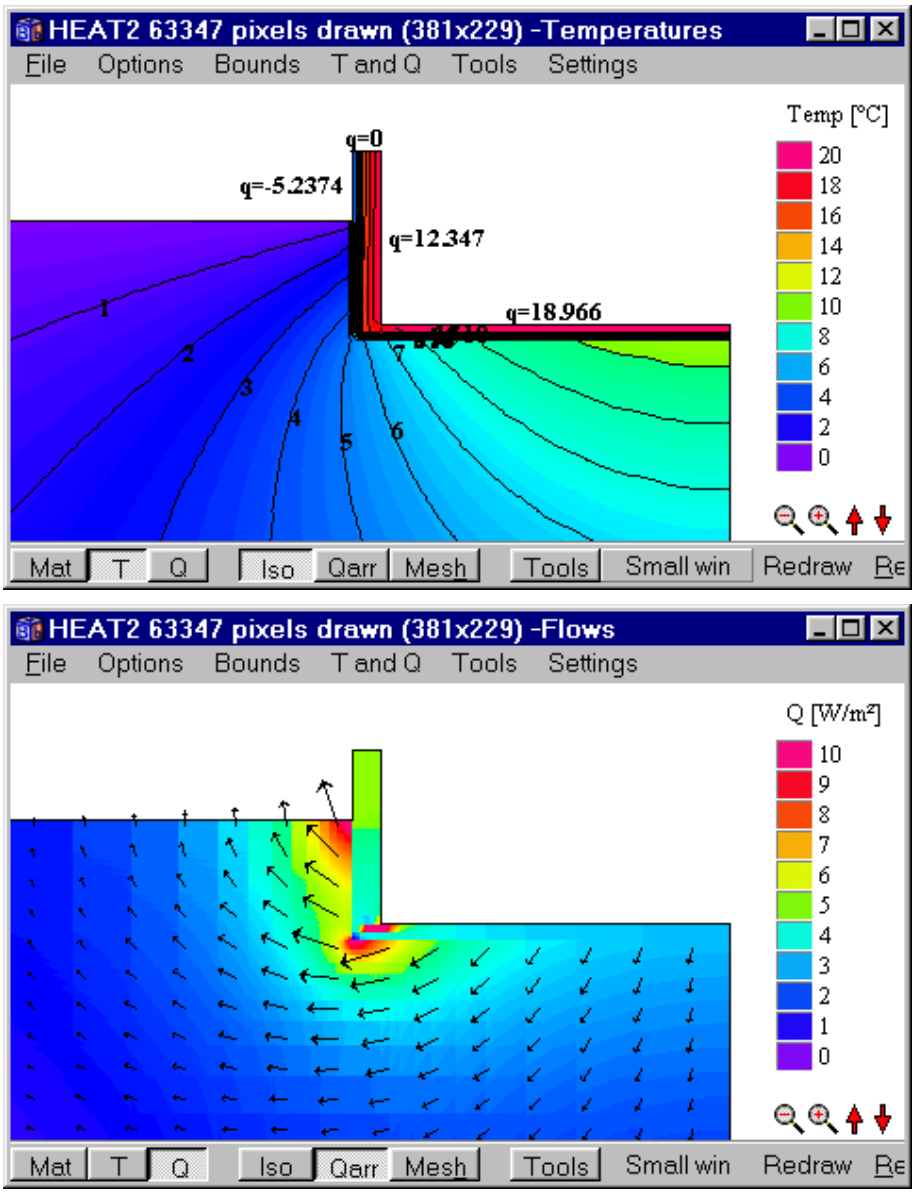


Figure 8.23: Temperature field, isotherms, and boundary heat flows (top figure) and heat flows (bottom figure).

8.2.6 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 8.24. Give initial temperatures (average, undisturbed temperature deep in the ground) of +5° C, see Figure 8.25.

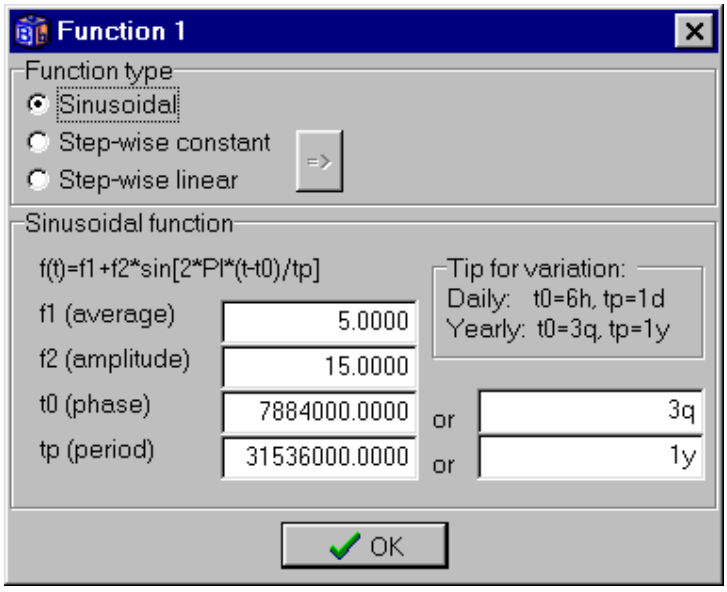


Figure 8.24: A sinus function with a daily variation between -10 and +20 °C with maximum occurring at noon. The time for the phase and period may be given either in seconds or with a time string.

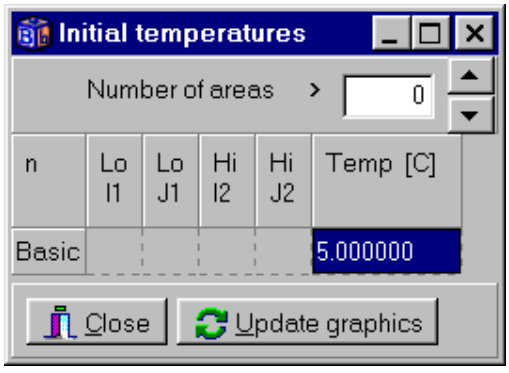


Figure 8.25: Initial temperatures.

Next step is to couple the function to a boundary condition type (*Input/Boundary conditions*), see Figure 8.26. Give the stop time 10 years (*Solve/Options for transient*), see Figure 8.27.

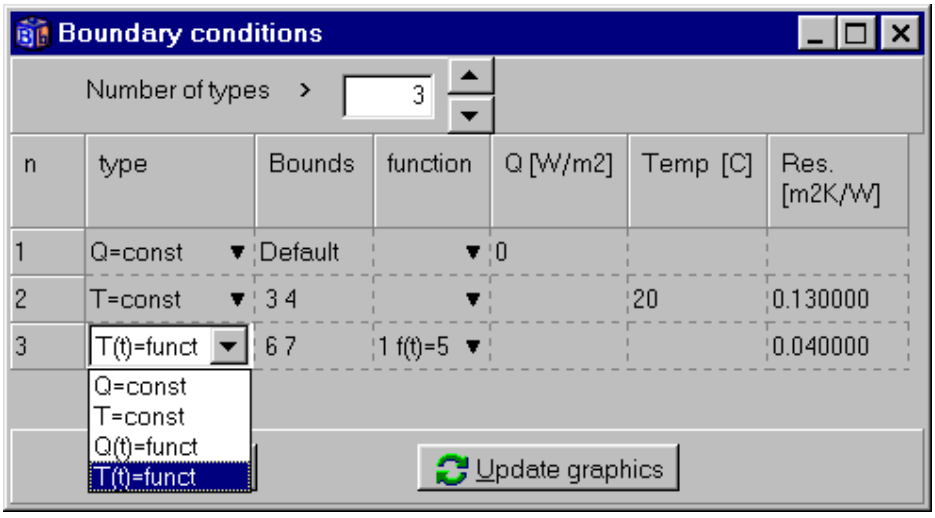


Figure 8.26: A temperature function is chosen in the pick list for the boundary condition type. The function number is selected in the function pick list (number 1 is default).

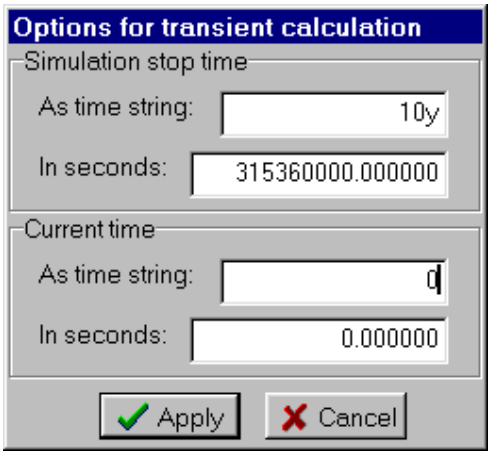


Figure 8.27: The stop time is 10 years.

In this example we will also look at the flows through the internal boundaries during each hour. Enable the record (*Output/Recorder*) by clicking *Record enable* and pick the second item in the list as in Figure 8.28 (top). Click the *Data* entry field and give the numbers 3 and 4 for the internal boundaries as in Figure 8.28 (bottom). This will record the sum of the two boundary heat flows at each screen update (*Solve/Update*) as given in Figure 8.29, 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).

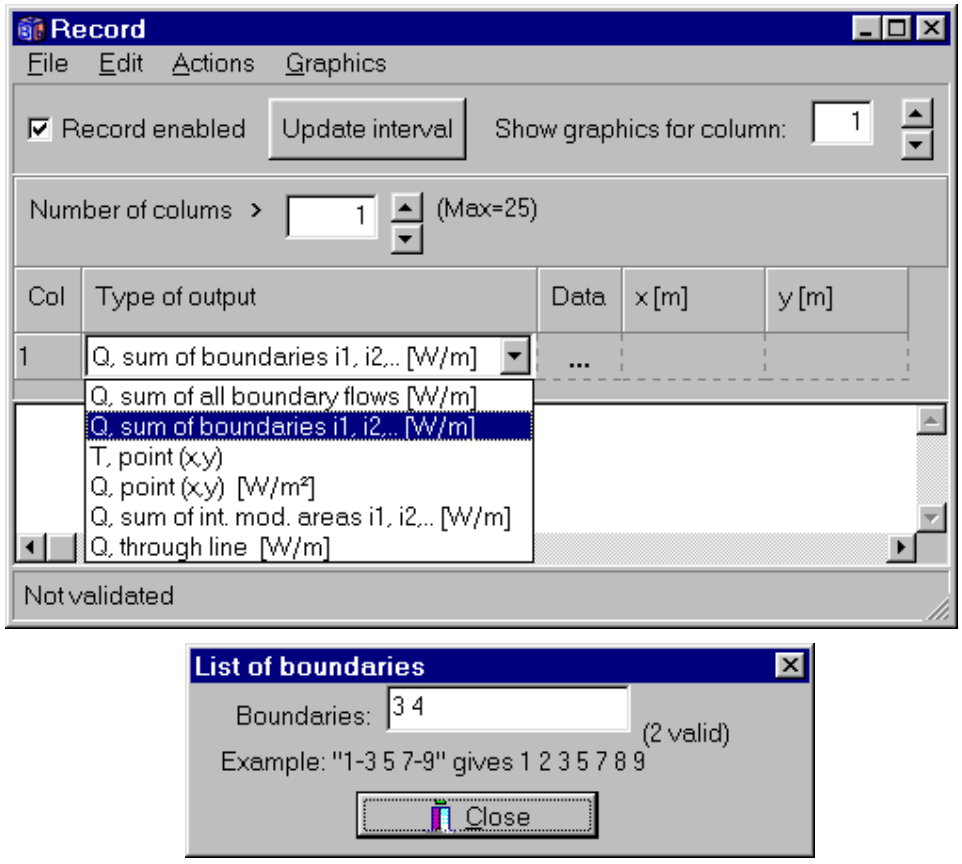


Figure 8.28: Flows and temperature may be saved to a list during the simulation. In this case it is the sum of the heat flows for boundaries 3 and 4 that will be recorded.

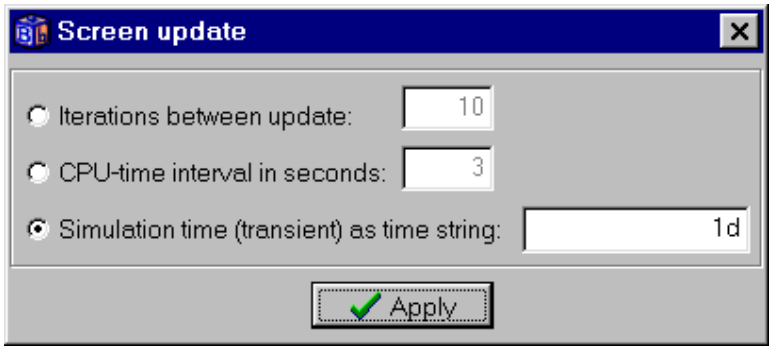


Figure 8.29: The time interval for recorded data is given in item *Solve/Update*).

To start the transient simulation select *Solve/Start transient calculation*. Consider

Figure 8.30. 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.

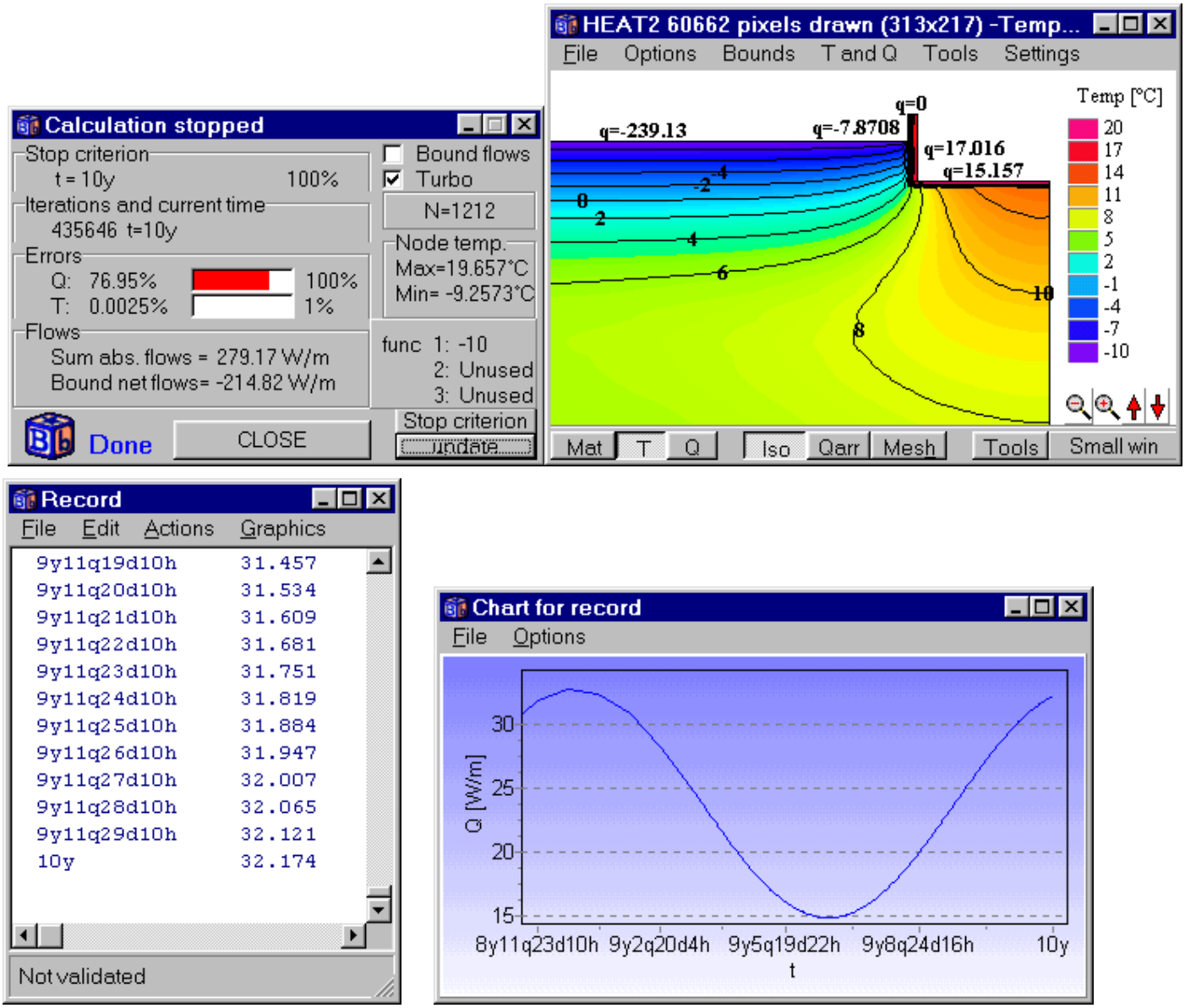


Figure 8.30: *A window (top left) will be displayed showing information such as the chosen stop criterion, number of iterations, and error for temperature and flows. The record window (bottom left) and the chart window (bottom right) show the internal heat flows during each day. The minimum heat flow occurs in the end of July (6q20d). See also Figure 8.31. The time to simulate10 years is a few minutes on a Pentium II (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 11 years, and remove boundary 4 from the recorder, see Figure 8.28 (right). Continue the simulation for the last year, see Figure 8.32.

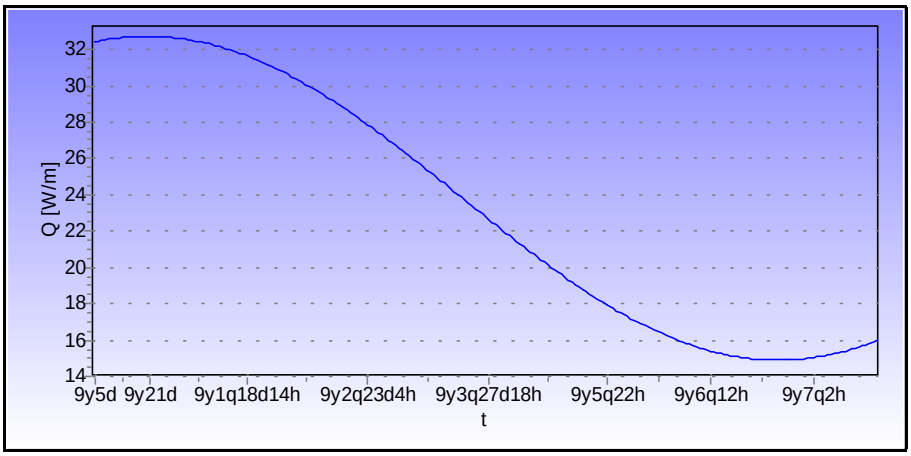


Figure 8.31: The max and min heat flow through the internal boundaries occur in the middle of January "9y20d" (32.7 W/m) and in the middle of July "9y6q20d" (14.9 W/m), respectively. Note that the string "9y6q20d" means that 6 months and 20 days has elapsed on the ninth year, i.e. the middle of July. The max and min outdoor temperature occur at t=0 (Jan 1) and at t="6q" (July 1), respectively. [3]

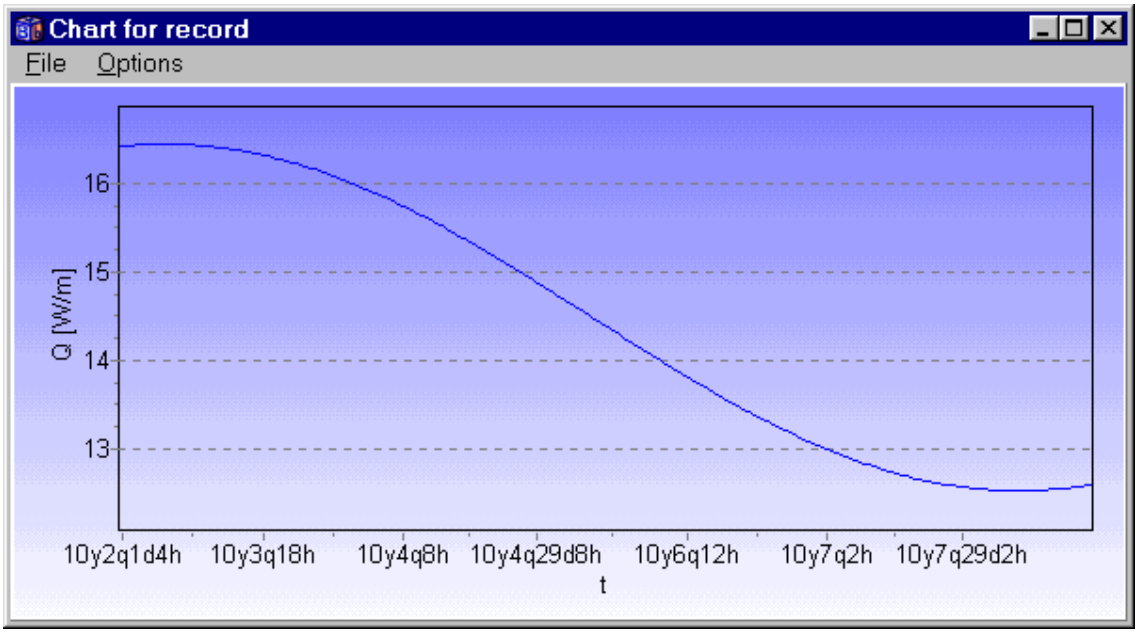


Figure 8.32: The maximum and minimum heat flow through the floor occurs in the middle of March "10y2q10d" (16.5 W/m) and in the middle of September "10y8q10d" (12.5 W/m), respectively. Note the time delay between extreme outdoor temperatures and extreme heat flows through the floor of about 2.5 months.

8.2.7 Input using the pre-processor

8.2.7.1 Drawing the whole geometry

Figure 8.33 shows the geometry given in the pre-processor (see file *Example 2 pre 1.dat*). Boundary conditions are specified as before. The heat flow becomes 31.1 with 2473 cells in the equidistant mesh

[3] The chart in Figure 8.31 was copied to the clipboard in metafile format (EMF). The chart in Figure 8.32 was copied to the clipboard in bitmap format using Alt-Print Screen, see Section 5.18.3.1.

generated by the pre-processor (50*50 cells), Figure 8.34. The difference is less than 1 percent as compared with the result for the expansive mesh 31.3, see Figure 8.16.

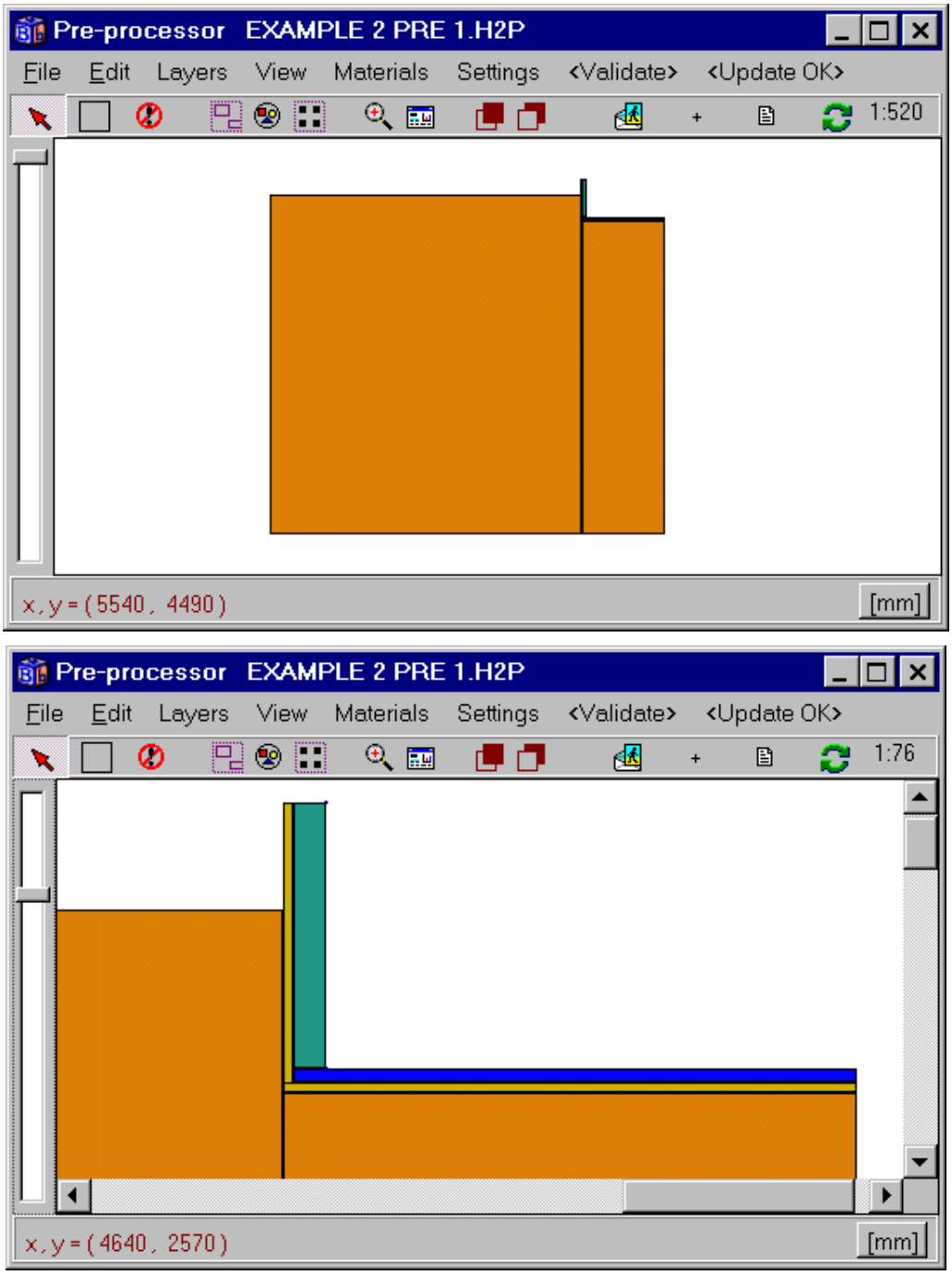


Figure 8.33: Whole geometry (top) and enlarged area (bottom).

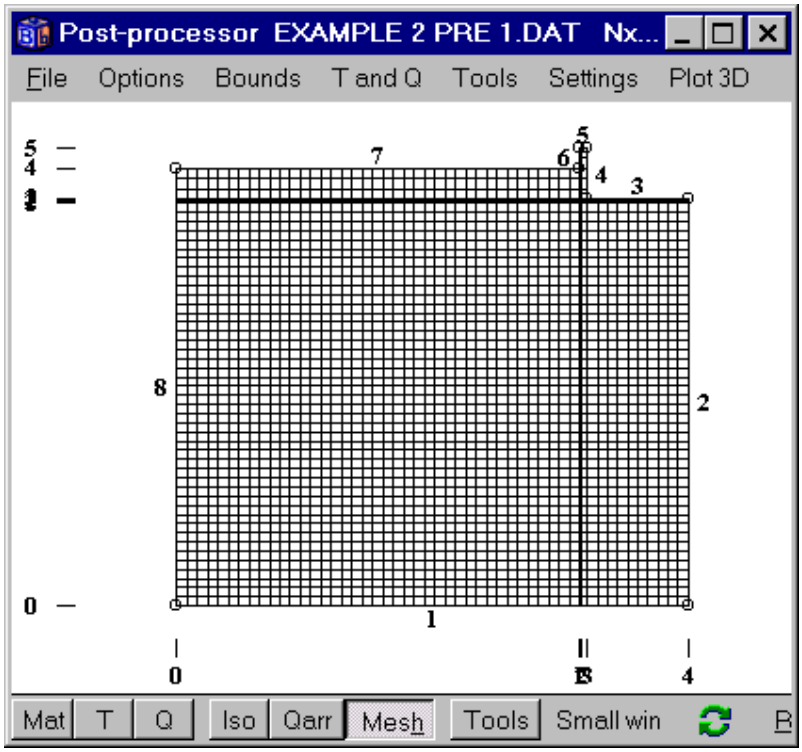


Figure 8.34: The heat flow becomes 31.1 with 2473 cells in the equidistant mesh generated by the pre-processor (50*50 cells).

8.2.7.2 Drawing parts of geometry

It may be quicker to draw only parts of the soil in the pre-processor, see Figure 8.35, and afterwards change the size for the soil by text input in menu items *Input/Mesh in x-dir* and *Input/Mesh in y-dir*, see Figure 8.36.

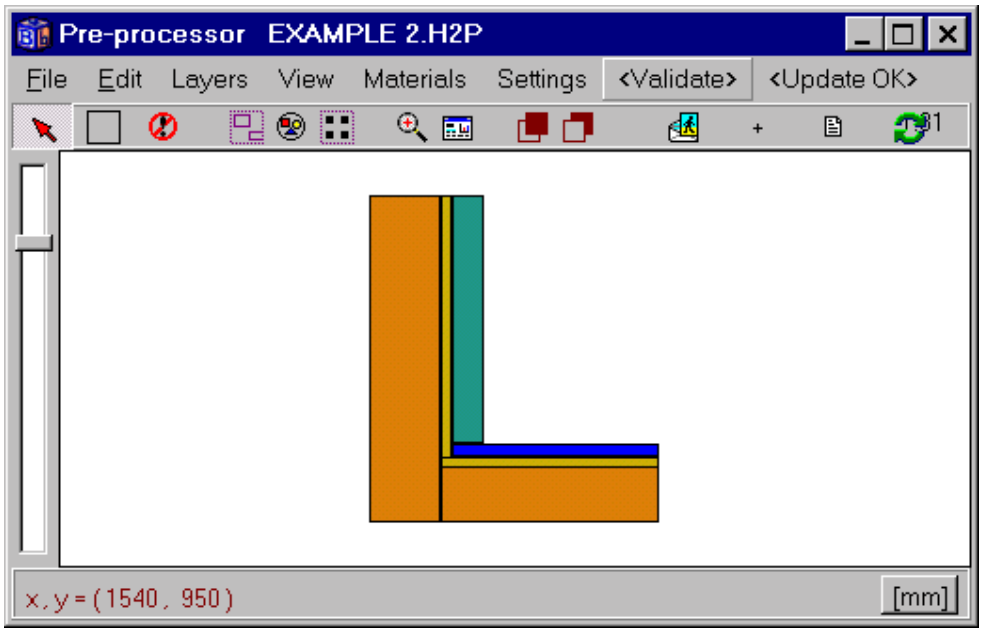


Figure 8.35: Parts of the geometry drawn.

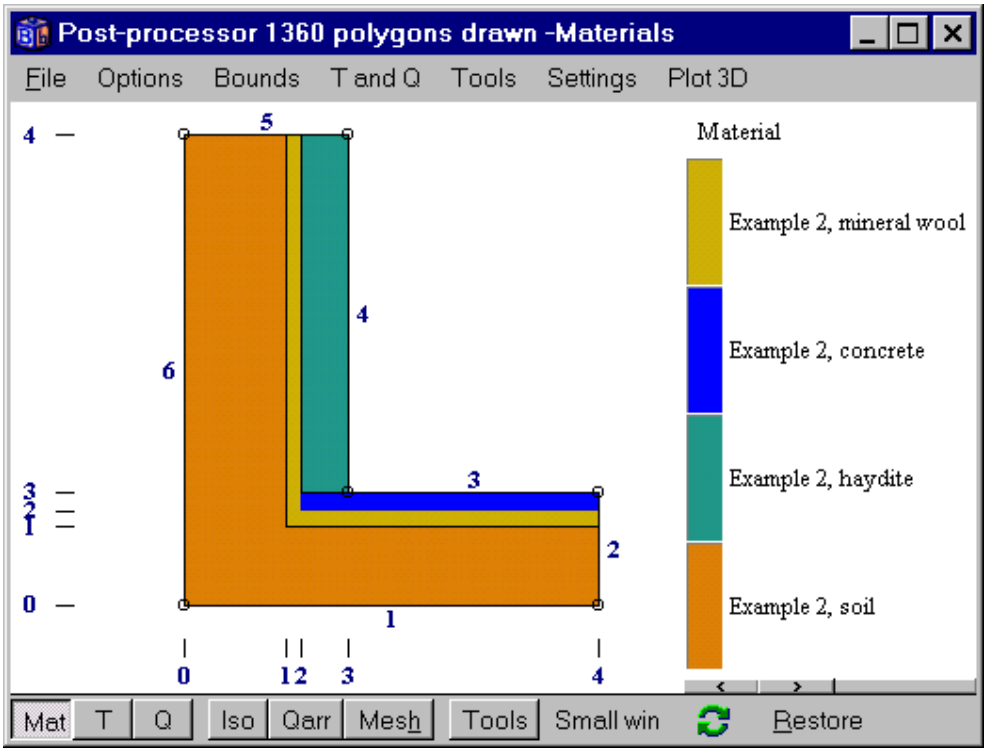


Figure 8.36: Lengths and mesh can be adjusted after the updating.

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

8.3.1 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 8.37 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.

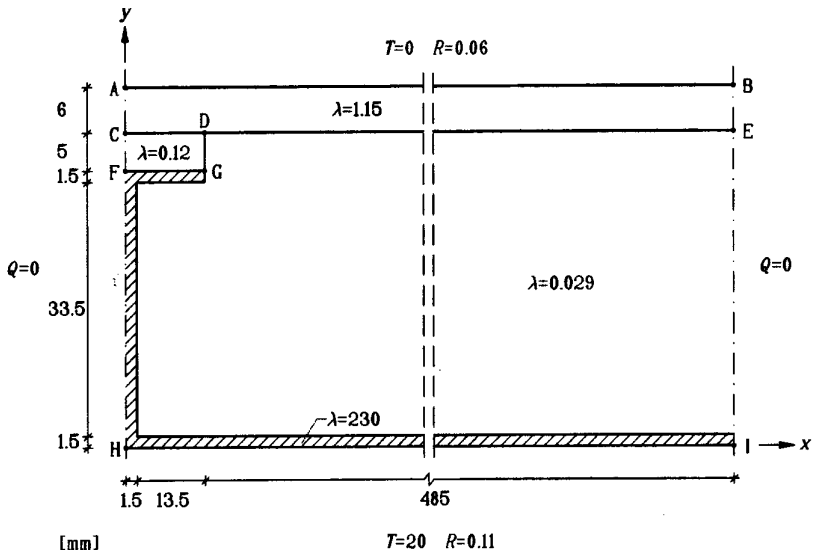


Figure 8.37: Test reference case in the European standards.

8.3.2 Pre-processor input

This example is available as *Example 3.dat*. Figure 8.38 and Figure 8.39 shows the problem as drawn in the pre-processor. The materials are chosen from the list. The smallest length is 1.5 mm, so the scale factor (*item Settings/Scale factor*) may be set to 1:0.5. This will make it possible to draw lengths of 0.5 mm, 1.0 mm, 1.5 mm …etc. Figure 8.40 shows the automatically mesh created with 50*50 cells (*item Settings/Numerical mesh*). Only the left-hand part is shown here. Figure 8.41 shows the boundary conditions (press F6). The heat flow becomes 9.38 for the equidistant mesh, see Figure 8.42.

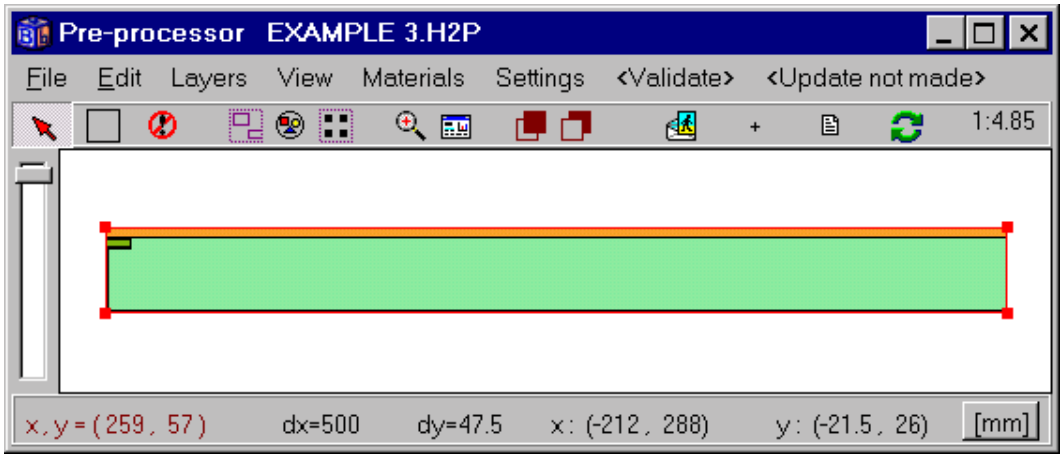


Figure 8.38: Problem as drawn in the pre-processor.

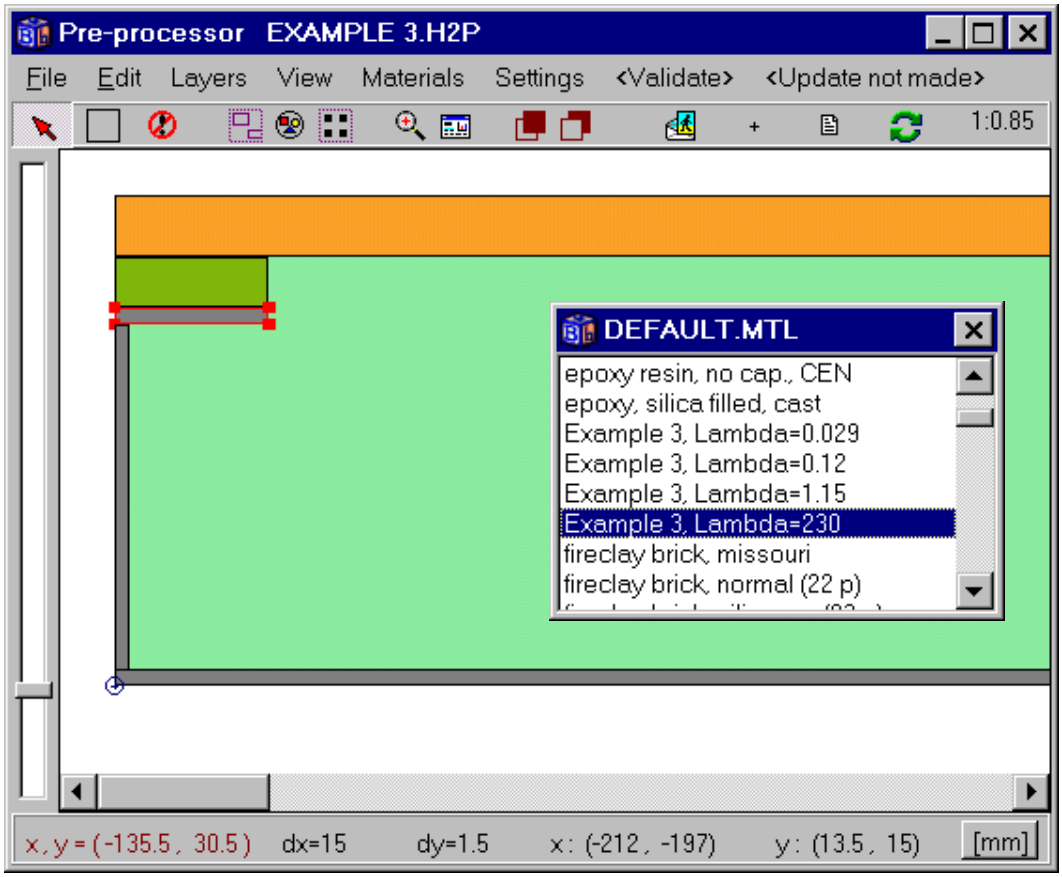


Figure 8.39: Left-hand side enlarged. The material pick list shows the defined materials.

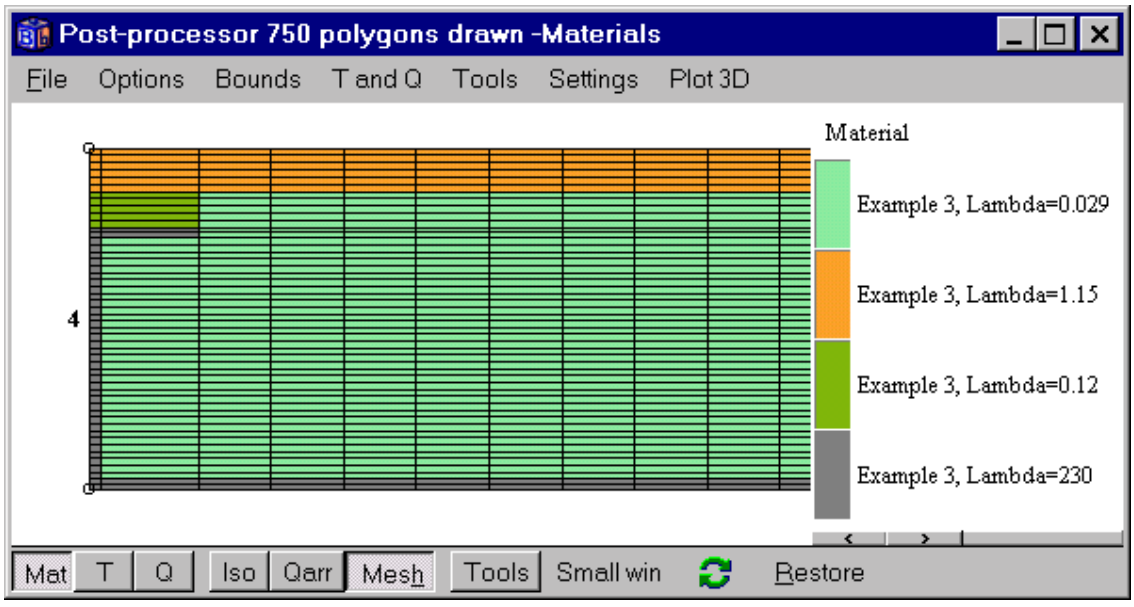


Figure 8.40: The automatically mesh created with 50*50 cells. Only the left-hand part is shown here.

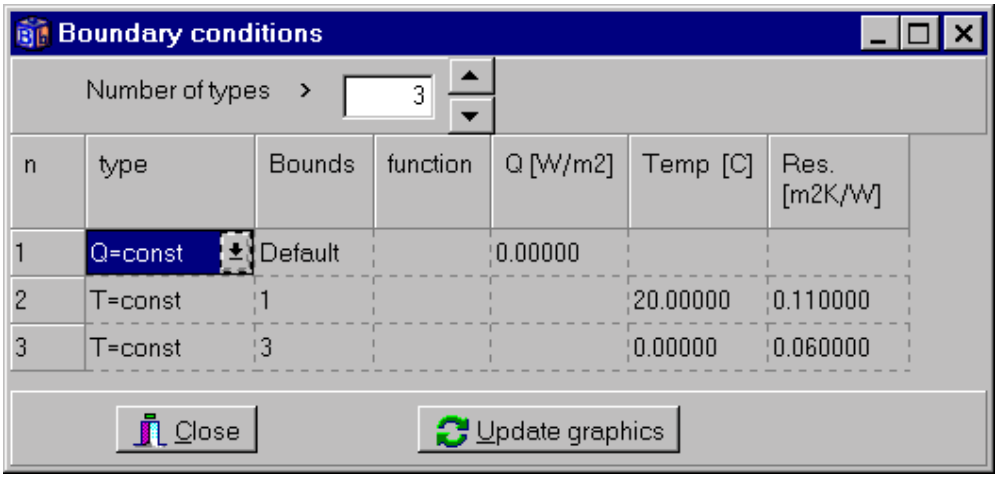


Figure 8.41: Boundary conditions.

The calculation takes a few seconds with the equidistant mesh. The heat flow becomes 9.38 W/m.

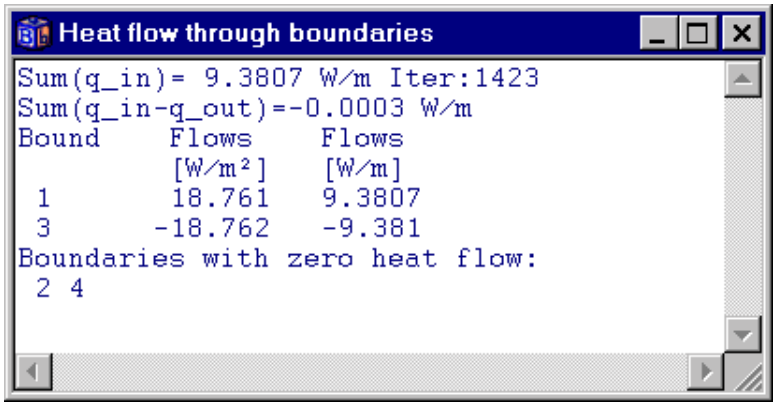


Figure 8.42: The heat flow becomes 9.38 for the equidistant mesh.

In Figure 8.43, the numerical mesh is changed to an expansive mesh in the *x*-direction (menu item *Input/mesh in x-direction*). Here 15 cells are specified between mesh coordinates 1 and 2, and an expansion coefficient of 1.08 between 2 and 3. This will give smaller cells where the temperature gradient is large. The heat flow now becomes 9.48 W/m. Figure 8.45 shows temperature field and isotherms. Figure 8.46 shows heat flows.

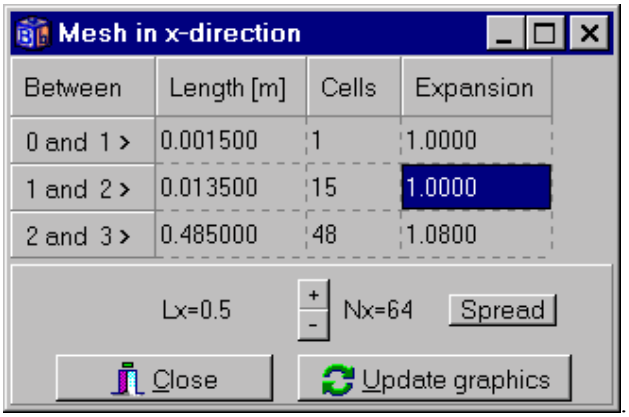


Figure 8.43: The numerical mesh is changed to an expansive mesh in the *x*-direction.

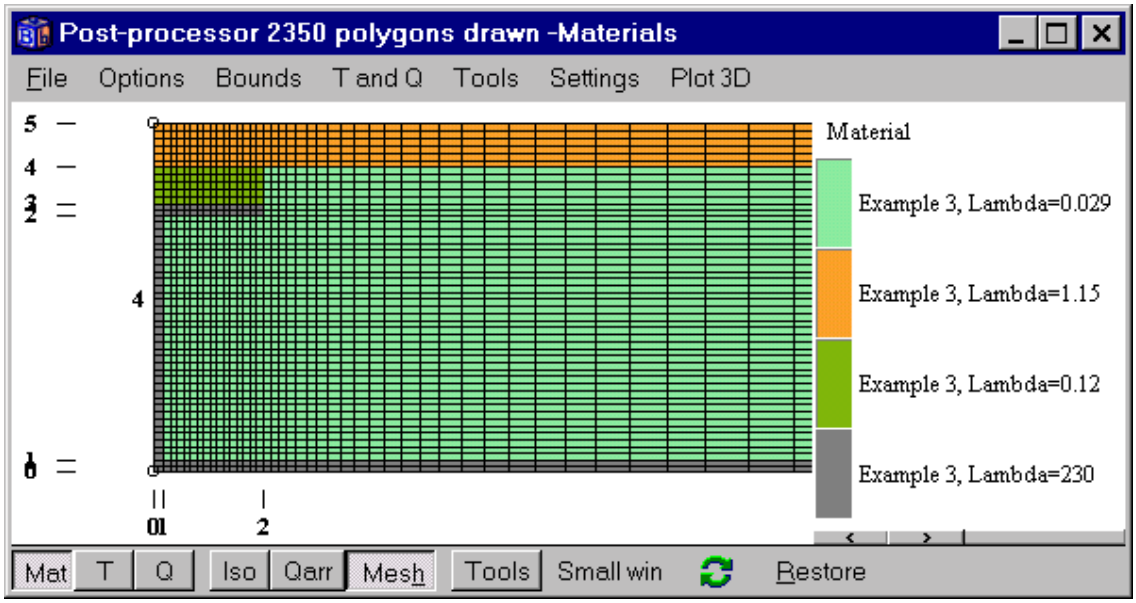


Figure 8.44: An expansive mesh with small cells where the temperature gradient is large. The heat flow now becomes 9.483 W/m.

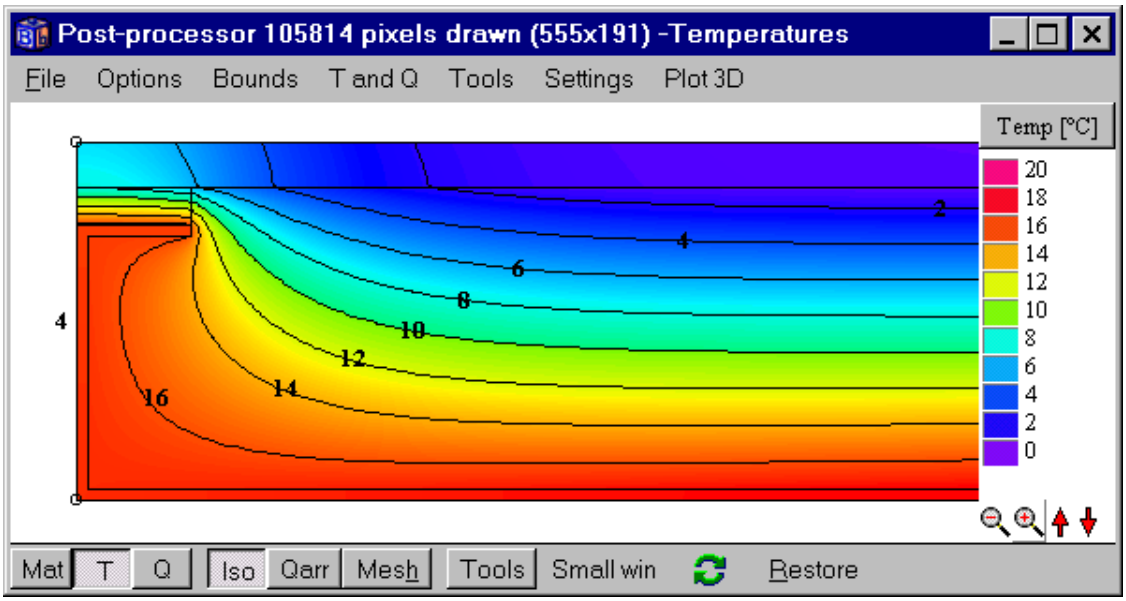


Figure 8.45: Temperature field and isotherms.

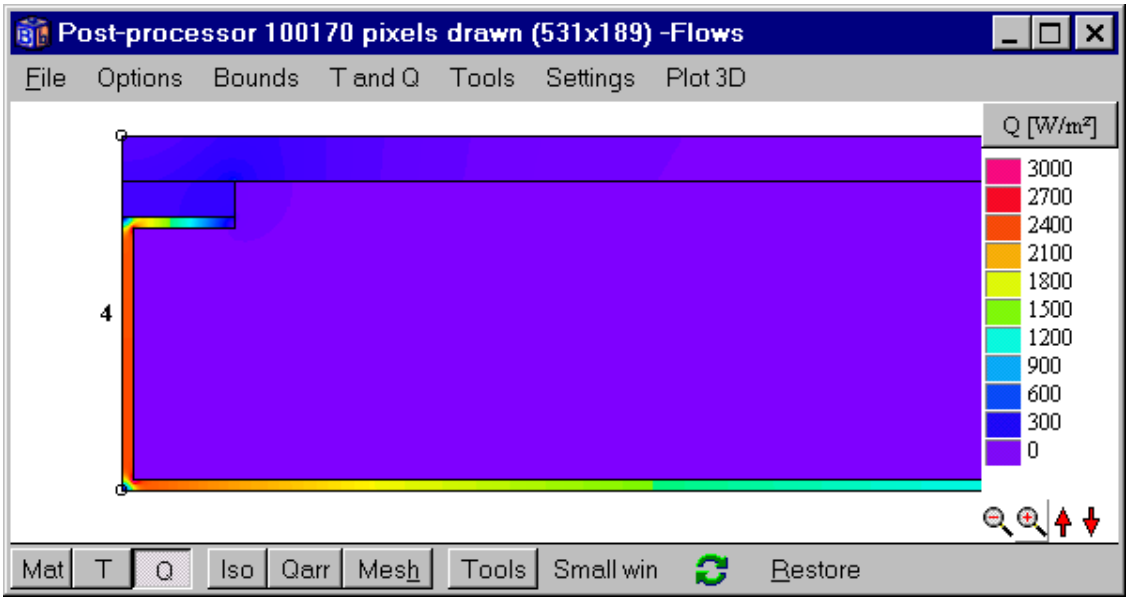


Figure 8.46: Heat flows for the left part.

Using about 60.000 cells in an expansive mesh gives a heat flow of 9.492 W/m. 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 in an expansive mesh suffice to fulfill this requirement.

8.3.3 Alternative input mesh

Figure 8.47 shows an example of another 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. In the *y*-direction, three coordinates have been inserted in the same way (coordinates 2, 5 and 7).

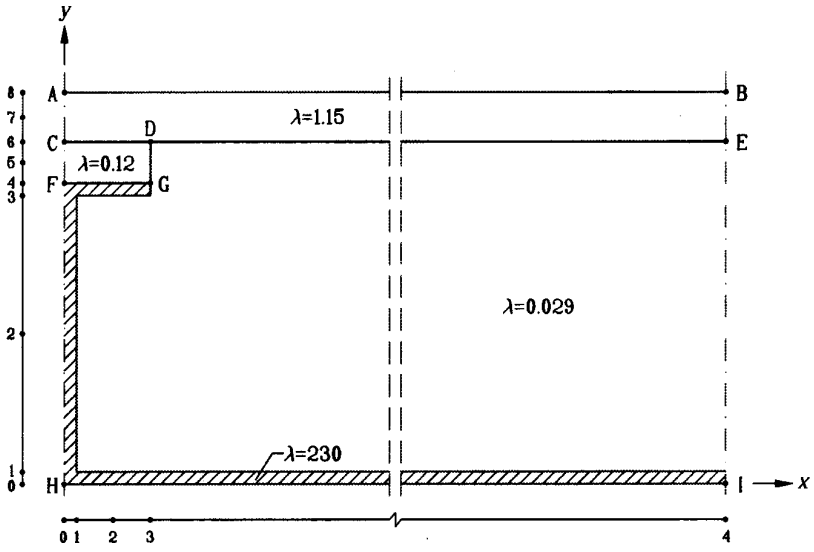


Figure 8.47: An example of example of a suitable input mesh.

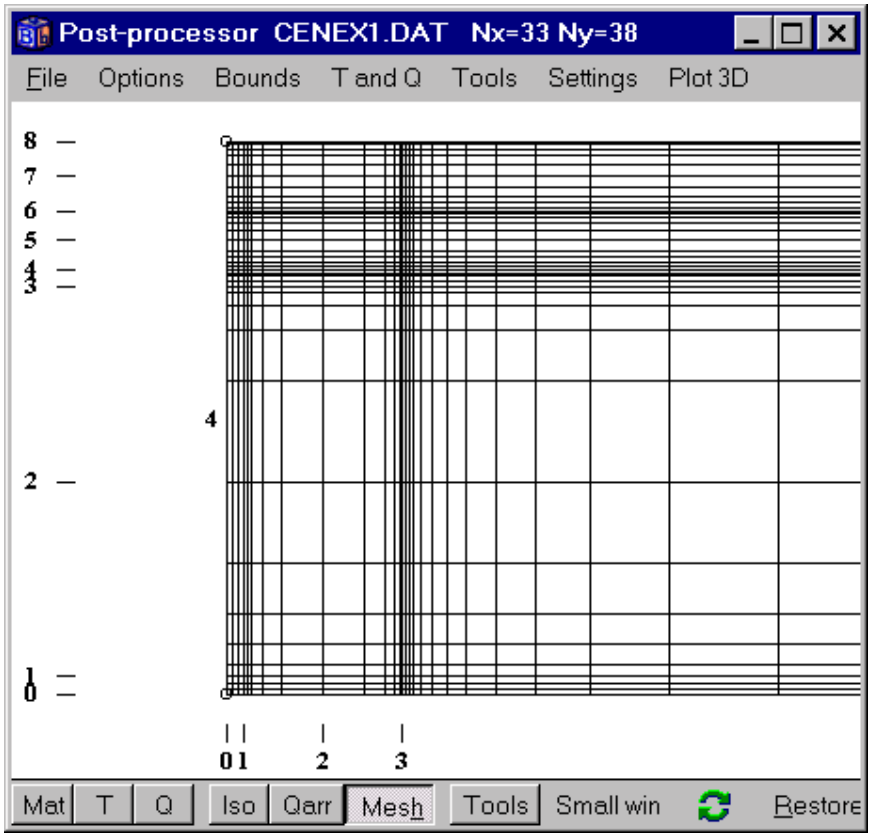


Figure 8.48: Expansive mesh for the left part (total of 1254 cells).

8.3.4 Calculations for different meshes

Table 8:1 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 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, see Figure 8.48. Using ω=1.995 gives a calculation time of 2 seconds. Using ω=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 8:1: Calculated heat flows for different meshes.

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 8:2 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 8.37. The temperature gradients are large at this location. This makes the interpolation difficult. About 4000 cells are needed to meet the CEN requirement. Figure 8.49 shows the isotherms for the left-hand part of the problem.

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

Table 8:2: Calculated surface temperatures (°C).

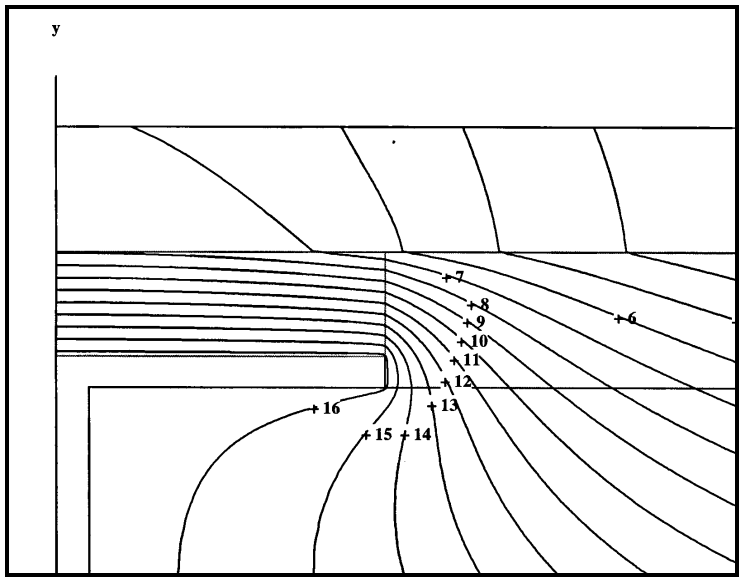


Figure 8.49: Isotherms for the left part of the problem.

8.3.5 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 400 numerical cells suffice to fulfill the requirement for heat flows.

8.4 Example 4 - Roof section with hollow bricks

8.4.1 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 8.50. The thermal conductivities of different areas are shown. The steady-state process in the two-dimensional section is to be simulated.

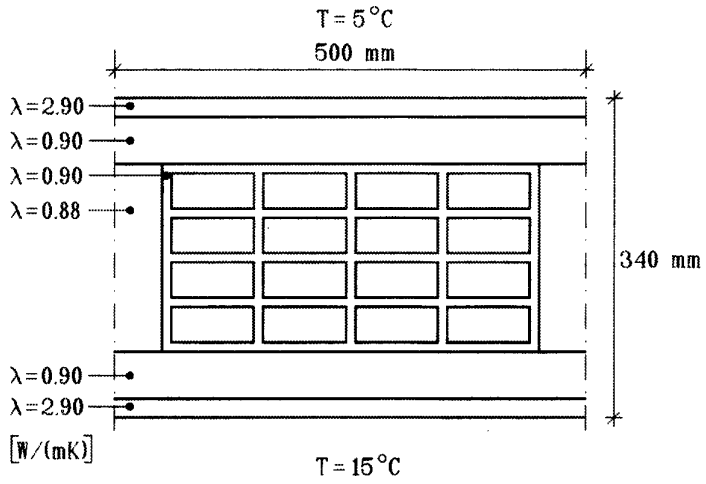


Figure 8.50: Considered section of a brick roof with 16 cavities.

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}, see section 9, is therefore put to 10 °C for all 16 cavities.

8.4.2 Geometry drawn in the pre-processor

The file *Example 4.dat* shows the input for this case, see Figure 8.51. Each modification area has a number with a reference to the type number (in this case type 7), see Figure 8.52. Figure 8.53 shows the input data for the modification type.

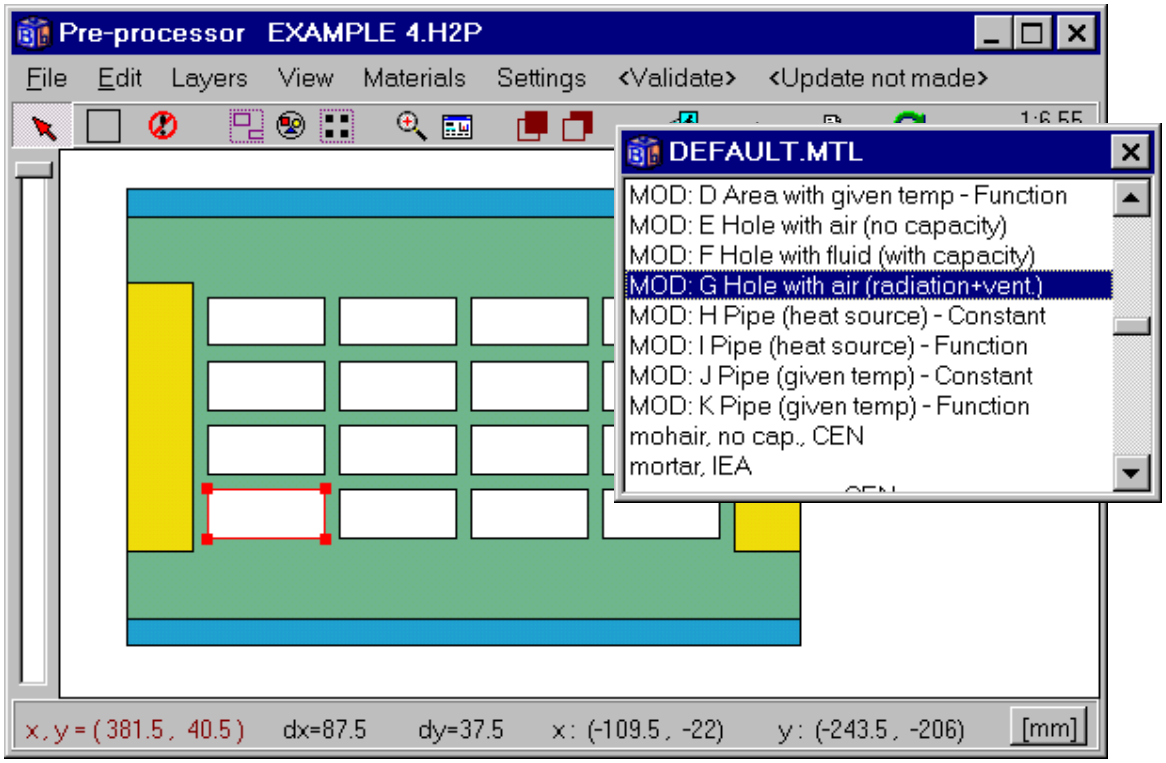


Figure 8.51: Geometry drawn in the pre-processor.

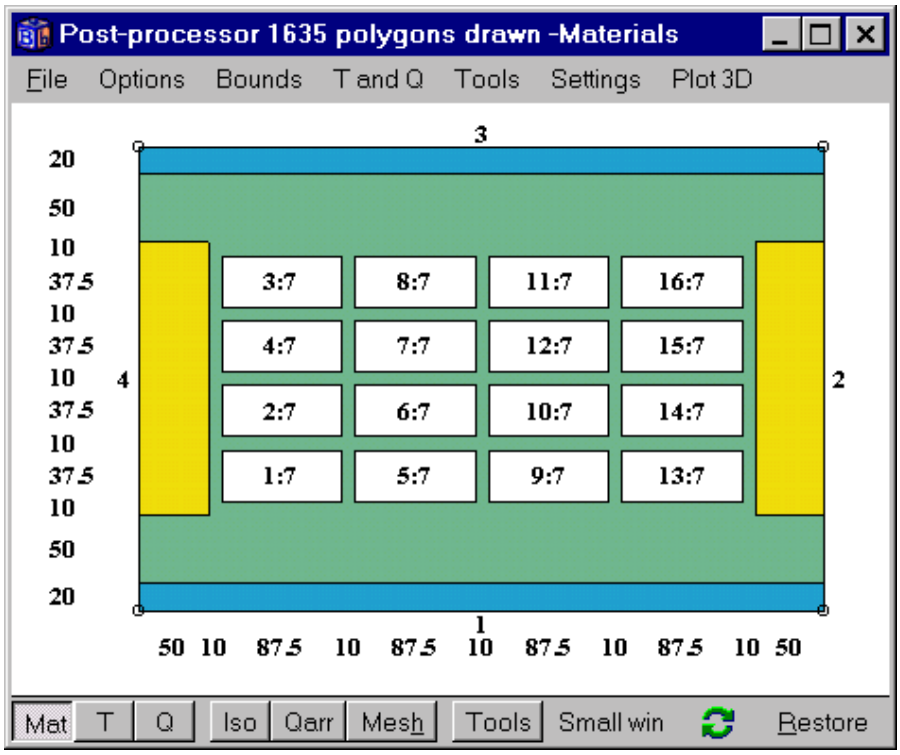


Figure 8.52: Each modification area has a number with a reference to the type number.

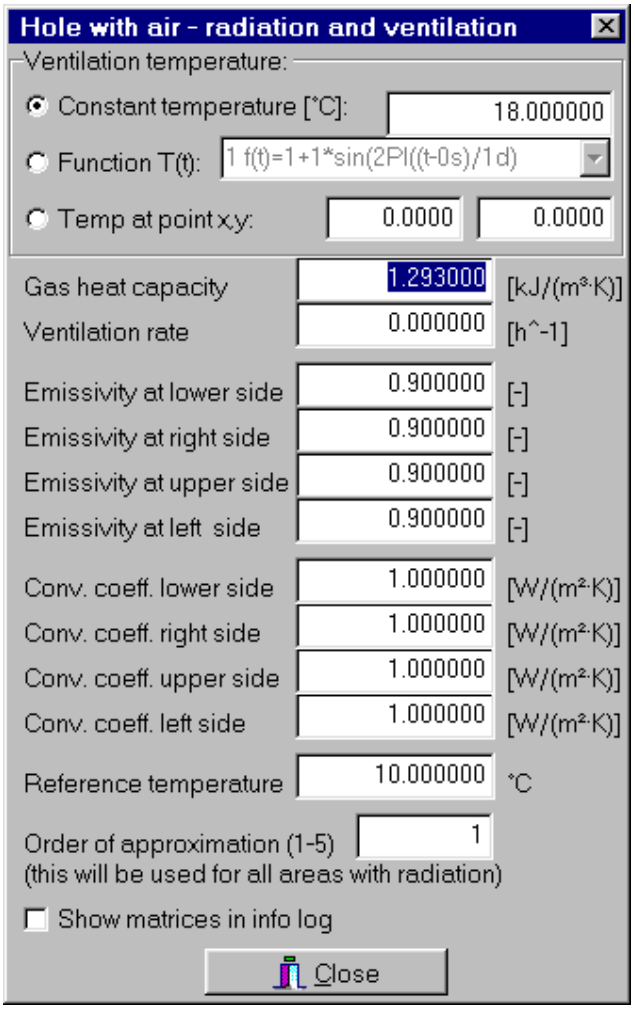


Figure 8.53: Input data for the cavities.

8.4.3 Results for different numerical meshes

The heat flow becomes 5.936 W/m with 1635 numerical cells placed in an equidistant mesh.

The process has been simulated for other numerical meshes to assess the numerical error. Figure 8.54 shows one of the finer meshes. The number of computational cells is here 3860. Each cavity has N_R=40 surface elements.

Table 8:3 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 9.4.1. 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.

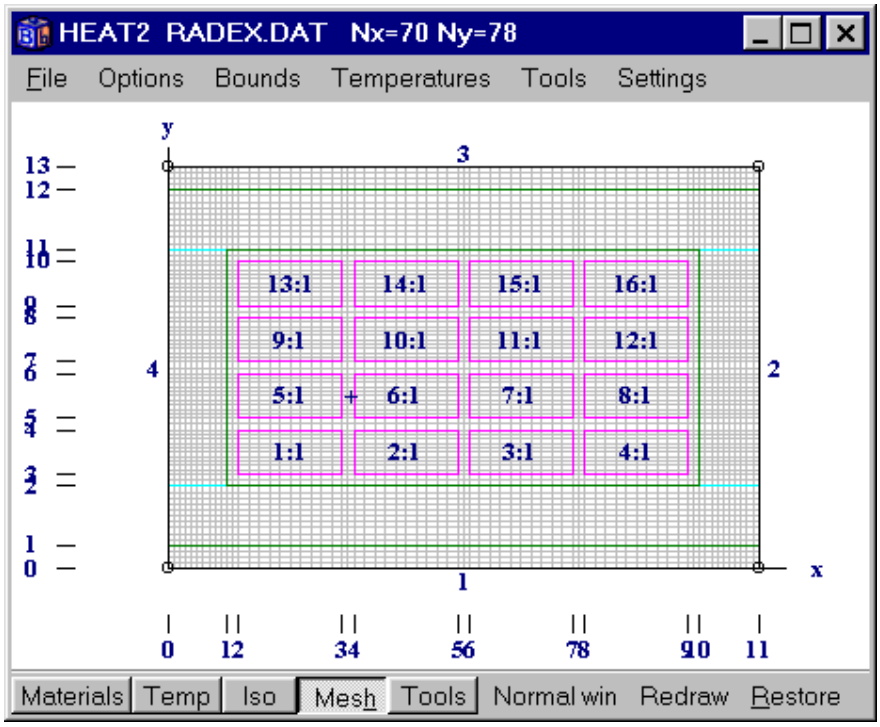


Figure 8.54: The numerical mesh with 3860 computational cells and 16 cavities with radiation (N_R=40).

Figure 8.55 shows the solve window (left) and boundary heat flows (right). Figure 8.56 shows the calculated isotherms. Figure 8.57 - Figure 8.59 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 8:3: Heat flow, error, iterations and CPU-time for four different meshes.

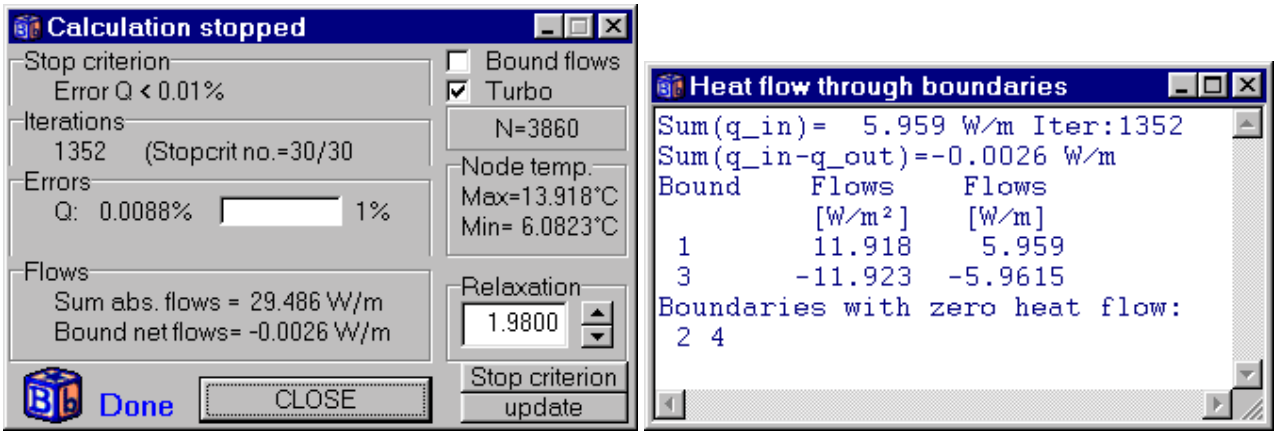


Figure 8.55: The solve window (left) and boundary heat flows (right).

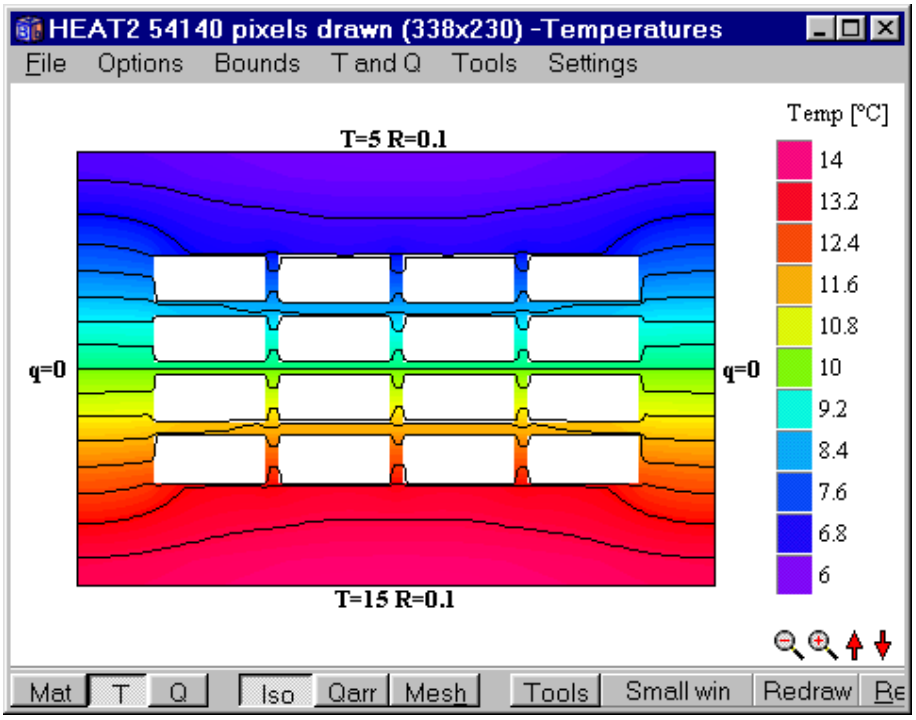


Figure 8.56: Calculated isotherms shown with the interval 0.5 °C.

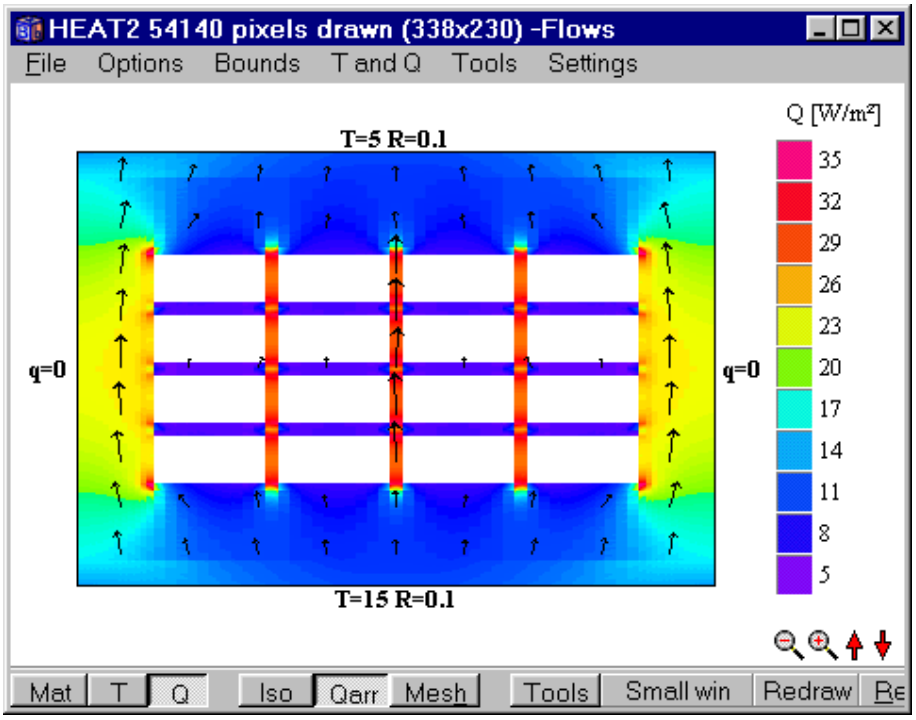


Figure 8.57: Heat flows.

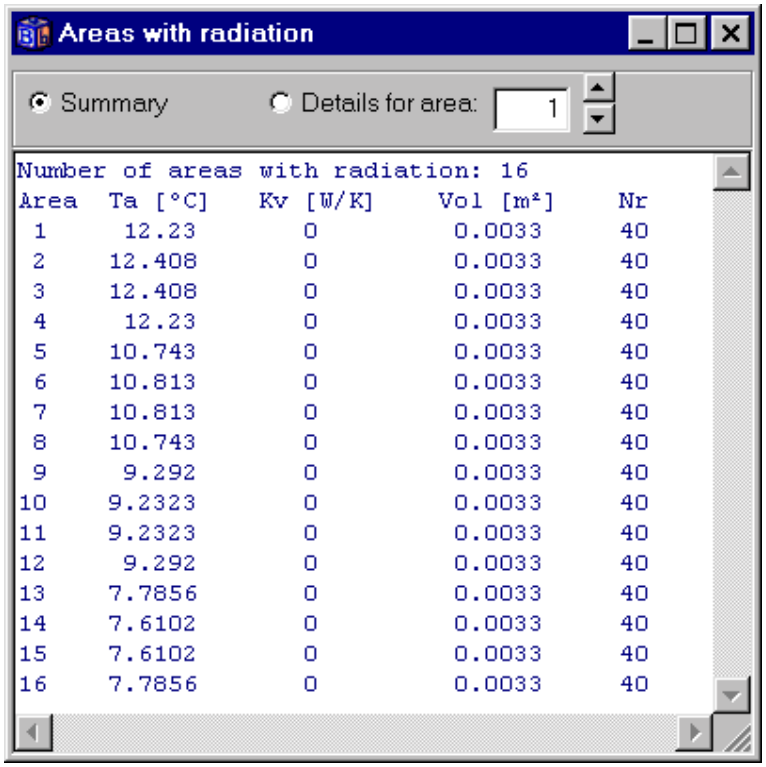


Figure 8.58: Air temperatures in each of the 16 cavities.

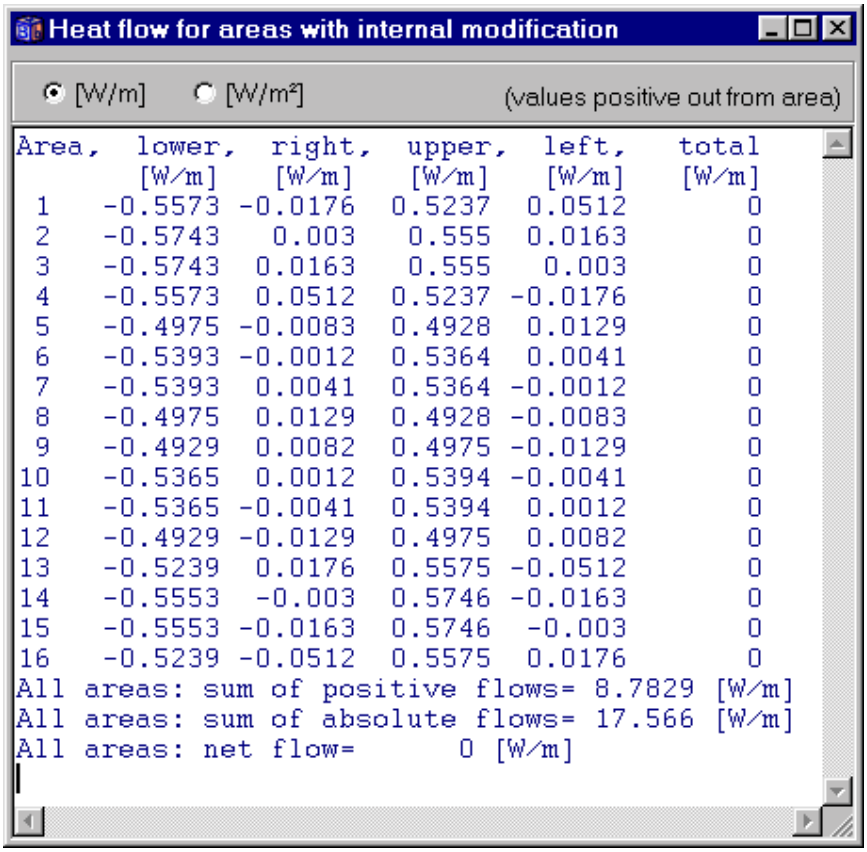


Figure 8.59: Heat flows through the sides of the 16 cavities.

8.4.4 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 8:4 shows the calculated heat flow for different iteration steps for the mesh with 3860 computational cells and N_R=40.

	T_{so} (°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 8:4: Calculated heat flows (W/m) for the mesh with 3860 computational cells and N_R=40.

Table 8:4 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).

8.4.5 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.

9. Radiation and ventilation in cavities

9.1 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).

9.2 Input data

Figure 9.1 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, (J/(m³·K)), which is the density times the specific heat capacity ($C=\rho \cdot c_p$).

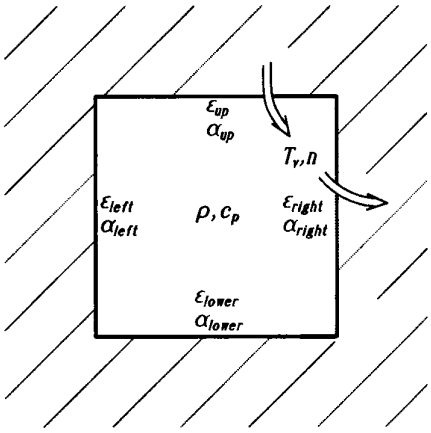


Figure 9.1: Input parameters for radiation inside a cavity.

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 9.2 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 9.3 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.

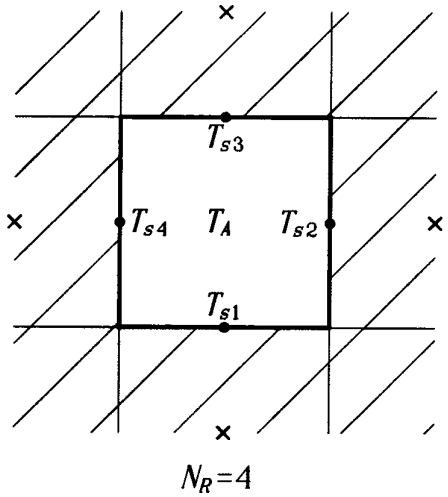


Figure 9.2: Four radiation surface elements in a cavity.

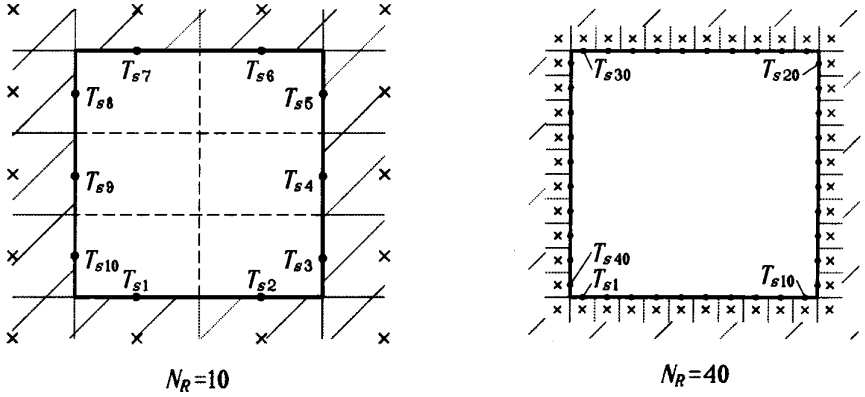


Figure 9.3: The number of radiation elements N_R is defined by the numerical mesh.

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 5.8.2 for more info.

9.3 Output data

The following results are shown for each cavity: calculated surface temperatures T_{sj} for j=1..N_R, see Figure 9.4; temperatures T_{bj} in the cells closest to the cavity, the conductances K_{bj} between the center of the cell and the surface of the enclosing cells, the heat flows q_{bj} (W/m), and the gas temperature T_a. The volume of the cavity per meter (m³/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.

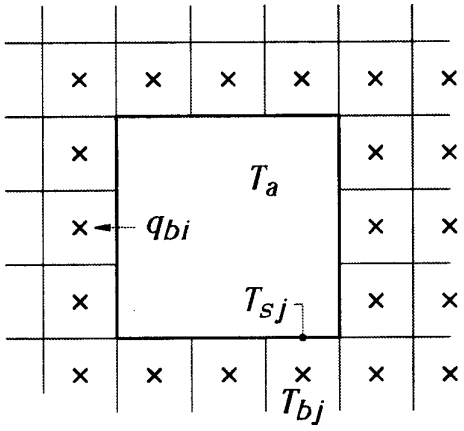


Figure 9.4: Calculated results for each cavity with radiation.

9.4 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):

$$\frac{\varepsilon_i A_1}{1-\varepsilon_i} \cdot \left(\sigma \tilde{T}_{si}^4 - J_i\right) = \sum_{j=1}^{N} A_i F_{ij} (J_i - J_j) \qquad i = 1,\ldots,N_R \qquad (5.1)$$

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 ~ denotes temperature in Kelvin. The view factors are denoted by F_{ij}.

9.4.1 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_{si} - T_{sj}) 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.

10. Analysis of window frames

10.1 Introduction

Routines for simplified window frame analysis according to the proposed European Standard prEN ISO 10077-2:2.2000 (**Thermal performance of windows, doors and shutters – Calculation of thermal transmittance – Part 2: Numerical method for frames)** have been implemented in HEAT2. Note that this standard is preliminary. Please check for update information for HEAT2 in this aspect.

The particular problem of heat transmittance through the window frame concerns how to treat the cavities in the frame. The heat flow rate in each cavity is calculated using an equivalent thermal conductivity λ_{eq} according to the standards. This equivalent thermal conductivity includes the heat flux by conduction, by convection and by radiation and depends on the geometric and material characteristics of the cavity as well as the temperature distribution inside the cavity.

The maximum number of frame cavities is 250.

10.2 Frame cavity calculation options

The equivalent thermal conductivity in rectangular and non-rectangular cavities depends on a given min and max temperature. There are four options in HEAT2 what temperatures to use, see Figure 10.1. The CEN/ISO proposal is to use the actual iteratively calculated minimum and maximum temperatures in each cavity (option 4). The equivalent thermal conductivities are updated iteratively at arbitrarily chosen iteration interval steps (the default value is 100). Alternatively, one can use the fix boundary air temperatures as min and max temperatures for all cavities (option 1), or to set T_{min}=5 and T_{min}=15 ºC (option 2). HEAT2 gives another option to use arbitrarily chosen temperatures (option 3).

Note that all frame calculations can be done for steady-state or transient analysis. If detailed transient analysis is required, a more frequent update of the equivalent thermal conductivities could be made. Ideally, an update should be made every iteration (the step is 1) but this would increase calculation time since new conductances are calculated for each update. It might be a good idea to see the effect of different update steps. However, in most cases it should be enough to keep the update interval at 100.

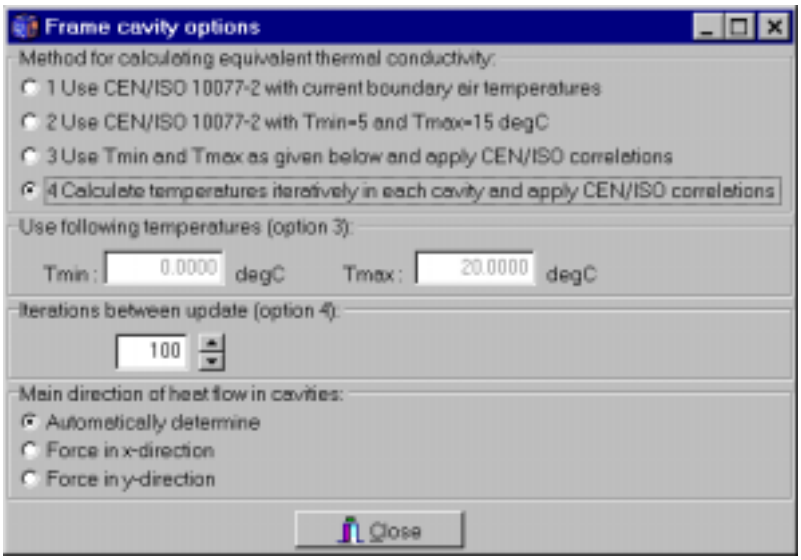


Figure 10.1: Frame cavity options.

10.3 Defining frame cavities

HEAT2 identifies the internal cavities and calculates the equivalent thermal conductivities according to the standards. Frame cavities are specified using solid material boxes with the reserved name “Frame cavity”, see Figure 10.2. If two or more boxes overlap they will be recognized as one cavity. The total number of found cavities is shown at the bottom line in the HEAT2 main window.

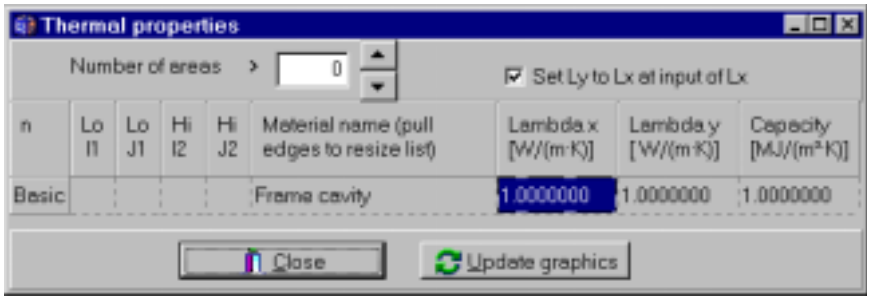


Figure 10.2: Cavities are given as solid materials with the name “Frame cavity” (the string may be written directly or taken from the material pick list.

Frame cavities may also be entered in the pre-processor by using the material “Frame cavity” in the material pick list. An empty space surrounded by material will also be recognized as a frame cavity. Figure 10.3 and Figure 10.4 show a schematic example how input is made.

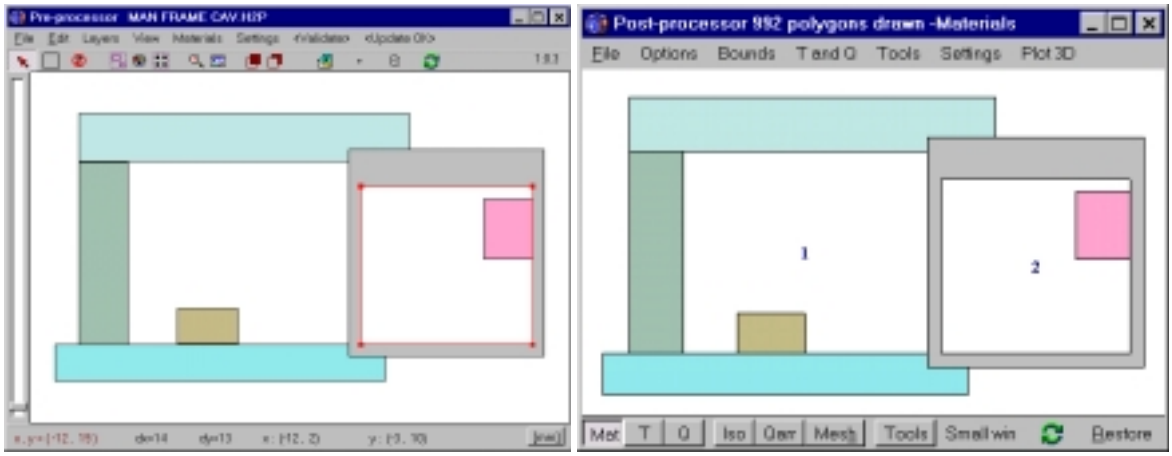


Figure 10.3: Geometry with two non-rectangular air cavities drawn in the pre-processor (left). The two identified cavities are shown in the post-processor (right). The first cavity (1) is never explicitly drawn but is recognized as a cavity since it is surrounded by material. The second cavity (2) is drawn using the material “Frame cavity” from the pick list.

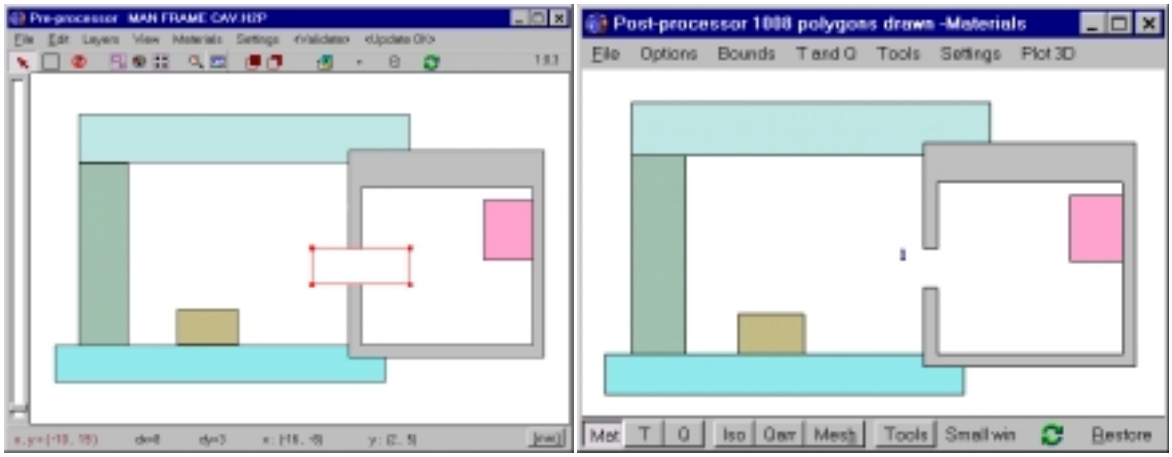


Figure 10.4: The two cavities are here connected resulting in one defined cavity.

10.4 Example 1 - A PVC shutter profile with five cavities

Figure 10.5 shows a PVC shutter profile with five cavities. This is a test case from the proposed standards. The input is made by drawing eight "Frame cavity" boxes on the PVC rectangle, see input file **Example Frame.dat**. Alternatively, the geometry could be given using only PVC boxes and the enclosed space would be detected as frame cavities. Figure 10.6 shows calculated heat flows for the PVC shutter profile with the five cavities. Note that the heat flow is not shown graphically in the cavities. This can however be shown by enabling menu item **T and Q/Show T and Q in cavities**.

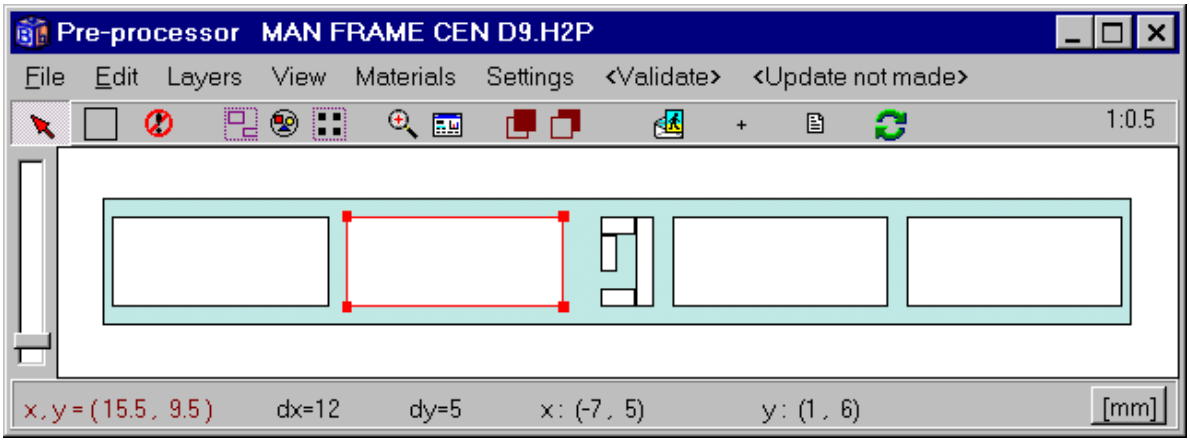


Figure 10.5: A PVC shutter profile with five cavities. The cavities are given as eight boxes using "Frame cavity" in the material list.

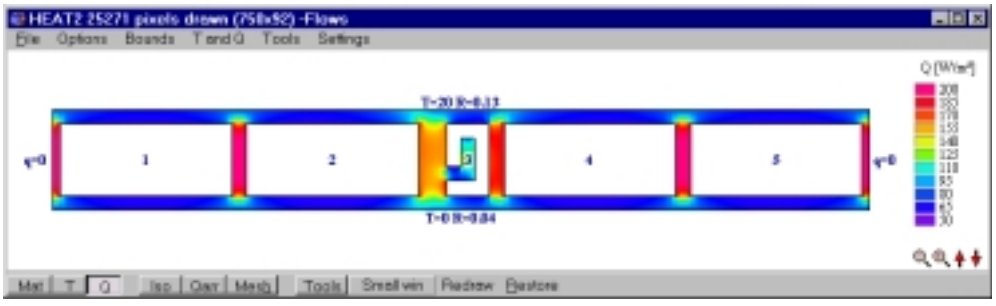


Figure 10.6: Calculated heat flows for the PVC shutter profile with five cavities. The calculation with about 12000 numerical cells takes a few seconds on a Pentium II.

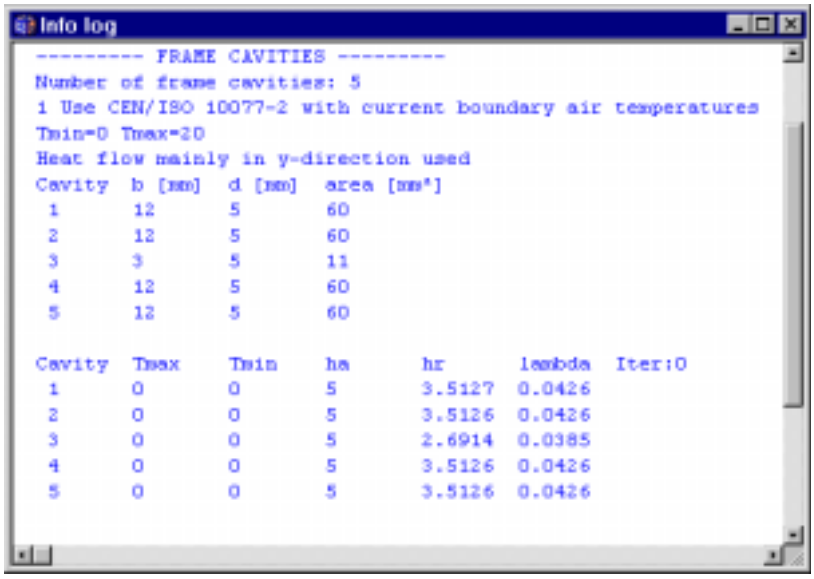


Figure 10.7: The info log shows data for frame cavities, such as equivalent thermal conductivities.

Figure 10.7 shows the info log when option 1 is used, see option list in Figure 10.1, and Figure 10.8 shows the info log when option 4 is used.

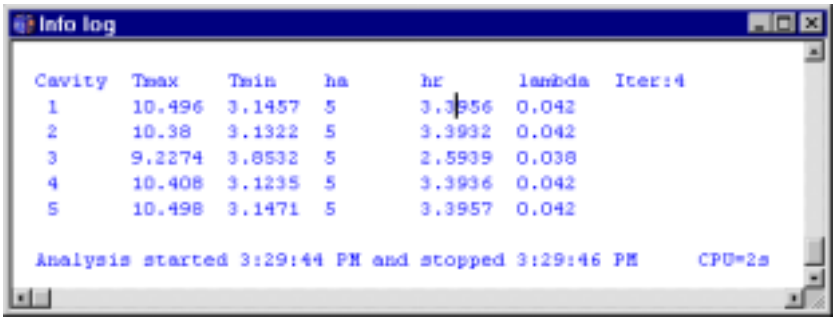


Figure 10.8: The info log shows here iteratively calculated temperatures when option 4 is used. The text indicates that four updates have been made for calculating the equivalent thermal conductivities. Compare equivalent thermal conductivities with those in the above figure.

Table 10:1 shows calculated heat flows through the boundaries for option 1 and 4 with different numerical meshes and relaxation factors ω. The CPU-time for a Pentium II 400 MHz is shown in the last column. An error less than 0.01% is used for the heat flows as a stop criterion, see Eq. 5.1.

Table 10:1: Heat flow through frame and benchmarks using a Pentium II 400 MHz.

Option	Numerical cells	*q*, (W/m)	HEAT2 4.0 CPU-time
1	11799	4.148	2 sec.
1	62500	4.149	195 sec. (ω=1.95)
			36 sec. (optimized ω=1.99)
4	11799	4.134	2 sec.
4	62500	4.134	203 sec. (ω=1.95)
			37 sec. (optimized ω=1.99)

10.5 Example 2 – a window frame

The following example shows how to input and model a window frame. It is a reference case in the proposed standards. See input file **Frame_Example.dat**.

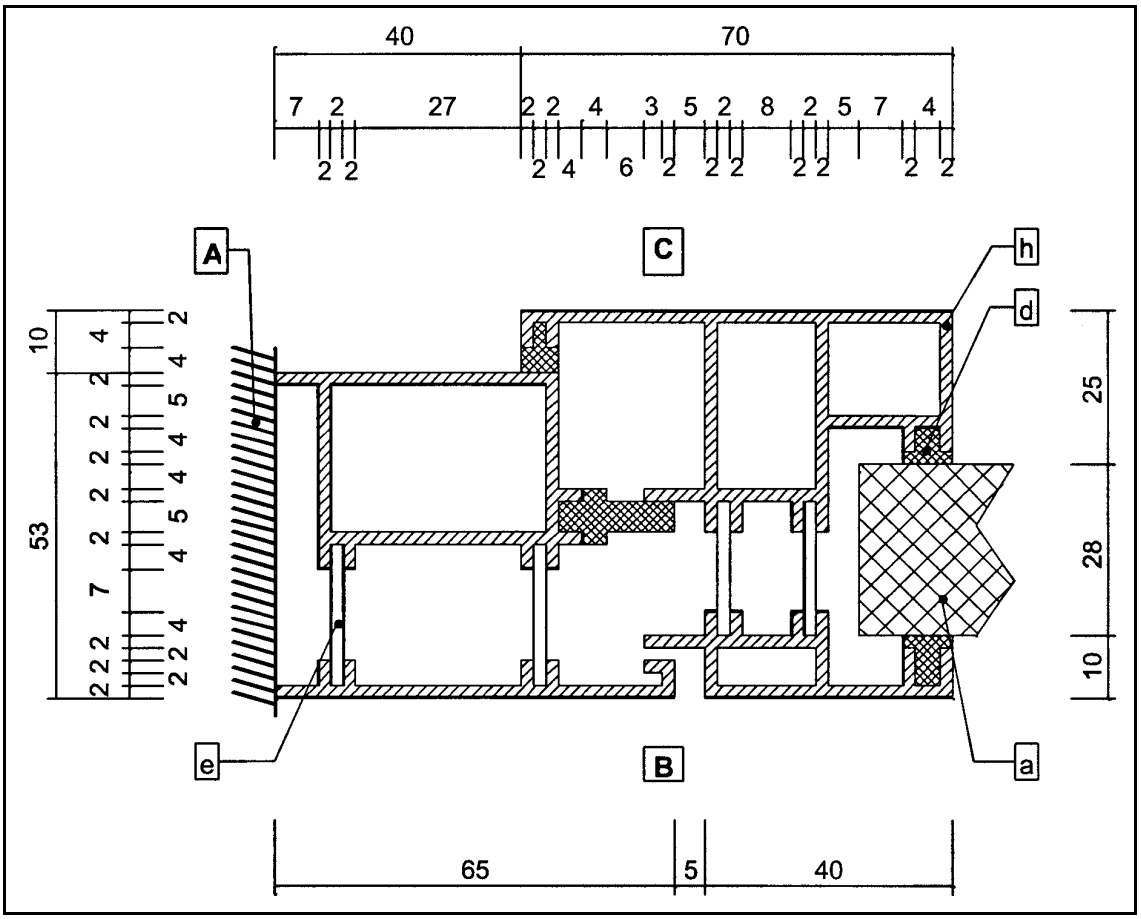


Figure 10.9: The window frame.

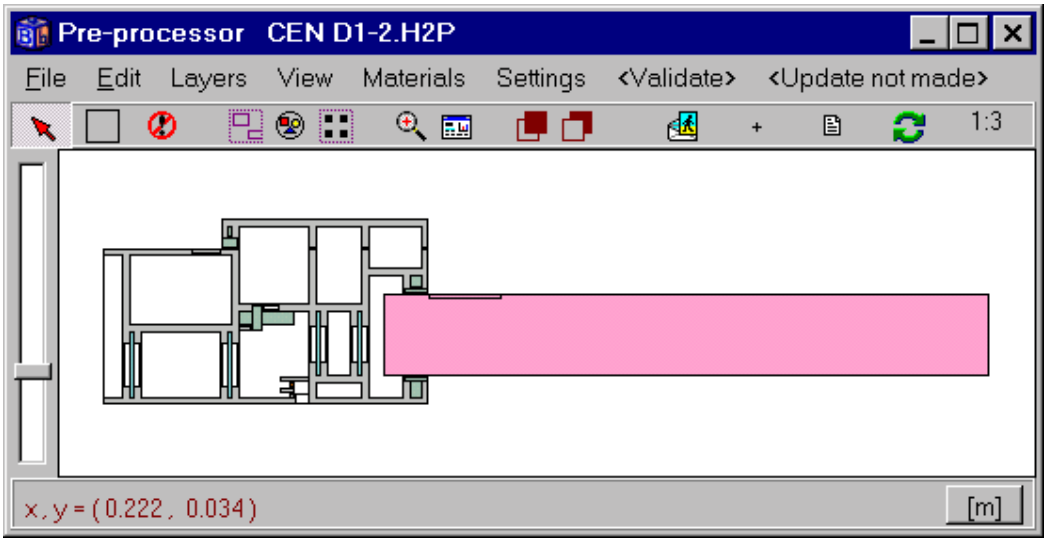


Figure 10.10: Window frame with insulation board.

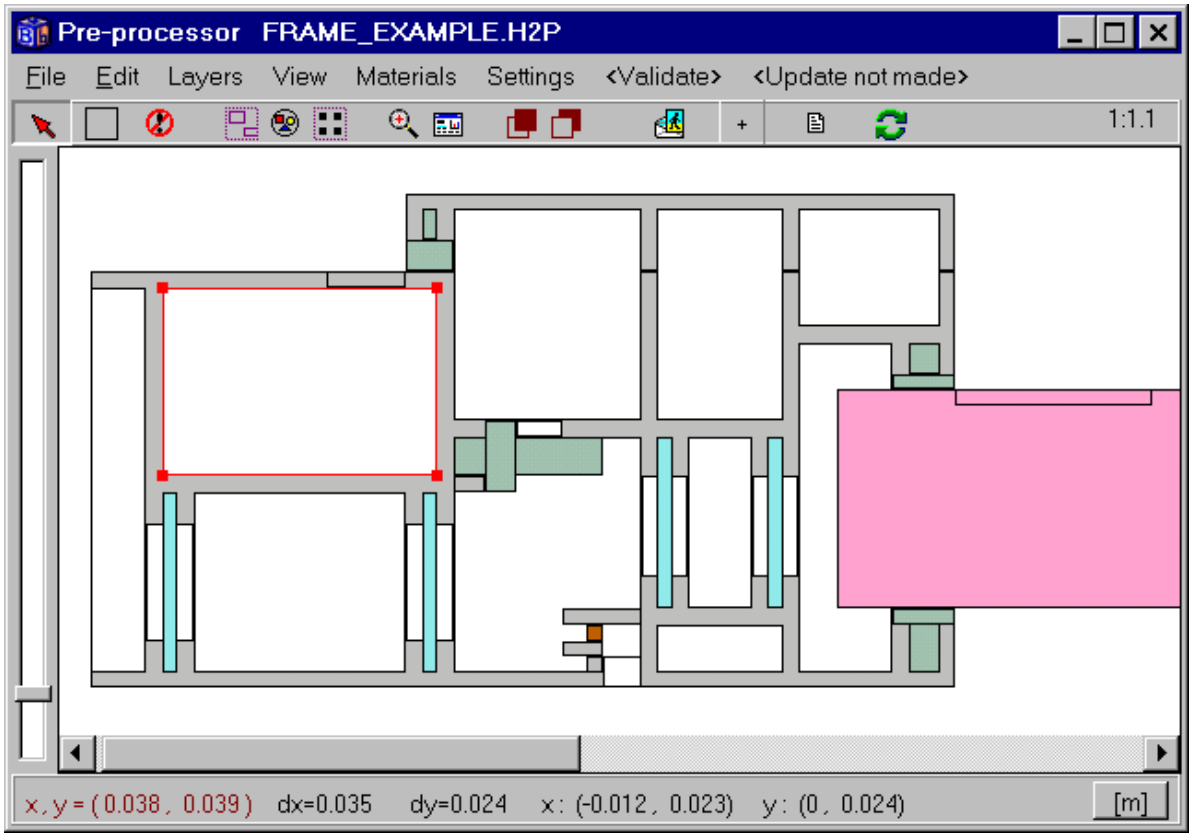


Figure 10.11: The window frame enlarged. Note that it is sufficient to draw one of the four blue bars and then copy this to the three other.

In this example the frame cavities are explicitly drawn using "Frame cavity" in the material pick list. Consider Figure 10.11 and Figure 10.12. Cavity number 2 is drawn here overlapping another material box (two rectangles drawn in total), but it would also have been possible to just draw the surrounding material using four bars (four rectangles drawn in total).

The length of the insulation board is set according to the CEN standard.

Boundary conditions are given in HEAT2 (T=20 and R=0.13 on upper segments and T=0 and R=0.04 on the lower ones, adiabatic condition on the vertical sides).

The standard proposes a "reduced radiation/convection" boundary condition of 0.20 m²·K/W at the internal sides in edges or junctions between two surfaces, in this case the boundaries 44-52 and 74-77. These boundary segments are not generated by default. This is done in the following way. Use short boundary segments (menu item *Use long boundary segments* is unchecked in the pre-processor *Settings* menu). Draw an overlapping rectangle of the same material, see Figure 10.11. It would also have been possible to insert some mesh points, adjusting the lengths in the new mesh segments, and defining four resistance lines of 0.07 m²·K/W that is added to the boundary condition surface resistance of 0.13 m²·K/W.

Figure 10.13 shows the boundary conditions.

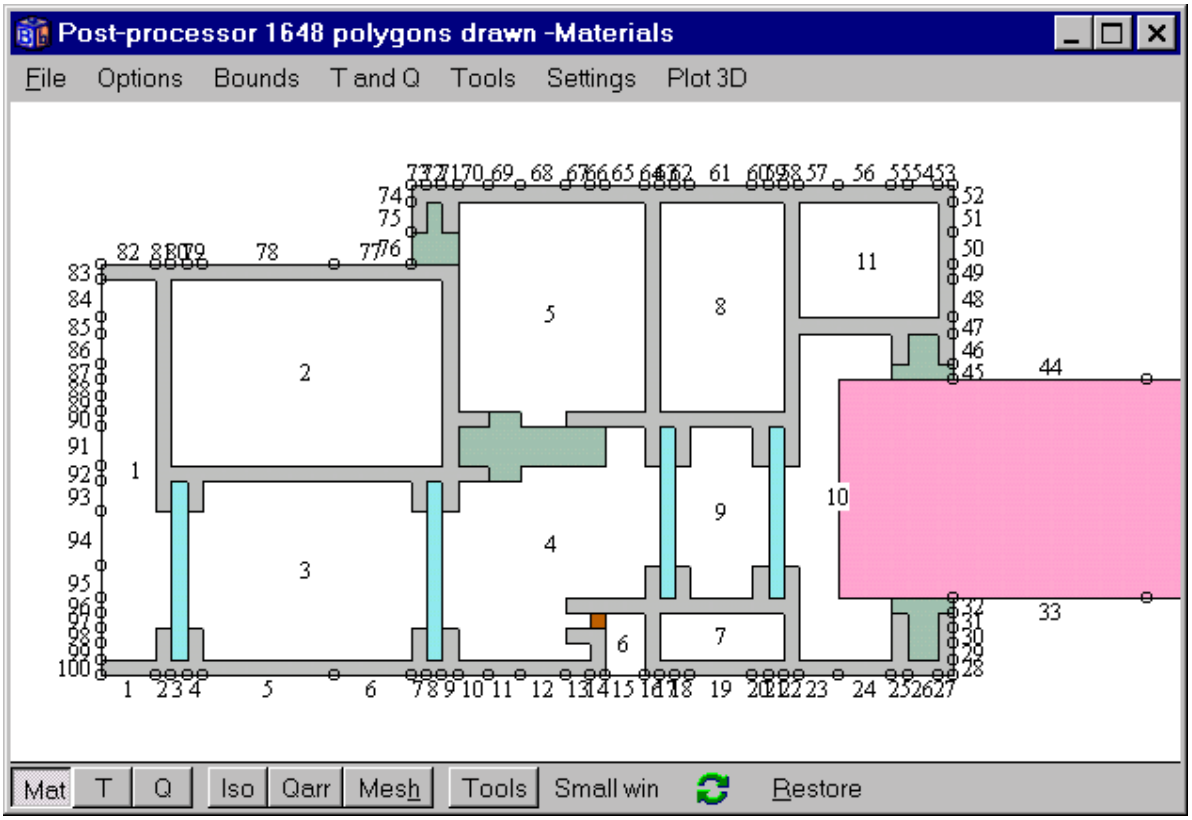


Figure 10.12: Generated boundary segment numbers (1..100), and identified cavities (1..10).

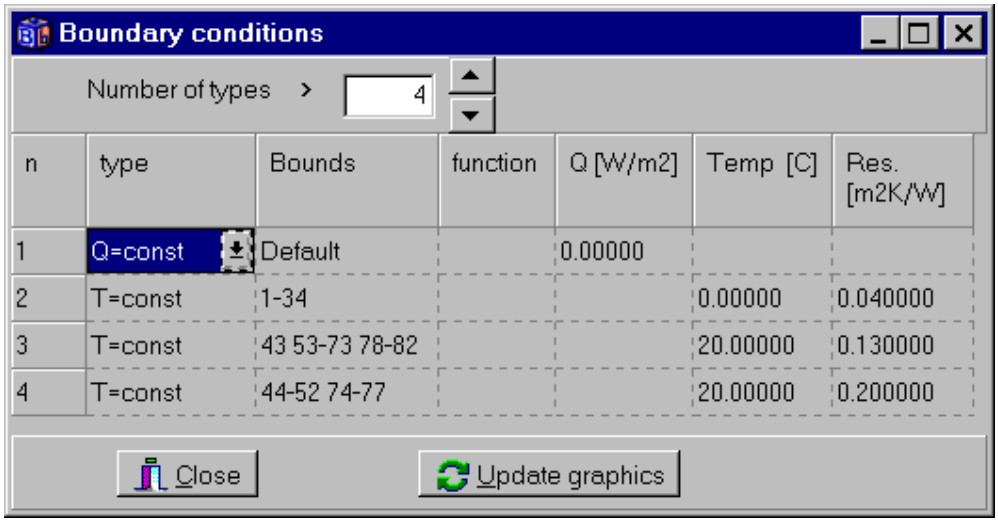


Figure 10.13: Boundary conditions.

When the calculation is started, HEAT2 will ask whether or not to use *twice* the equivalent thermal conductivity for cavities at external boundaries with a prescribed temperature, such as in the case of cavity 6. See the standard for special rules on these so-called ventilated cavities. Cavity 1 will automatically be treated as a normal cavity since the boundary condition is adiabatic.

The info log, see Figure 10.14 and Figure 10.15, shows the frame cavity option, what temperatures are used, the width, height, and area of each cavity, the calculated max and min temperatures and the radiative (hr) and convective (h_a) heat transfer coefficients and the equivalent thermal conductivity for each cavity.

Figure 10.16 shows temperatures and calculated equivalent thermal conductivities when option 4 is used. Figure 10.17 and Figure 10.18 show heat flow intensities (W/m²).

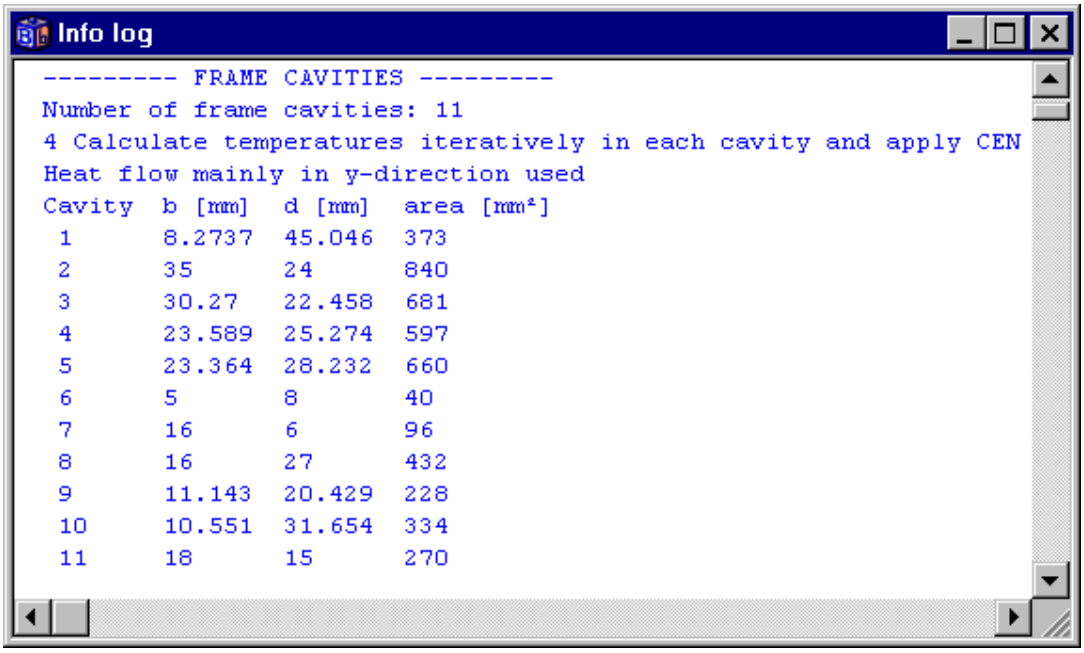


Figure 10.14: The info log shows data for frame cavities, such as equivalent thermal conductivities. Option 1 is used here. In this case, twice the thermal conductivity is used for cavity 6 according to the standards.

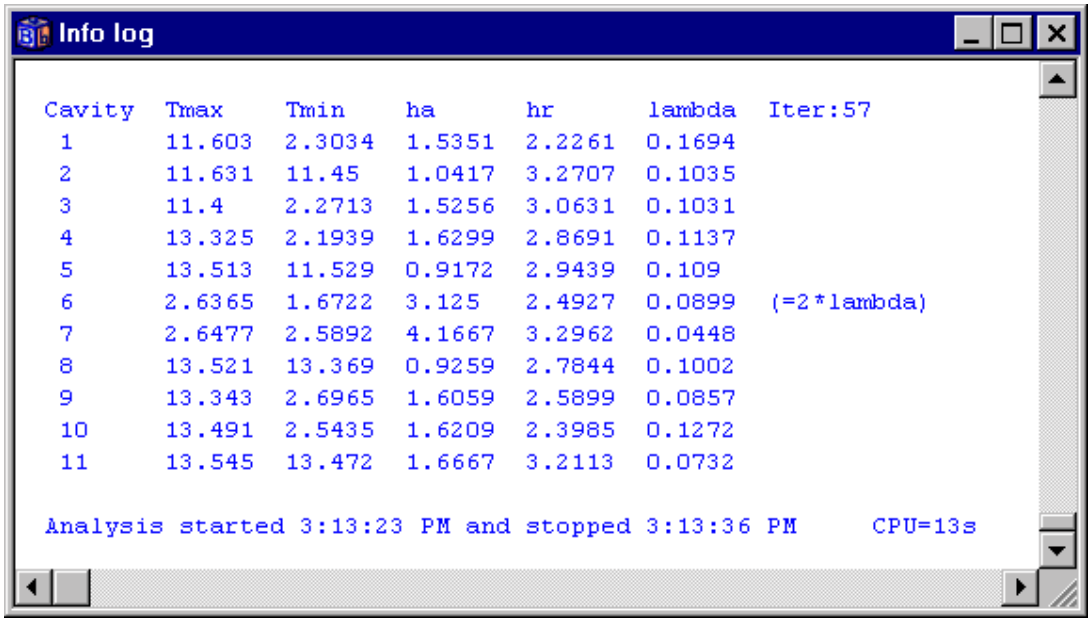


Figure 10.15: The info log shows here iteratively calculated temperatures when option 4 is used. The text indicates that 57 updates have been made.

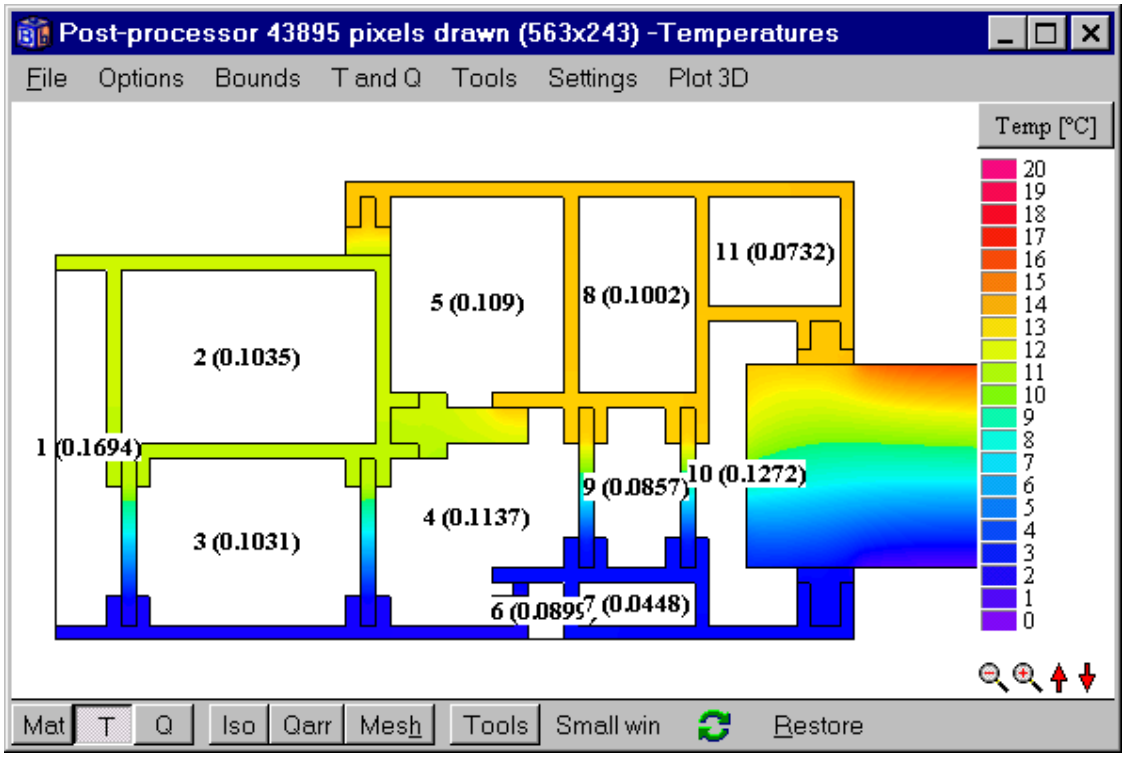


Figure 10.16[col]: Temperatures and calculated equivalent thermal conductivities.

Table 10:2 shows calculated heat flows through the boundaries for option 1 and 4 with different numerical meshes and relaxation factors ω. The CPU-time for a Pentium III 500 MHz is shown in the last column. An error less than 0.01% is used for the heat flows as a stop criterion, see Eq. 5.1.

[col] See Appendix D for color image

Table 10:2: Heat flow through frame and benchmarks using a Pentium III 500 MHz.

Option	Numerical cells	*q*, (W/m)	HEAT2 4.0 CPU-time
4	6435	10.90	81 sec. (ω=1.95)
			13 sec. (ω=2.0)
			6 sec. (optimized ω=1.998)
4	39795	11.00	1.5 hours (ω=1.95)
			149 sec. (optimized ω=2.0)
1	6435	11.18	75 sec. (ω=1.95)
			12 sec. (ω=2.0)
			6 sec. (optimized ω=1.998)
1	39795	11.28	1.5 hours (ω=1.95)
			125 sec. (optimized ω=2.0)

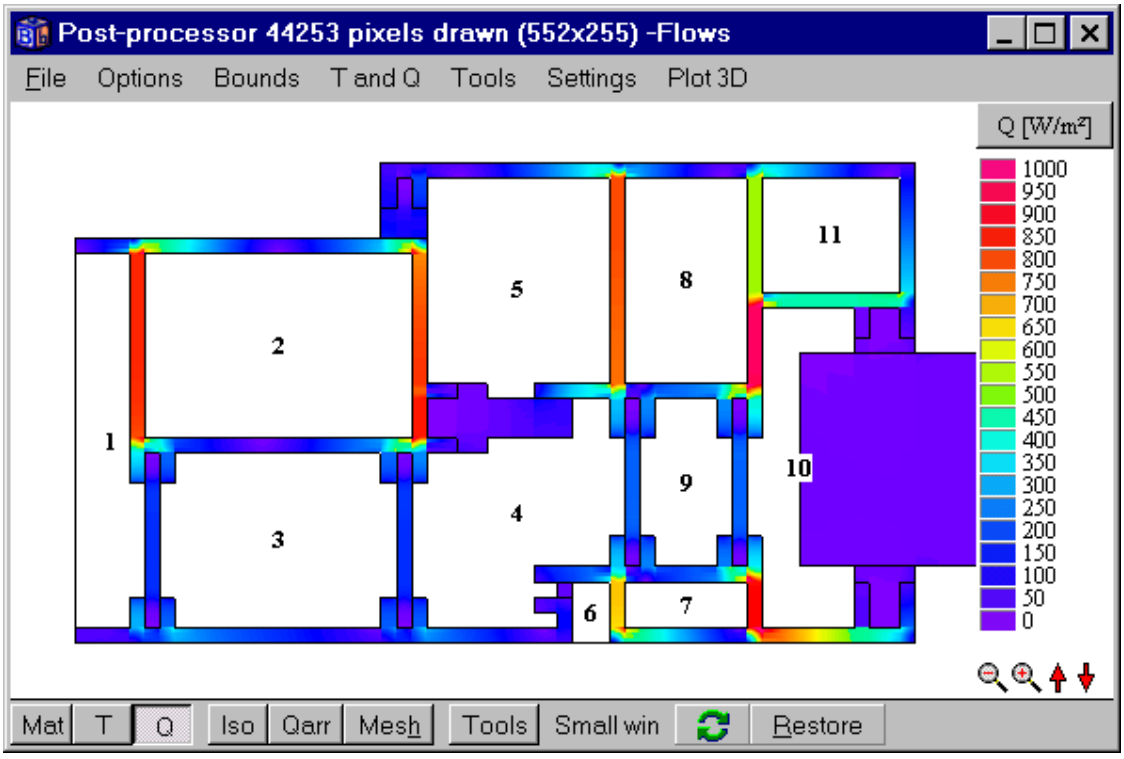


Figure 10.17[col]: Heat flow intensities (W/m²).

[col] See Appendix D for color image

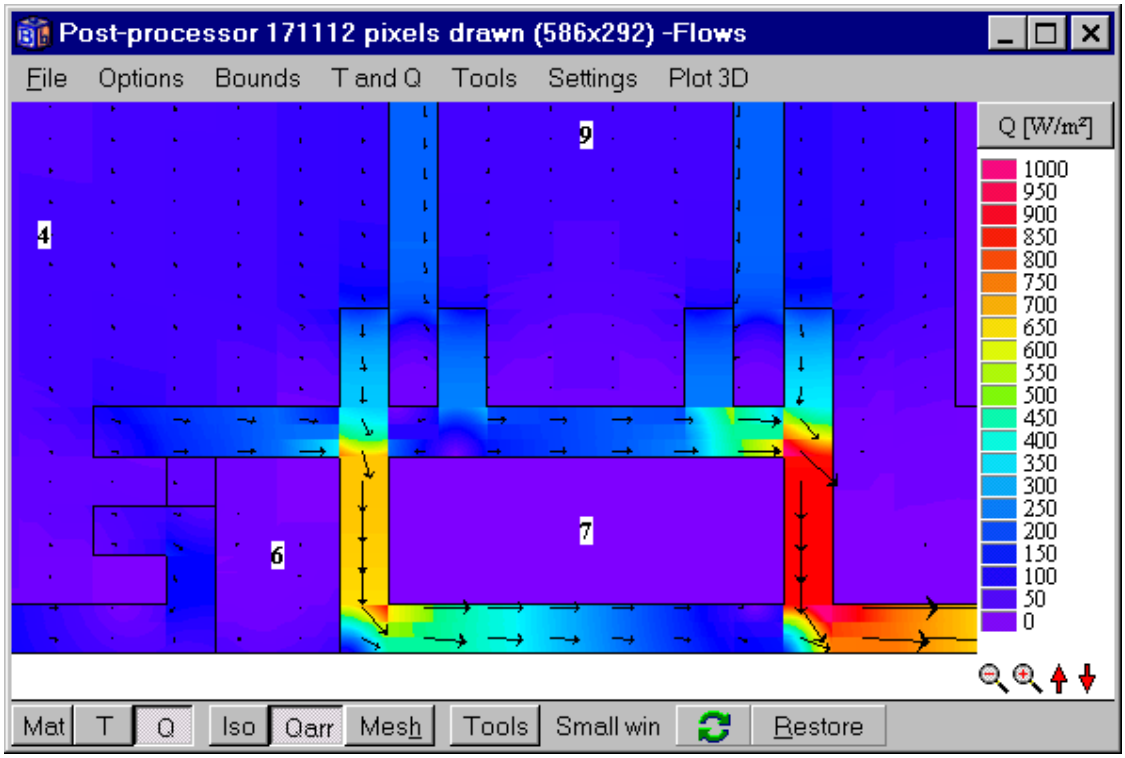


Figure 10.18[col]: Heat flow intensity and direction (here menu item **T and Q/Show T and Q in cavities** is enabled to show it also in the cavities). The area around cavity 7 is enlarged.

[col] See Appendix D for color image

11. A few tips

11.1 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.

11.2 Numerical aspects

11.2.1 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 5.55. 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.

11.2.2 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.

11.2.3 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.

11.2.4 Symmetric cases

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

11.2.5 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).

11.2.6 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 11.2.1. 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.

11.2.7 Sloping boundaries

Sloping internal and external boundaries may be approximated by steps. It is often relatively easy to do this using the pre-processor. Note that the boundary condition values and the surface resistance have to be modified taking the slope into account.

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. Doctoral 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.

DIN (Deutsches Institut für Normung, DIN V 4108-4).

Edwards, A.L. *A Compilation of Thermal Property Data for Computer Heat Conduction Calculations*, UCRL-50589, February 24, 1969

Eftring, 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 on input 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	100
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
Number of CEN/ISO frame cavities	250

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	$1 \le I_{\max} \le 250$	
$J_{\max}$	Num. of ims in y-dir	$1 \le J_{\max} \le 250$	
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	$1 \le N_I \le 250$	
N_J	Number of cells in each ims in y-dir	$1 \le N_J \le 250$	
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/(m³·K)
R	Surface resistance	≥ 0	m²·K/W
R_x	Internal resistance in x-dir	≥ 0	m²·K/W
R_y	Internal resistance in y-dir	≥ 0	m²·K/W
T	Temperature	no restriction	°C
q	Heat flow over boundary	no restriction	W/m²

Appendix B. How to add and edit materials

How to add and edit materials in HEAT2

Section 5.2 shows how to add and edit material properties. The following materials are available in the standard material list (Default.MTL).

Material	***C*** λ_x λ_y **(λ_y does not have to be specified if $\lambda_x=\lambda_y$)**
	MJ/(m³·K), m²·K/W, m²·K/W
IEA Report Annex XIV	
acrylic resin, no cap., CEN	1 0.2 0.2
air	0.0013 0.0251 0.0251
aluminum	2.4835 225.94 225.94
brick, IEA	1.49 0.45 0.45
butyl (hot melt), no cap., CEN	1 0.24 0.24
butyl rubber (solid), no cap., CEN	1 0.24 0.24
concrete, cellular, IEA	0.53 0.52 0.52
concrete, cinder	1.051 0.3347 0.3347
concrete, IEA	1.83 2.7 2.7
concrete, lightweight	0.6241 0.2092 0.2092
concrete, lightweight, IEA	1.83 0.49 0.49
copper	3.441 397.48 397.48
cork, ground	0.3012 0.0431 0.0431
cork, ground, regranulated	0.261 0.0448 0.0448
cork, IEA	0.21 0.042 0.042
epoxy fibre, no cap., CEN	1 0.23 0.23
epoxy resin, no cap., CEN	1 0.2 0.2
epoxy, silica filled, cast	1.8072 0.6276 0.6276
Example 1, concrete	1.83 2.7 2.7
Example 1, cross bars	1.7 0.19 0.19
Example 1, insulation	0.094 0.037 0.037
Example 1, lightweight concrete	1.83 0.49 0.49
Example 2, concrete	1 1.7 1.7
Example 2, haydite	1 0.34 0.34
Example 2, mineral wool	1 0.036 0.036
Example 2, soil	1 2.3 2.3
Example 3, Lambda=0.029	1 0.029 0.029
Example 3, Lambda=0.12	1 0.12 0.12
Example 3, Lambda=1.15	1 1.15 1.15
Example 3, Lambda=230	1 230 230
Example 4, Lambda=0.88	1 0.88 0.88
Example 4, Lambda=0.90	1 0.9 0.9
Example 4, Lambda=2.90	1 2.9 2.9
fireclay brick, missouri	1.5062 1.0042 1.0042
fireclay brick, normal (22 p)	1.4498 1.297 1.297
fireclay brick, siliceous (23 p)	1.5062 1.0878 1.0878
foam rubber, no cap., CEN	1 0.06 0.06
Frame cavity	1 1 1
Frame ex. (a) Insulation panel	1 0.035 0.035
Frame ex. (b) Soft wood	1 0.13 0.13
Frame ex. (c) PVC	1 0.17 0.17
Frame ex. (d) EPDM	1 0.25 0.25
Frame ex. (e) Polyamid 6.6	1 0.3 0.3
Frame ex. (f) Glass	1 1 1
Frame ex. (g) Steel	1 50 50
Frame ex. (h) Aluminium	1 160 160
Frame ex. (i) Mohair (polyester)	1 0.14 0.14
Frame ex. (k) Polyamid Nylon	1 0.25 0.25

Frame ex. (l) PU (polyurethane), resin	1 0.25 0.25
Frame ex. (m) Polysulfid	1 0.4 0.4
Frame ex. (n) Silica gel (desiccant)	1 0.13 0.13
Frame ex. (o) see D.10	1 0.034 0.034
FRP Nylon, no cap., CEN	1 0.23 0.23
glass (avg prop)	1.9246 1.046 1.046
glass fibre resin, no cap., CEN	1 0.3 0.3
glass fibre, no cap., CEN	1 0.23 0.23
glass, borosilicate crown	1.6869 1.046 1.046
glass, ceramic, pyroceram 9606	2.0342 3.9748 3.9748
glass, ceramic, pyroceram 9608	2.0187 2.0502 2.0502
glass, diabase (artificial)	1.8074 1.1715 1.1715
glass, flint	1.205 0.7531 0.7531
glass, float glass, no cap., CEN	1 1 1
glass, foamed (d = 0.144)	0.1084 0.0347 0.0347
glass, fused quartz (sio2)	1.6386 1.3807 1.3807
glass, fused silica or vitrous silica	1.6386 1.3807 1.3807
glass, IEA	2.3 1 1
glass, lead	1.5264 1.1297 1.1297
glass, lime window	1.8677 1.318 1.318
glass, obsidian	1.8074 1.3724 1.3724
glass, plexiglass, no cap., CEN	1 0.2 0.2
glass, pyrex	1.6162 1.1297 1.1297
glass, soda plate	1.8827 1.2134 1.2134
glass, soda-lime	1.7321 1.3389 1.3389
glass, vycor	1.6311 1.3389 1.3389
glass-wool, IEA	0.062 0.033 0.033
gold	2.4704 317.98 317.98
granite (av prop) (see ref 51)	2.2175 2.5104 2.5104
granite (high k)	2.2175 3.9748 3.9748
granite (low k)	2.2175 1.7154 1.7154
gypsum (caso4.4h2o) (artificial)	2.2957 0.7531 0.7531
gypsum (caso4.4h2o) (mineral)	2.5242 1.297 1.297
gypsum board, IEA	0.88 0.1 0.1
ice (h2o) (solid)	3.867 2.092 2.092
iron (0 to 3000 deg c)	3.5234 71.965 71.965
limestone (dense, dry)	2.3012 1.6736 1.6736
limestone (h2o 15.3)	1.5188 0.9205 0.9205
masonry, IEA	0.88 0.1 0.1
mohair, no cap., CEN	1 0.14 0.14
mortar, IEA	1.21 0.53 0.53
neoprene, no cap., CEN	1 0.25 0.25
nylon, no cap., CEN	1 0.25 0.25
plaster, building (molded, dry)	1.36 0.431 0.431
plastic laminate, various types	2.259 0.2092 0.2092
plate glass	1.8827 1.2134 1.2134
plywood, IEA	1.17 0.125 0.125
polyamide, no cap., CEN	1 0.3 0.3
polycarbonate, no cap., CEN	1 0.23 0.23
polyester resin, no cap., CEN	1 0.3 0.3
polyethylene (high dens), no cap., CEN	1 0.52 0.52
polyethylene, no cap., CEN	1 0.4 0.4
polyisobutylene, no cap., CEN	1 0.24 0.24
polystyrene expanded, IEA	0.039 0.033 0.033
polystyrene extruded, IEA	0.059 0.024 0.024
polystyrene foam (d = 0.038) (1 atm)	0.0429 0.0328 0.0328
polystyrene foam (d = 0.046) (1 atm)	0.052 0.0259 0.0259
polystyrene foam (d = 0.046) (vacuum)	0.052 0.0081 0.0081
polystyrene, foamed-in-place, rigid	0.113 0.0347 0.0347
polystyrene, general purpose	1.4941 0.1255 0.1255
polystyrene, modified	1.4461 0.0837 0.0837
polystyrene, no cap., CEN	1 0.17 0.17
polystyrene, prefoamed, rigid, dow	0.0452 0.0418 0.0418

polysulfone	1.5688 0.2594 0.2594
polysulphide, no cap., CEN	1 0.19 0.19
polyurethane foam, flexible	0.1054 0.0418 0.0418
polyurethane foamed-in-place, rigid	0.067 0.0325 0.0325
polyurethane rubber l-100	2.0925 0.2929 0.2929
polyurethane, no cap., CEN	1 0.3 0.3
polyvinyl, no cap., CEN	1 0.4 0.4
porous wood fibre panel, no cap., CEN	1 0.08 0.08
PTFE, no cap., CEN	1 0.24 0.24
PVC, no cap., CEN	1 0.17 0.17
PVC/Vinyl (rigid), no cap., CEN	1 0.17 0.17
pyrex glass	1.6162 1.1297 1.1297
quartz sand (dry) (av prop) (see ref)	1.205 0.3347 0.3347
quartz sand (wet) (h2o 4-23) (av prop)	1.2803 1.6736 1.6736
rock or stone (average properties)	2.1757 1.7573 1.7573
rock-wool, IEA	0.094 0.037 0.037
rubber, buna, with carbon black	1.757 0.2427 0.2427
rubber, butyl	1.7694 0.0879 0.0879
rubber, dielectric mix	2.3012 0.2092 0.2092
rubber, high k	2.3012 0.2929 0.2929
rubber, natural	1.9456 0.1381 0.1381
rubber, natural, foam	0.2092 0.0418 0.0418
rubber, neoprene	2.72 0.1925 0.1925
rubber, nitrile	1.966 0.2427 0.2427
rubber, polyurethane elastomer l-100	2.0925 0.2929 0.2929
sand (dry) (d = 1.36 to 1.84) (av prop)	1.205 0.3347 0.3347
sand, northway (h2o 4-10) (av prop)	1.4226 0.8368 0.8368
sand, quartz (wet) (h2o 4-23) (av prop)	1.2803 1.6736 1.6736
sandstone (av prop) (see ref 51)	2.2133 2.5104 2.5104
sandstone (high density)	2.502 4.184 4.184
sandstone (low density)	2.1171 1.841 1.841
silica brick, hard fired (sio2 94-95)	1.3556 1.6736 1.6736
silica gel (desiccant), no cap., CEN	1 0.13 0.13
silica glass, fused or vitrous	1.6386 1.3807 1.3807
siliceous brick (sio2 89, al2o3 9)(25p)	1.4535 0.9372 0.9372
silicon	1.6378 125.52 125.52
silicon boride (sib4)	2.5732 9.8324 9.8324
silicon carbide (sic) (brick al2o3 1.7)	1.7013 11.715 11.715
silicone foam, no cap., CEN	1 0.12 0.12
silicone, no cap., CEN	1 0.35 0.35
snow, fresh	0.2092 0.1046 0.1046
snow, packed	1.1506 0.4686 0.4686
soda-lime glass	1.7321 1.3389 1.3389
soil (av props)	1.3598 0.8368 0.8368
soil, clay (wet)	4.3935 1.5062 1.5062
soil, fine quartz flour (dry)	0.6554 0.1674 0.1674
soil, fine quartz flour (h2o 21 pc)	2.6645 2.2175 2.2175
soil, loam (dry) (av prop) (see refs)	1.0042 0.251 0.251
soil, loam (h2o 4-27 pc) (av, see refs)	1.6736 0.4184 0.4184
soil, mars surface (see ucrl-50309)	1.9246 0.0134 0.0134
soil, sandy (h20 8)	1.757 0.5858 0.5858
soil, sandy dry	1.3117 0.2636 0.2636
steel, alloy and mild (4130, 4340)	3.8913 41.84 41.84
steel, alloy, cast	3.6034 46.024 46.024
steel, IEA	3.7 60 60
steel, stainless ca15, ca40 (cast)	3.5021 25.104 25.104
urethane (liquid), no cap., CEN	1 0.3 0.3
vinyl (flexible), no cap., CEN	1 0.12 0.12
water (h2o) (gas) (1 atm)	0.0012 0.0234 0.0234
water (h2o) (gas) (satd)	0.0011 0.0178 0.0178
water (h2o) (liquid)	4.184 0.6025 0.6025
water (h2o) (solid) (ice)	3.867 2.092 2.092
water, heavy (h2o) (liquid)	4.6255 0.5607 0.5607

window glass, lime	1.8677 1.318 1.318
wood chip board, no cap., CEN	1 0.13 0.13
wood, balsa (across grain)	0.7363 0.0837 0.0837
wood, cypress (across grain)	1.0585 0.0962 0.0962
wood, hardwood, no cap., CEN	1 0.2 0.2
wood, mahogany (across grain)	1.2425 0.1297 0.1297
wood, maple (across grain)	1.6337 0.1757 0.1757
wood, norway pine (across grain)	1.2656 0.1506 0.1506
wood, oak	1.7 0.19 0.19
wood, oak, red, black (across grain)	1.4548 0.1464 0.1464
wood, oak, white, live (across grain)	1.9676 0.2092 0.2092
wood, oregon pine (across grain)	1.1735 0.113 0.113
wood, pine (with grain)	1.2656 0.3431 0.3431
wood, plywood, no cap., CEN	1 0.15 0.15
wood, redwood/cedar, no cap., CEN	1 0.11 0.11
wood, softwood, no cap., CEN	1 0.14 0.14
wood, spruce (across grain)	0.5146 0.1255 0.1255
wood, spruce (with grain)	0.5146 0.2301 0.2301
wood, teak (across grain)	1.4726 0.1715 0.1715
wood, virginia pine (across grain)	1.2425 0.1423 0.1423
wood, white fir (across grain)	1.0354 0.1088 0.1088
wood, white pine (across grain)	1.2195 0.1297 0.1297
woodwool cement	0.64 0.071 0.071
zinc	2.7782 111.71 111.71

Appendix C. Input data file example

```
**********************************************************************
* HEAT2 version 5.00 - Input data file EXAMPLE 3.DAT
* Date: 5/5/00  Time: 11:59:34 AM
* Thomas Blomberg, Lund Group for Computational Building Physics
**********************************************************************
-MEMORY NOTES FOR PROJECT-




* 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: 3
* Number of input mesh intervals in Y-direction: 5
* Number of boundaries: 4
* Description of boundaries
  bound   from   I     J
    1          :0    :0
    2          :3    :0
    3          :3    :5
    4          :0    :5

* Lengths, number of computation cells and expanding coefficient
  between the mesh points
* X-direction
              length [m]    comp cells    exp coeff
  0 and  1   : 0.001500      : 1         : 1
  1 and  2   : 0.013500      : 15        : 1
  2 and  3   : 0.485000      : 48        : 1.08

* Y-direction
              length [m]    comp cells    exp coeff
  0 and  1   : 0.001500      : 2         : 1
  1 and  2   : 0.033500      : 35        : 1
  2 and  3   : 0.001500      : 2         : 1
  3 and  4   : 0.005000      : 5         : 1
  4 and  5   : 0.006000      : 6         : 1

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

* Basic properties for whole area

  Lambda(x)              : 1 [W/(m·K)]
  Lambda(y)              : 1 [W/(m·K)]
  Volumetric heat capacity : 1 [MJ/(m³·K)]
Frame cavity

* Number of partial areas :6
* 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    :0   :0    :3   :5     :0.029  :0.029 :1 Example 3, Lambda=0.029
   2    :0   :4    :3   :5     :1.15  :1.15 :1 Example 3, Lambda=1.15
```

```
  3   :0  :3   :2  :4    :0.12  :0.12 :1 Example 3, Lambda=0.12
  4   :0  :2   :2  :3    :230  :230 :1 Example 3, Lambda=230
  5   :0  :0   :1  :2    :230  :230 :1 Example 3, Lambda=230
  6   :0  :0   :3  :1    :230  :230 :1 Example 3, Lambda=230

***********************************
*   FUNCTIONS BLOCK               *
***********************************

* Combinations
*  A  Periodic f(t)=f1+f2*Sin(2*pi*(t-t0)/tp)
*  B  Stepwise Constant
*  C  Stepwise Linear

* Number of Functions:3
* Typenumber/Combination
  1/ :A f1:1 f2:1 t0:0.000 tp:86400.000
  2/ :A f1:1 f2:1 t0:0.000 tp:86400.000
  3/ :A f1:1 f2:1 t0:0.000 tp:86400.000

***********************************
*   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 : 20   °C  R : 0.11 [m²·K/W]
  3/ :B Temperature : 0   °C  R : 0.06 [m²·K/W]

* Typenumber/Corresponding Boundaries (list ends with zero)
  1/ :2 :4 :0
  2/ :1 :0
  3/ :3 :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)
& 0.0001 (Flow crit.)
& 0.0001 (Temp crit.)
& 100000 (Max iter.)
& 1 (Update)
& 10 (Iter)
& 3 (CPU)
& 3600 (Time)
& 0 (record disabled)
& 1423 (Number of iterations)
& 1 (notify when solved enabled)
& 0 (beep when solved disabled)
& 1 (Order of approximation)
& 3 (Frame cavity option)
& 0 (Frame cavity Tmin)
& 20 (Frame cavity Tmax)
& 0 (Frame cavity direction of heat flows)
& 100 (Frame cavity iter between update)
& 1 (Column for grahics)
& 1 (Num Record columns)
& 2 (index)
& 0 (x)
& 0 (y)
Material file :DEFAULT.MTL
H2P file :EXAMPLE 3.H2P
& 50 (grid x)
& 50 (grid y)
& 0.5 (scale factor)
& 1 (long boundaries)
& 1 (Update made)