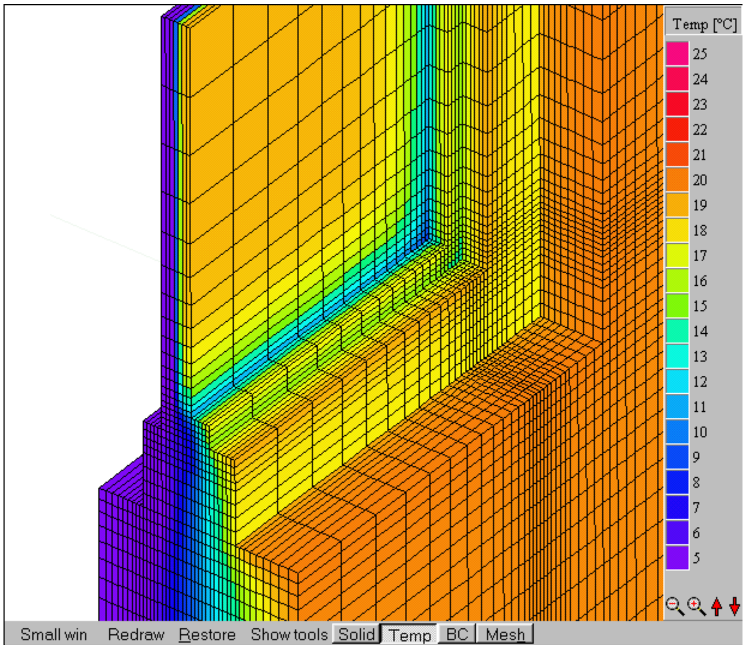
Preliminary version

HEAT3

A PC-program for heat transfer in three dimensions.

Manual with brief theory and examples.

Version 4.0



September 28, 2001

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 7

1.1 HOW TO ALMOST AVOID READING THIS MANUAL 7
1.2 INTRODUCTION TO HEAT3 7
1.2.1 Overview 7
1.2.2 Features 7
1.2.3 HEAT3 compared to HEAT2 8
1.2.4 Versions of HEAT3 and system requirements 8
1.2.5 Cartesian coordinates 8
1.2.6 Numerical method 8
1.2.7 Pre-processor 9
1.3 NEW VERSION 4.0 COMPARED TO VERSION 3 14
1.3.1 Integrated pre-processor 14
1.3.2 Extensive post-processing capabilities: 14
1.3.3 Recorder 14
1.3.4 Material properties 14
1.3.5 Import of climate files 14
1.3.6 Faster calculation 14
1.3.7 Important news for users of version 3 14
1.4 INSTALLATION 14
1.5 TECHNICAL SUPPORT 15

2. MATHEMATICAL DESCRIPTION 17

2.1 GOVERNING DIFFERENTIAL EQUATIONS 17
2.2 BOUNDARY CONDITIONS 17
2.3 INITIAL CONDITIONS 18

3. NUMERICAL FORMULATION 19

3.1 INTRODUCTION 19
3.2 THERMAL CONDUCTANCES 19
3.3 HEAT FLOWS 20
3.4 NEW TEMPERATURES 21
3.5 CHOICE OF TIME-STEP 22
3.6 ITERATIVE CALCULATION 22
3.7 STEADY-STATE CASE 22

4. OVERVIEW OF INPUT USING THE PRE-PROCESSOR 23

4.1 PRE-PROCESSOR 23
4.2 BOUNDARY CONDITIONS 28
4.3 SOLVING A STEADY-STATE PROBLEM 39
4.4 SOLVING A TRANSIENT PROBLEM 39

5. WORKING WITH HEAT3 45

5.1 THE PRE-PROCESSOR 45
5.1.1 Introduction 45
5.1.2 Generating input data 45
5.1.3 Describing 3D geometry using 2D drawings 45
5.1.4 The object inspector 45
5.1.5 The tool bar 46
5.1.6 The tool bar 46

5.1.7 Copying objects *48*
5.1.8 Picking material *48*
5.1.9 Snap to grid *48*
5.1.10 Zoom scale *48*
5.1.11 Scale factor *49*
5.1.12 Drawing in metric or English units *49*
5.1.13 Numerical mesh *49*
5.1.14 Importing pictures *50*
5.1.15 Importing geometry from H3P- and H2P-files. *50*
5.2 THE MATERIAL EDITOR 52
5.2.1 Introduction *52*
5.2.2 Saving material files *55*
5.2.3 Importing and merging other material files *55*
5.3 BOUNDARY CONDITIONS 55
5.3.1 Introduction *55*
5.3.2 Disabling sides of a BC box *56*
5.4 INITIAL TEMPERATURES 57
5.5 HEAT SOURCES 58
5.6 FUNCTIONS 59
5.6.1 Sinusoidal function *60*
5.6.2 Step-wise constant and step-wise linear function *61*
5.6.3 Import from other file formats *62*
5.6.4 Example 1 - sinusoidal function with a time period of one year *62*
5.6.5 Example 2 - sinusoidal function with a time period of one day *63*
5.6.6 Example 3 - step-wise constant function *64*
5.6.7 Example 4 - step-wise linear function *64*
5.7 SOLVING THE PROBLEM 64
5.7.1 Steady-state stop criteria *64*
5.7.2 Successive over-relaxation used for the steady-state calculations *66*
5.7.3 Simulation time for transient calculation *67*
5.7.4 Simulation window *68*
5.7.5 Screen update *69*
5.7.6 Output file options *69*
5.7.7 Reset *69*
5.8 OUTPUT DATA - THE RECORDER 70
5.8.1 Introduction *70*
5.8.2 What can be recorded? *74*
5.8.3 Changing the record chart *74*
Editor design 75
5.9 OTHER OUTPUT DATA 76
5.9.1 Flows and temperatures for sets *76*
5.9.2 Temperature and heat flows at a given point *76*
5.9.3 Output files with temperature planes *77*
5.9.4 Output files for Matlab 3D figures *79*
5.10 SAVING CONDUCTANCES, CAPACITIES, AND TEMPERATURES TO FILE 80
5.11 POST-PROCESSOR 80
5.11.1 Introduction *80*
5.11.2 Basic commands *81*
5.11.3 Menu item Details *82*
5.11.4 Viewing thermal conductivities and heat capacities *83*
5.11.5 Viewing boundary conditions *84*
5.11.6 Viewing temperatures and heat flows *86*
5.11.7 Removing material boxes in the picture *90*
5.11.8 Cut-out areas *91*

5.11.9 Saving images 92
5.11.10 Maximizing scales 92
5.12 INFO LOG 93

6. NUMERICAL PERFORMANCE 95

6.1 BENCHMARKS 95
Steady-state calculations 95

7. FORMAT FOR TEXT INPUT DATA 97

7.1 INTRODUCTION 97
7.2 EDITING DATA 97
7.2.1 The editor 97
7.2.2 Inserting mesh coordinates 98
7.3 INPUT MESH 100
7.4 SURFACES DEFINING BOUNDARY CONDITIONS 103
7.5 NUMERICAL MESH 104
7.6 INTERNAL RESISTANCES 105
7.7 INPUT DATA FILE 105
7.8 EXAMPLE 1 106
7.9 EXAMPLE 2 108
7.10 EXAMPLE 3 110
7.11 EXAMPLE 4 111
7.12 THE INFO LOG 113

8. MORE EXAMPLES USING TEXT INPUT 115

8.1 INTRODUCTION TO EXAMPLES 115
8.1.1 Example 5, metal studs in a wall 115
8.1.2 Example 6, slotted metal studs in a wall 115
8.1.3 Example 7, heat flow through a corner 115
8.1.4 Example 8, slab on the ground 115
8.1.5 Example 9, ground heat storage 115
8.2 EXAMPLE 5, METAL STUDS IN A WALL, STEEL1.DAT 116
8.2.1 Introduction 116
8.2.2 Input file steel1.dat 118
8.3 RESULTS 119
8.4 EXAMPLE 6, SLOTTED METAL STUDS, STEEL2.DAT 121
8.4.1 Introduction 121
8.4.2 Data for the numerical calculation 121
8.4.3 Numerical mesh 122
8.4.4 Input file, steel2.dat 124
8.4.5 Results 125
8.5 EXAMPLE 7, HEAT FLOW THROUGH A CORNER, CEN.DAT 126
8.5.1 Introduction 126
8.5.2 Input file, cen.dat 129
8.5.3 Results 131
8.6 EXAMPLE 8, SLAB ON THE GROUND, SLAB.DAT 133
8.6.1 Introduction 133
8.7 INPUT FILE, SLAB.DAT 136
8.8 RESULTS 137
8.9 EXAMPLE 9, GROUND HEAT STORAGE, CAV.DAT 139
8.9.1 Introduction 139
8.9.2 Input file, cav.dat 141
8.9.3 Results 143

9. A FEW TIPS 145

9.1 INTRODUCTION 145
9.2 NUMERICAL ASPECTS 145
9.2.1 Numerical mesh 145
9.2.2 Expansive meshes 145
9.2.3 Steady-state calculations 145
9.2.4 Symmetric cases 145
9.2.5 Computational area for calculations of heat losses to the ground 145
9.2.6 Transient calculations for heat losses to the ground 145

REFERENCES 145

APPENDIX A. THERMAL DATA 147

APPENDIX B. MATLAB EXAMPLES 149

APPENDIX C. IMPORTANT NEWS FOR USERS OF VERSION 3 151

1. Introduction

1.1 How to almost avoid reading this manual

For a quick start read Chapter 4 *Overview of input using the pre-processor.* The examples in Chapter 5 would also be useful.

1.2 Introduction to HEAT3

1.2.1 Overview

HEAT3 is a PC-program for three-dimensional transient and steady-state heat conduction. The heat equation is solved with explicit forward finite differences. The successive over-relaxation technique is used in the steady-state case.

The program can be used for analyses of thermal bridges, heat transfer through corners of a window, heat loss from a house to the ground, to mention a few applications. One important restriction is that the problem has to be described in a parallelepipedical mesh, i.e. all boundary surfaces are parallel to one of the Cartesian coordinate planes.

For a reasonably complicated case, 10-15 minutes work is sufficient for an experienced user to describe the geometry, the numerical mesh, and the boundary conditions. A problem often takes less than a few minutes to solve for steady-state.

The input and output may be viewed graphically (geometry, numerical mesh, boundary conditions, temperature field). The three-dimensional figure may be rotated in space, and details of particular interest can be enlarged.

The output temperature field may be written in a special format and imported directly in Matlab (MathWorks Inc., 1992). This program has powerful options concerning plots in two and three dimensions.

HEAT3 offers high calculation speed and allows for memory allocation for large three-dimensional problems. Up to one million nodes may be used in the standard version but separate compilations are available with for example 50 million cells.

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.

1.2.2 Features

The following list shows some features:

- An integrated pre-processor facilitates the input procedure, see section 1.2.7.
- Material properties may easily be edited and added. Several material lists 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 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. 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.
- The recorder may save at specified intervals (transient or steady-state) temperatures and heat flows at given points, heat flow through boundaries.

- A simple numerical mesh generation option is available. The mesh can easily be changed.
- Any structure consisting of adjacent or overlapping parallelepipeds with any combination of materials may be simulated.
- 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.
- Heat sources/sinks may be specified.
- 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.
- Various formats with climatic data may be imported (TRNSYS, METEONORM, HELIOS, DOE, TMY2 - Typical Meteorological Year, SUNCODE, MATCH) for dynamic calculations

1.2.3 HEAT3 compared to HEAT2

There are many similarities between HEAT2 (Blomberg, 2000), HEAT2R (Blomberg, 1994) and HEAT3. These programs are based on the same numerical method, they automatically calculate the maximum stable time-step, and they use the successive over-relaxation technique for the steady-state case, to mention a few of the similarities.

1.2.4 Versions of HEAT3 and system requirements

There are two different versions of HEAT3 on the diskette, see Table 1:1. The maximum number of computational cells depends on the amount of memory (RAM) available. One million require about 44 MB RAM, and 125000 cells can be addressed with 14 MB. Versions for other memory configurations are available upon request. A version with 50 million cells require a PC with about 2 GB RAM.

It may be a good idea to close other programs (especially those that require much memory) when running HEAT3.

HEAT3 requires Windows 95/98/2000/NT (or compatible) and 5 MB hard disk space.

Version	Maximum cells	Required memory	Recommended memory in PC
HEAT3-50	50·50·50=125000	14 MB	32 MB
HEAT3-C	100·100·100=1000000	44 MB	64 MB

Table 1:1: Standard versions of HEAT3.

1.2.5 Cartesian coordinates

HEAT3 solves most of the heat conduction problems that can be described in Cartesian coordinates, i.e. all internal and external boundaries are parallel with one of the directions *x,y,z*. This tie to rectangular coordinates may be a limitation in some cases, but it often reduces the amount of input needed from the user. In addition to this, building technology is often a “Cartesian world”. Sloped boundaries, such as sloping insulation under a concrete slab edge have to be modeled by step-wise pieces. It is however quite easy to cut and paste multiple step-wise pieces in the pre-processor.

1.2.6 Numerical method

The heat conduction equation is solved 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 100 000 computational nodes is usually less than a few minutes on a Pentium.

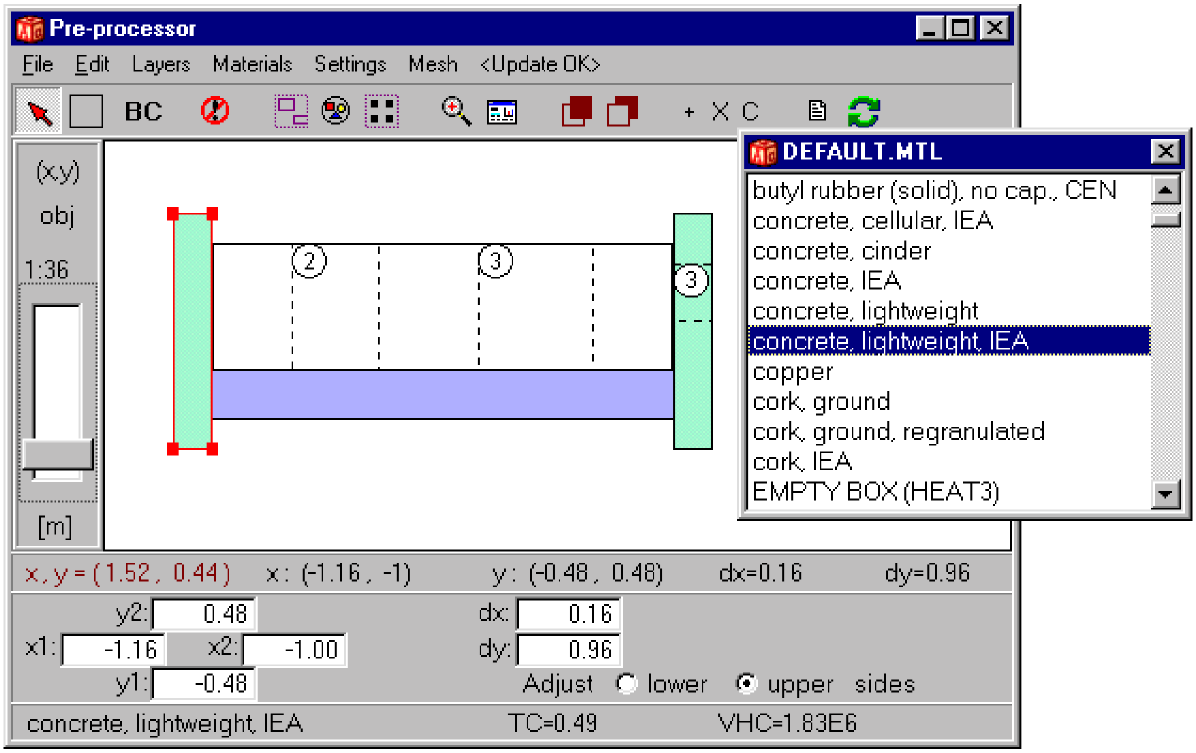
1.2.7 Pre-processor

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 boxes 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 an example with simple geometry as described in the (*x*,*y*)-plane and the (*x*,*z*)-plane. The post-processor shows the result in a three-dimensional view, see Figure 1.2 (materials and boundary conditions) and Figure 1.3 (calculated temperatures and heat flows, respectively).

The following editing features are available in the pre-processor: 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.

Objects drawn in the pre-processor can be selected and copied. Figure 1.4 shows one bench copied three times.

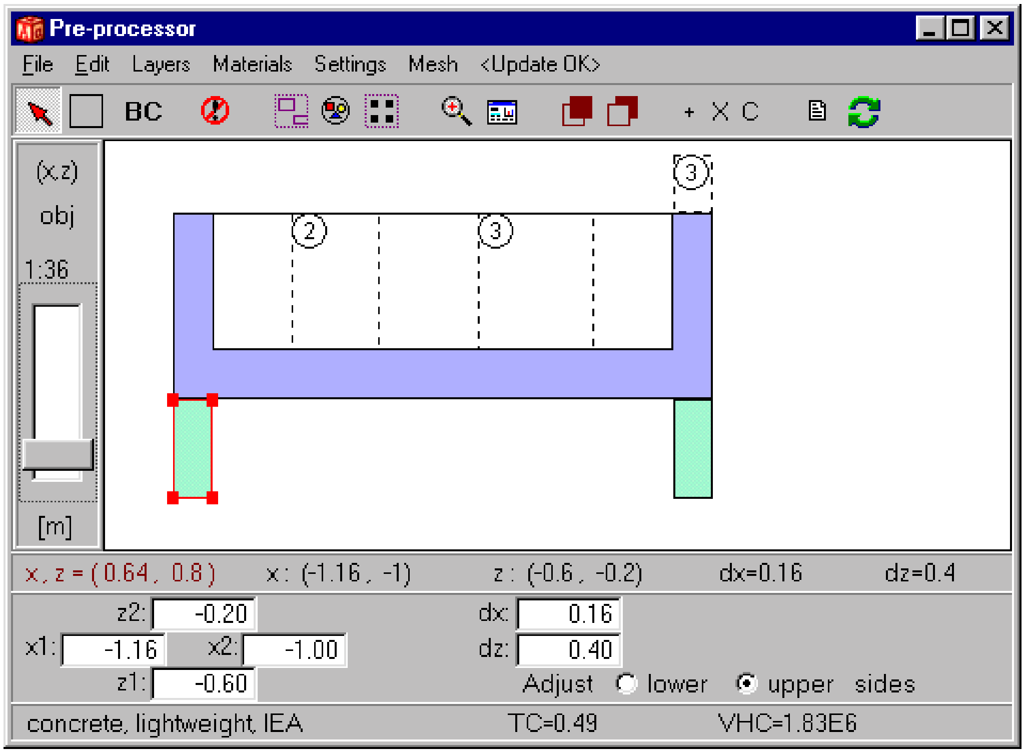


butyl rubber (solid), no cap., CEN
concrete, cellular, IEA
concrete, cinder
concrete, IEA
concrete, lightweight
concrete, lightweight, IEA
copper
cork, ground
cork, ground, regranulated
cork, IEA
EMPTY BOX (HEAT3)

x, y = (1.52 , 0.44)   x : (-1.16 , -1)   y : (-0.48 , 0.48)   dx=0.16   dy=0.96

y2: 0.48
x1: -1.16   x2: -1.00
y1: -0.48
dx: 0.16
dy: 0.96
Adjust ○ lower ● upper sides

concrete, lightweight, IEA   TC=0.49   VHC=1.83E6



x, z = (0.64 , 0.8)   x : (-1.16 , -1)   z : (-0.6 , -0.2)   dx=0.16   dz=0.4

z2: -0.20
x1: -1.16   x2: -1.00
z1: -0.60
dx: 0.16
dz: 0.40
Adjust ○ lower ● upper sides

concrete, lightweight, IEA   TC=0.49   VHC=1.83E6

Figure 1.1: A park bench drawn in the pre-processor. The top figure shows the (*x*,*y*)-plane and the lower one the (*x*,*z*)-plane. The material pick list is shown to the right.

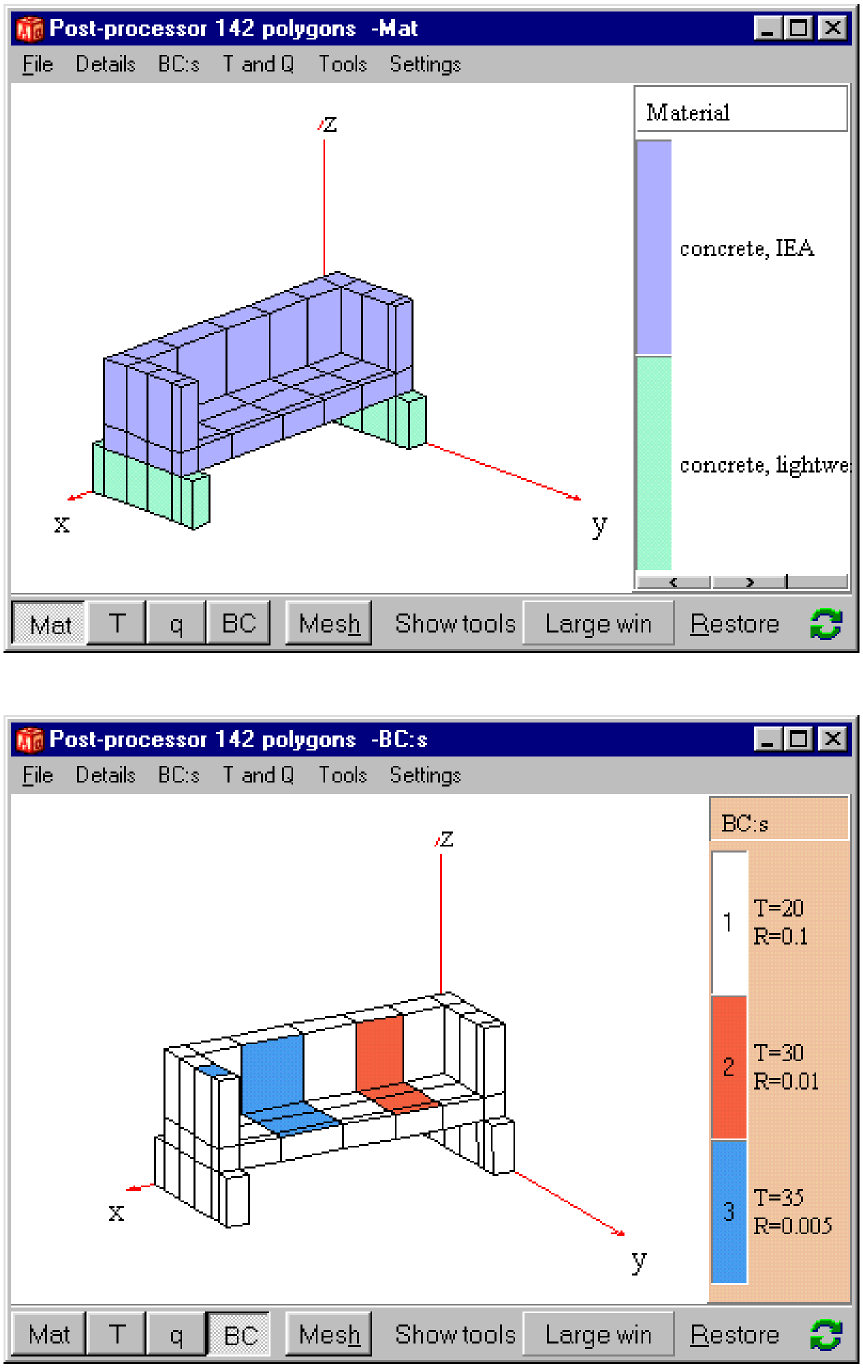
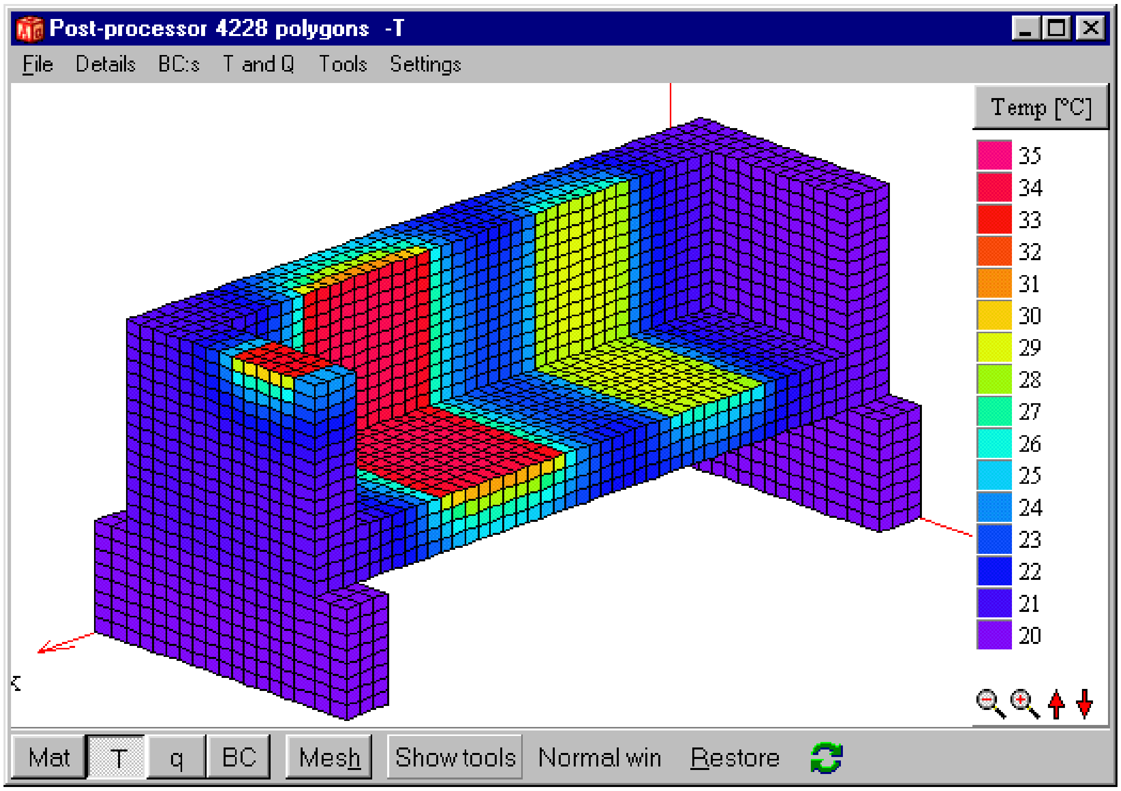

Post-processor 142 polygons -Mat
File Details BC:s T and Q Tools Settings
z
x
y
Material
concrete, IEA
concrete, lightwe
Mat T q BC Mesh Show tools Large win Restore
Post-processor 142 polygons -BC:s
File Details BC:s T and Q Tools Settings
z
x
y
BC:s
1 T=20 R=0.1
2 T=30 R=0.01
3 T=35 R=0.005
Mat T q BC Mesh Show tools Large win Restore

Figure 1.2: The post-processor shows a 3D view, here of the materials (top figure) and boundary conditions (lower figure).


Post-processor 4228 polygons -T
File Details BC:s T and Q Tools Settings
Temp [°C]
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
X
Mat T q BC Mesh Show tools Normal win Restore

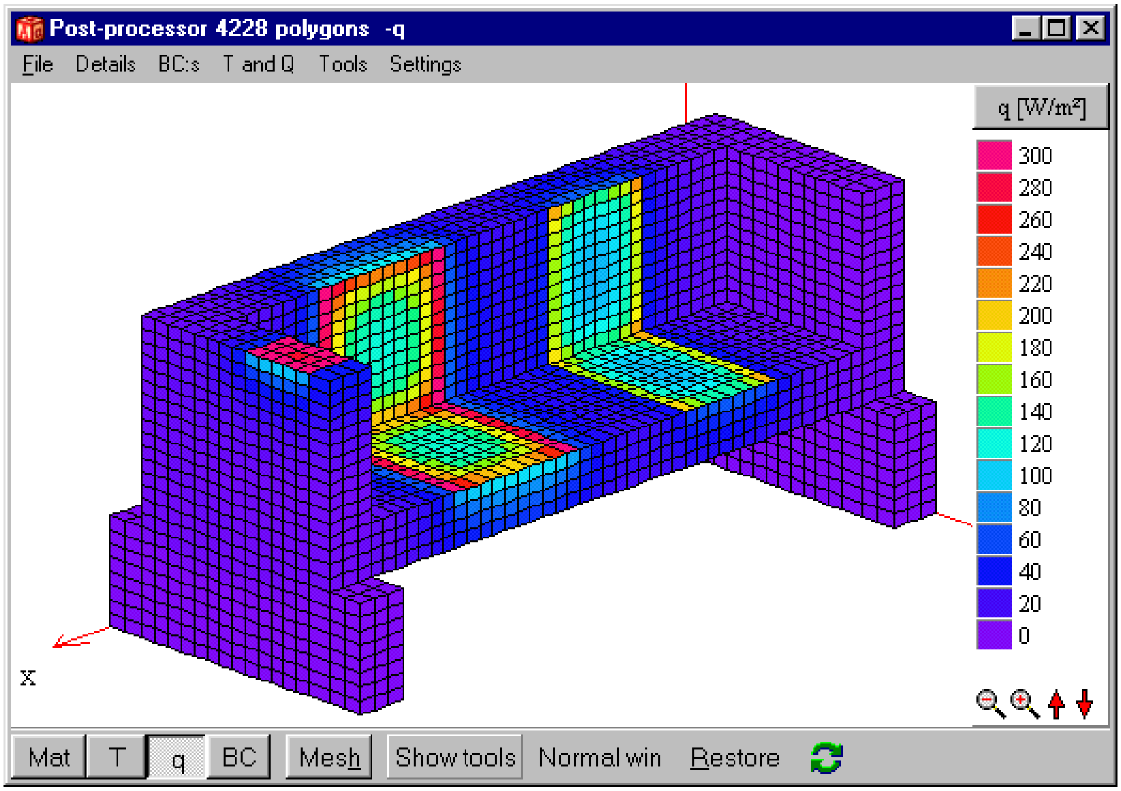
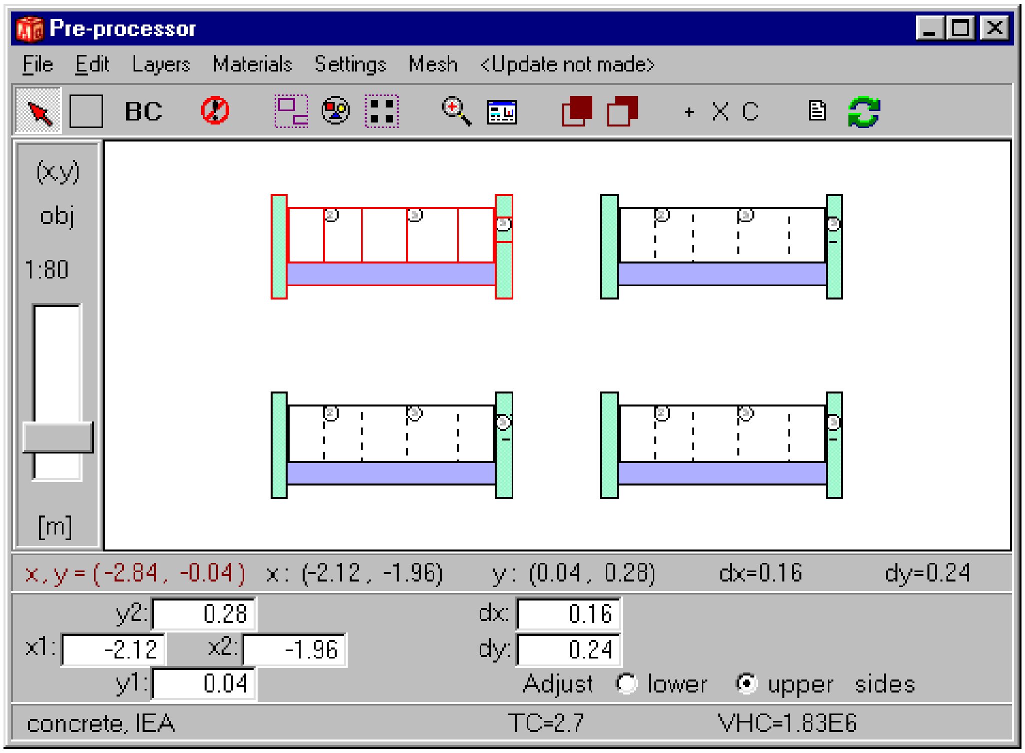

Post-processor 4228 polygons -q
File Details BC:s T and Q Tools Settings
q [W/m²]
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0
X
Mat T q BC Mesh Show tools Normal win Restore

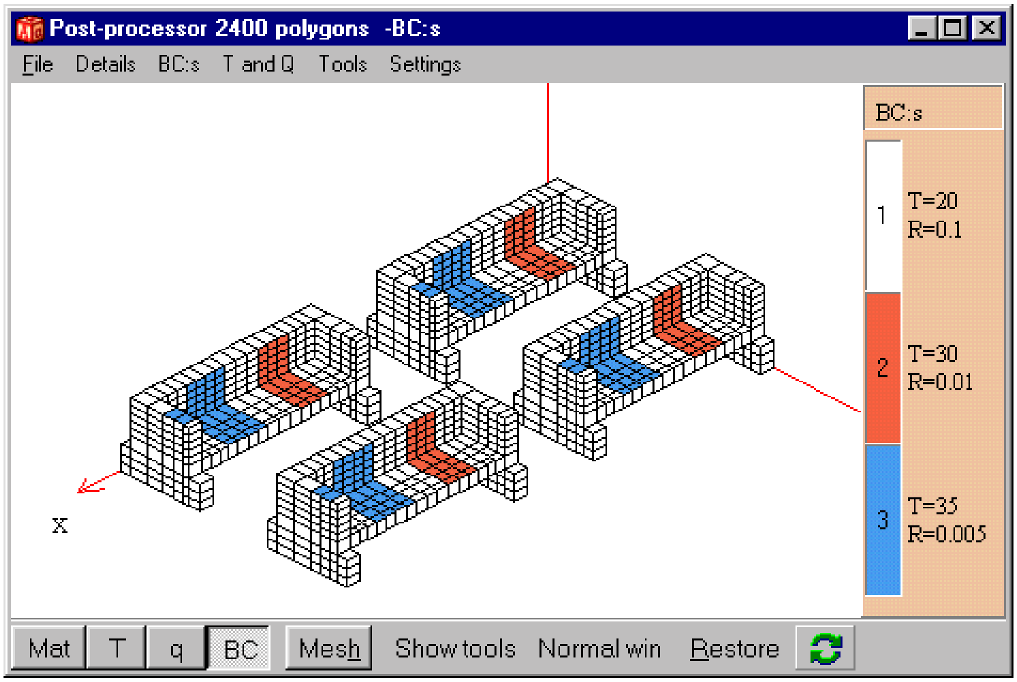
Figure 1.3: Calculated temperature field (top) and heat flows (bottom).



```
x,y = (-2.84, -0.04)  x: (-2.12, -1.96)   y: (0.04, 0.28)   dx=0.16   dy=0.24
```

y2: 0.28		dx: 0.16	
x1: -2.12	x2: -1.96	dy: 0.24	
y1: 0.04		Adjust ○ lower ● upper sides	

```
concrete, IEA          TC=2.7        VHC=1.83E6
```


Post-processor 2400 polygons -BC:s
File Details BC:s T and Q Tools Settings
X
Mat T q BC Mesh Show tools Normal win Restore

BC:s

1	T=20
	R=0.1

2	T=30
	R=0.01

3	T=35
	R=0.005

Figure 1.4: One bench is here copied three times in the pre-processor (top). The postprocessor shows the position of the benches in 3D.

1.3 New version 4.0 compared to version 3

The following list shows the most important new features for version 4 compared to version 3.

1.3.1 Integrated pre-processor

A new CAD-like integrated pre-processor has been developed. See Section 4.1. The pre-processor makes input much simpler. The geometry is built using boxes of different materials that may overlap each other. Boundary conditions, numerical mesh, and heat sources are also easily given by graphical input. Materials may be cut-out by drawing "empty boxes".

It is (as in previous versions) possible to give input data in text mode using the text editor, see Section 5.1.2.

HEAT3 4 can import 2D geometry given in HEAT2 (H2P-files from version 5).

1.3.2 Extensive post-processing capabilities:

The post-processor shows geometry, materials, numerical mesh, boundary conditions, temperature and heat flow fields. Features: zoom, panning, rotation, color/gray-scale, high-resolution printing. Geometry may be cut-out allowing areas inside the material to be drawn. One or more material boxes can be made invisible.

1.3.3 Recorder

Heat flows and temperatures can be recorded and shown during the simulation. The recorder can at specified intervals (transient or steady-state) save temperatures and heat flows at given points, and heat flow through boundary surfaces.

1.3.4 Material properties

Material properties may easily be edited and added. Over 1200 materials are available. There is also over 200 materials available in German from the German DIN standard (Deutsches Institut für Normung, DIN V 4108-4). The format of the material list file is the same as in HEAT2 version 5. If you have edited/added a list in HEAT2 you will be able to use this right away. See Section 5.2.

1.3.5 Import of climate files

Various formats with climatic data may be imported, see Section 5.6.3.

1.3.6 Faster calculation

HEAT3 version 4 is faster than version 3. The improvement is greater for Pentium II, III and IV than for ordinary Pentium (I) processors. Section 6.1 shows benchmarks for some different problems. HEAT3 version 4 are for the steady-state problems 1.3-3.5 times faster compared with HEAT3 version 3. A transient calculation is as much as 9 times faster.

1.3.7 Important news for users of version 3

Please see Appendix C.

1.4 Installation

Before installation it is recommended to make a backup copy of the installation file. To install HEAT3 run the SETUP.EXE program from Windows. The program will be installed in directory *HEAT3_40* 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:

- HEAT3.EXE is the main program.
- Files with extension MTL are files with materials.
- Files with extension H3P are files created by the pre-processor.
- Files with extension DAT are input data files in text format.

1.5 Technical support

News is regularly updated on www.buildingphysics.com or 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 three dimensions for the temperature *T(x,y,z,t)* is

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

Here *I*, (W/m³), is the rate of internal heat generation. The thermal conductivity is denoted by λ, (W/(m·K)). 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 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 which can be applied to the boundary surfaces (*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):

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

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

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

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

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

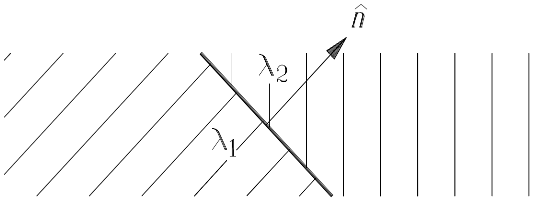


Figure 2.1: The normal $\hat{n}$ at an internal boundary.

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

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

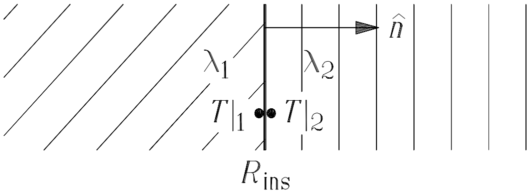


Figure 2.2: Case involving internal insulation.

2.3 Initial conditions

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

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. Only Cartesian meshes are used in this formulation. The increments in the x-, y-, and z-directions are denoted by Δx_i, Δy_j, and Δz_k, respectively. The smaller these increments are, the better is the agreement with the ”true” temperature distribution.

3.2 Thermal conductances

Figure 3.1 shows a cell (i,j,k) with the side lengths Δx_i, Δy_j, and Δz_k. There are six adjacent cells. The figure shows also the cell $(i,j,k+1)$ located directly above. The heat flow $Q_{i,j,k+1/2}$, (W), from cell (i,j,k) to cell $(i,j,k+1)$ is given by the thermal conductance multiplied by the temperature difference between these two cells:

$$Q_{i,j,k+1/2}=K_{i,j,k+1/2}(T_{i,j,k}- T_{i,j,k+1}) \qquad \text{(W)} \qquad (3.1)$$

where $K_{i,j,k+1/2}$, (W/K), is the conductance between the two cells (i,j,k) and $(i,j,k+1)$. The other five heat flows pertaining to cell (i,j,k) are calculated in the same way.

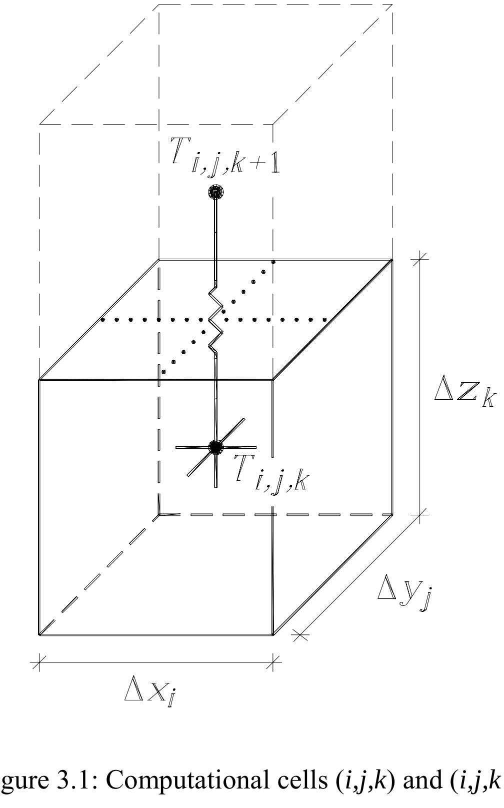


Figure 3.1: Computational cells (i,j,k) and $(i,j,k+1)$.

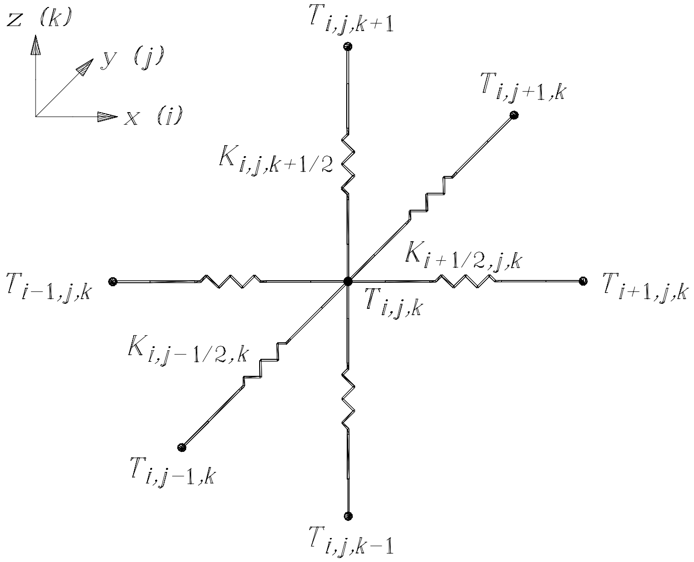


Figure 3.2: Thermal conductances connected to cell (*i,j,k*).

Figure 3.2 shows the six thermal conductances to cell (*i,j,k*). The conductance $K_{i,j,k+1/2}$, (W/K), between the two cells (*i,j,k*) and (*i,j,k+1*) is calculated as, (Claesson et al, 1994):

$$K_{i,j,k+1/2} = \frac{\Delta x_i \cdot \Delta y_j}{\Delta z_k /(2\cdot\lambda_{i,j,k}) + \Delta z_{k+1}/(2\cdot\lambda_{i,j,k+1}) + R_{i,j,k+1/2}} \qquad \text{(W/K)} \qquad (3.2)$$

Here $\lambda_{i,j,k}$, (W/(m·K)), is the thermal conductivity in cell (*i,j,k*). The conductance refers to the total heat flow through the area $\Delta x_i \cdot \Delta y_j$. The first term in the denominator is the thermal resistance in the z-direction for half of the cell (*i,j,k*), the second term being the resistance for half of the cell (*i,j,k+1*). The third term $R_{i,j,k+1/2}$, (m²·K/W), is an optional additional thermal resistance at the interface between the two cells (*i,j,k*) and (*i,j,k+1*).

Equation (3.2) is valid for all internal cells (an internal cell has at least one cell on each side). For boundary cells, the equation is modified in the following way. Consider cell (*1,j,k*) that lies at a boundary. The conductance which couples the temperature $T_{1,j,k}$ with a boundary temperature is

$$K_{1/2,j,k} = \frac{\Delta y_j \cdot \Delta z_k}{\Delta x_1 /(2\cdot\lambda_{1,j,k}) + R_{1/2,j,k}} \qquad \text{(W/K)} \qquad (3.3)$$

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

3.3 Heat flows

An energy balance is made for each cell. The total heat flow to cell (*i,j,k*) from the six adjacent cells is put in the variable $H_{i,j,k}$, (W),

$$\begin{aligned} H_{i,j,k} = {} & K_{i-1/2,j,k}\cdot(T_{i-1,j,k}-T_{i,j,k})+K_{i+1/2,j,k}\cdot(T_{i+1,j,k}-T_{i,j,k})+ \qquad \text{(W)} \qquad (3.4) \\ & K_{i,j-1/2,k}\cdot(T_{i,j-1,k}-T_{i,j,k})+K_{i,j+1/2,k}\cdot(T_{i,j+1,k}-T_{i,j,k})+ \\ & K_{i,j,k-1/2}\cdot(T_{i,j,k-1}-T_{i,j,k})+K_{i,j,k+1/2}\cdot(T_{i,j,k+1}-T_{i,j,k}) \end{aligned}$$

Changes in the energy of a cell due to heat sources/sinks I, (W/m³), are directly made by adding or subtracting the values to the variable $H_{i,j,k}$ as $I_{i,j,k}\cdot\Delta x_i\cdot\Delta y_j\cdot\Delta z_k$. This method simplifies the calculation procedures as well as further modifications that may be needed.

Another advantage of the introduction of $H_{i,j,k}$ is that less data need to be allocated in the computer memory. The arrays that are needed are normally eight:

$$T_{i,j,k} \quad C_{i,j,k} \quad K_{i+1/2,j,k}\ K_{i,j+1/2,k}\ K_{i,j,k+1/2} \quad Q_{i+1/2,j,k}\ \ Q_{i,j+1/2,k}\ \ Q_{i,j,k+1/2}$$

Introducing the $H_{i,j,k}$-variable gives instead six three-dimensional arrays:

$$T_{i,j,k} \quad C_{i,j,k} \quad K_{i+1/2,j,k}\ K_{i,j+1/2,k}\ K_{i,j,k+1/2} \quad H_{i,j,k}$$

Note that only three of the products (conductance multiplied by temperature difference in Eq. (3.4) have to be calculated at each time-step. This may save almost half the computer time compared with a direct use of Eq. (3.4). The following procedure which is used for all internal cells (i,j,k) illustrates this. Three local variables, Q_x, Q_y, and Q_z are introduced in order to decrease the number of arithmetic operations. The heat flows from the three cells "upstream" ($i+1,j,k$), ($i,j+1,k$), and ($i,j,k+1$) to cell (i,j,k) are put in the variables

$$Q_x= K_{i+1/2,j,k}\cdot(T_{i+1,j,k}-T_{i,j,k})$$

$$Q_y= K_{i,j+1/2,k}\cdot(T_{i,j+1,k}-T_{i,j,k})$$

$$Q_z= K_{i,j,k+1/2}\cdot(T_{i,j,k+1}-T_{i,j,k})$$

The change in heat for the cells "upstream" is directly made as

$$H_{new,\ i+1,j,k}=H_{i+1,j,k}-Q_x$$

$$H_{new,\ i,j+1,k}=H_{i,j+1,k}-Q_x$$

$$H_{new,\ i,j,k+1}=H_{i,j,k+1}-Q_x$$

At the same time-step, the energy is increased in cell (i,j,k) as

$$H_{new,\ i,j,k}=H_{i,j,k}+Q_x+Q_y+Q_z$$

3.4 New temperatures

The net heat flow during a time-step Δt results in an increase or a decrease in temperature. The temperature at the new time-step becomes

$$T_{i,j,k}^{new} = T_{i,j,k} + \frac{\Delta t}{C_{i,j,k}\Delta x_i \Delta y_j \Delta z_k}\cdot H_{i,j,k} \qquad (3.5)$$

3.5 Choice of time-step

The stable time-step Δ*t* for cell (*i*,*j*,*k*) is determined using the following stability criterion:

$$\Delta t < \frac{C_{i,j,k}\Delta x_i \Delta y_j \Delta z_k}{K_{i-1/2,j,k} + K_{i+1/2,j,k} + K_{i,j-1/2,k} + K_{i,j+1/2,k} + K_{i,j,k-1/2} + K_{i,j,k+1/2}} \tag{3.6}$$

This criterion must be satisfied for all cells (*i*,*j*,*k*). 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).

3.6 Iterative calculation

Equation (3.5) gives the new temperature based on the change in energy during the time-step Δ*t*. This calculation is made for all cells. The updated temperatures give the new heat flow to the cell according to Eq. (3.4), which in turn changes the temperatures again, and so on.

3.7 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 $T_{i,j,k}^{new}$ are calculated by Eq. (3.5).

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 □ may give calculation times between 1/70th 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 HEAT3 (and in HEAT2 and HEAT2R) this factor is initially set to 1.95. Equation (3.5) is modified to

$$T_{i,j,k}^{new} = T_{i,j,k} + \frac{\Delta t \cdot \omega}{C_{i,j,k}\Delta x_i \Delta y_j \Delta z_k} \cdot H_{i,j,k} \tag{3.7}$$

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 $C_{i,j,k}$ are chosen to give the same time-step for all cells. The heat capacities are determined by putting Δ*t* equal to the right-hand side in (3.6). The chosen capacities are then

$$C_{i,j,k} = \Delta t \cdot \frac{K_{i-1/2,j,k} + K_{i+1/2,j,k} + K_{i,j-1/2,k} + K_{i,j+1/2,k} + K_{i,j,k-1/2} + K_{i,j,k+1/2}}{\Delta x_i \Delta y_j \Delta z_k} \tag{3.9}$$

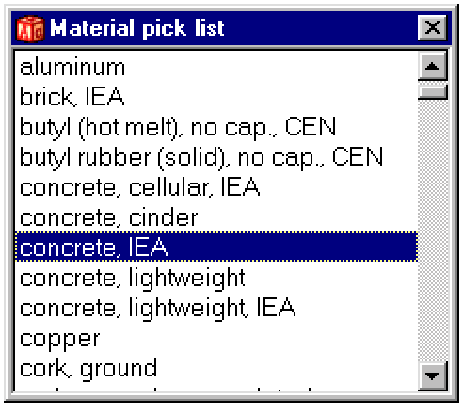
Actually, the choice of Δ*t* does not matter since it cancels in Eq. (3.7).

4. Overview of input using the pre-processor

4.1 Pre-processor

The geometry is built by overlapping boxes that are drawn in the (*x*,*y*)- and (*x*,*z*)-planes. The pre-processor shows continuously the result in a three-dimensional view. It is quite easy after some practice to give the full three-dimensional geometry.

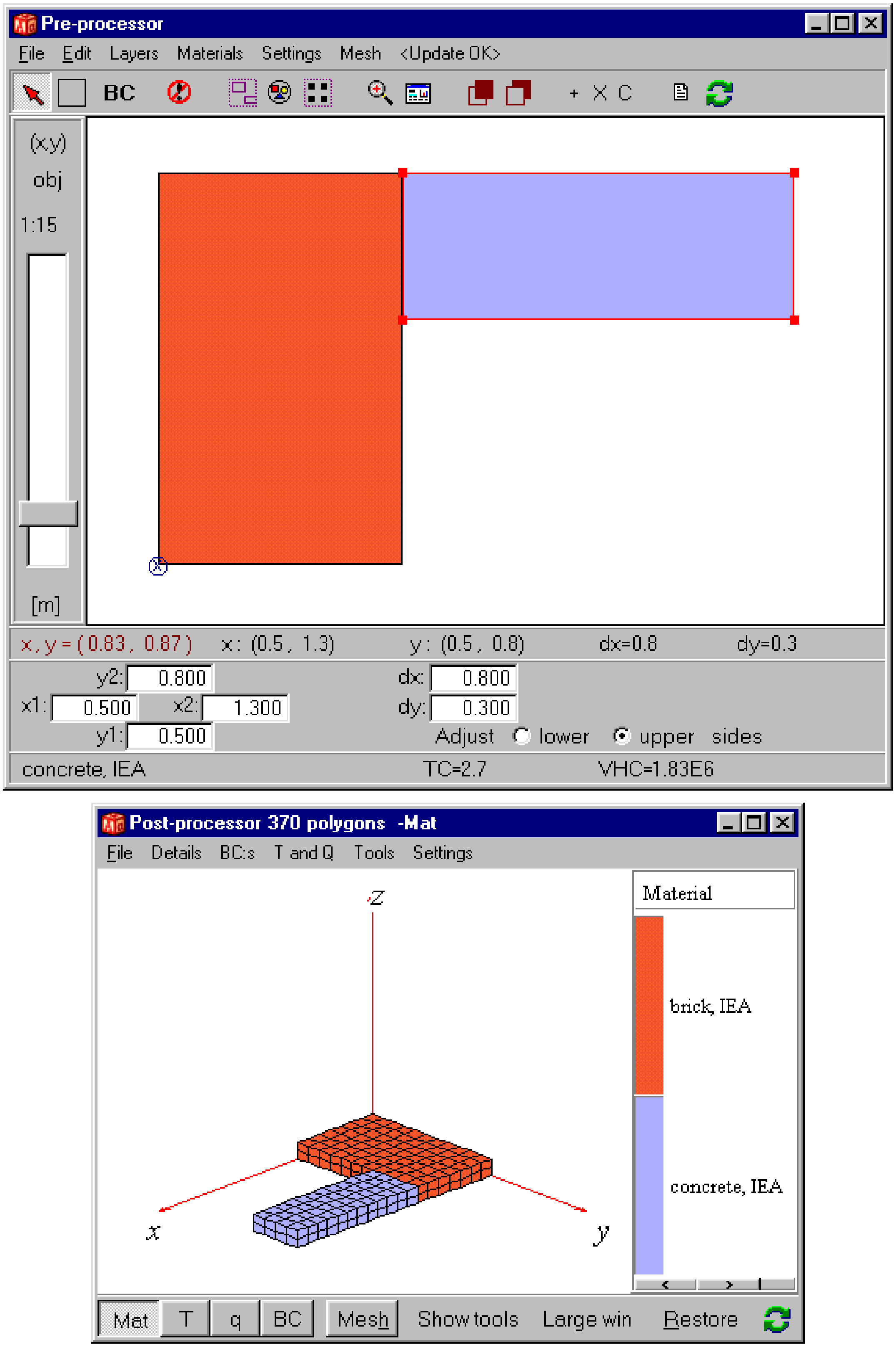
Materials are chosen from the material pick list, see Figure 4.1.



aluminum
brick, IEA
butyl (hot melt), no cap., CEN
butyl rubber (solid), no cap., CEN
concrete, cellular, IEA
concrete, cinder
concrete, IEA
concrete, lightweight
concrete, lightweight, IEA
copper
cork, ground

Figure 4.1: Material pick list.

Figure 4.2 shows two boxes drawn in the pre-processor in the (*x*,*y*)-plane. 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. The geometry is automatically updated and shown in the post-processor.


Pre-processor
File Edit Layers Materials Settings Mesh <Update OK>
BC
+ X C
(x,y)
obj
1:15
[m]

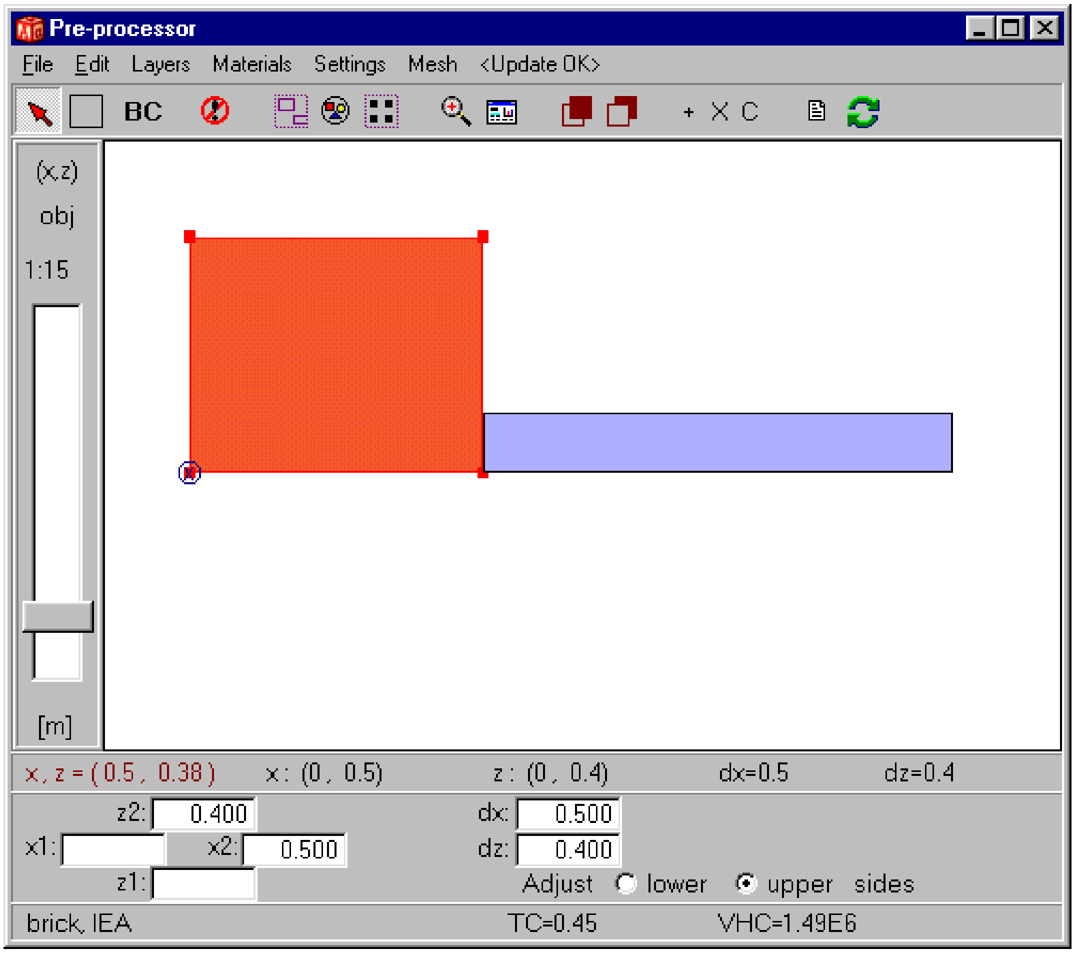
x, y = (0.83 , 0.87)   x : (0.5 , 1.3)   y : (0.5 , 0.8)   dx=0.8   dy=0.3

y2: 0.800
x1: 0.500   x2: 1.300
y1: 0.500
dx: 0.800
dy: 0.300
Adjust ○ lower ● upper sides

concrete, IEA   TC=2.7   VHC=1.83E6

Post-processor 370 polygons -Mat
File Details BC:s T and Q Tools Settings
Z
x
y
Material
🟥 brick, IEA
🟦 concrete, IEA
Mat T q BC Mesh Show tools Large win Restore

Figure 4.2: Two boxes of different materials drawn in the pre-processor in the (*x*,*y*)-plane. The post-processor (bottom figure) shows the generated numerical mesh with material names.



x, z = (0.5 , 0.38) x : (0 , 0.5) z : (0 , 0.4) dx=0.5 dz=0.4

z2: 0.400
x1: x2: 0.500
z1:
dx: 0.500
dz: 0.400
Adjust ○ lower ● upper sides

brick, IEA TC=0.45 VHC=1.49E6

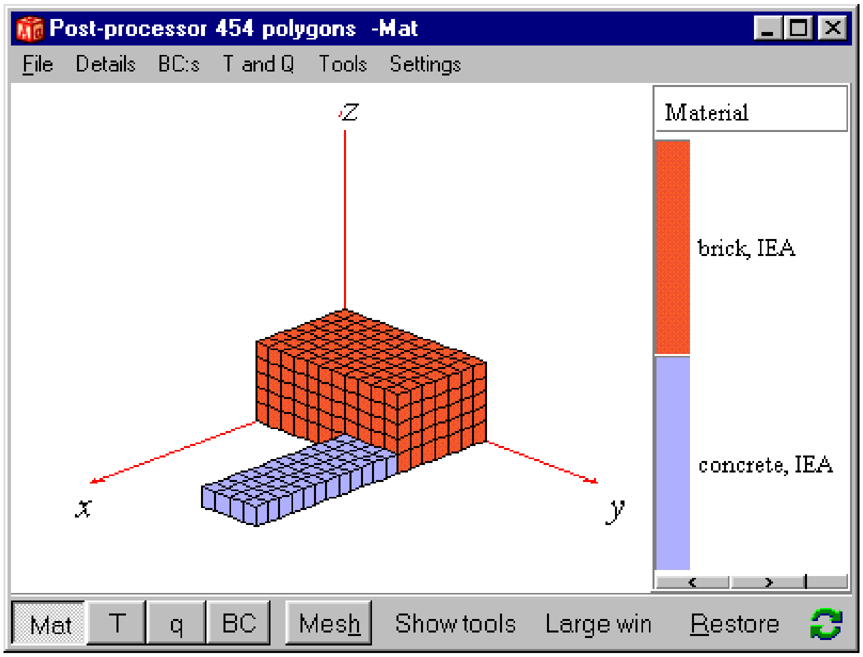
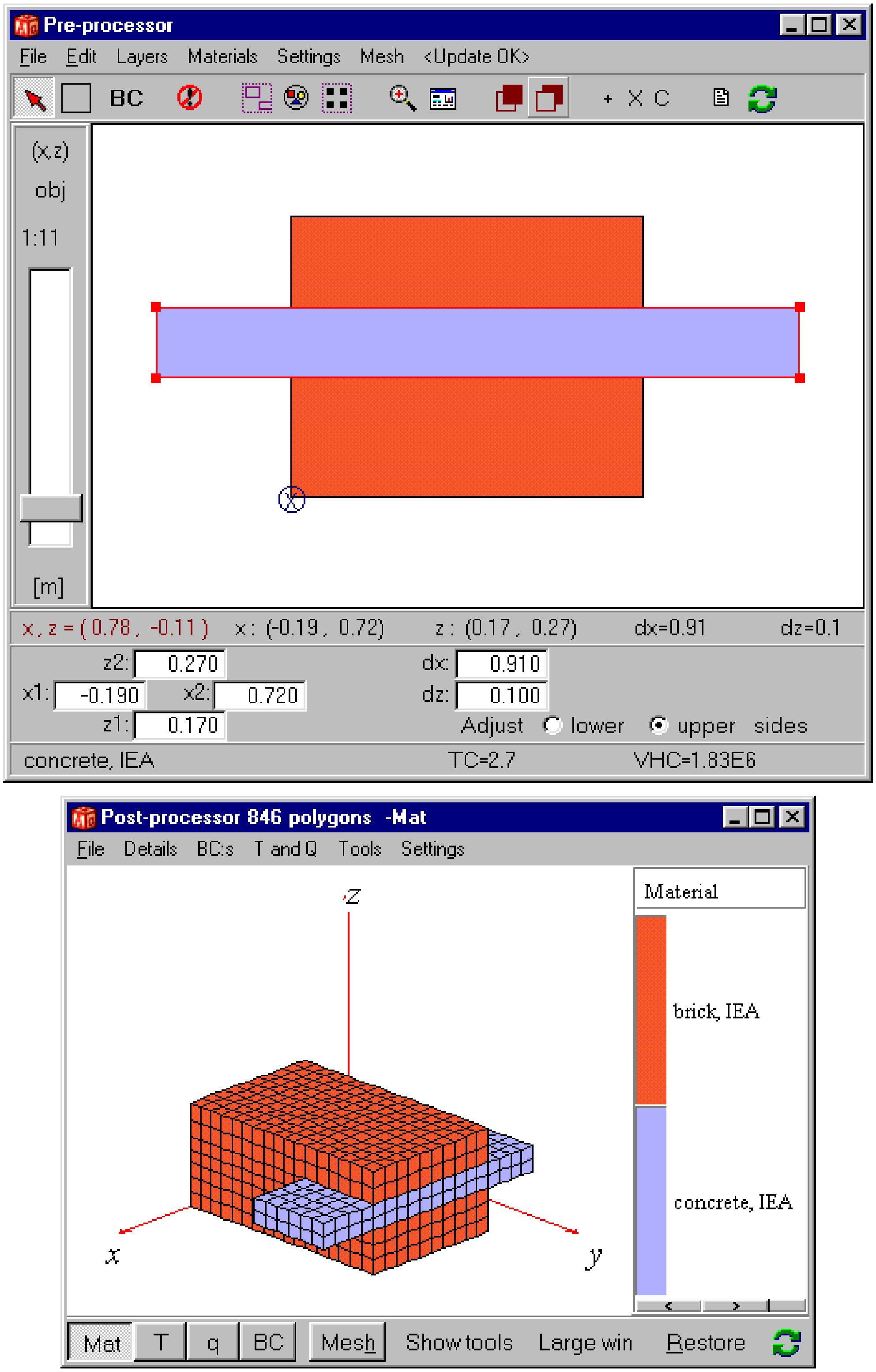

Post-processor 454 polygons -Mat
File Details BC:s T and Q Tools Settings
z
x
y
Material
brick, IEA
concrete, IEA
Mat T q BC Mesh Show tools Large win Restore

Figure 4.3: The view in the pre-processor is here changed to (*x*,*z*) and the red box (brick) is resized in the *z*-direction. The 3D-view is shown in the pre-processor (bottom).


Pre-processor
File Edit Layers Materials Settings Mesh <Update OK>
BC
+ X C
(x,z)
obj
1:11
[m]

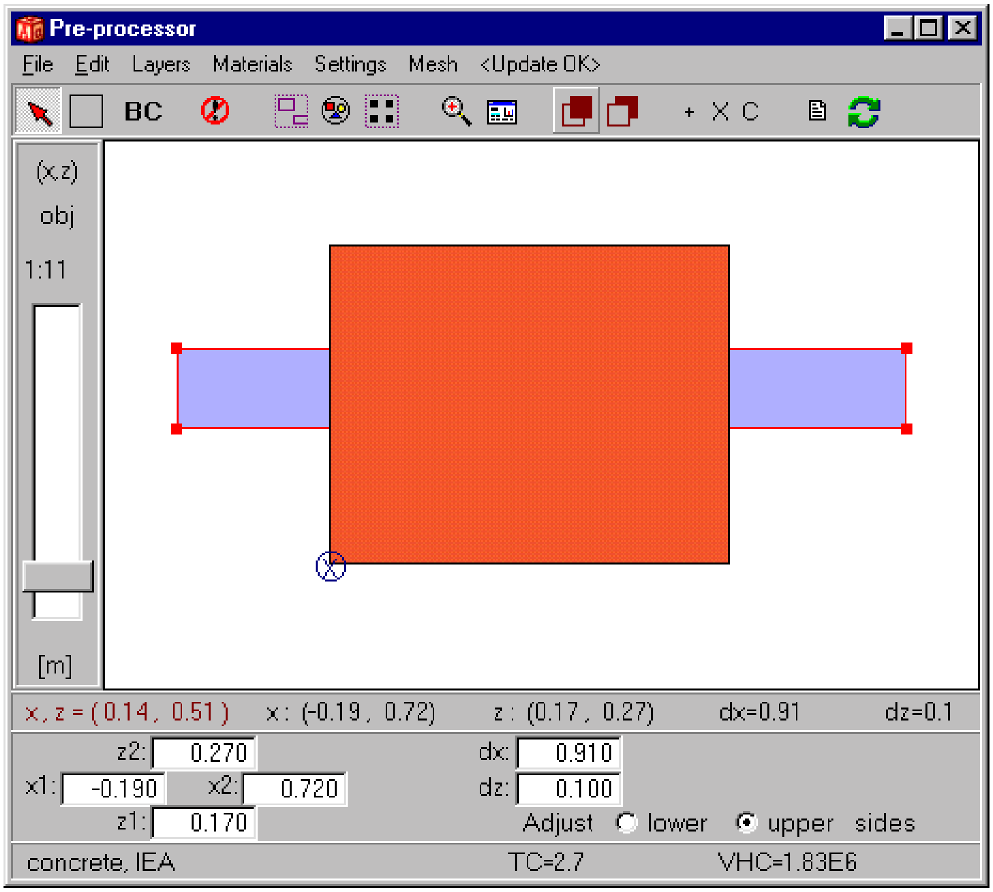
x, z = (0.78 , -0.11)   x : (-0.19 , 0.72)   z : (0.17 , 0.27)   dx=0.91   dz=0.1

z2: 0.270
x1: -0.190   x2: 0.720
z1: 0.170
dx: 0.910
dz: 0.100
Adjust ○ lower ● upper sides

concrete, IEA   TC=2.7   VHC=1.83E6

Post-processor 846 polygons -Mat
File Details BC:s T and Q Tools Settings
Z
x
y
Material
brick, IEA
concrete, IEA
Mat T q BC Mesh Show tools Large win Restore

Figure 4.4: The blue box overlaps the red one. This means that the blue box will cut through the red one as shown in the 3D-view (bottom).



```
x, z = ( 0.14 , 0.51 )   x : (-0.19 , 0.72)   z : (0.17 , 0.27)   dx=0.91   dz=0.1
```

z2:	0.270
x1:	-0.190
x2:	0.720
z1:	0.170
dx:	0.910
dz:	0.100

Adjust ○ lower ● upper sides

concrete, IEA   TC=2.7   VHC=1.83E6

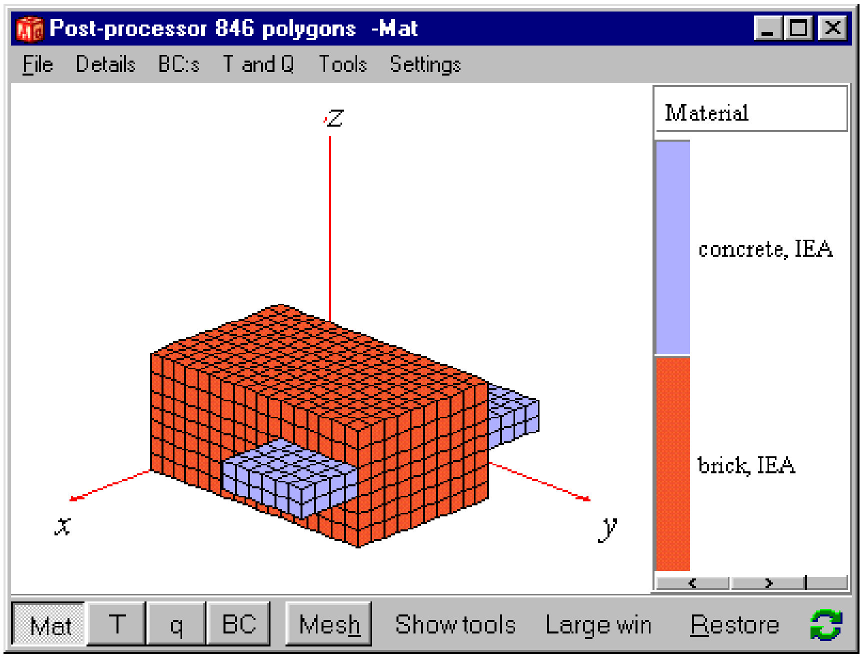
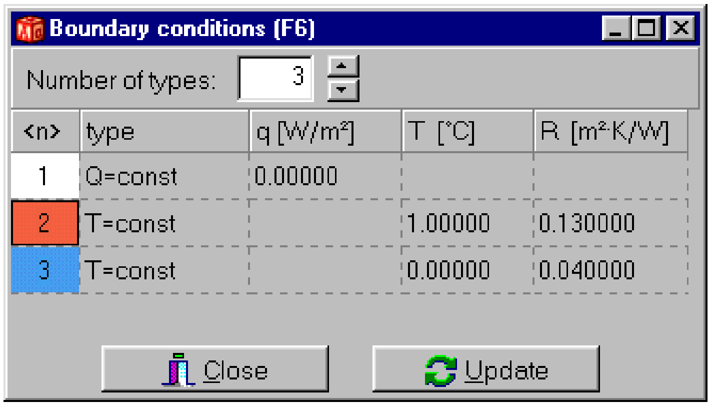


Figure 4.5: The red box overlaps the blue one. This means that the red box will cut through the blue one as shown in the 3D-view (bottom).

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*, ($m^2 \cdot K/W$), between the air and the surface, or a given heat flow *q*, (W/m^2). The temperatures and the heat flows may be constant in time, or time-dependent using a sinusoidal, a step-wise constant, or a step-wise linear function. See definitions in Section 5.6.

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 4.19.



Number of types: 3

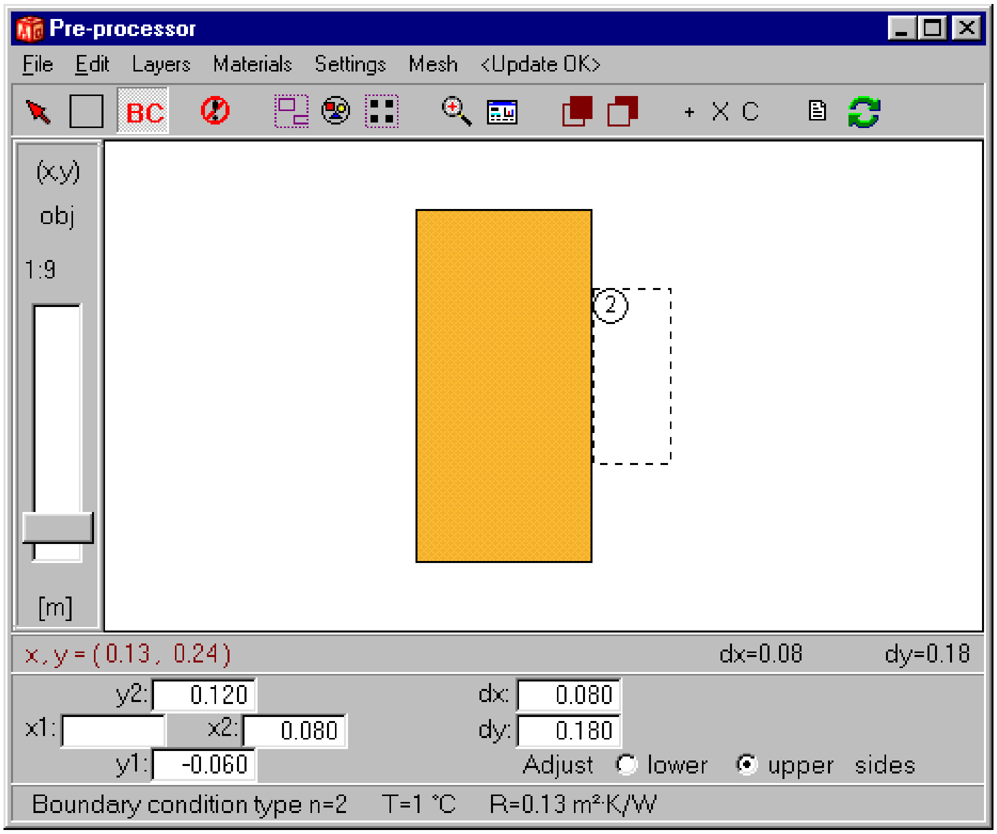
<n>	type	q [W/m²]	T [°C]	R [m²K/W]
⬜ 1	Q=const	0.00000		
🟥 2	T=const		1.00000	0.130000
🟦 3	T=const		0.00000	0.040000

Figure 4.6: The three default defined boundary condition types.

Boundary conditions are given by using boxes (from now on referred to as BC boxes). A BC box is drawn by picking the menu item BC in the top tool row in the pre-processor and choosing the type in the BC window (as shown in Figure 4.6) by clicking on the type color under column “<n>”. Each of the six sides of the box that touches a material boundary surface will apply its boundary type to the particular surface.

Figure 4.7 shows one BC box. Boundary condition type 2 will be applied to the boundary surface that touches the BC box (outlined rectangle). Any interface between a material surface and a BC box surface will be shown as an outlined rectangle in the post-processor, see Figure 4.8. If no outline can be seen in the post-processor then the BC box does not fit to any material surface and the boundary condition is not accounted for. The boundary condition type may also be shown in the post-processor by pressing the menu tool item “BC”, see Figure 4.9. A few examples of boundary conditions are shown in Figure 4.10-Figure 4.26.

Note that the first boundary condition (here adiabatic i.e. q=0) will apply to all surfaces not defined by any BC boxes. The first type may of course be changed from the default *q*=0.



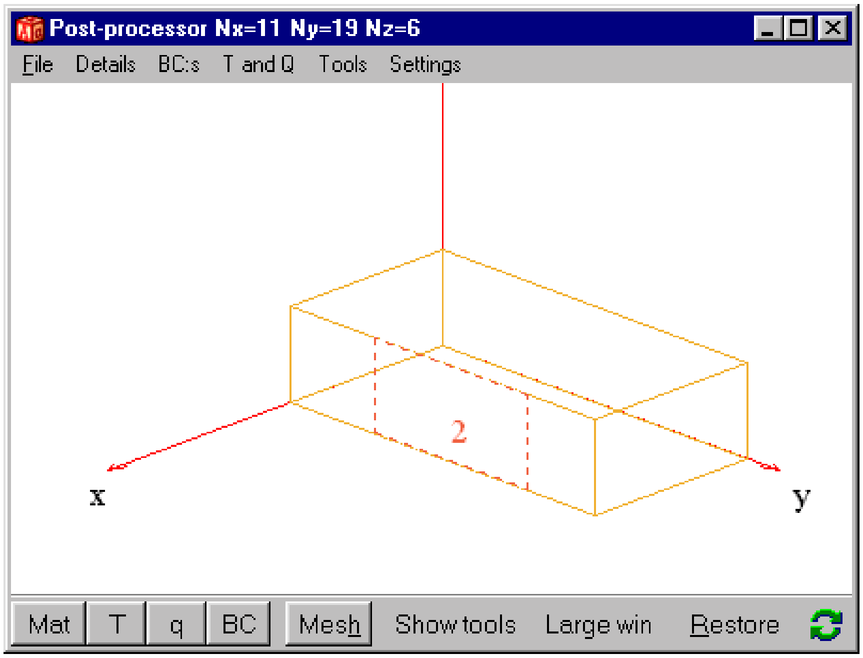
x , y = (0.13 , 0.24) dx=0.08 dy=0.18

y2:	0.120	dx:	0.080
x1:			
x2:	0.080	dy:	0.180
y1:	-0.060		

Adjust ○ lower ● upper sides

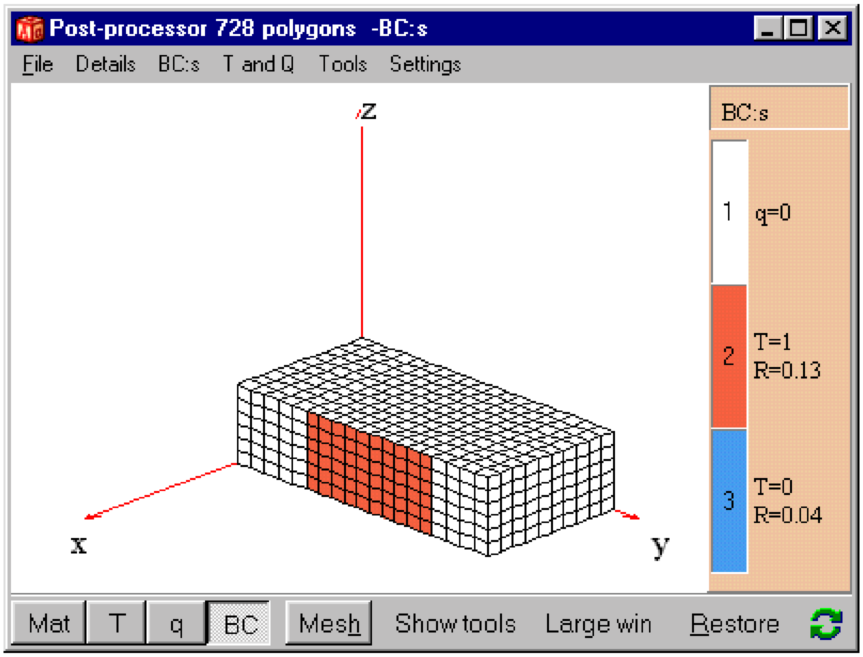
Boundary condition type n=2 T=1 °C R=0.13 m²·K/W

Figure 4.7: Boundary condition type 2 will be applied to the boundary surface that touches the BC box (outlined rectangle).


Post-processor Nx=11 Ny=19 Nz=6

File Details BC:s T and Q Tools Settings
2
x
y
Mat T q BC Mesh Show tools Large win Restore

Figure 4.8: Any valid interface between a material surface and BC box surface will be shown as an outlined rectangle in the post-processor.

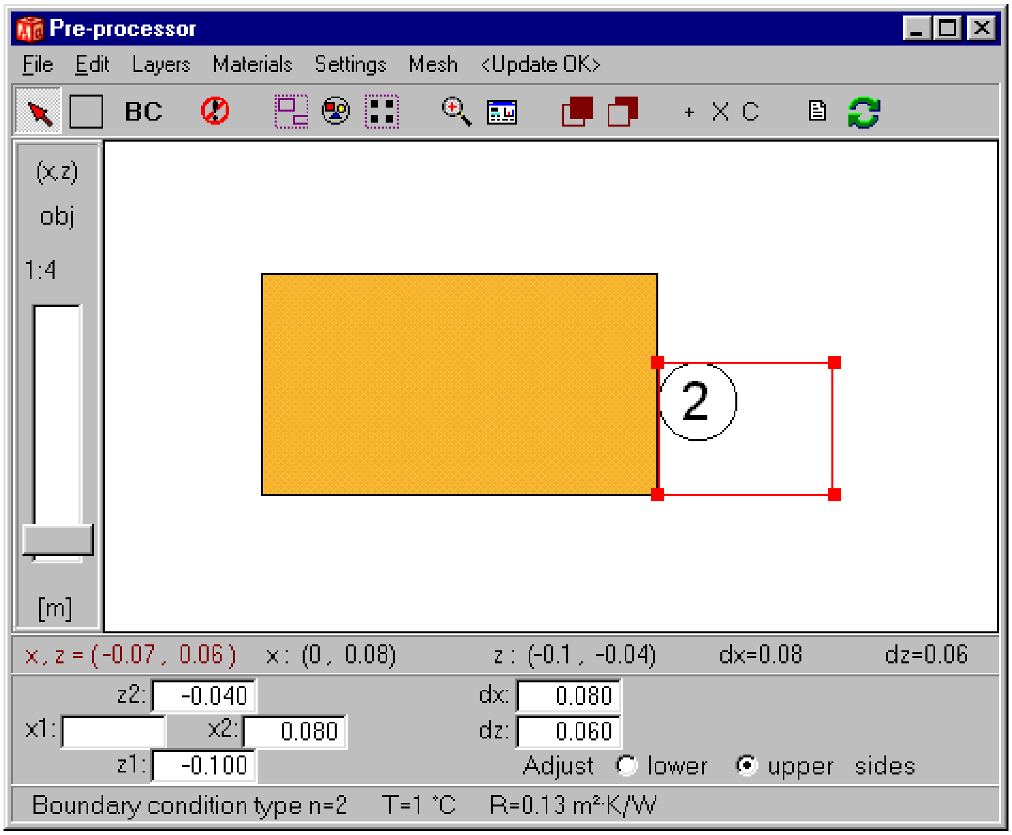

Post-processor 728 polygons -BC:s
File Details BC:s T and Q Tools Settings
z
x
y
BC:s
Mat T q BC Mesh Show tools Large win Restore

BC:s	
1	q=0

2	T=1
	R=0.13

3	T=0
	R=0.04

Figure 4.9: The boundary condition type may also be shown in the post-processor by pressing the menu tool item "BC". In this case the numerical grid is also shown.



x, z = (-0.07 , 0.06) x : (0 , 0.08) z : (-0.1 , -0.04) dx=0.08 dz=0.06

z2: -0.040
x1: x2: 0.080
z1: -0.100
dx: 0.080
dz: 0.060
Adjust ○ lower ● upper sides

Boundary condition type n=2 T=1 °C R=0.13 m²K/W

Figure 4.10: The BC box is here resized in the z-direction.

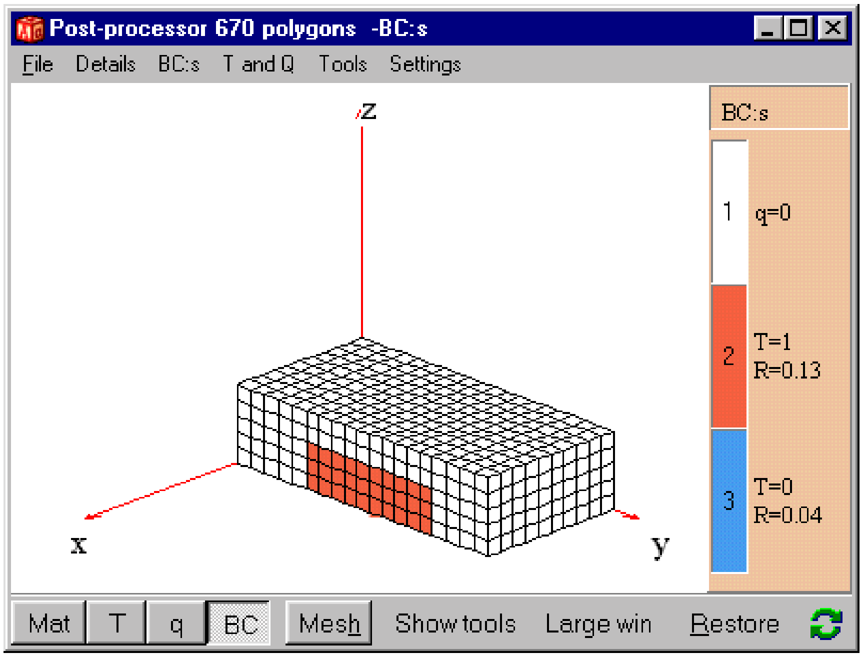
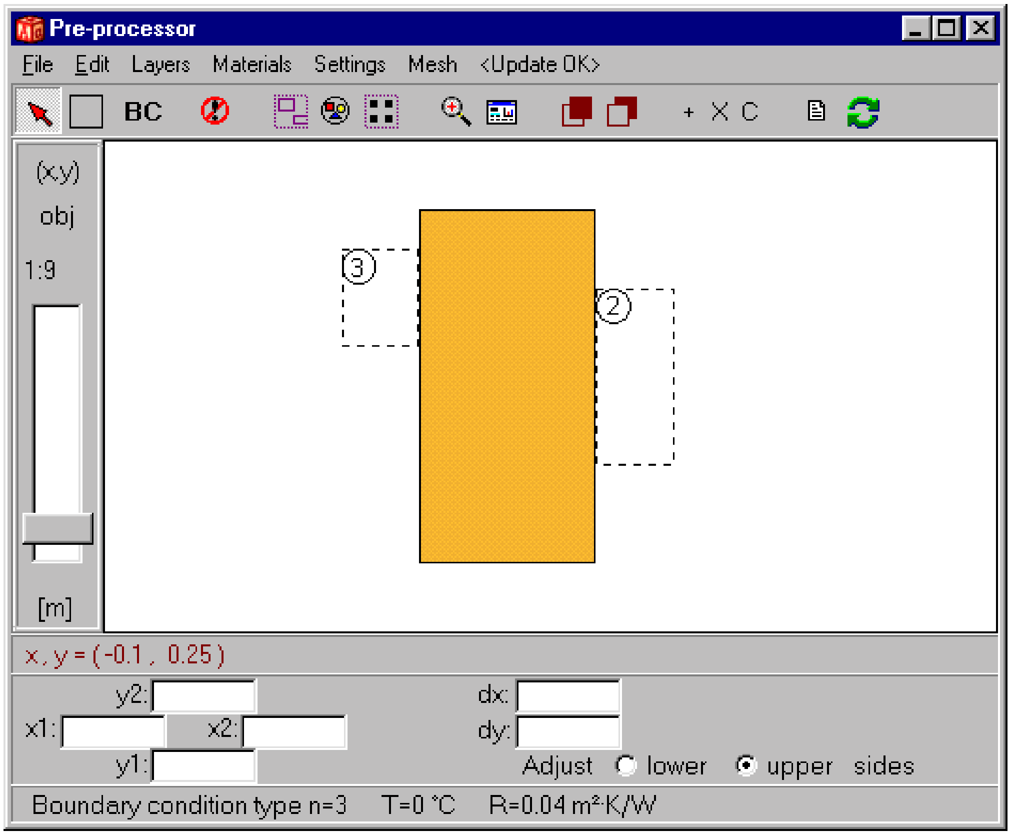


Figure 4.11: The boundary condition types shown in the post-processor.



x, y = (-0.1 , 0.25)

Adjust ○ lower ● upper sides

Boundary condition type n=3 T=0 °C R=0.04 m²K/W

Figure 4.12: Two BC boxes drawn.

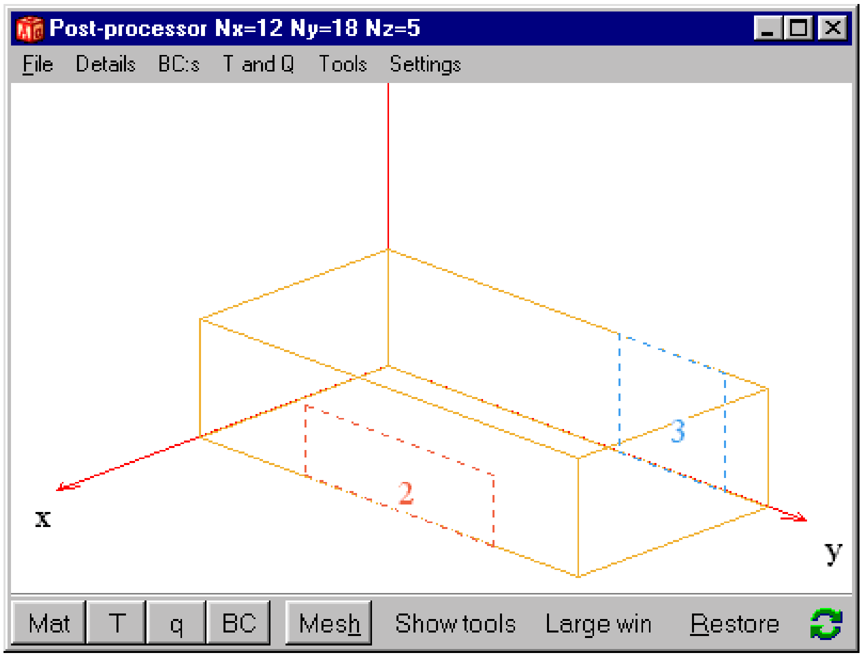
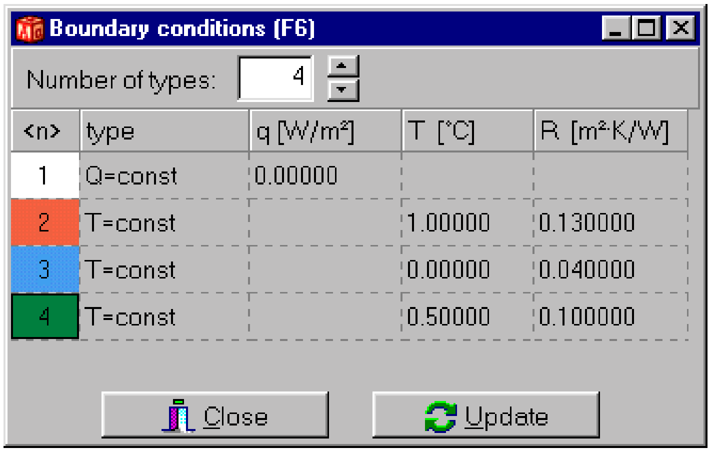

Post-processor Nx=12 Ny=18 Nz=5
File Details BC:s T and Q Tools Settings
x
y
2
3
Mat T q BC Mesh Show tools Large win Restore

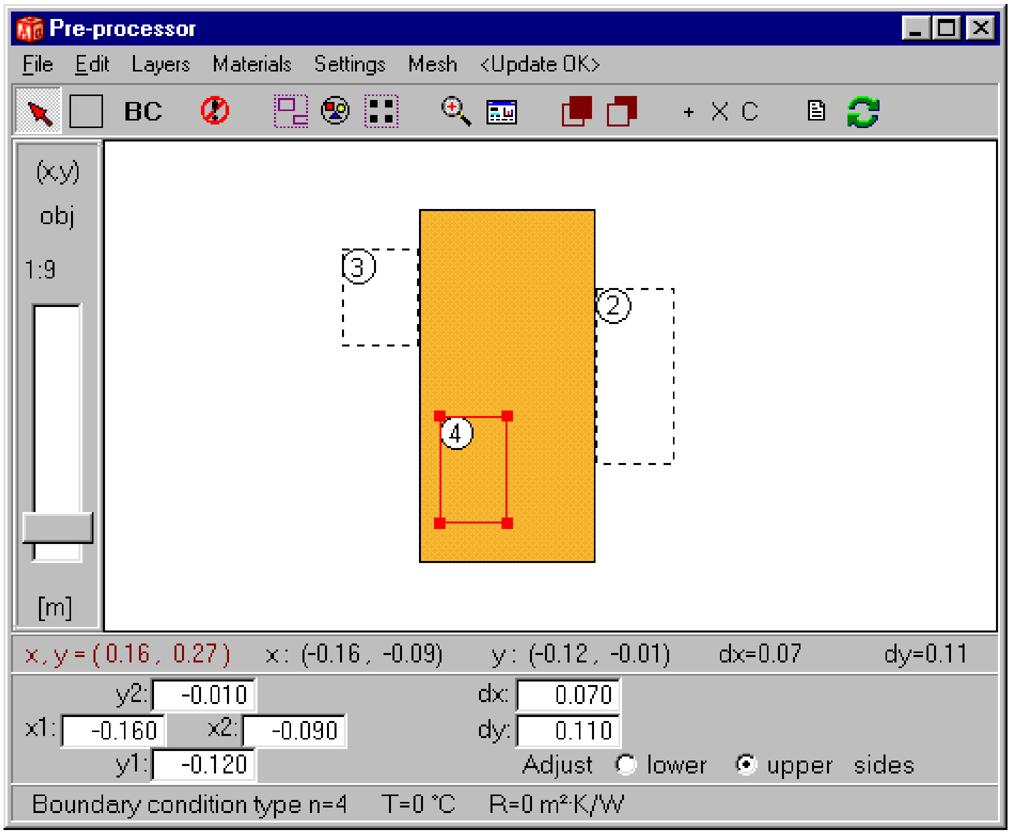
Figure 4.13: Two boundary condition areas as an outlined rectangle in the post-processor.



Number of types: 4

<n>	type	q [W/m²]	T [°C]	R [m²K/W]
1	Q=const	0.00000		
2	T=const		1.00000	0.130000
3	T=const		0.00000	0.040000
4	T=const		0.50000	0.100000

Figure 4.14: A fourth boundary condition defined.



x, y = (0.16, 0.27) x: (-0.16, -0.09) y: (-0.12, -0.01) dx=0.07 dy=0.11

y2: -0.010
x1: -0.160 x2: -0.090
y1: -0.120
dx: 0.070
dy: 0.110
Adjust ○ lower ● upper sides

Boundary condition type n=4 T=0 °C R=0 m²·K/W

Figure 4.15: Three BC boxes drawn.

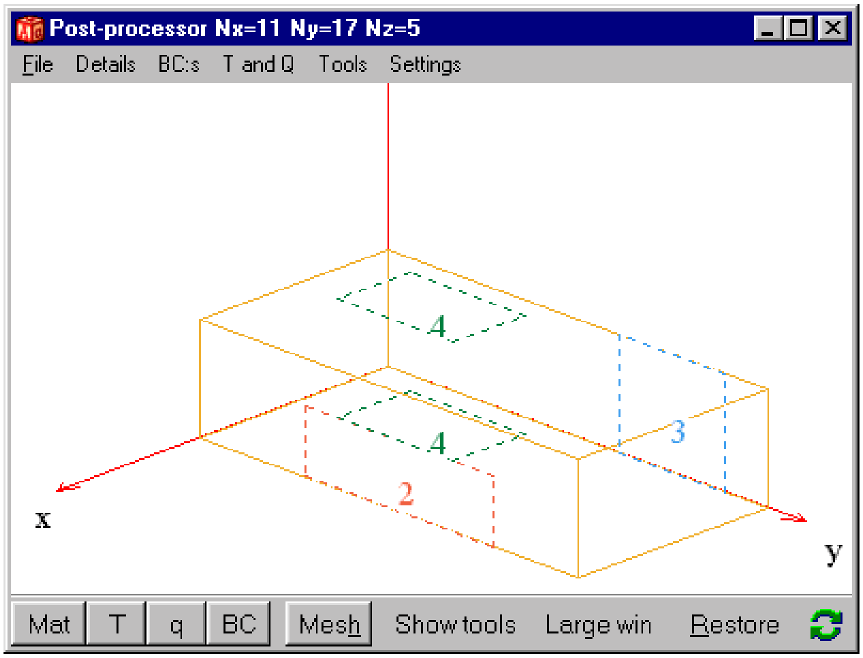

Post-processor Nx=11 Ny=17 Nz=5
File Details BC:s T and Q Tools Settings
4
4
3
2
x
y
Mat T q BC Mesh Show tools Large win Restore

Figure 4.16: Notice that the BC box of type 4 here touches two separate surfaces.

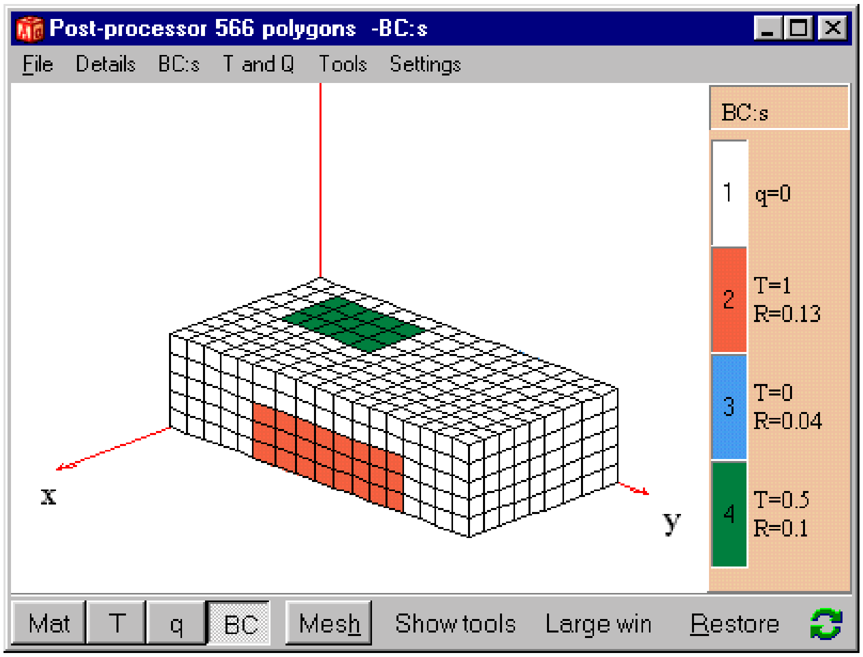

Post-processor 566 polygons -BC:s
File Details BC:s T and Q Tools Settings
BC:s
1 q=0
2 T=1 R=0.13
3 T=0 R=0.04
4 T=0.5 R=0.1
x
y
Mat T q BC Mesh Show tools Large win Restore

Figure 4.17: The boundary condition type shown in the post-processor.

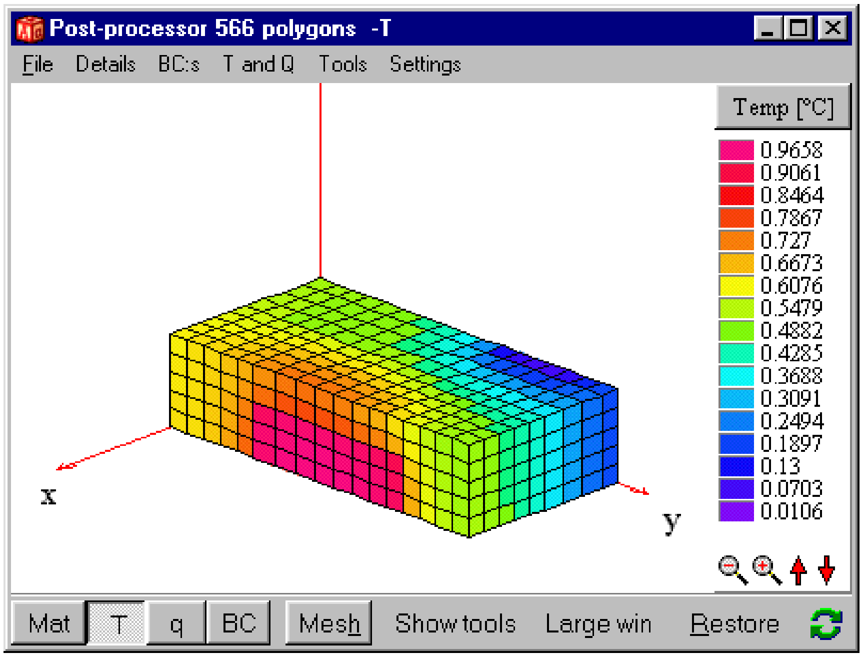

Post-processor 566 polygons -T
File Details BC:s T and Q Tools Settings
Temp [°C]
0.9658
0.9061
0.8464
0.7867
0.727
0.6673
0.6076
0.5479
0.4882
0.4285
0.3688
0.3091
0.2494
0.1897
0.13
0.0703
0.0106
x
y
Mat T q BC Mesh Show tools Large win Restore

Figure 4.18: Calculated temperatures for the current boundary conditions.

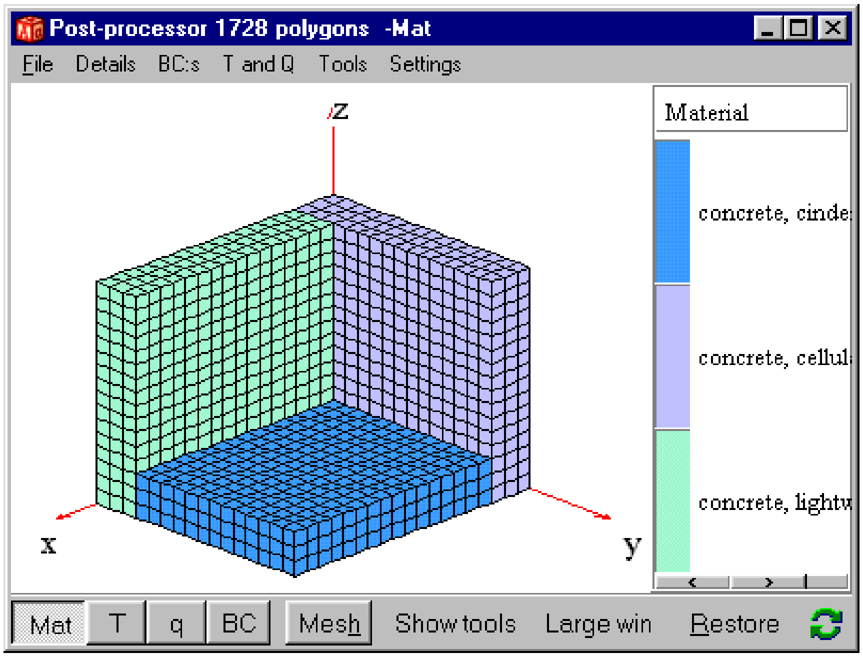
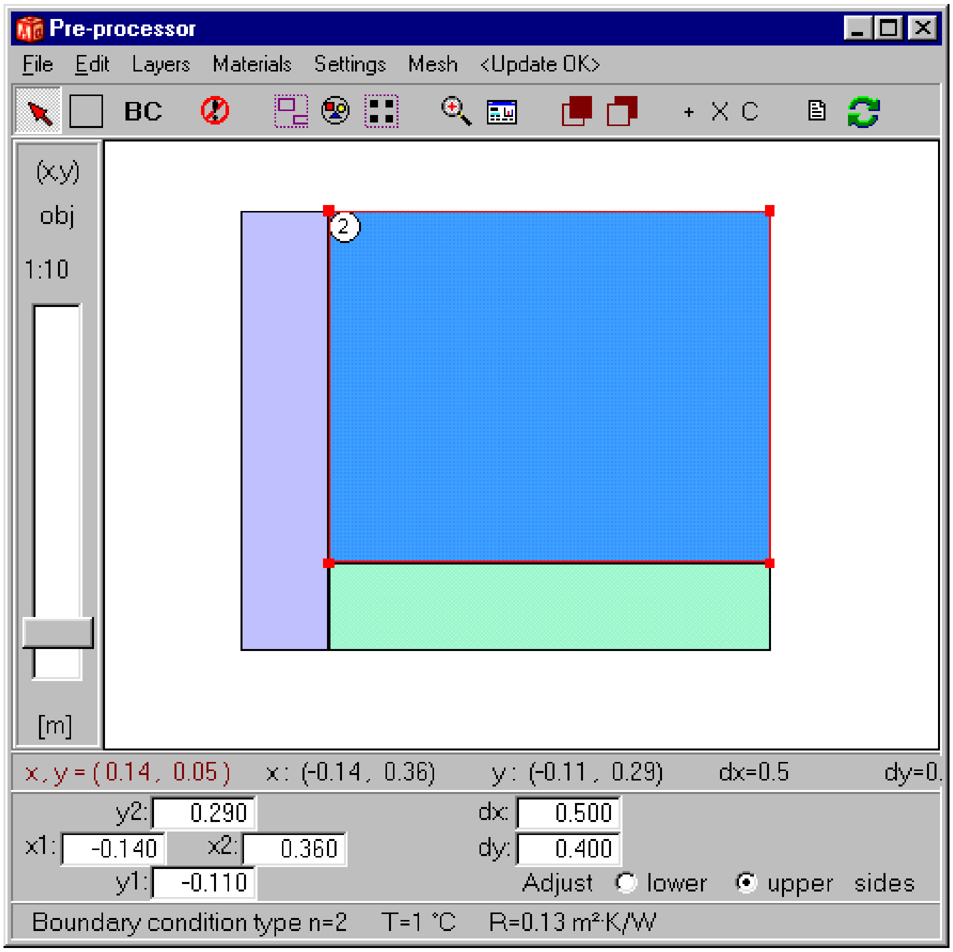


Figure 4.19: A corner built by three material boxes.



x, y = (0.14 , 0.05)   x : (-0.14 , 0.36)   y : (-0.11 , 0.29)   dx=0.5   dy=0.

y2: 0.290
x1: -0.140   x2: 0.360
y1: -0.110
dx: 0.500
dy: 0.400
Adjust ○ lower ● upper sides

Boundary condition type n=2   T=1 °C   R=0.13 m²·K/W

Figure 4.20: A BC box defining the internal boundary condition. The box is fitted in each direction for both planes (*x*,*y*) and (*x*,*z*).

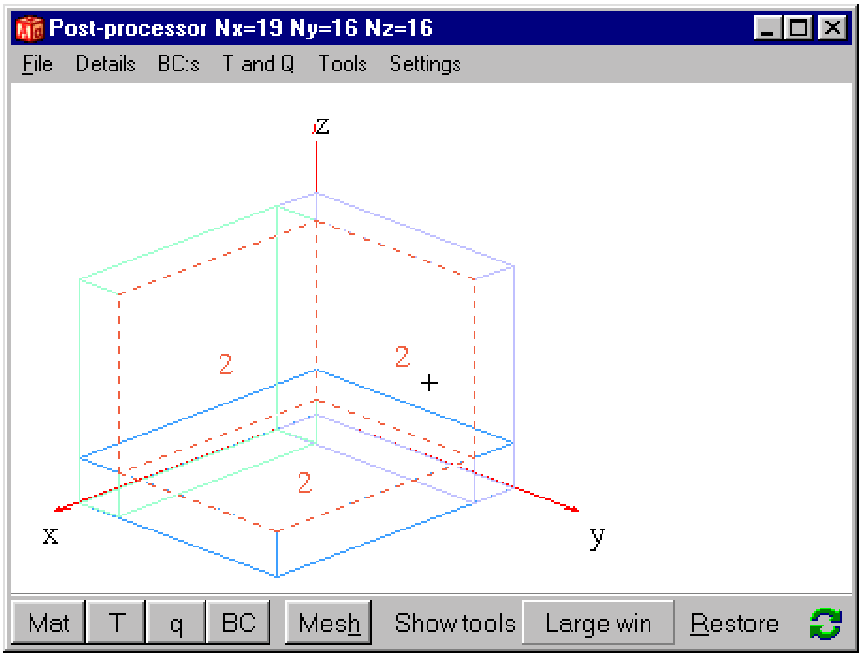


Figure 4.21: It is sufficient to draw one BC box since three of the six sides will interface with the three internal wall surfaces.

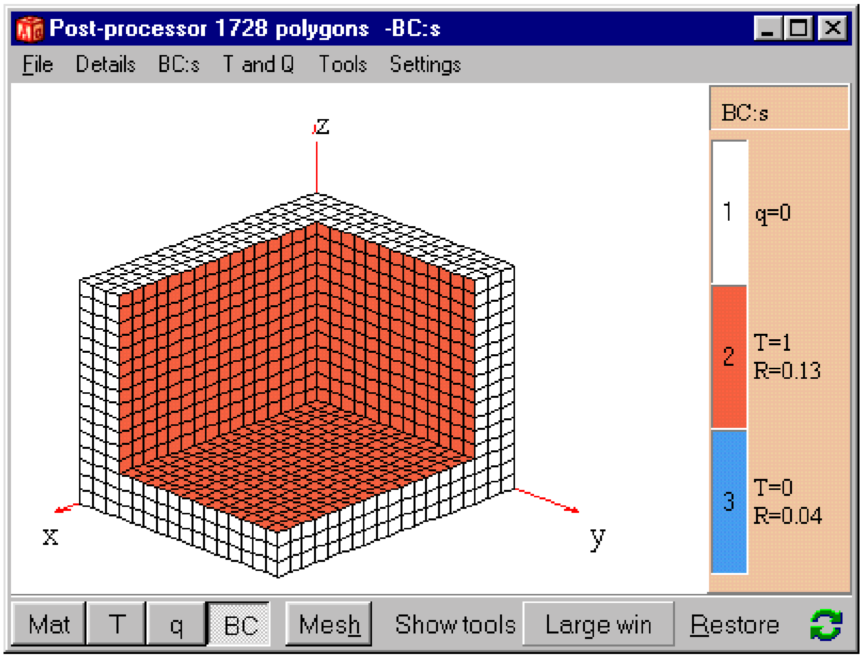
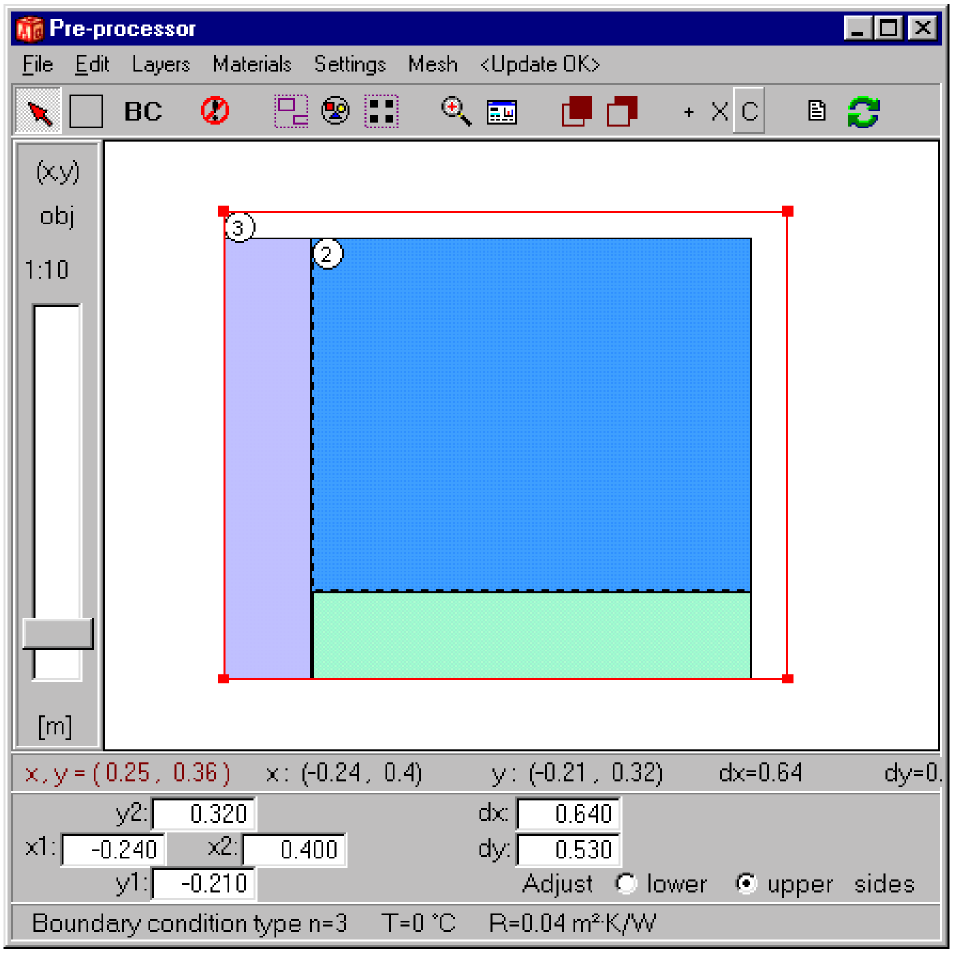


Figure 4.22: The boundary condition type shown in the post-processor.

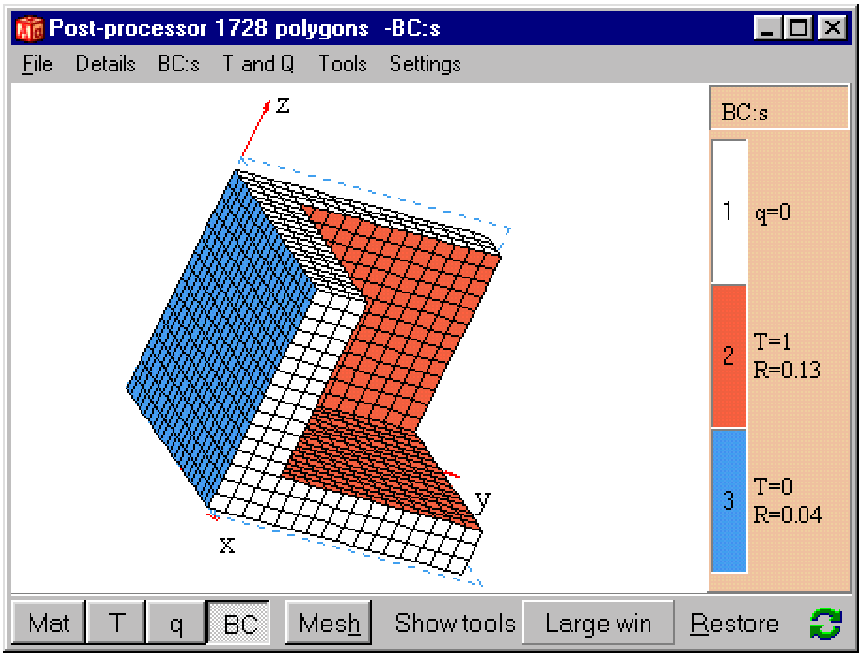


x, y = (0.25 , 0.36)   x : (-0.24 , 0.4)   y : (-0.21 , 0.32)   dx=0.64   dy=0.

y2: 0.320
x1: -0.240   x2: 0.400
y1: -0.210
dx: 0.640
dy: 0.530
Adjust ○ lower ● upper sides

Boundary condition type n=3   T=0 °C   R=0.04 m²·K/W

Figure 4.23: A BC box defining the external boundary condition. The box is fitted in each direction for both planes (*x*,*y*) and (*x*,*z*) so that three sides touches the external wall.



Post-processor 1728 polygons -BC:s
File Details BC:s T and Q Tools Settings
z
y
x
BC:s
Mat T q BC Mesh Show tools Large win Restore

BC:s	
1	q=0

2	T=1
	R=0.13

3	T=0
	R=0.04

Figure 4.24: The boundary condition type shown in the post-processor. Note that the first boundary condition (here adiabatic i.e. q=0) will apply to all surfaces not defined by any BC box.

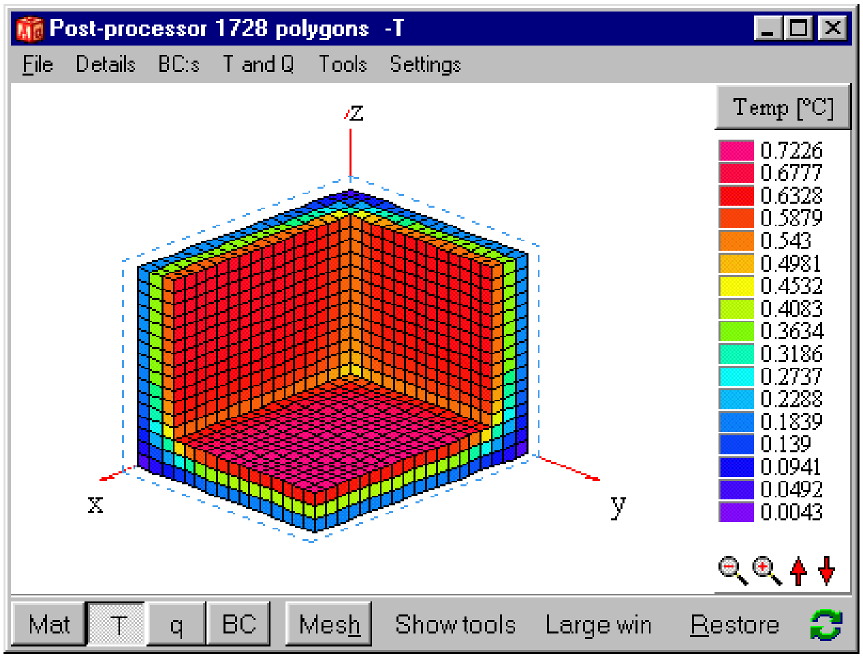


Figure 4.25: Calculated temperatures for the current boundary conditions.

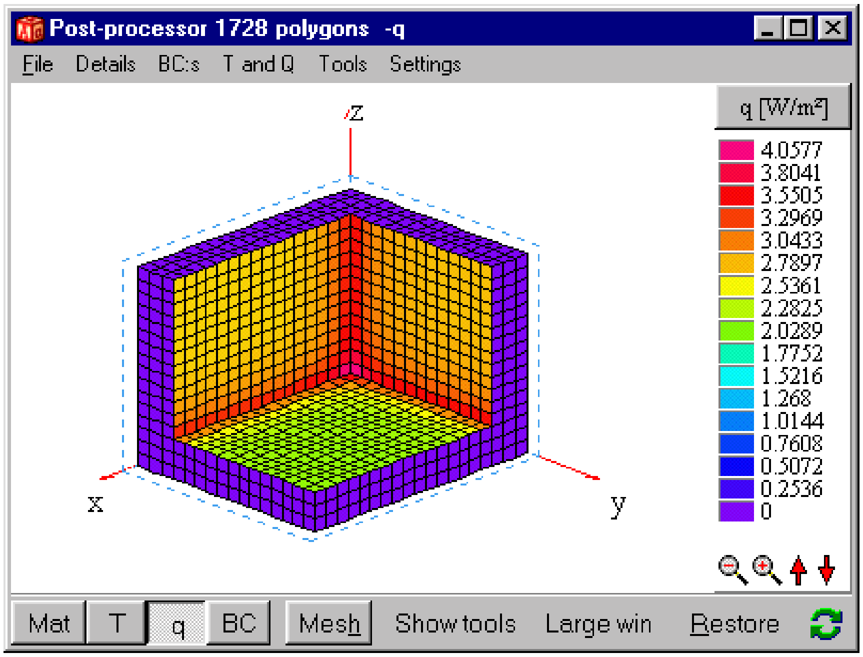
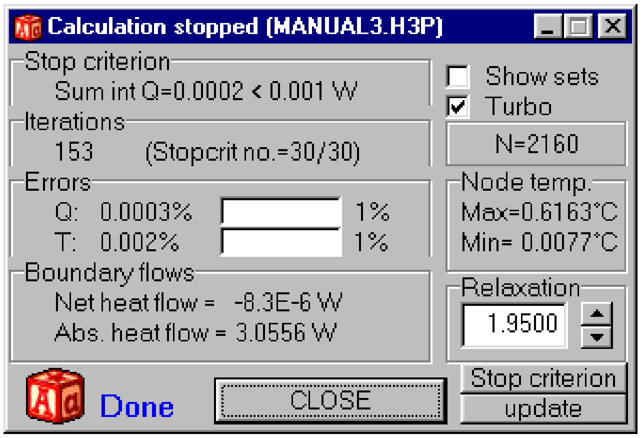


Figure 4.26: Calculated heat flows for the current boundary conditions.

4.3 Solving a steady-state problem

To start a steady-state simulation select *Solve/Start steady-state calculation*. A window, see Figure 4.27, will be displayed showing information such as the chosen stop criterion, number of iterations, and errors for temperature and flow, respectively. A list is shown in Figure 4.28 (*Output/Boundary flows*).



Calculation stopped (MANUAL3.H3P)

Stop criterion
Sum int Q=0.0002 < 0.001 W

Iterations
153 (Stopcrit no.=30/30)

Errors
Q: 0.0003% 1%
T: 0.002% 1%

Boundary flows
Net heat flow = -8.3E-6 W
Abs. heat flow = 3.0556 W

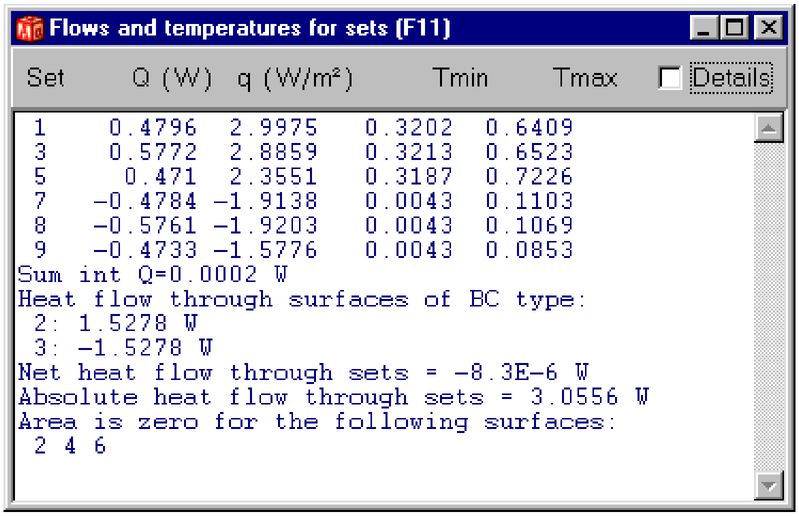
□ Show sets
☑ Turbo

N=2160

Node temp.
Max=0.6163°C
Min= 0.0077°C

Relaxation
1.9500

Figure 4.27: Window shown during steady-state calculation.



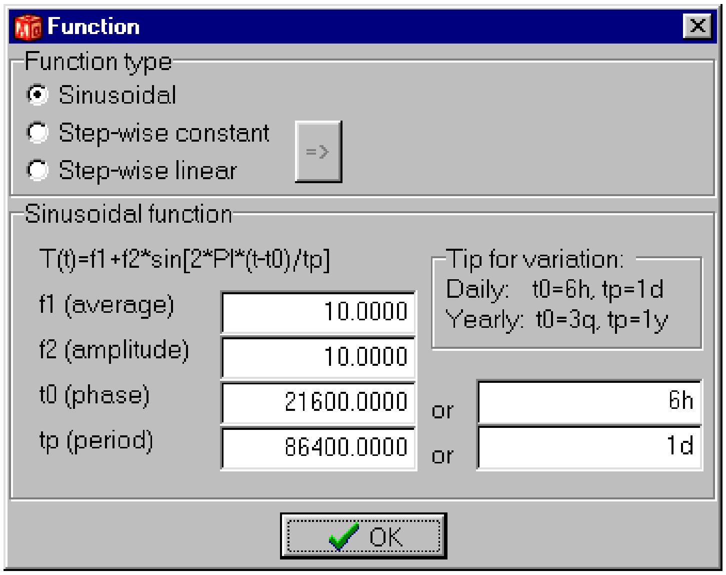
Flows and temperatures for sets (F11)

```
Set      Q (W)  q (W/m²)     Tmin      Tmax     □ Details
 1     0.4796  2.9975   0.3202  0.6409
 3     0.5772  2.8859   0.3213  0.6523
 5      0.471  2.3551   0.3187  0.7226
 7    -0.4784 -1.9138   0.0043  0.1103
 8    -0.5761 -1.9203   0.0043  0.1069
 9    -0.4733 -1.5776   0.0043  0.0853
Sum int Q=0.0002 W
Heat flow through surfaces of BC type:
 2: 1.5278 W
 3: -1.5278 W
Net heat flow through sets = -8.3E-6 W
Absolute heat flow through sets = 3.0556 W
Area is zero for the following surfaces:
 2 4 6
```

Figure 4.28: Heat flows through the boundaries for the corner problem.

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 4.29.



Function type

● Sinusoidal

○ Step-wise constant

○ Step-wise linear

Sinusoidal function

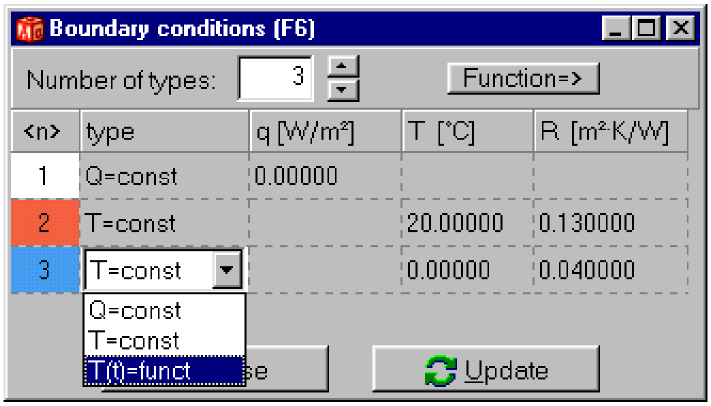
T(t)=f1+f2*sin[2*PI*(t-t0)/tp]

f1 (average)	10.0000		
f2 (amplitude)	10.0000		
t0 (phase)	21600.0000	or	6h
tp (period)	86400.0000	or	1d

Tip for variation:
Daily: t0=6h, tp=1d
Yearly: t0=3q, tp=1y

Figure 4.29: 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.

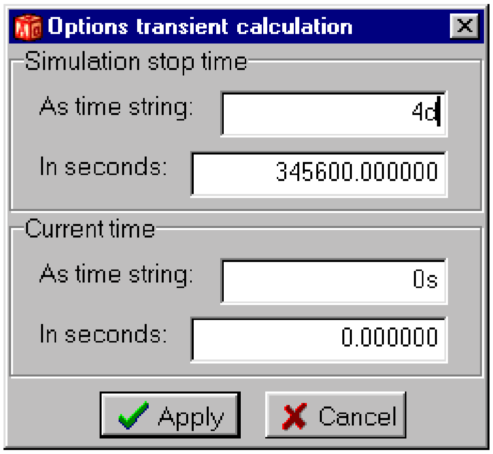
Next step is to couple the function to a boundary condition type, see Figure 4.30. Finally, the stop time 4 days (Solve/Options for transient) is given, see Figure 4.31.



Number of types: 3

<n>	type	q [W/m²]	T [°C]	R [m²K/W]
1	Q=const	0.00000		
2	T=const		20.00000	0.130000
3	T=const		0.00000	0.040000

Figure 4.30: A temperature function is chosen in the pick list for the boundary condition type. The function number is selected in the function pick list. Type number 1 (Q=0) is default and will initially apply to all external boundaries.


Options transient calculation

Simulation stop time

As time string: 4d

In seconds: 345600.000000

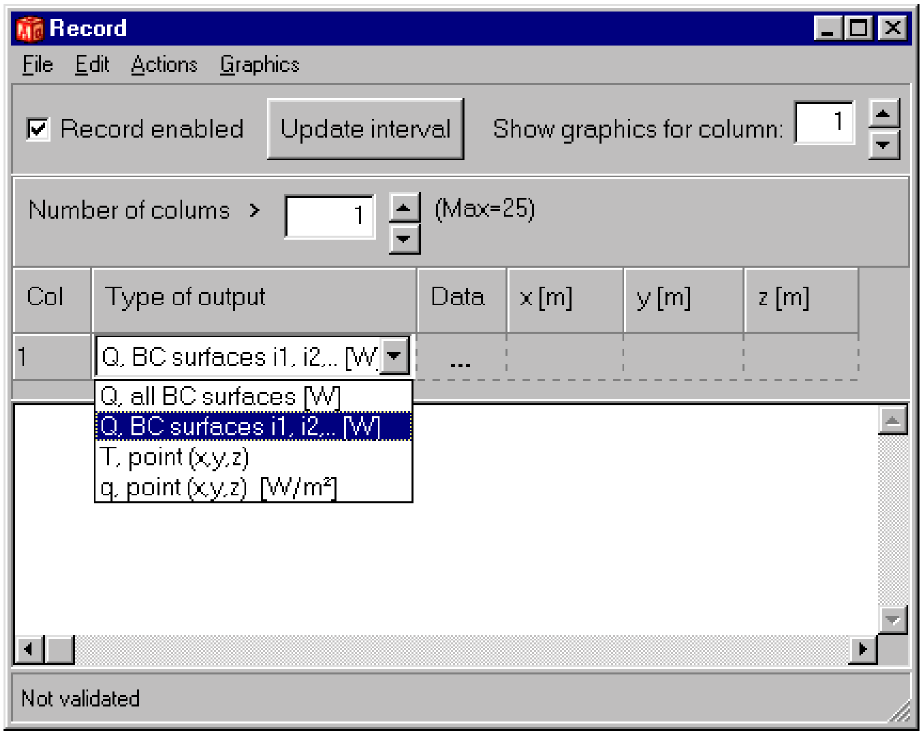
Current time

As time string: 0s

In seconds: 0.000000

Figure 4.31: 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 4.32 (top). Click in the data column and give the numbers 1, 3 and 5 for the internal boundaries as in Figure 4.32 (bottom). This will record the sum of the two boundary heat flows at each screen update (*Solve/Update*) as given in Figure 4.33, 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).


Record

☑ Record enabled

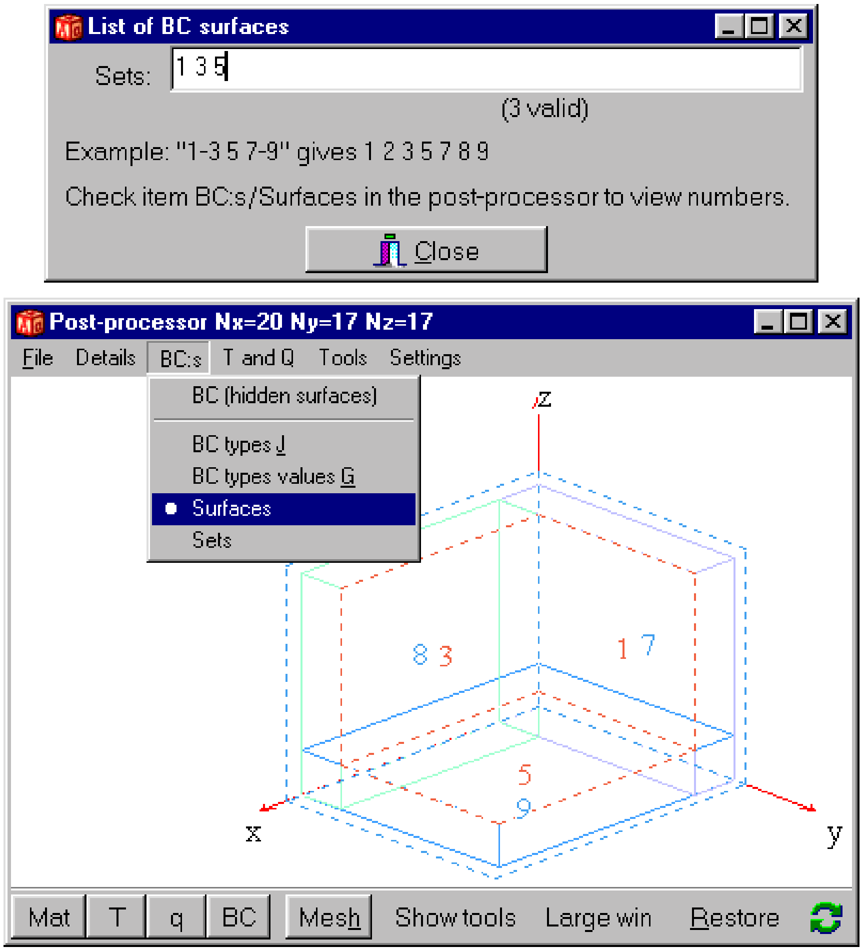
Show graphics for column: 1

Number of colums > 1 (Max=25)

Col	Type of output	Data	x [m]	y [m]	z [m]
1	Q, BC surfaces i1, i2,.. [W]	...			

Q, all BC surfaces [W]
Q, BC surfaces i1, i2,.. [W]
T, point (x,y,z)
q, point (x,y,z) [W/m²]

Not validated



List of BC surfaces

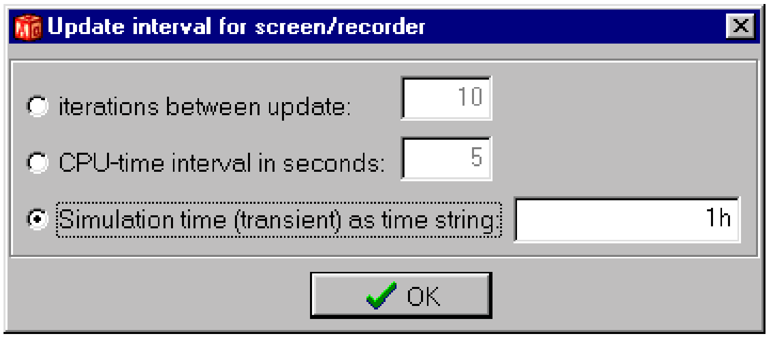
Sets: 1 3 5

(3 valid)

Example: "1-3 5 7-9" gives 1 2 3 5 7 8 9

Check item BC:s/Surfaces in the post-processor to view numbers.

Figure 4.32: Flows and temperatures may be saved to a list during the simulation. In this case the sum of the heat flows through the internal boundaries (1, 3 and 5) will be recorded.



Update interval for screen/recorder

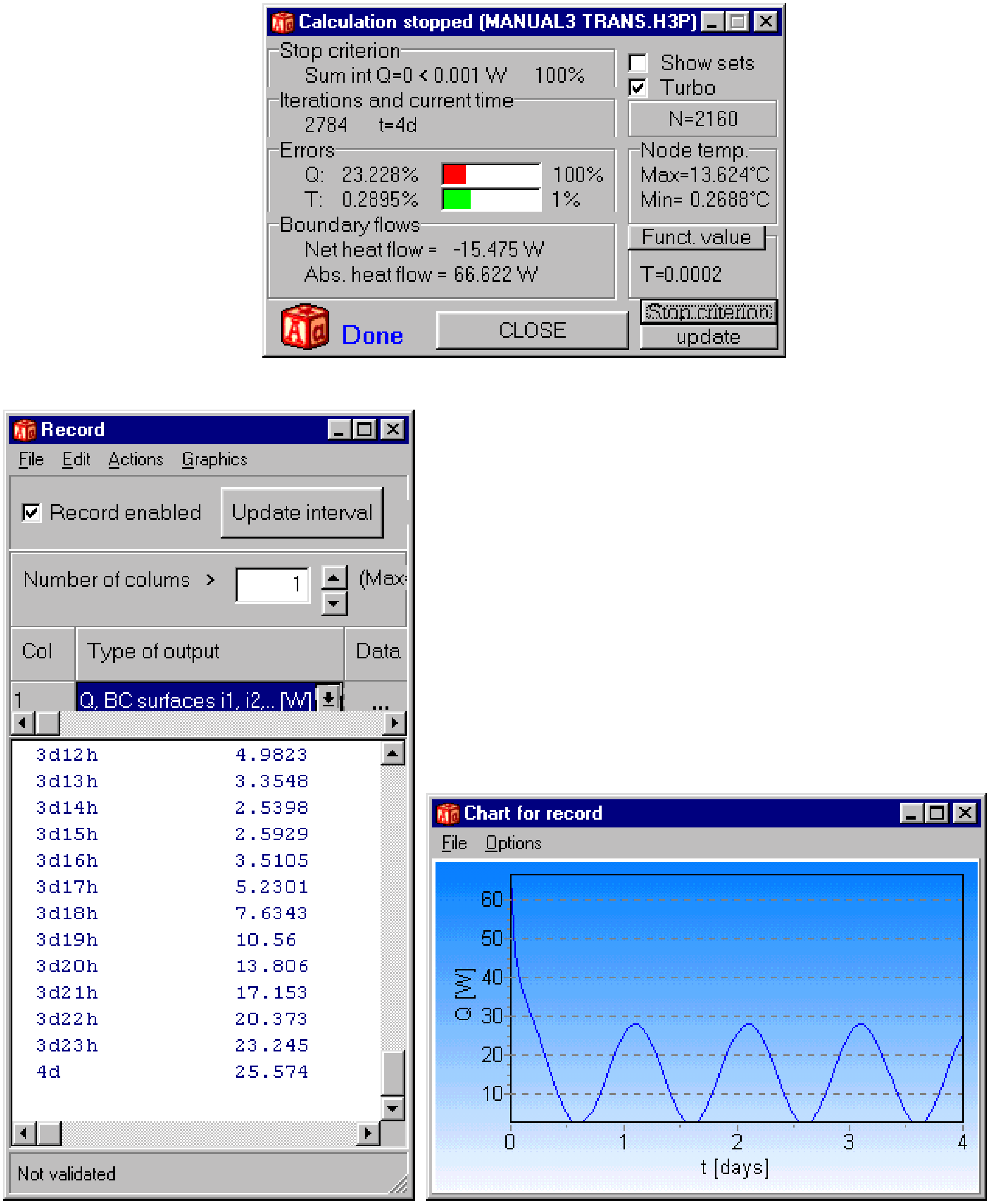
○ iterations between update: 10

○ CPU-time interval in seconds: 5

● Simulation time (transient) as time string: 1h

Figure 4.33: The time interval for recorded data is given in item *Solve/Update*.

Start the transient simulation by selecting *Solve/Start transient calculation*. Consider Figure 4.34. 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.



Stop criterion
Sum int Q=0 < 0.001 W 100%

Iterations and current time
2784 t=4d

Errors
Q: 23.228% 🟥 100%
T: 0.2895% 🟩 1%

Boundary flows
Net heat flow = -15.475 W
Abs. heat flow = 66.622 W

□ Show sets
☑ Turbo
N=2160

Node temp.
Max=13.624°C
Min= 0.2688°C

T=0.0002

☑ Record enabled

Number of colums > 1 (Max

Col	Type of output	Data
1	Q, BC surfaces i1, i2,.. [W]	...

```
3d12h       4.9823
3d13h       3.3548
3d14h       2.5398
3d15h       2.5929
3d16h       3.5105
3d17h       5.2301
3d18h       7.6343
3d19h       10.56
3d20h       13.806
3d21h       17.153
3d22h       20.373
3d23h       23.245
4d          25.574
```

Not validated

Figure 4.34: 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.

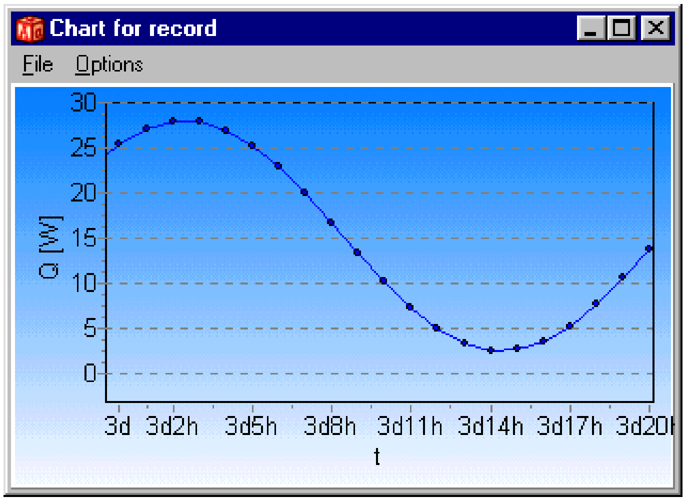


Figure 4.35: The maximum heat flow through the internal boundaries into the wall occurs at two-o-clock in the morning and the minimum at two-o-clock in the afternoon.

5. Working with HEAT3

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.

5.1.2 Generating input data

The pre-processor generates input data in text format that can be viewed by pressing the main menu item *Editor*. Normally the user does not have to edit data in text mode, but it is possible to make changes in text mode (and also to describe a whole new problem). Note that changes in the text editor will **not** update the graphical input given in the pre-processor. Also note that all text in the editor will be over-written when the pre-processor content is updated.

However, a couple of reasons to edit data in text mode may be to fit a better numerical mesh or to change some lengths if there are very large differences in scale (e.g. a small metal nail in a wall that stands on a foundation where a large extent of the underlying soil is taken into account in the simulation).

The automatic updating that occur upon changes in the pre-processor may be disabled by item *Setting/Update text editor upon changes in the pre-processor*. If this item is disabled, updating may manually be made by pressing the upper right menu item <*Update not made*>.

Tip: The updating in the post-processor will be quicker (especially while dragging objects around) if few numerical cells are given, see menu item *Settings/Mesh*. It may be a good idea to specify a proper full numerical mesh when the geometry and boundary conditions have been finalized.

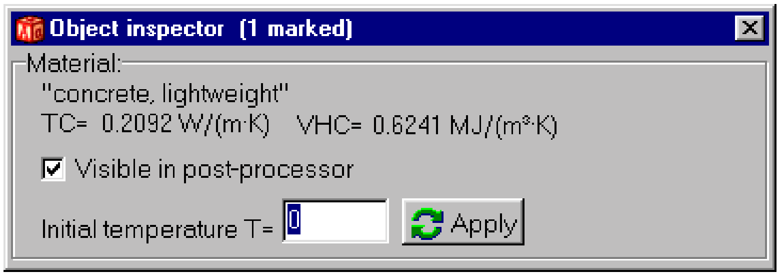
5.1.3 Describing 3D geometry using 2D drawings

The full 3D geometry is given by overlapping boxes. Each box is drawn two-dimensionally in two planes: (*x*,*y*) and (*x*,*z*) (use menu item on the tool bar to the left to toggle between them). The length and position in each direction may be given in text mode or by dragging/resizing the selected boxes. Sometimes it is necessary to bring an object to the front (see below) in order to adjust its properties.

Different objects are drawn on separate layers, see menu item *Layers*. The command Bring to Front /Send to back applies only to objects in the same layer. Each layer may be turned off. For example, BC boxes are on separate layers above the layer with material boxes. **Tip:** This means that to be able to select a material box it is sometimes required to turn the layer *BC* off .

5.1.4 The object inspector

The object inspector may be viewed for currently selected objects by clicking on the item *obj* on the left tool bar, or by double-clicking an object. Figure 5.1 shows the properties for a material box. The box can be made invisible in the post-processor (it is however taken into account in the calculation). The initial temperature can also be set. Figure 5.2 shows the properties for a heat source.



Object inspector (1 marked)

Material:

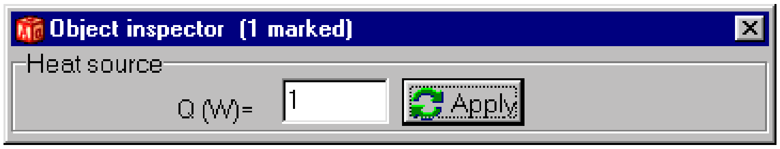
"concrete, lightweight"

TC= 0.2092 W/(m·K) VHC= 0.6241 MJ/(m³·K)

☑ Visible in post-processor

Initial temperature T= 0

Figure 5.1: Properties for a material box. The box can be made invisible in the post-processor (it is however taken into account in the calculation). The initial temperature can also be set.



Object inspector (1 marked)

Heat source

Q (W)= 1

Figure 5.2: The only properties for a heat source is the source term.

5.1.5 The tool bar

There is a tool bar shown above the drawing area, see Figure 4.2. 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.

The material is chosen from the material pick list, see Figure 4.1 (left). Double-clicking a rectangle brings up a window with material name and properties, see Figure 4.1 (right).

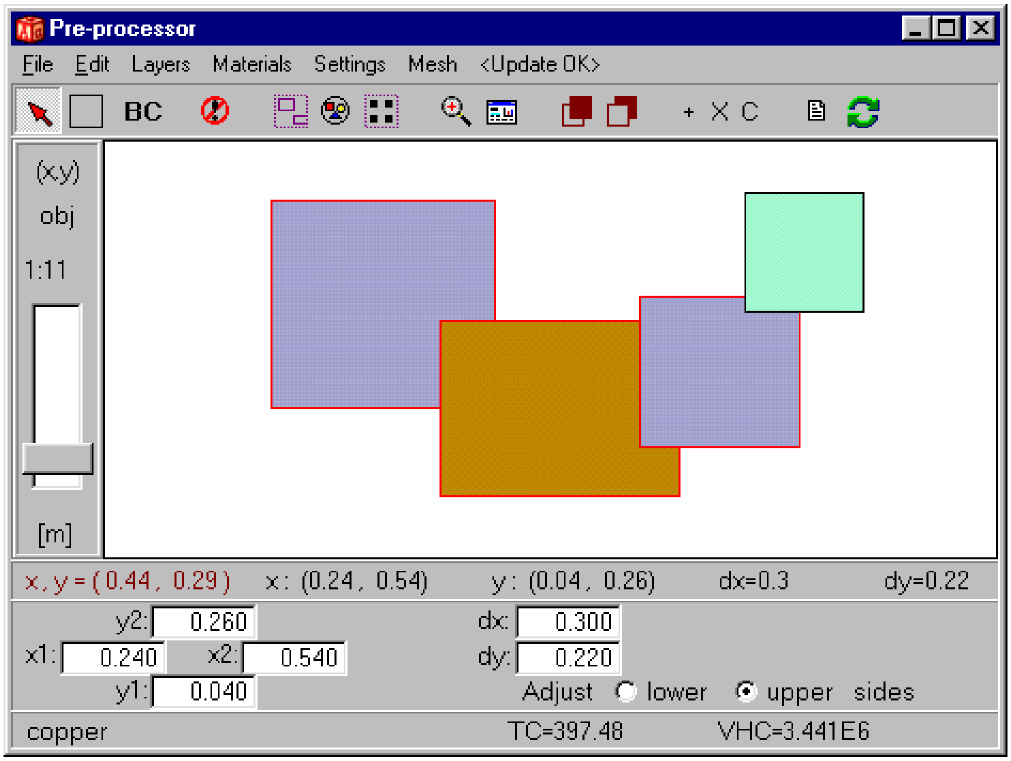
5.1.6 The tool bar

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).

Tip: Hold down the Shift-key and click on an object to add it to the current selection container. If the object already is selected, it will be de-selected.

Tip: Click on an already selected object to get its properties. This may e.g. be used to align several objects. Figure 5.3 shows three marked objects outlined in red. Clicking on an already marked object will fetch its properties (e.g. position and material name as shown here). Press <Enter> at the bottom coordinate y1=0.040 to align the lower sides of all selected objects, see Figure 5.4.

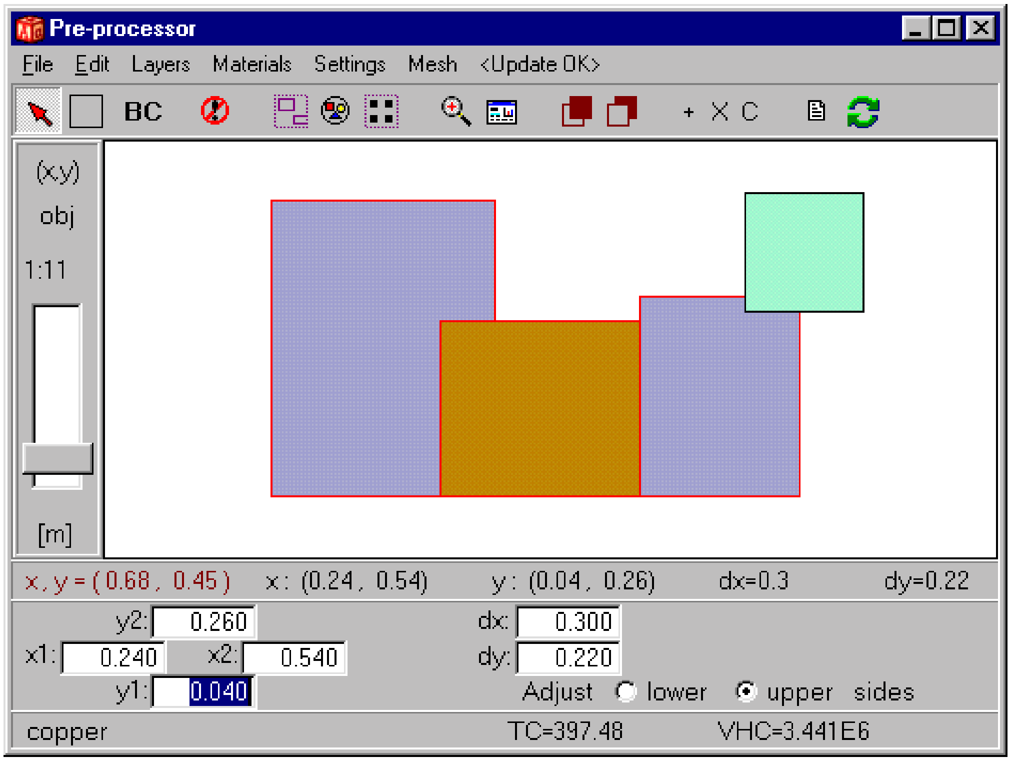


x, y = (0.44 , 0.29)   x : (0.24 , 0.54)   y : (0.04 , 0.26)   dx=0.3   dy=0.22

y2: 0.260
x1: 0.240   x2: 0.540   dx: 0.300
y1: 0.040   dy: 0.220
Adjust ○ lower ● upper sides

copper   TC=397.48   VHC=3.441E6

Figure 5.3: Three marked objects outlined with red. Clicking on an already marked object will fetch its properties (e.g. position and material name as shown here).



x, y = (0.68 , 0.45)   x : (0.24 , 0.54)   y : (0.04 , 0.26)   dx=0.3   dy=0.22

y2: 0.260
x1: 0.240   x2: 0.540   dx: 0.300
y1: 0.040   dy: 0.220
Adjust ○ lower ● upper sides

copper   TC=397.48   VHC=3.441E6

Figure 5.4: Press <Enter> at the bottom coordinate y1=0.040 to align the lower sides of all selected objects.

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 current selected object type (e.g. material box, or BC box with same type). 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.

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.

X **Expansion point for numerical grid**: An expansive mesh with cells being smaller (or larger) towards the expansion point. Give expansion factor in item *Mesh*.

C **Cut-out box**: Only material inside volume will be visible in the post-processor.

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

Redraw: Invalidates the drawing area.

5.1.7 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.

5.1.8 Picking material

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

5.1.9 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*.

5.1.10 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.

5.1.11 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 the text editor (main menu item Editor).** See also Section 5.1.2.

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.

5.1.12 Drawing in metric or English units

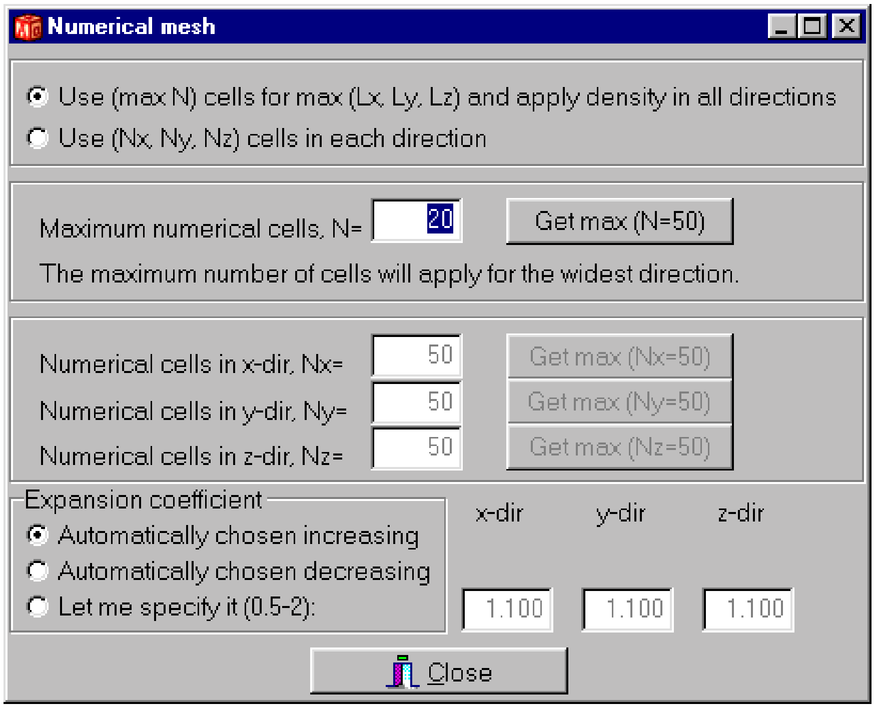
By default lengths are drawn using metric units. It is also possible to change the unit to inch, foot or yard. To do this click on the button located in the lower right corner. The lengths will automatically be calculated to the metric unit when updating occurs. Note that the drawn rectangles still are 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.

5.1.13 Numerical mesh

The number of computational cells is specified in each direction, see Figure 5.5. The cells will be spread out in an equidistant mesh. Note that there always has to be a minimum number of cells for the computational region.

It is also possible to give an expansive mesh by putting an expansion point in the pre-processor, see Section 5.1.6. A mesh with cell sizes increasing or decreasing towards the point may be given, see Figure 5.5.



Numerical mesh

● Use (max N) cells for max (Lx, Ly, Lz) and apply density in all directions

○ Use (Nx, Ny, Nz) cells in each direction

Maximum numerical cells, N= 20 | Get max (N=50)

The maximum number of cells will apply for the widest direction.

Numerical cells in x-dir, Nx= 50 | Get max (Nx=50)

Numerical cells in y-dir, Ny= 50 | Get max (Ny=50)

Numerical cells in z-dir, Nz= 50 | Get max (Nz=50)

Expansion coefficient

● Automatically chosen increasing

○ Automatically chosen decreasing

○ Let me specify it (0.5-2):

x-dir	y-dir	z-dir
1.100	1.100	1.100

Figure 5.5: The number of numerical cells are given.

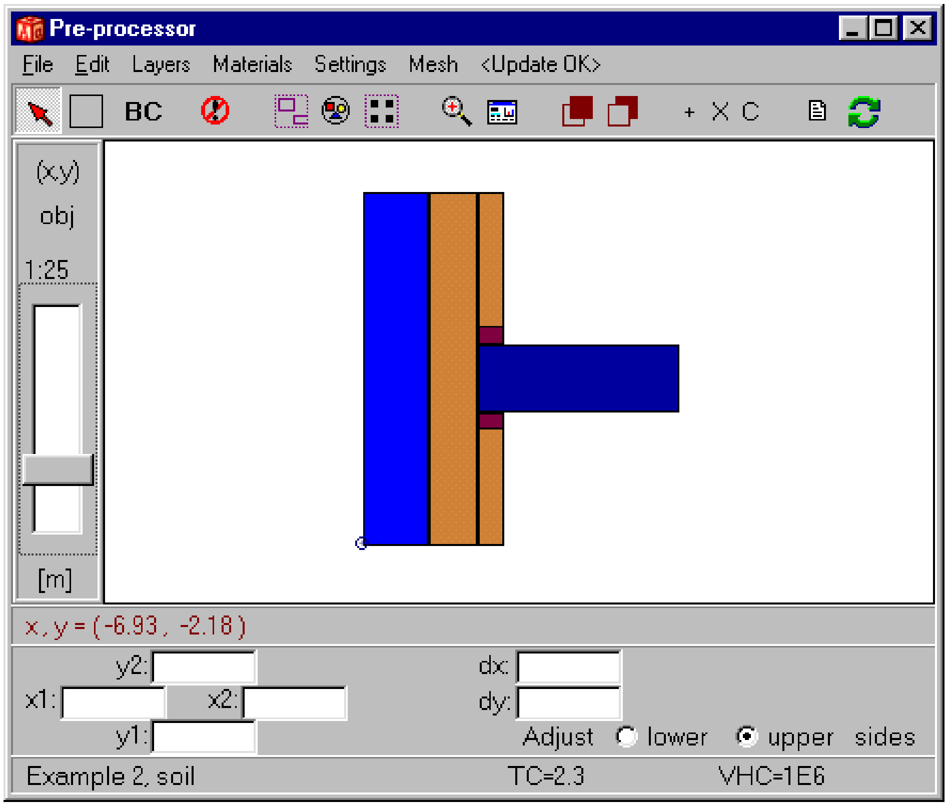
5.1.14 Importing pictures

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. To import a picture (BMP, EMF, WMF, ICO), click menu item *Edit/Import picture*.

5.1.15 Importing geometry from H3P- and H2P-files.

A geometry from another H3P file may be imported, see menu item *File/Import H3P*. The geometry will be added to the current drawn objects if menu item *File/Import H3P 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 H3P-file, see menu item *File/Export H3P*.

It is possible to import 2D geometry from HEAT2 (version 5 required). Figure 5.6 shows the HEAT2 example EXAMPLE 1.H2P imported in the (*x*,*y*)-plane. The height in the z-direction may be changed by viewing the (*x*,*z*)-plane, marking all objects and changing the width (dz). The height is in Figure 4.5 changed to dz=0.5 m. The geometry can easily be edited and new material boxes can be added. Boundary conditions have to be specified before simulation is possible.



x, y = (-6.93 , -2.18)

Adjust ○ lower ● upper sides

Example 2, soil TC=2.3 VHC=1E6

Figure 5.6: Imported 2D geometry from a HEAT2 H2P-file.

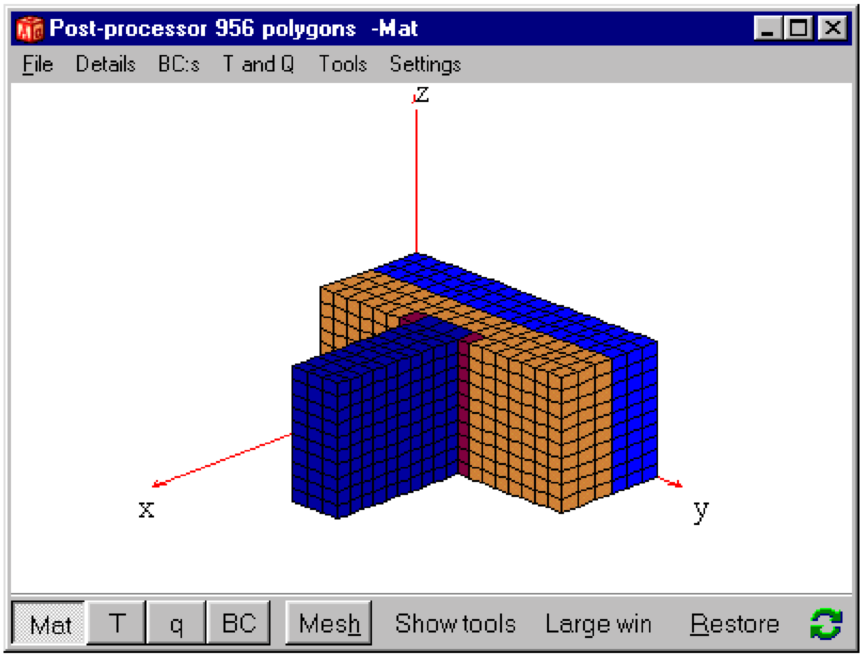

Post-processor 956 polygons -Mat
File Details BC:s T and Q Tools Settings
z
x
y
Mat T q BC Mesh Show tools Large win Restore

Figure 5.7: The height in the z-direction is here changed to dz=0.5 m.

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 Institut für Normung, DIN V 4108-4) is available. **See www.buildingphysics.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).

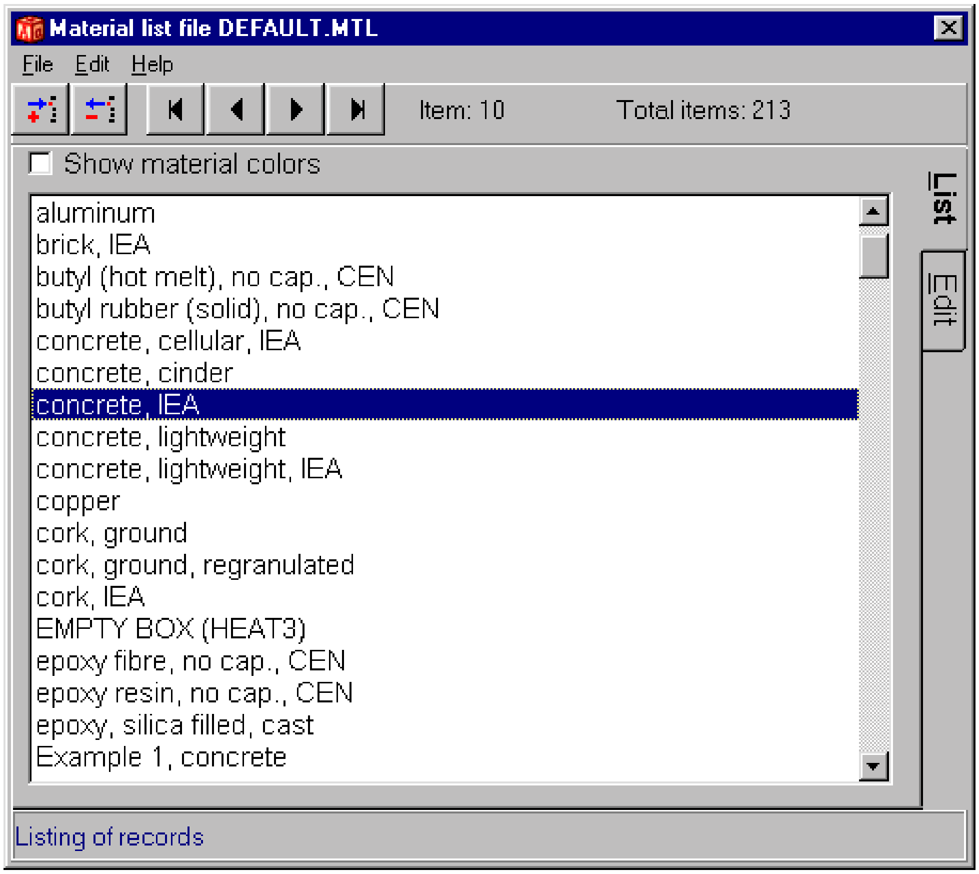
In some cases heat capacities are not given (marked with "No Cap."), and the heat capacity is set to 1. These materials should only be used in steady-state calculations.

Note that the format of the material list file is the same as for HEAT2 version 5. If you have edited/added a list in HEAT2 you will be able to use this right away. Also note that value for the thermal conductivity in the *x*-direction applies to all directions (*x,y,z*). This is due to the fact that HEAT2 handles different properties in the (x,y)-directions, but HEAT3 can only handle the same property in all three directions (x,y,z).

Figure 5.8 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.

The volumetric heat capacity is denoted by C, (J/(m^3·K)), which is the density ρ, (kg/m^3), 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/(m^3·K)), that is $1.0 \cdot 10^6$ J/(m^3·K). The default value of C is 1.0 MJ/(m^3·K). For the steady-state simulation the value of C is neglected.



Item: 10

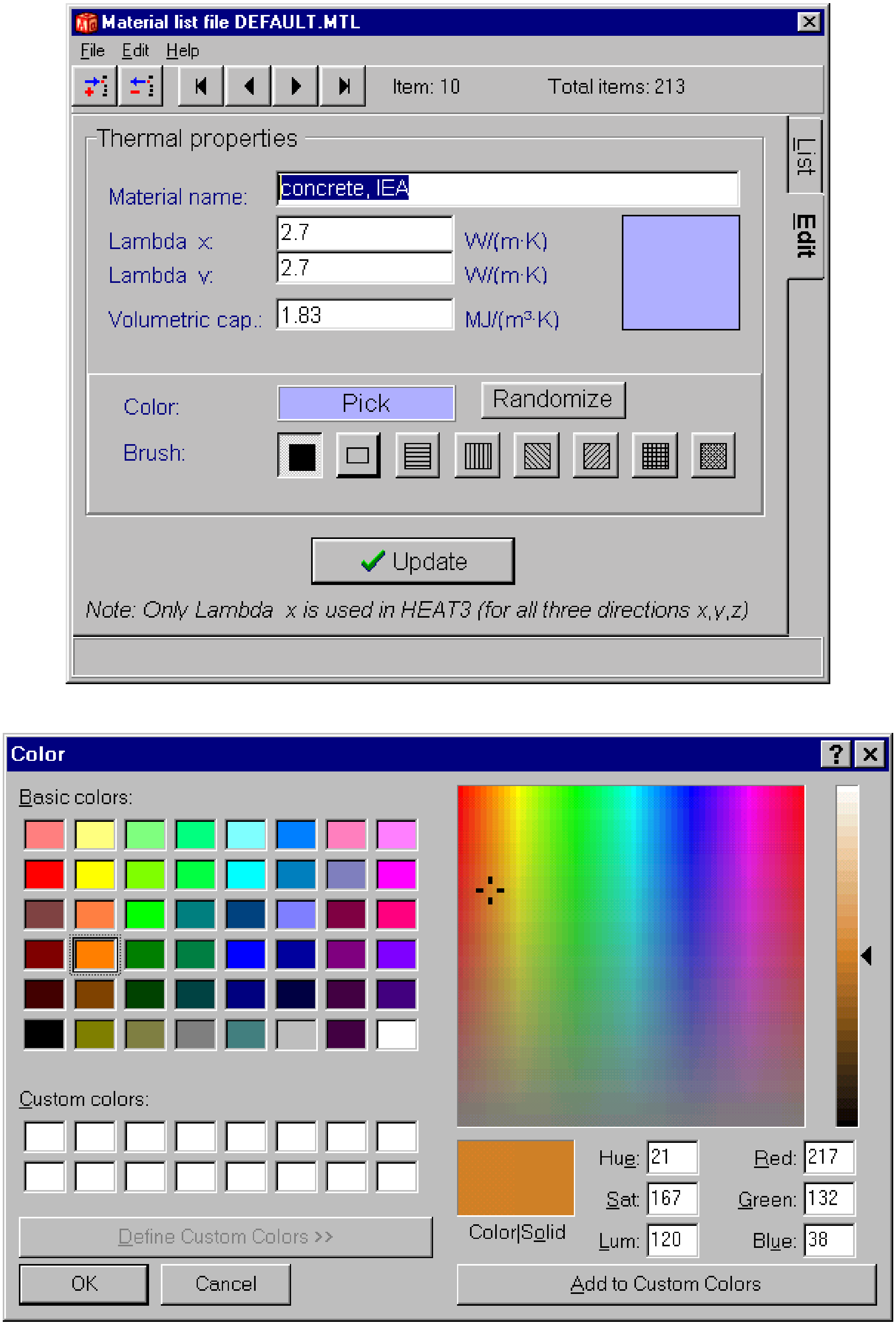
Total items: 213

□ Show material colors

aluminum
brick, IEA
butyl (hot melt), no cap., CEN
butyl rubber (solid), no cap., CEN
concrete, cellular, IEA
concrete, cinder
concrete, IEA
concrete, lightweight
concrete, lightweight, IEA
copper
cork, ground
cork, ground, regranulated
cork, IEA
EMPTY BOX (HEAT3)
epoxy fibre, no cap., CEN
epoxy resin, no cap., CEN
epoxy, silica filled, cast
Example 1, concrete

Listing of records

Figure 5.8: The material editor.



Material list file DEFAULT.MTL

Item: 10 Total items: 213

Thermal properties

Material name: concrete, IEA

Lambda x: 2.7 W/(m·K)

Lambda y: 2.7 W/(m·K)

Volumetric cap.: 1.83 MJ/(m³·K)

Note: Only Lambda x is used in HEAT3 (for all three directions x,y,z)

Color

Hue: 21 Red: 217

Sat: 167 Green: 132

Lum: 120 Blue: 38

Figure 5.9: 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 HEAT3 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 HEAT3.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 HEAT3.

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. 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 Boundary conditions

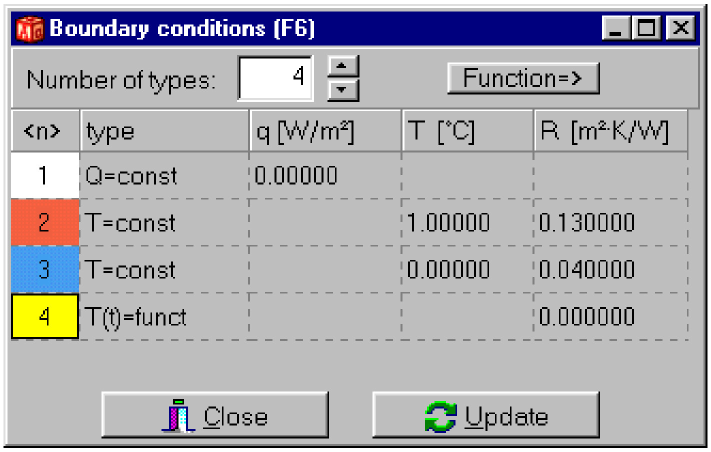
5.3.1 Introduction

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 5.6.

The user defines the boundary conditions in a list and assigns them to the boundary segments. Consider Figure 5.10. There are in this case four boundary conditions with list numbers n=1-4 in the first column.

The first type (here q=0) is default. This means that all boundaries not assigned to any type will use the first type.

The third column links to a function for temperatures (sinusoidal, stepwise constant or step-wise linear). The last column gives the surface thermal resistance (applies to a type with temperature).



Number of types: 4

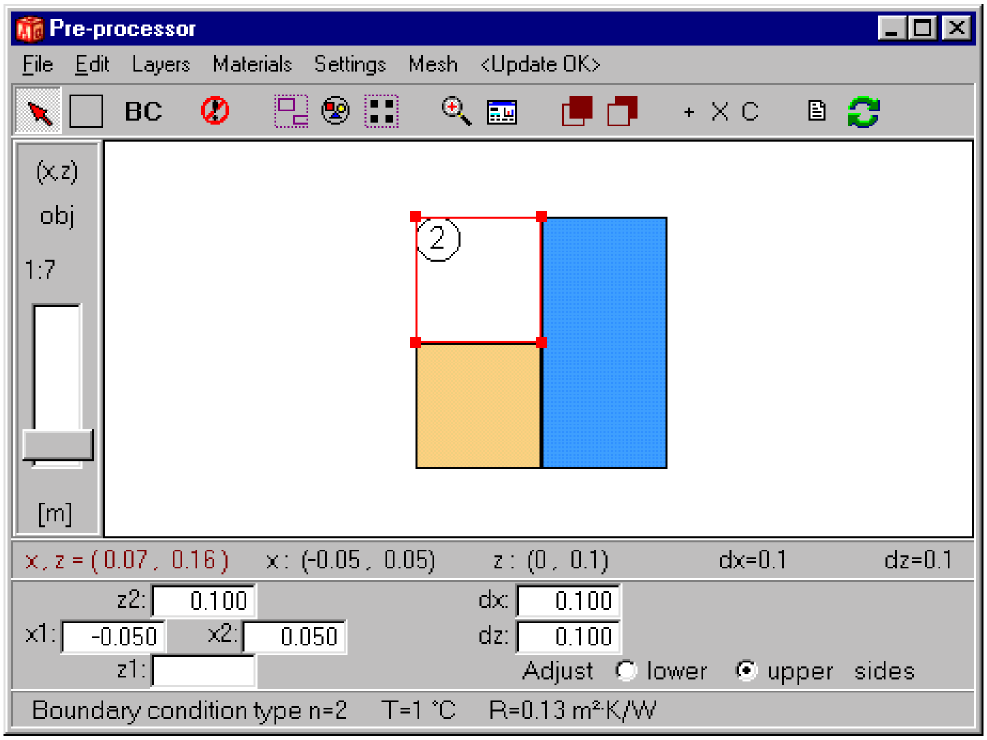
<n>	type	q [W/m²]	T [°C]	R [m²K/W]
1	Q=const	0.00000		
2	T=const		1.00000	0.130000
3	T=const		0.00000	0.040000
4	T(t)=funct			0.000000

Figure 5.10: The boundary types are linked to each boundary segment.

5.3.2 Disabling sides of a BC box

In rare cases it may be easier to specify boundary conditions if one or a few of the six sides of a BC box can be disabled. Figure 5.11 shows a BC box drawn in the pre-processor. Two of the six sides will be recognized as shown in the post-processor.

Double-clicking on the BC box will show a dialog box where each of the six surfaces can be disabled, see Figure 5.12. The upper side in the x-direction (*x*+) has been disabled and will not be used here.



x, z = (0.07 , 0.16)   x : (-0.05 , 0.05)   z : (0 , 0.1)   dx=0.1   dz=0.1

z2: 0.100

x1: -0.050   x2: 0.050

z1:

dx: 0.100

dz: 0.100

Adjust ○ lower ● upper sides

Boundary condition type n=2   T=1 °C   R=0.13 m²K/W

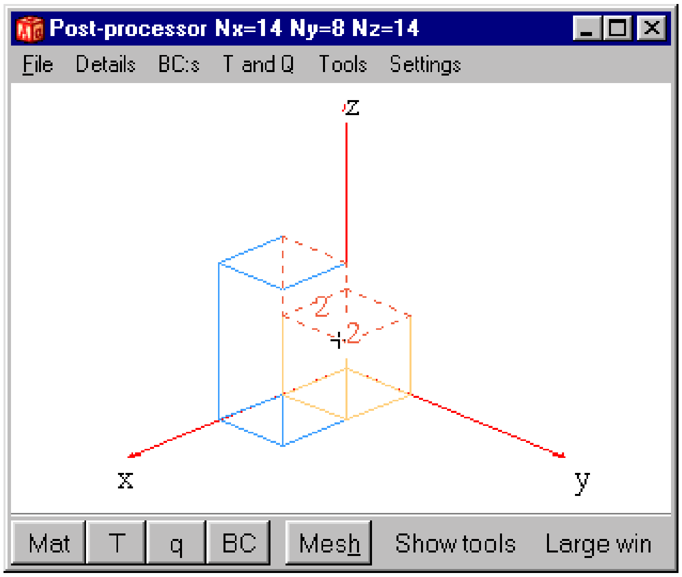
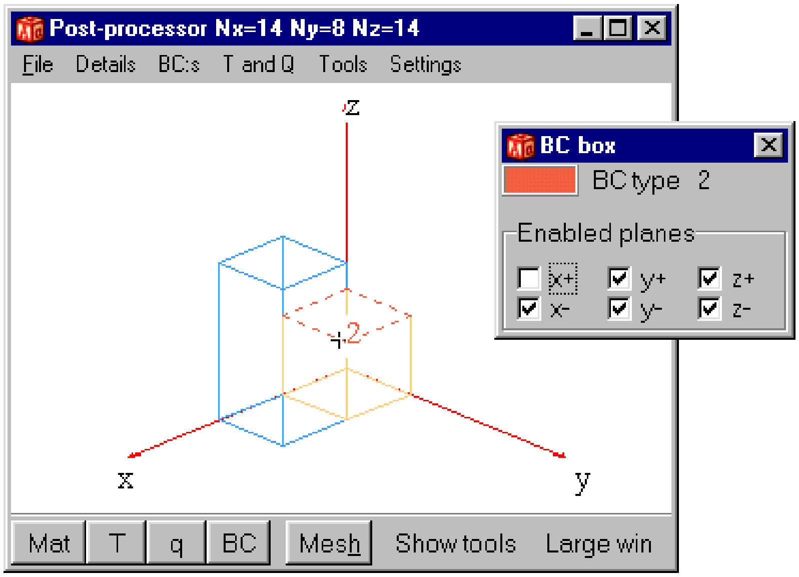


Figure 5.11: A BC box drawn in the pre-processor (top figure). Two of the six sides will be recognized in this case as shown in the post-processor (bottom figure).



🟥 BC type 2

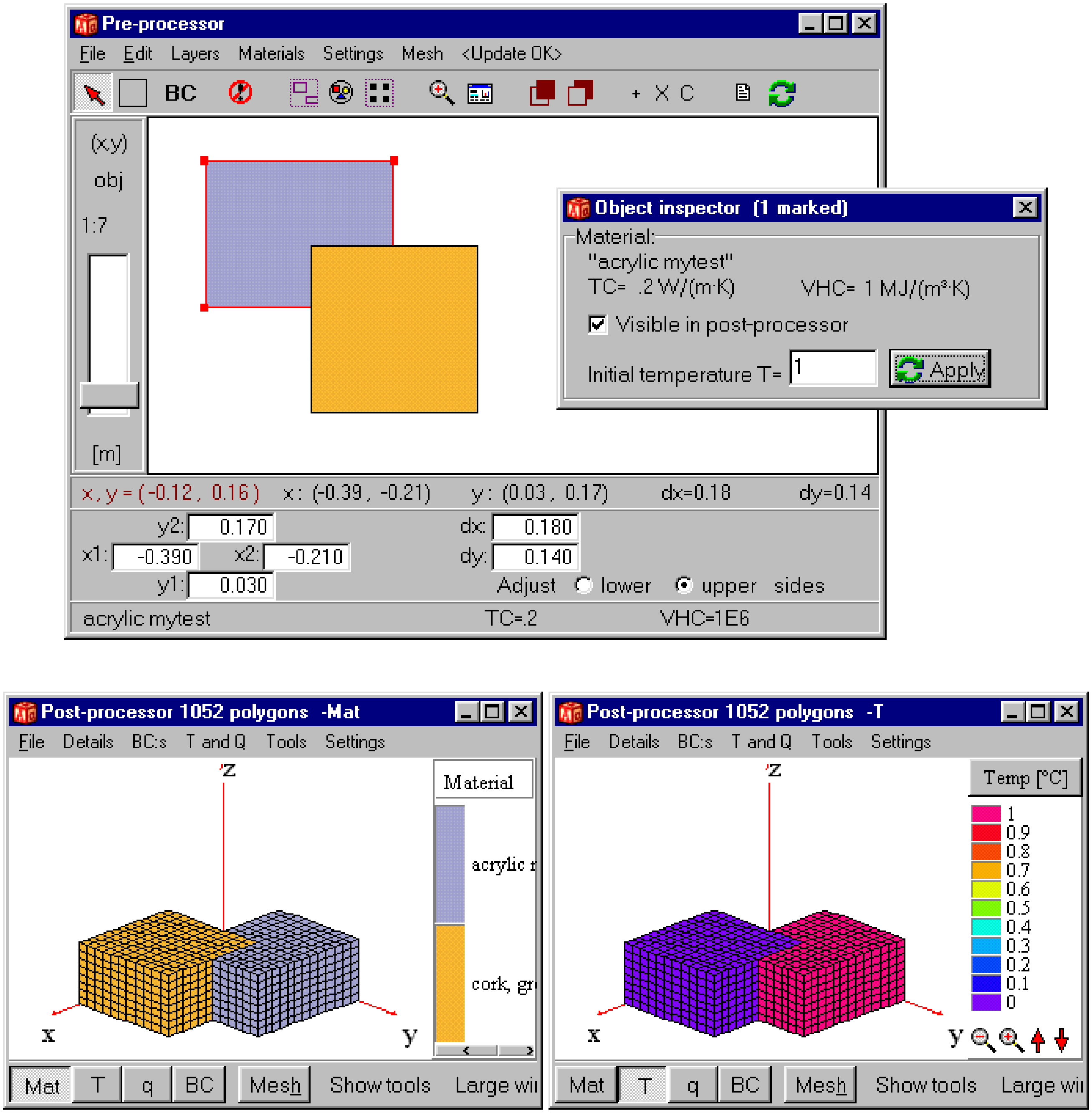
Enabled planes

☐ x+ ☑ y+ ☑ z+
☑ x- ☑ y- ☑ z-

Figure 5.12: The upper side in the x-direction (*x*+) has been disabled and will not be used. In this case it is actually only the side (*z*-) that will be recognized.

5.4 Initial temperatures

Initial temperatures may be given for each material box by using the object inspector, see Figure 5.13.


Pre-processor
File Edit Layers Materials Settings Mesh <Update OK>
BC
+ X C
(x,y)
obj
1:7
[m]
Object inspector (1 marked)
Apply
Post-processor 1052 polygons -Mat
File Details BC:s T and Q Tools Settings
Z
x
y
Material
acrylic r
cork, gr
Mat T q BC Mesh Show tools Large wi
Post-processor 1052 polygons -T
File Details BC:s T and Q Tools Settings
Z
x
y
Temp [°C]
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Mat T q BC Mesh Show tools Large wi

Material:
"acrylic mytest"
TC= .2 W/(m·K) VHC= 1 MJ/(m³·K)
☑ Visible in post-processor
Initial temperature T= 1

x,y=(-0.12, 0.16) x: (-0.39, -0.21) y: (0.03, 0.17) dx=0.18 dy=0.14

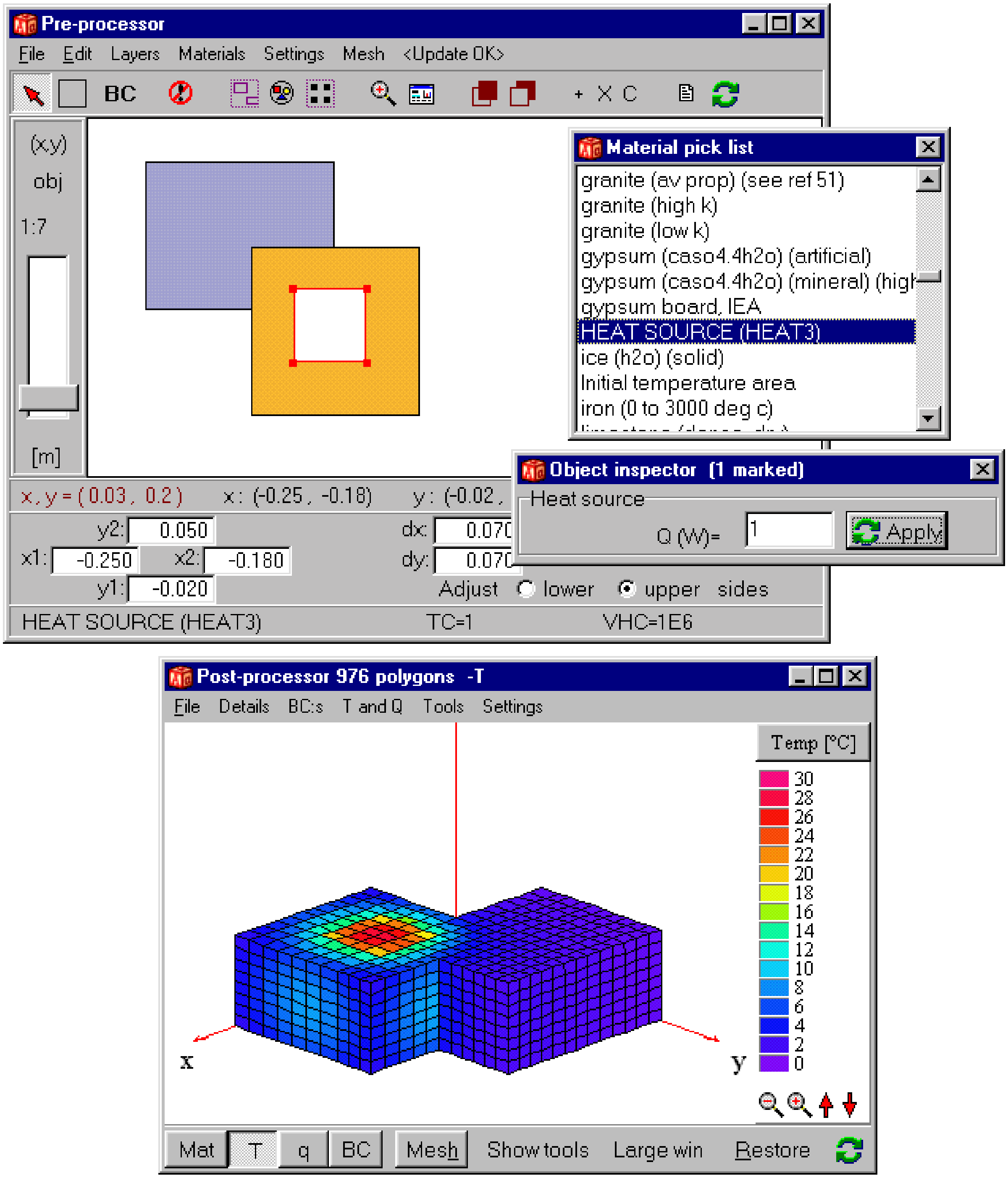
y2: 0.170
x1: -0.390 x2: -0.210
y1: 0.030
dx: 0.180
dy: 0.140
Adjust ○ lower ● upper sides

acrylic mytest TC=.2 VHC=1E6

Figure 5.13: Initial temperatures may be given for each material box. The top figure shows the pre-processor and the object inspector. In this case the gray box is given T=1 (the yellow box has as default T=0). The bottom figure shows materials and temperatures.

5.5 Heat sources

Heat sources may be given for each material box by picking "HEAT SOURCE (HEAT3)" in the material list, see Figure 5.13. The value of the heat source is given in the object inspector. A negative value gives a heat sink.


Pre-processor
File Edit Layers Materials Settings Mesh <Update OK>
BC
+ X C
(x,y)
obj
1:7
[m]
Material pick list
Object inspector (1 marked)
Apply
Post-processor 976 polygons -T
File Details BC:s T and Q Tools Settings
Temp [°C]
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
x
y
Mat T q BC Mesh Show tools Large win Restore

granite (av prop) (see ref 51)
granite (high k)
granite (low k)
gypsum (caso4.4h2o) (artificial)
gypsum (caso4.4h2o) (mineral) (high
gypsum board, IEA
HEAT SOURCE (HEAT3)
ice (h2o) (solid)
Initial temperature area
iron (0 to 3000 deg c)

Heat source
Q (W)= 1

x, y = (0.03 , 0.2)   x : (-0.25 , -0.18)   y : (-0.02 ,

y2: 0.050   dx: 0.070
x1: -0.250   x2: -0.180   dy: 0.070
y1: -0.020   Adjust ○ lower ● upper sides

HEAT SOURCE (HEAT3)   TC=1   VHC=1E6

Figure 5.14: Heat sources are drawn using "HEAT SOURCE (HEAT3)" in the material list. The value of the heat source/sink is given in the object inspector. The lower figure shows calculated temperatures (all boundaries have T=0 and R=1 here).

5.6 Functions

Functions may be used to describe changes of temperatures in time for boundary conditions. The following functions are available:

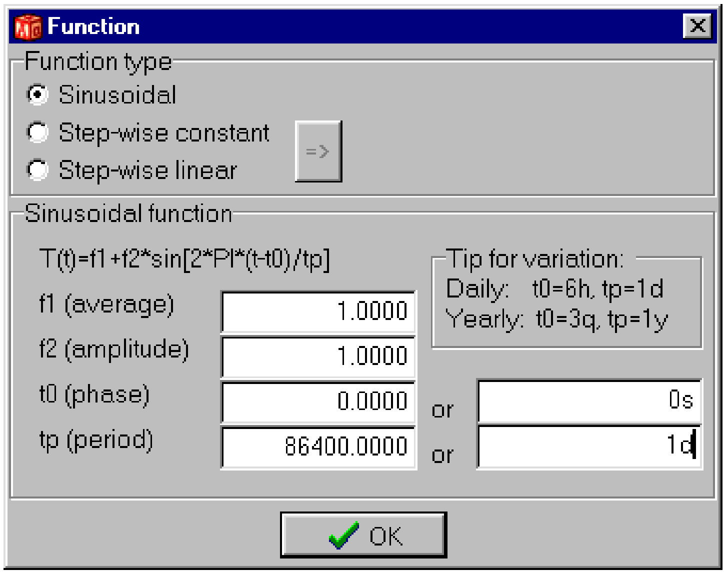
- $f(t) = f_1 + f_2 \cdot \sin\left(\dfrac{2\pi(t-t_0)}{t_p}\right) t_p > 0$ *(sinusoidal)*

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

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

5.6.1 Sinusoidal function

Figure 5.15 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.



Function

Function type

● Sinusoidal

○ Step-wise constant

○ Step-wise linear

Sinusoidal function

T(t)=f1+f2*sin[2*PI*(t-t0)/tp]

Tip for variation:
Daily: t0=6h, tp=1d
Yearly: t0=3q, tp=1y

f1 (average) 1.0000

f2 (amplitude) 1.0000

t0 (phase) 0.0000 or 0s

tp (period) 86400.0000 or 1d

Figure 5.15: 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)

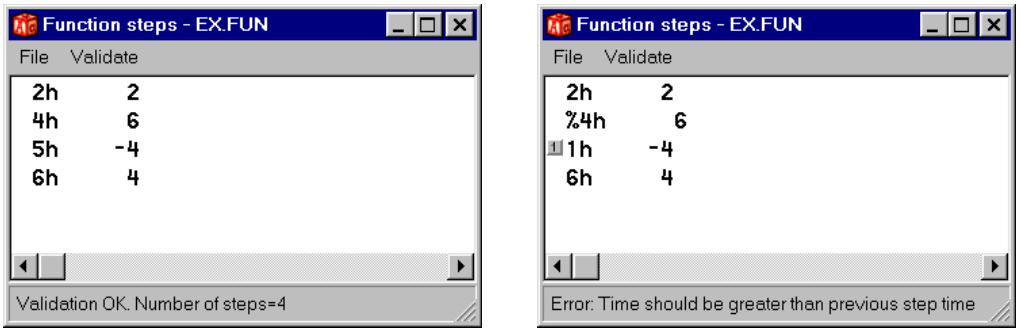
Please note that *m* is minute and *q* is month. Here are some examples:

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

5.6.2 Step-wise constant and step-wise linear function

The step-wise constant and step-wise linear values are given in an editor, see Figure 5.16. 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 5.16 (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 5.21 and Figure 5.22. 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.



```
2h      2
4h      6
5h     -4
6h      4
```

Validation OK. Number of steps=4

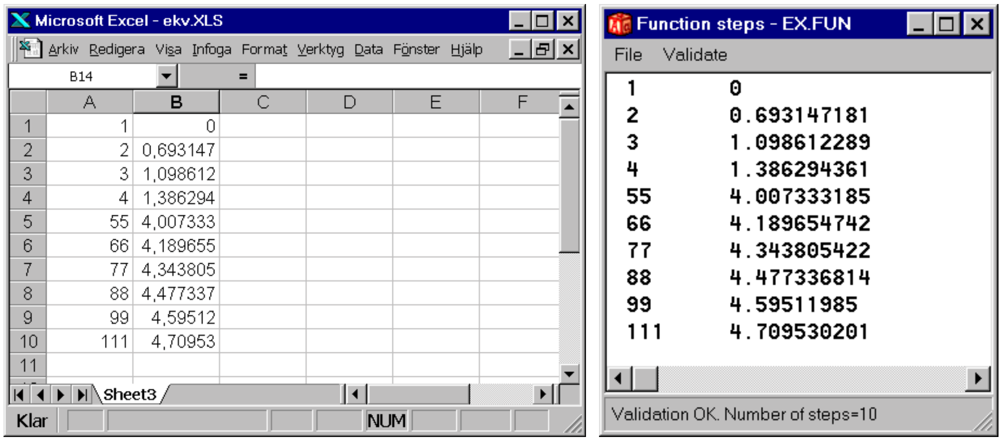
```
2h      2
%4h      6
1h     -4
6h      4
```

Error: Time should be greater than previous step time

Figure 5.16: 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 5.17 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 5.8.


Microsoft Excel - ekv.XLS
Arkiv Redigera Visa Infoga Format Verktyg Data Fönster Hjälp
B14
=
Sheet3
Klar
NUM
Function steps - EX.FUN
File Validate

	A	B	C	D	E	F
1	1	0				
2	2	0,693147				
3	3	1,098612				
4	4	1,386294				
5	55	4,007333				
6	66	4,189655				
7	77	4,343805				
8	88	4,477337				
9	99	4,59512				
10	111	4,70953				
11						

```
1       0
2       0.693147181
3       1.098612289
4       1.386294361
55      4.007333185
66      4.189654742
77      4.343805422
88      4.477336814
99      4.59511985
111     4.709530201
```

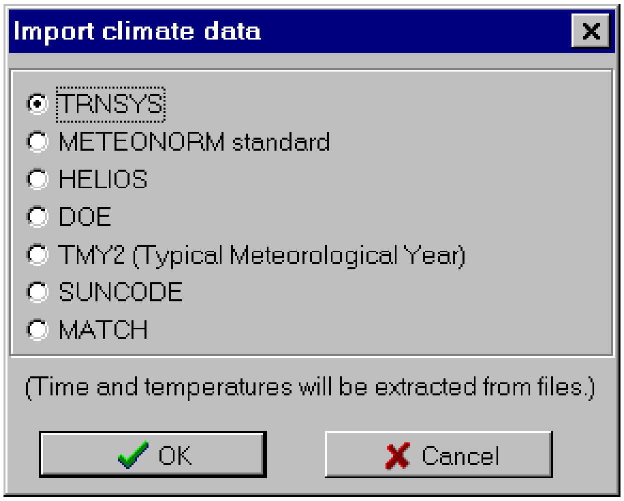
Validation OK. Number of steps=10

Figure 5.17: 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).

5.6.3 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 5.18 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.



● TRNSYS
○ METEONORM standard
○ HELIOS
○ DOE
○ TMY2 (Typical Meteorological Year)
○ SUNCODE
○ MATCH

(Time and temperatures will be extracted from files.)

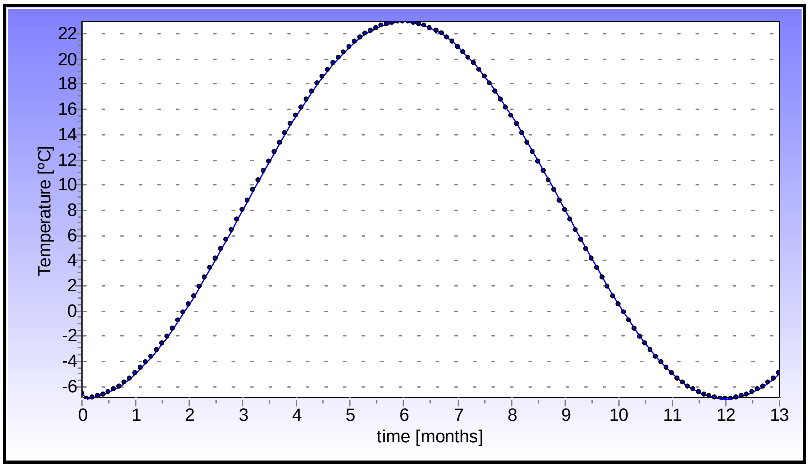
Figure 5.18: Time and temperatures can be extracted from several climatic files.

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

Figure 5.19 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].

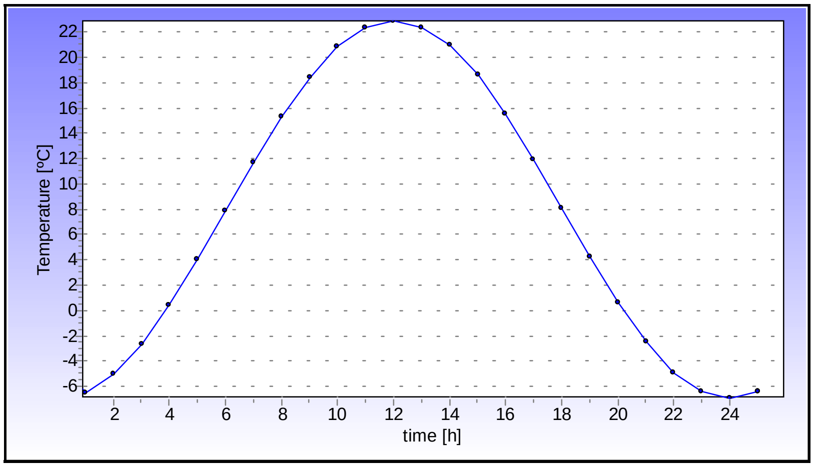
5.6.5 Example 2 - sinusoidal function with a time period of one day

Figure 5.20 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 5.19: Sinusoidal function with f_1=8, f_2=15, t_o=*3q*, and t_p=*1y*.



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

Figure 5.20: Sinusoidal function with f_1=8, f_2=15, t_o=*6h*, and t_p=*1d*.

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

5.6.6 Example 3 - step-wise constant function

Figure 5.21 shows a step-wise constant function with four steps using the data as given to the right. Note that the value of the function is zero before the first defined step. If t_1=0, instead of t_1=*2h*, the first value f_1 will be used from start. Also note that the function value for the last step will sustain after this point of time.

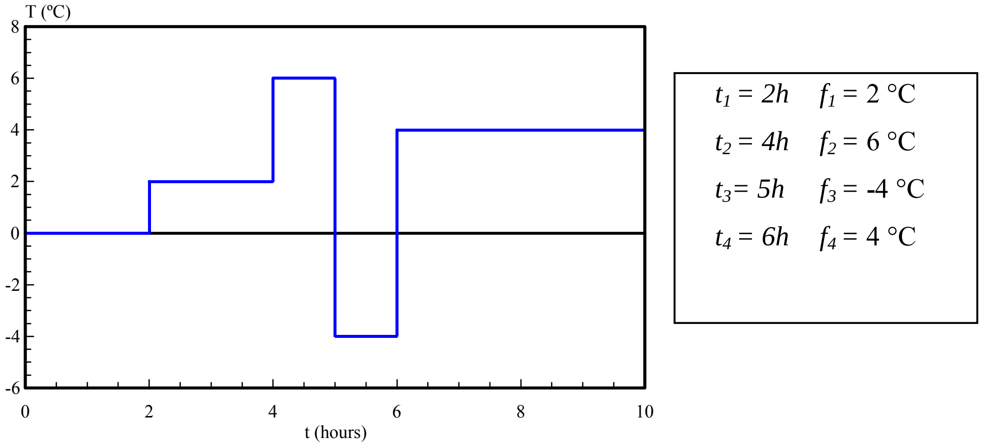


Figure 5.21: Step-wise constant function for the four steps given to the right.

5.6.7 Example 4 - step-wise linear function

Figure 5.22 shows the step-wise linear function for the four steps. Note that the function value is zero for *t*=0 (may be changed by giving a function value for t_1=0). The function value for the last step will be sustained.

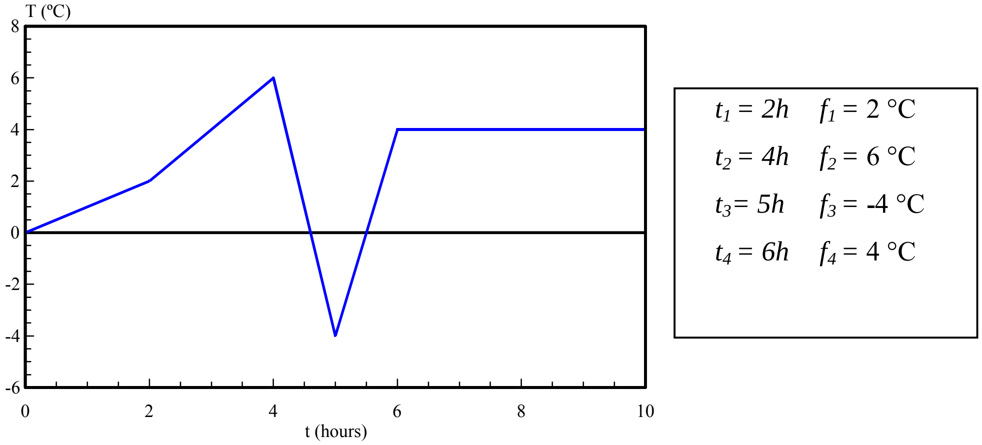
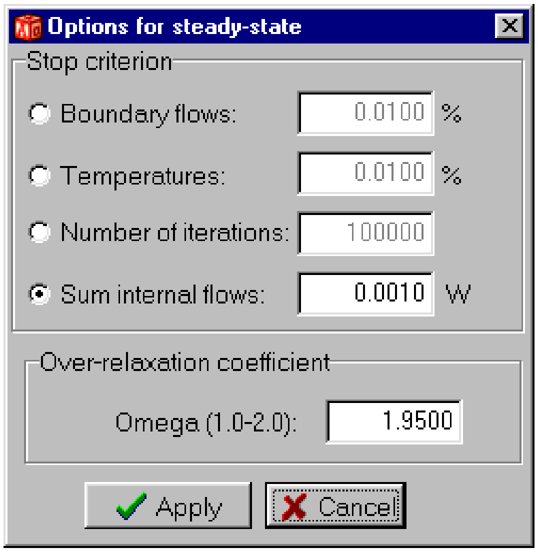


Figure 5.22: Step-wise linear function for the four steps given to the right.

5.7 Solving the problem

5.7.1 Steady-state stop criteria

There are four ways to give a stop criterion for a steady-state simulation, see Figure 5.23. 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.



Options for steady-state

Stop criterion

○ Boundary flows: 0.0100 %

○ Temperatures: 0.0100 %

○ Number of iterations: 100000

● Sum internal flows: 0.0010 W

Over-relaxation coefficient

Omega (1.0-2.0): 1.9500

Figure 5.23: Steady-state stop criteria.

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

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*:

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

Smaller values of *F* and *R* give in general smaller numerical errors, but longer computational run-time. Smaller values of *R* and *F* give in general smaller numerical errors, but longer computational run-time. Consider as an example the example *cen.dat* using 7600 cells. Table 5:2 shows the sum of the heat flows through the surfaces on the interior and the exterior sides, the required number of iterations, and the calculation time in seconds on a Pentium/233 for different values of *F* (relaxation coeff. is 1.85) The accurate choice of *F* depends on the problem. For practical cases a larger value may be sufficient. For numerical case studies a smaller value should be preferred.

F	Heat flow (W) (interior side)	Heat flow (W) (exterior side)	Iterations	CPU-time (s)
10^{-1}	57.87	60.29	27	2
10^{-2}	59.66	59.74	40	3
10^{-3}	59.60	59.67	50	4
10^{-4}	59.60	59.60	71	5
10^{-5}	59.60	59.60	80	6

Table 5:2: Results for some choices of criterion *F*.

It is normally sufficient to put *F* to values between 10^{-5} and 10^{-3}. The default value is 10^{-4}. 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 criterion *F* is calculated for all defined sets. This means that all surfaces with a nonzero heat flow must be defined when this criteria is used, and also that each surface belongs to *one* (and one only) set. The effect from optional heat sources will also be accounted for when *F* is calculated.

The third way is a maximum number of iterations.

The fourth way calculates the sum of all internal flows, i.e. the sum of all net flows flowing to each cell. This is default option. It may be a good idea to change the limit (default is 0.0010 W) to suit the problem.

5.7.2 Successive over-relaxation used for 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).

Consider the five examples in Chapter 8. The calculation time on a Pentium/233 as a function of the relaxation coefficient ω is shown in Table 5:3. Note that HEAT3 version 3 is used for these benchmarks. The number of computational cells is indicated. The stop criterion F=10^{-5}, see Section 5.7.1, is used.

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

	CPU-time (s)				
ω	*Steel1.dat* *N*=11616	*Steel2.dat* *N*=28560	*Cen.dat* *N*=7600	*Slab.dat* *N*=25221	*Cav.dat* *N*=15968
1.00	393	913	69	3300	435
1.80	45		6		
1.85	33		6		
1.90	21	102	8	195	48
1.91	19				
1.92	17				
1.93	15				
1.94	12				
1.95	10	79	14	106	36
1.96	7				
1.97	6			69	
1.98	8			45	30
1.99	11	61	54	70	28
2.00	51	53			28

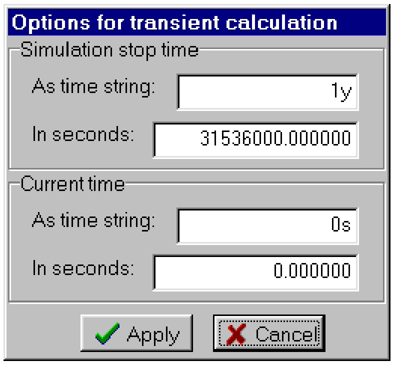
Table 5:3: CPU-time in seconds as a function of the over-relaxation coefficient.

The minimum CPU-time is underlined in the table. As shown, the method of over-relaxation is very efficient. An optimal ω gives calculation times between 1/73 and 1/12 of that required for a calculation not using over-relaxation (ω=1.0).

The optimal ω lies typically in the range 1.8-2. The default value in HEAT3 is ω=1.95. In relation to the optimal value for the examples in Table 5:3, the default value gives no more than about twice the CPU-time.

5.7.3 Simulation time for transient calculation

Simulation stop time is given in *Solve/Options for transient*, see Figure 5.24. The start time, denoted by current time, may be arbitrarily chosen.



Options for transient calculation

Simulation stop time

As time string: 1y

In seconds: 31536000.000000

Current time

As time string: 0s

In seconds: 0.000000

Figure 5.24: Menu for simulation time.

A special format for time input may be used (instead of using seconds). A sequence of pairs with a number and one of the following letters is given:

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)

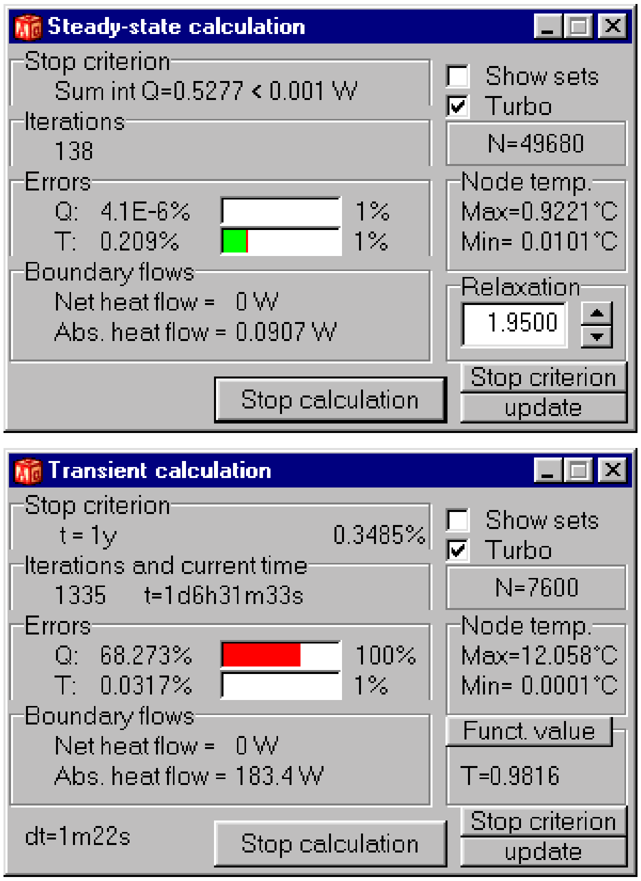
Please note that *m* is minute and *q* is month. Here are some examples:

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 below.
2d1h	This string is OK, meaning 2 days and 1 hour

5.7.4 Simulation window

An information window is shown during the simulation. Figure 5.25, shows the steady-state and the transient cases, respectively. Information is shown about stop criterion, iteration, errors, heat flows through the given sets, maximum and minimum node temperatures, and number of cells. The relaxation coefficient may be changed. There are short-cuts to *Show sets*, *Stop criterion* (see sections 5.2.1 and 5.2.3), *Screen update*. If a function is used, the current function value will be shown with a short-cut.

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.



Steady-state calculation

Stop criterion
Sum int Q=0.5277 < 0.001 W

Iterations
138

Errors
Q: 4.1E-6% 1%
T: 0.209% 1%

Boundary flows
Net heat flow = 0 W
Abs. heat flow = 0.0907 W

□ Show sets
☑ Turbo

N=49680

Node temp.
Max=0.9221°C
Min= 0.0101°C

Relaxation
1.9500

Transient calculation

Stop criterion
t = 1y 0.3485%

Iterations and current time
1335 t=1d6h31m33s

Errors
Q: 68.273% 100%
T: 0.0317% 1%

Boundary flows
Net heat flow = 0 W
Abs. heat flow = 183.4 W

dt=1m22s

□ Show sets
☑ Turbo

N=7600

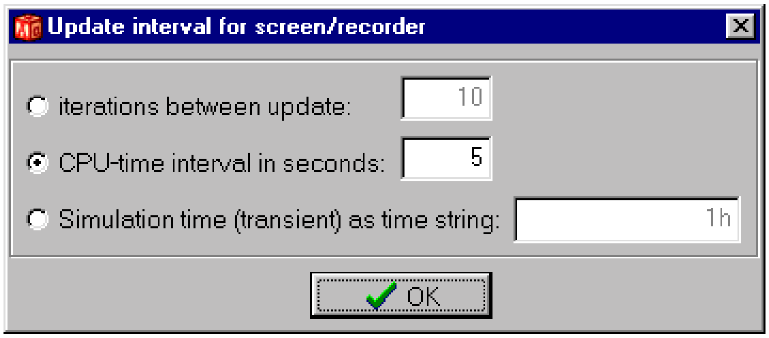
Node temp.
Max=12.058°C
Min= 0.0001°C

Funct. value
T=0.9816

Figure 5.25: Windows shown during steady-state and the transient simulations.

5.7.5 Screen update

Data shown during the calculation are updated as defined in item *Solve/Screen update*, see Figure 5.26. 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 5:th second. Note that the update interval also affects the recording interval, see Section 5.8.



Update interval for screen/recorder

○ iterations between update: 10

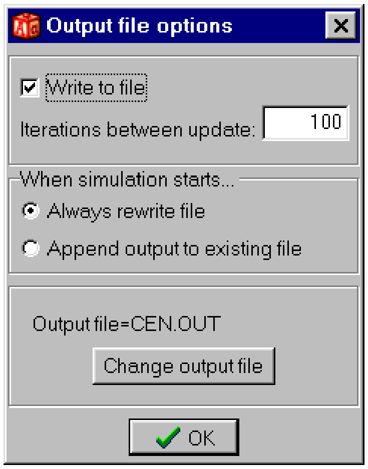
● CPU-time interval in seconds: 5

○ Simulation time (transient) as time string: 1h

Figure 5.26: Data shown during the calculation are updated as defined here.

5.7.6 Output file options

During the simulation heat flows may be written to an output file, see item *Solve/Output file options* and Figure 5.27.



Output file options

☑ Write to file

Iterations between update: 100

When simulation starts...

● Always rewrite file

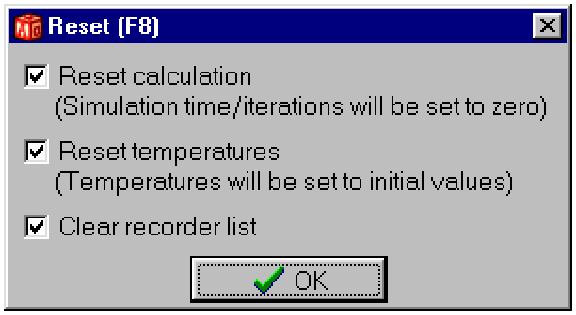
○ Append output to existing file

Output file=CEN.OUT

Figure 5.27: Output file options.

5.7.7 Reset

Figure 5.28 shows the menu item *Solve/Reset all*. There are three options to reset. The first one sets the iteration number and the current simulation time to zero. The second one initializes the temperature field. The third one clears the recorder list (will only appear if the recorder is enabled).



Reset (F8)

☑ Reset calculation
(Simulation time/iterations will be set to zero)

☑ Reset temperatures
(Temperatures will be set to initial values)

☑ Clear recorder list

Figure 5.28:

5.8 Output data - the recorder

5.8.1 Introduction

Output values may be saved to a list during the simulation. The recorder can save up to 25 different items as a function of time or number of iterations. What to be recorded is chosen from the pick list in Figure 5.30. 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 5.31, 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.

Consider Figure 5.29 that shows a cube with side lengths of 1 m. The initial temperature is zero. We want to record in time the heat flow through the upper boundary due to a a step change of temperature 1. We also want to see how the temperature in the middle of the cube changes.

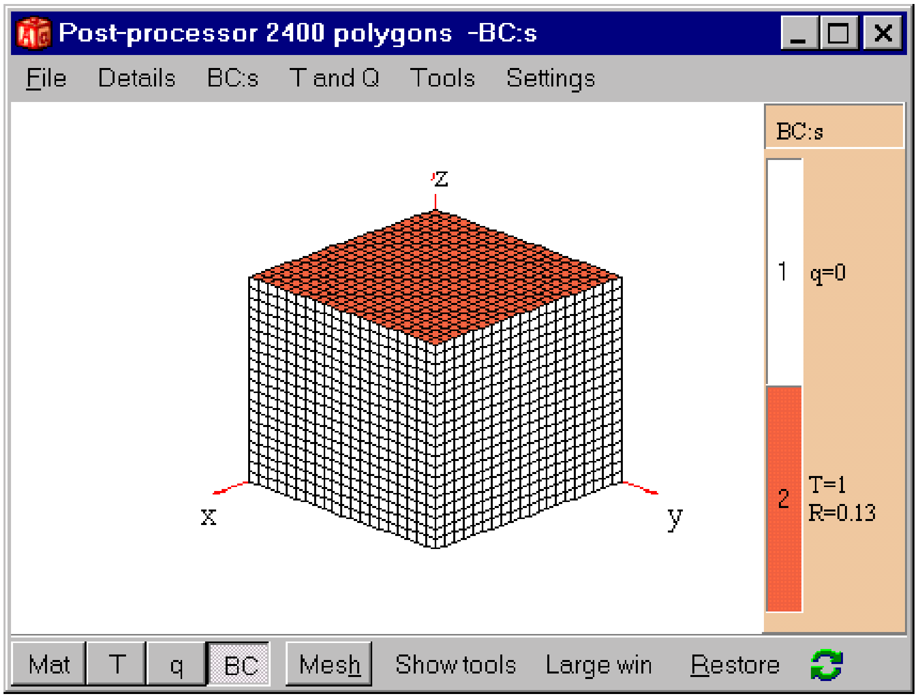
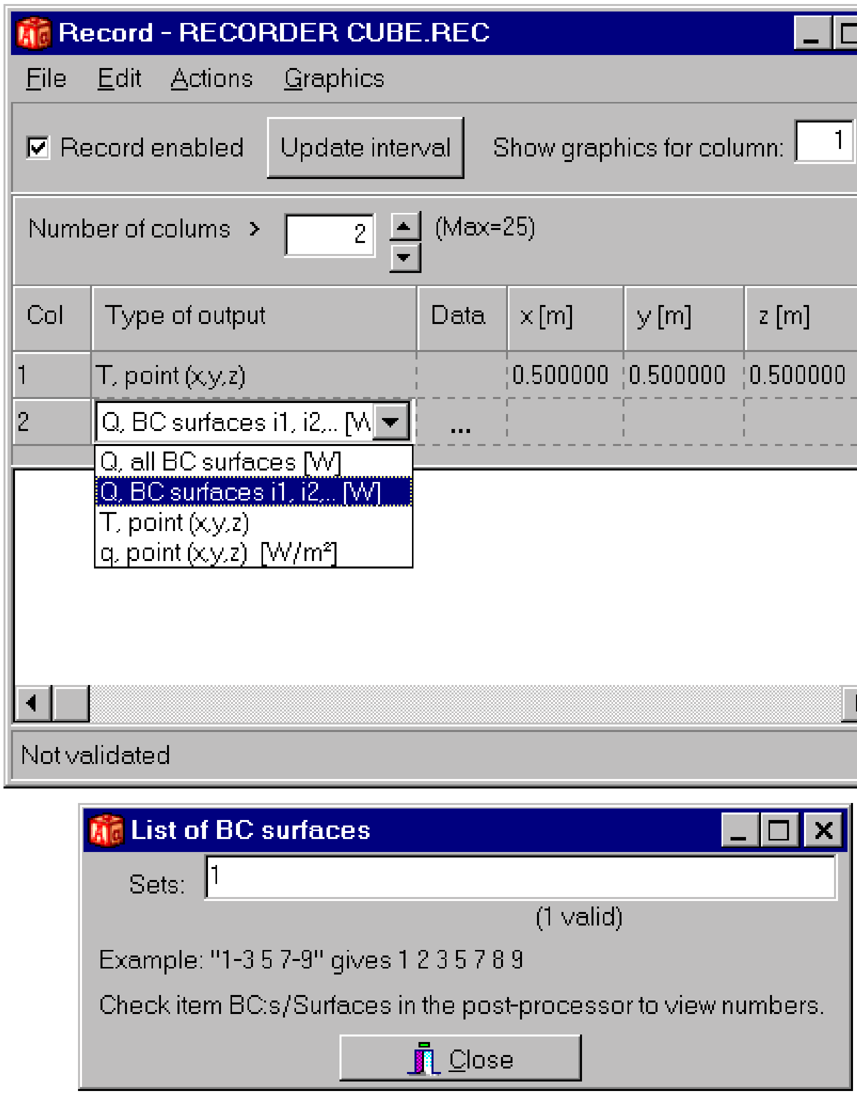


Figure 5.29: Recorder example. A cube with side lengths of 1 m.



Record - RECORDER CUBE.REC

☑ Record enabled

Show graphics for column: 1

Number of colums > 2 (Max=25)

Col	Type of output	Data	x [m]	y [m]	z [m]
1	T, point (x,y,z)		0.500000	0.500000	0.500000
2	Q, BC surfaces i1, i2... [W	...			

Not validated

List of BC surfaces

Sets: 1

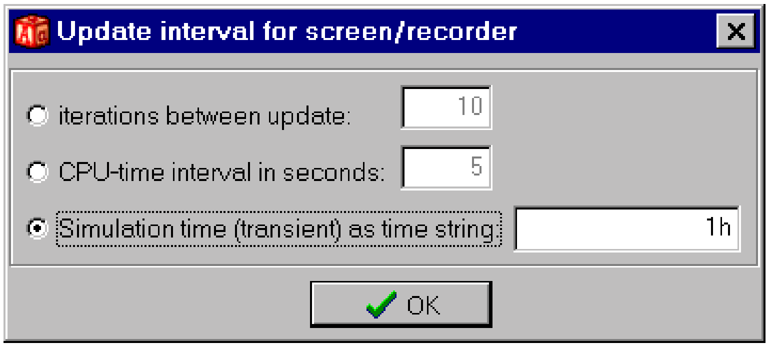
(1 valid)

Example: "1-3 5 7-9" gives 1 2 3 5 7 8 9

Check item BC:s/Surfaces in the post-processor to view numbers.

Figure 5.30: Data are recorded in columns as a function of time (transient) or iterations (steady-state) in the recorder.

Figure 5.30 shows the recorder with two columns defined. The first column will record the temperature at (*x*, *y*, *z*)=(0.5, 0.5, 0.5). The second column will record the flow through boundary 1 (upper side of cube given in the **Data** entry). Figure 5.32 shows the output with an update interval of 1 hour. Figure 5.33 and Figure 5.34 show the recorded columns (see menu item **Graphics** and menu box **Show graphics for column** in Figure 5.30).



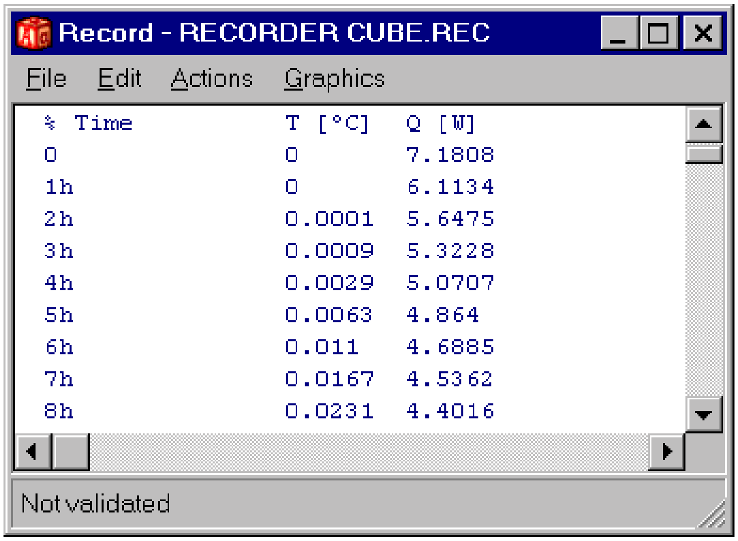
Update interval for screen/recorder

○ iterations between update: 10

○ CPU-time interval in seconds: 5

● Simulation time (transient) as time string: 1h

Figure 5.31: The time interval for recorded data is given in item *Solve/Update*.



```
% Time          T [°C]   Q [W]
0               0        7.1808
1h              0        6.1134
2h              0.0001   5.6475
3h              0.0009   5.3228
4h              0.0029   5.0707
5h              0.0063   4.864
6h              0.011    4.6885
7h              0.0167   4.5362
8h              0.0231   4.4016
```

Not validated

Figure 5.32: Two columns (one with temperatures and one with one heat flows) have been recorded with a update interval of 1 hour.

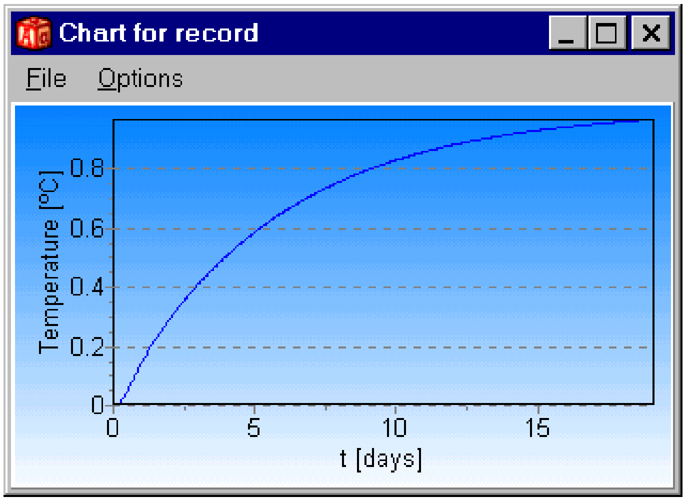


Figure 5.33: Left: Temperature in the middle of the cube (*x*, *y*, *z*)=(0.5, 0.5, 0.5).

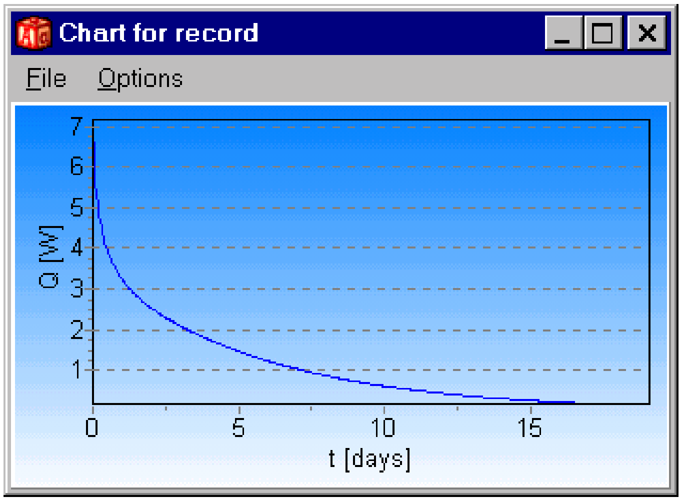


Figure 5.34: Heat flow through the upper boundary.

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 5.35 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 HEAT3 directly is also possible.

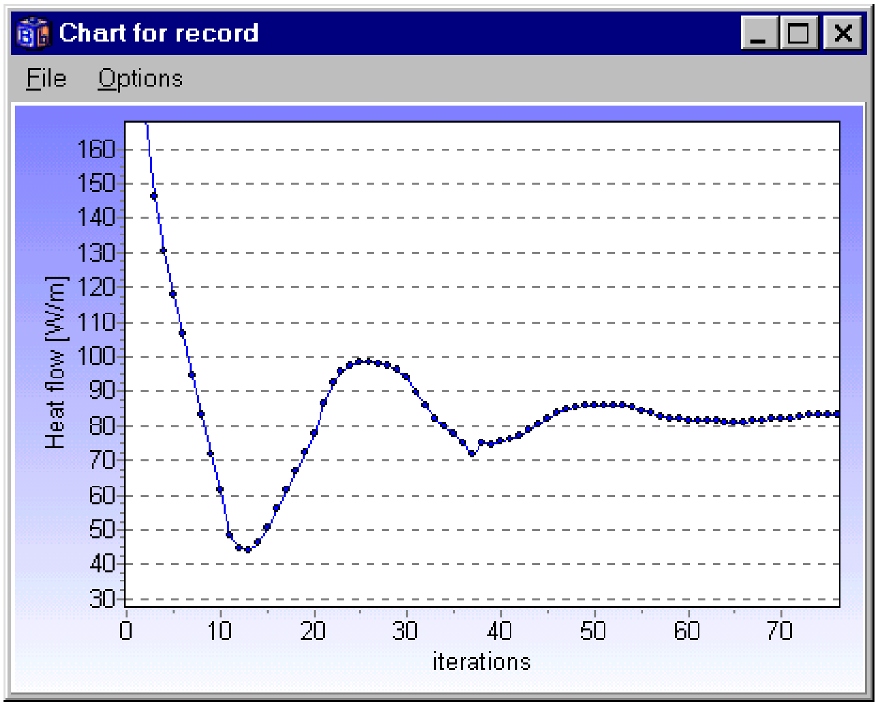


Figure 5.35: Chart showing recorded values.

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 5.8.3.

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

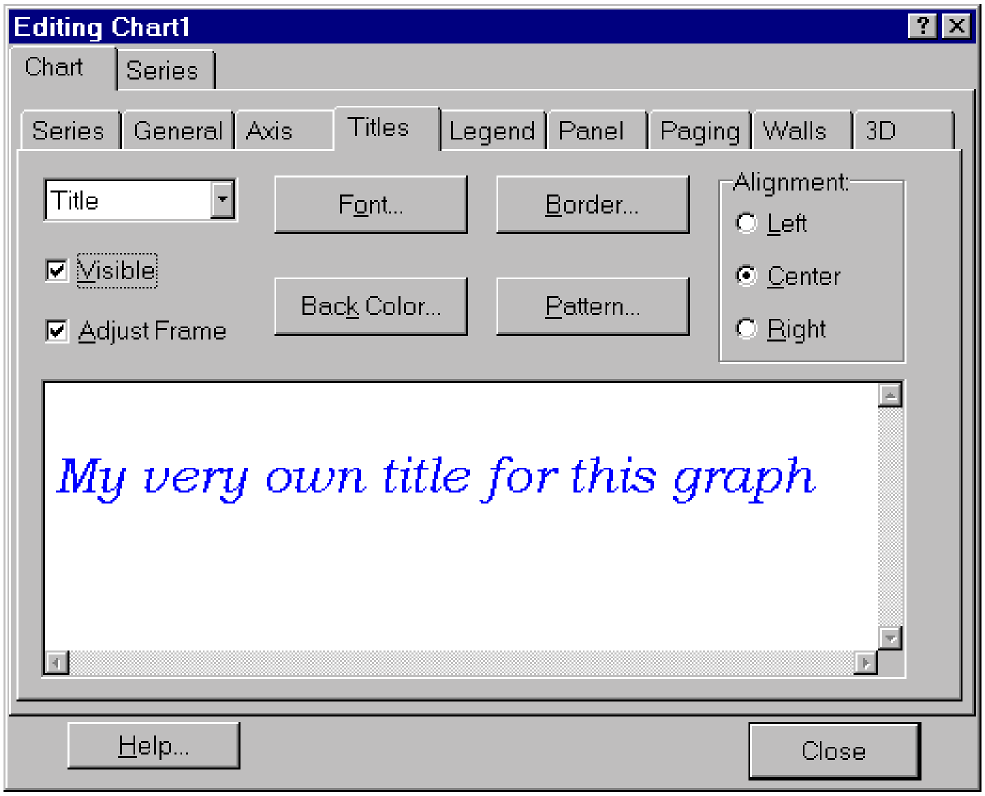
5.8.2 What can be recorded?

The pick list in Figure 5.30 shows the six options that may be recorded:

1. Q, sum of heat flows through all specified boundaries [W]
2. Q, sum of heat flows through boundaries i1, i2,.. [W]
3. T, point (x, y, z)
4. q, point (x, y,z) [W/m²]

5.8.3 Changing the record chart

The menu box **Show graphics for column** in Figure 5.30 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 5.36.



Title

☑ Visible

☑ Adjust Frame

Alignment: ○ Left ● Center ○ Right

My very own title for this graph

Figure 5.36: Chart settings may be changed in menu item **Options/Edit chart**.

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.

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

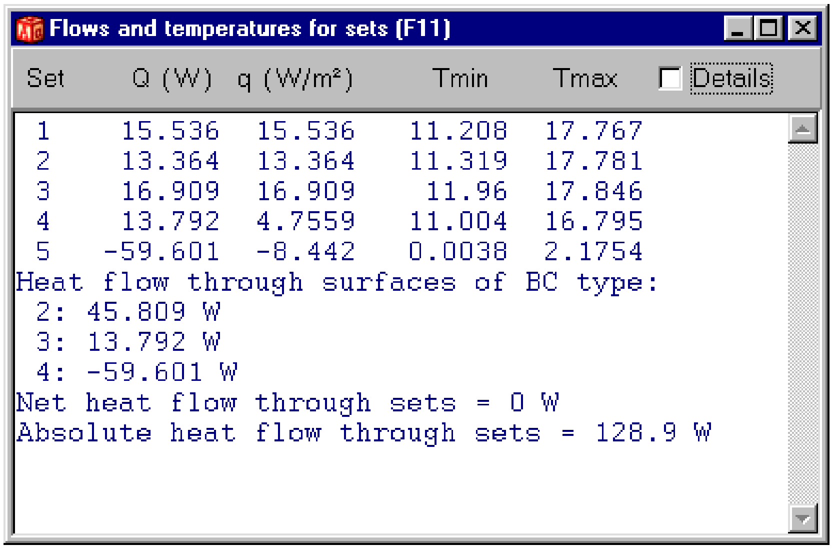
General Page - Series value format, axis association

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

5.9 Other output data

5.9.1 Flows and temperatures for sets

Item *Output/View sets* gives the heat flows for all sets, see Figure 5.37. If "**Details**" is checked, the position for the maximum and minimum temperature will also be indicated.

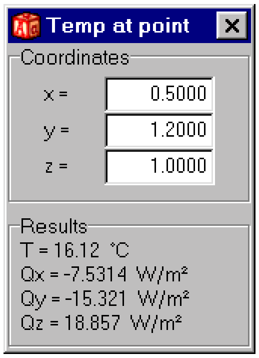


```
Set      Q (W)   q (W/m²)    Tmin     Tmax
 1     15.536   15.536    11.208   17.767
 2     13.364   13.364    11.319   17.781
 3     16.909   16.909     11.96   17.846
 4     13.792   4.7559    11.004   16.795
 5    -59.601   -8.442    0.0038   2.1754
Heat flow through surfaces of BC type:
 2: 45.809 W
 3: 13.792 W
 4: -59.601 W
Net heat flow through sets = 0 W
Absolute heat flow through sets = 128.9 W
```

Figure 5.37: Flows and temperatures for the sets.

5.9.2 Temperature and heat flows at a given point

The temperature and heat flows may be shown for a given point (*x*,*y*,*z)*, see item *Output/temp at point*, and Figure 5.38. The data are updated during the simulation, see Section 5.7.5, *Screen update*.


Temp at point

Coordinates

x = 0.5000

y = 1.2000

z = 1.0000

Results

T = 16.12 °C

Qx = -7.5314 W/m²

Qy = -15.321 W/m²

Qz = 18.857 W/m²

Figure 5.38: Temperature and heat flows shown for a given point.

5.9.3 Output files with temperature planes

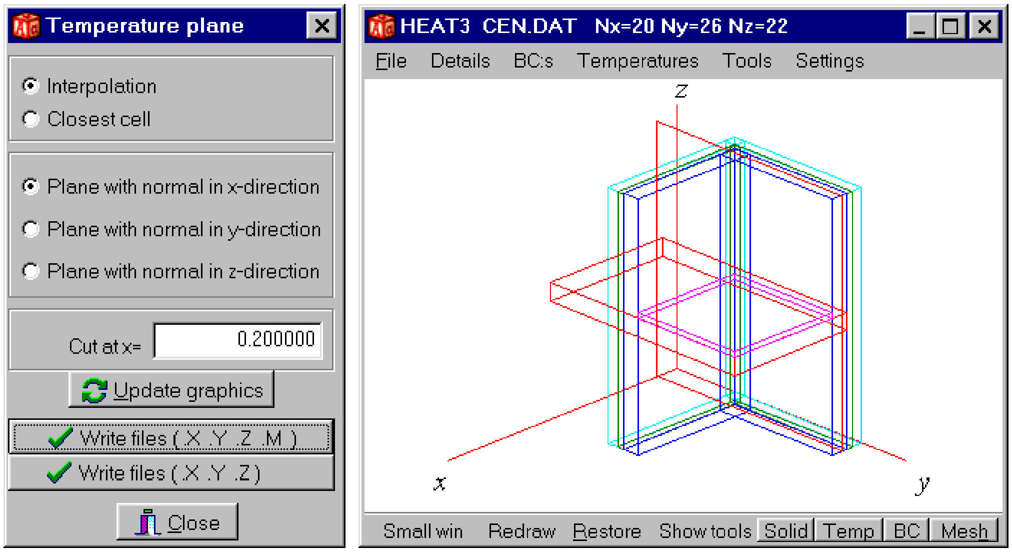
A temperature field for any plane with the normal in the *x*-, *y*-, or *z*-direction may be written to a file, see item *Output/Write temp plane* and Figure 5.39. Three files will be produced with the extensions *filename.x*, *filename.y*, and *filename.z*. The .z file contains the temperature of each cell (a matrix with $m{\cdot}n$ values). The files .x and .y hold the *x*- and *y*-positions of the cells counted from the origin (each has an array with *m* and *n* values, respectively).

The plane is defined by giving the normal direction of the plane and the distance *D* from the origin. Table 5:4 shows the data in the files .x .y and .z. Consider the last case with the *z*-direction. This means that the temperatures of the *x,y*-plane at the points (x_i,y_j) at the distance *z=D* from the origin are written to *filename.z*. The distances for each cell in the *x*-direction from the origin ($x_1..x_{Nx}$) are written to *filename.x*. The distances in the y-direction ($y_1..y_{Ny}$) are written to *filename.y*.

Plane with normal in	Data in file .x	Data in file .y	Data in file .z
x-dir	$y_1..y_{Ny}$	$z_1..z_{Nz}$	*T(D,y,z)*
y-dir	$z_1..z_{Nz}$	$x_1..x_{Nx}$	*T(x,D,z)*
z-dir	$x_1..x_{Nx}$	$y_1..y_{Ny}$	*T(x,y,D)*

Table 5:4: Data in the created files with extension .x, .y, and .z.

A second order interpolation of the temperatures in the three directions is used. As an option during the simulation, the value *D* may automatically be rounded to cut a plane leveled at the center of the vertically closest numerical cell. In this case, the temperatures in the nearest cells will be written without any interpolation.



Temperature plane

● Interpolation
○ Closest cell

● Plane with normal in x-direction
○ Plane with normal in y-direction
○ Plane with normal in z-direction

Cut at x= 0.200000

Figure 5.39: Temperature field for any plane may be written to a file. The plane (leveled at *x*=0.2) is indicated in red in the right figure.

As an option (*write files (.X .Y .Z .M)*), a Matlab script file may be written to file *filename.m*. The script (given below) will load the temperature field and the coordinates, and draw a mesh plot, see Figure 5.40.

```
figure
load Filename.z;
z=Filename;
load Filename.x;
x=Filename;
load Filename.y;
y=Filename;
mesh(x,y,z)
```

The axes in Figure 5.40 are labeled in Matlab using the following command:

```
xlabel('y (m)')
ylabel('z (m)')
zlabel('T (ºC)')
```

If the plane contains cells that are not part of the computational area, the word 'NAN' will be written to *filename.z* instead of a temperature $T_{i,j,k}$. The word 'NAN' is recognized by Matlab as a 'not available number', and no grid points will be drawn in the picture.

Appendix B shows a few examples of Matlab graphic functions.

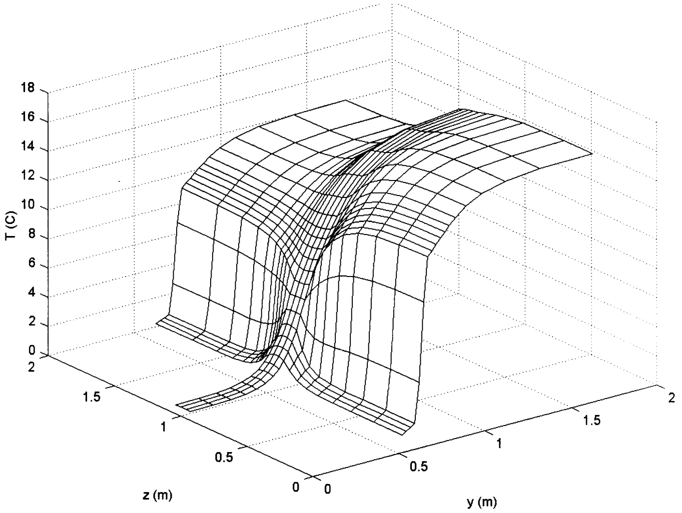
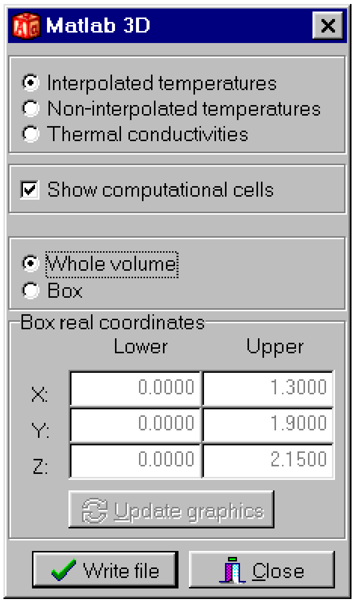


Figure 5.40: Surface plot of a temperature plane.

5.9.4 Output files for Matlab 3D figures

Before version 3.0 of HEAT3, Matlab was needed to produce pictures in 3D. Matlab is now more or less obsolete since HEAT3 manages most of this itself. However, some users may still want to use Matlab (there are e.g. optional isothermal plots, imaging routines such as shading, light sources, etc). Figure 5.41 shows the options.


Matlab 3D

● Interpolated temperatures
○ Non-interpolated temperatures
○ Thermal conductivities

☑ Show computational cells

● Whole volume
○ Box

Box real coordinates

	Lower	Upper
X:	0.0000	1.3000
Y:	0.0000	1.9000
Z:	0.0000	2.1500

Figure 5.41: Menu for Matlab 3D figures.

HEAT3 generates a Matlab script file (’m-file’) containing all commands and data to draw e.g. surface temperatures (see e.g. the front cover). It is also possible to view thermal conductivities (indicated in colors or in gray-scale). The figures may be treated according to Matlab instructions such as changing the point of view, the color map, zoom, two- or three-dimensional perspective, etc.

The surface temperature within the cells may be interpolated as an option to generate smoother figures. Computational cells may also be displayed. The whole problem, or just a part of it, may be chosen. The latter allows a box to be cut out (showing the internal temperatures). The box is defined by its lower and upper real coordinates, respectively. Note that the volume will be extended to include whole numerical cells that are divided by the borders of the box.

Note that Matlab version 4.0 is required. Versions from 4.2 are able to display a color bar.

The 'm-files' can be large (sometimes more than 1 Mb). If you temporarily do not use the files it may be a good idea to compress them with e.g. WinZip (often to 5-10 % of the original size).

5.10 Saving conductances, capacities, and temperatures to file

Conductances and capacities for each cell may be written to file (*Output/Write Cond & Cap & T*). The format is shown below. The first three rows show the maximum number of computational nodes in each direction. The fourth row shows the total number of computational cells N. The fifth row shows a headline. The next *N* rows show the indices *i,j,k* for the cell, the heat capacity (J/K), and the thermal conductances (W/K) for cell *i,j,k*, see Figure 3.2. Only cells part of the computational volume are listed. Note that six conductances are written for each cell. This means that the same conductance is written twice for internal cells.

```
10 =Nx
10 =Ny
10 =Nz
1000 =Number of computational cells
i j k  C(i,j,k)  K(i-1/2,j,k) K(i+1/2,j,k) K(i,j-1/2,k) K(i,j+1/2,k) K(i,j,k-1/2) K(i,j,k+1/2) T(i,j,k)
1 1 1 1000 0.2 0.1 0.2 0.1 0.2 0.1 0.0002
1 1 2 1000 0.2 0.1 0.2 0.1 0.1 0.1 0.0007
1 1 3 1000 0.2 0.1 0.2 0.1 0.1 0.1 0.0014
…
…
…
10 10 9 1000 0.1 0.2 0.1 0.2 0.1 0.1 0.0857
10 10 10 1000 0.1 0.2 0.1 0.2 0.1 0.2 0.3331
```

5.11 Post-processor

5.11.1 Introduction

The "post-processor window", see Figure 5.42, shows surface temperatures and heat flows in 3D. The picture may be rotated, moved, or zoomed (the mathematical routines for this are taken from (Foley et. Al., 1984)). Visualized data may be temperatures, materials, numerical mesh, or boundary conditions.

The graphic window will be updated when new input data are given. The temperatures may be updated during the simulation (item *Temperatures/Update figure on solve*). Colors and fonts may be changed in item *Settings*.

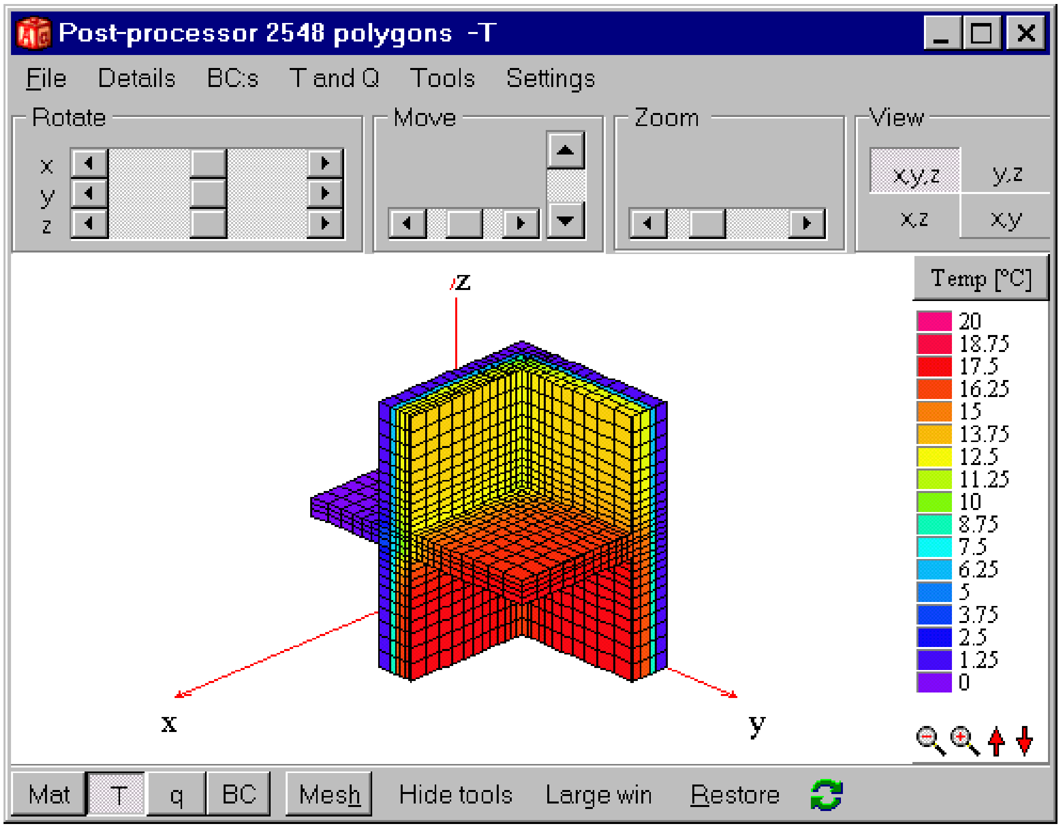
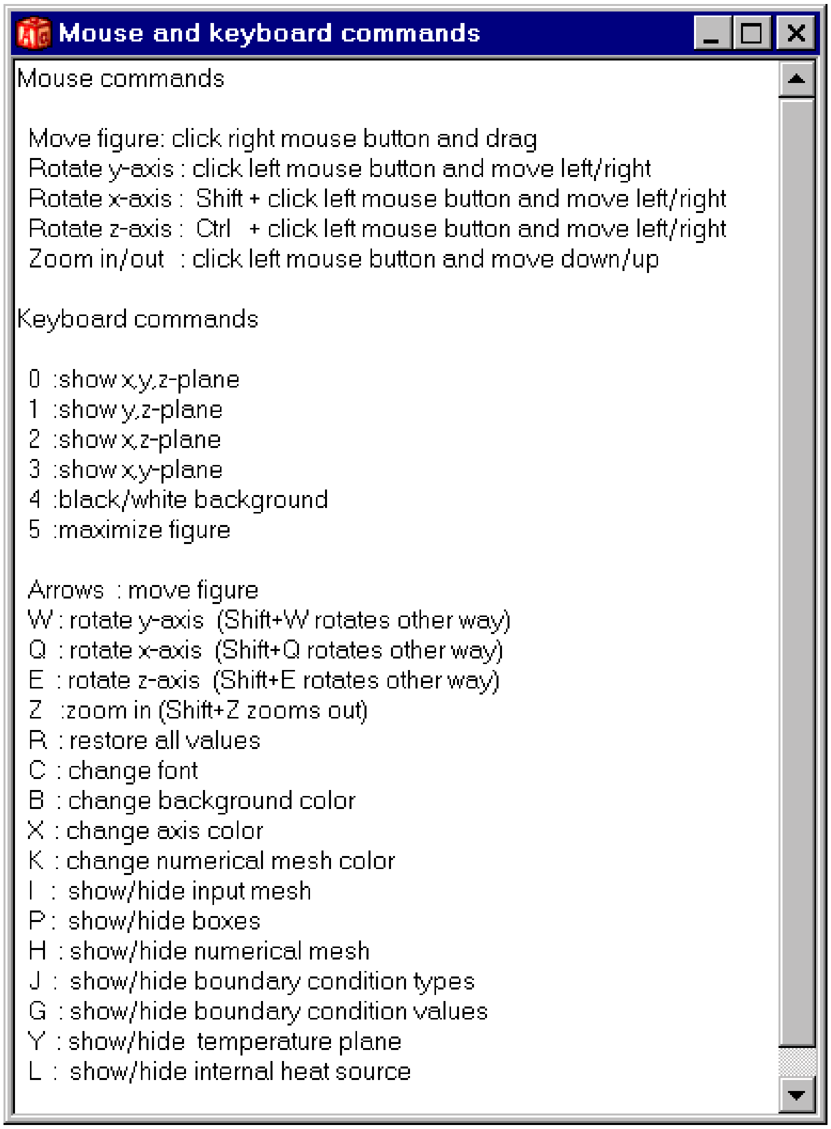


Figure 5.42: The post-processor window.

5.11.2 Basic commands

The picture may be rotated, moved, or zoomed using the top tool-bar, the mouse, or directly from the keyboard, see Figure 5.43 for mouse and keyboard commands.



```
Mouse commands

 Move figure: click right mouse button and drag
 Rotate y-axis : click left mouse button and move left/right
 Rotate x-axis :  Shift + click left mouse button and move left/right
 Rotate z-axis :  Ctrl   + click left mouse button and move left/right
 Zoom in/out   : click left mouse button and move down/up

Keyboard commands

 0  :show x,y,z-plane
 1  :show y,z-plane
 2  :show x,z-plane
 3  :show x,y-plane
 4  :black/white background
 5  :maximize figure

 Arrows  : move figure
 W : rotate y-axis  (Shift+W rotates other way)
 Q  : rotate x-axis  (Shift+Q rotates other way)
 E  : rotate z-axis  (Shift+E rotates other way)
 Z   :zoom in (Shift+Z zooms out)
 R  : restore all values
 C  : change font
 B  : change background color
 X  : change axis color
 K  : change numerical mesh color
 I   :  show/hide input mesh
 P :  show/hide boxes
 H  : show/hide numerical mesh
 J  :  show/hide boundary condition types
 G  : show/hide boundary condition values
 Y  : show/hide  temperature plane
 L  :  show/hide internal heat source
```

Figure 5.43: Shortcuts for basic commands.

The list in Figure 5.43 is displayed by item *Tools/Help shortcuts*.

5.11.3 Menu item *Details*

Figure 5.44 shows the menu item *Details* with its option. As an example, input mesh coordinates are marked in the picture.

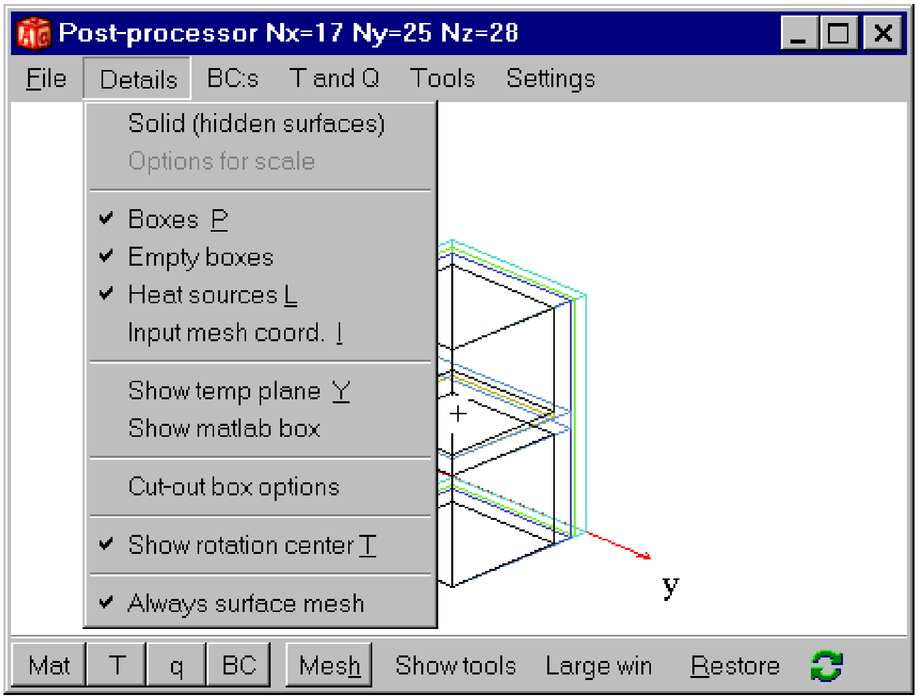
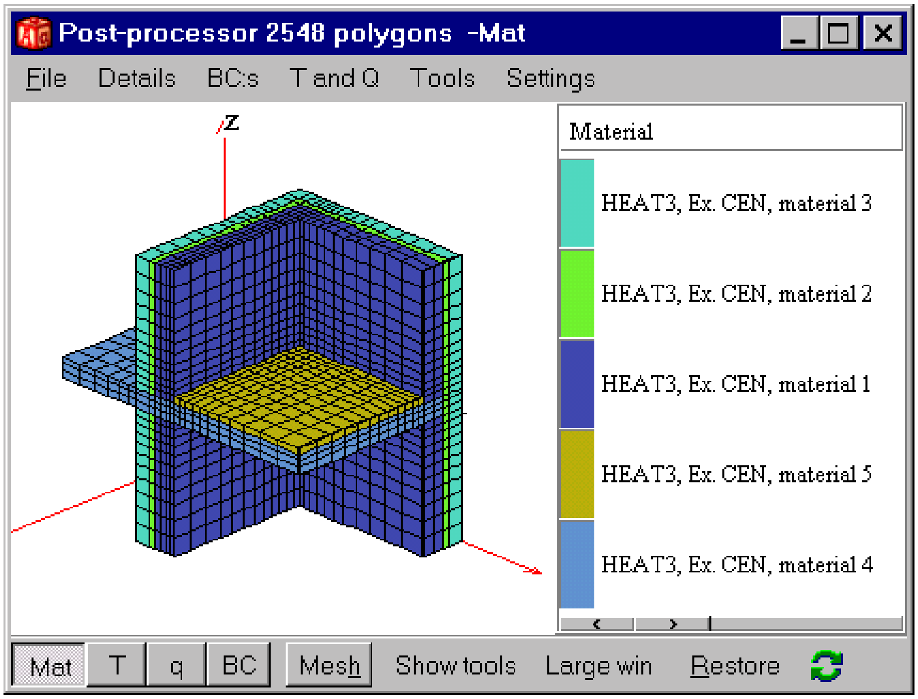


Figure 5.44: menu item *Details*.

5.11.4 Viewing thermal conductivities and heat capacities

Click "**Mat**" on the bottom tool bar (or check item *Details/Solid*) to view materials names, see Figure 5.45, top.

Clicking the scale area to the right (or *Details/Options for Scale*) brings up the window shown in Figure 5.46. Figure 5.45, bottom, shows the thermal conductivities.


Post-processor 2548 polygons -Mat
File
Details
BC:s
T and Q
Tools
Settings
z
Material
HEAT3, Ex. CEN, material 3
HEAT3, Ex. CEN, material 2
HEAT3, Ex. CEN, material 1
HEAT3, Ex. CEN, material 5
HEAT3, Ex. CEN, material 4
Mat
T
q
BC
Mesh
Show tools
Large win
Restore

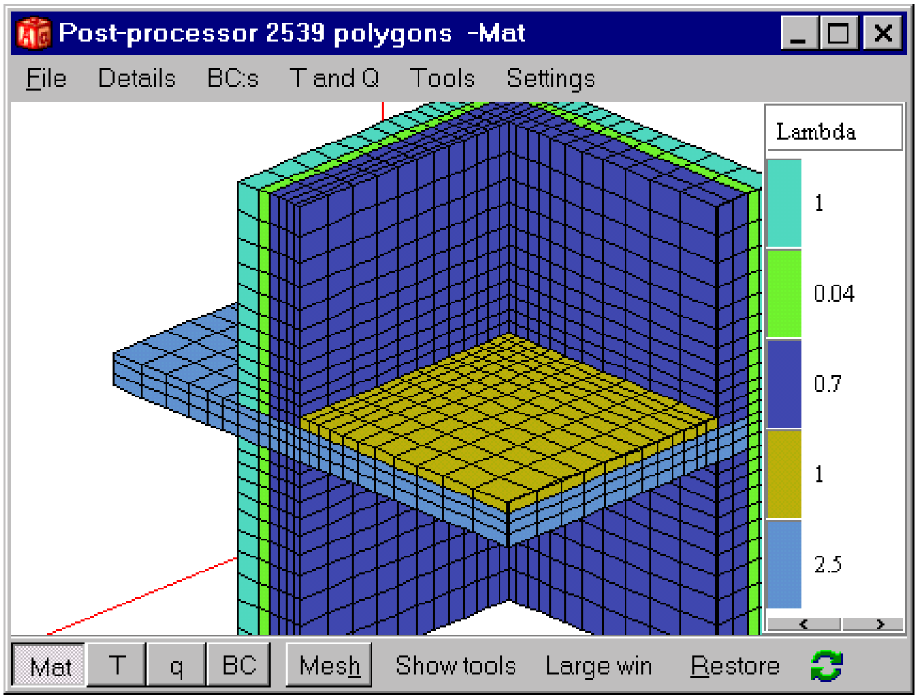
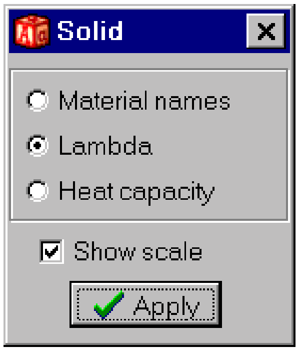


Figure 5.45: Material names (top) and thermal conductivities (bottom).



○ Material names
● Lambda
○ Heat capacity

☑ Show scale

Figure 5.46: Click the scale bar (or *Details/Options for Scale*) for options.

5.11.5 Viewing boundary conditions

Figure 5.47 shows the menu item BC:s with its option. The picture and the scale indicate the specified boundary conditions.

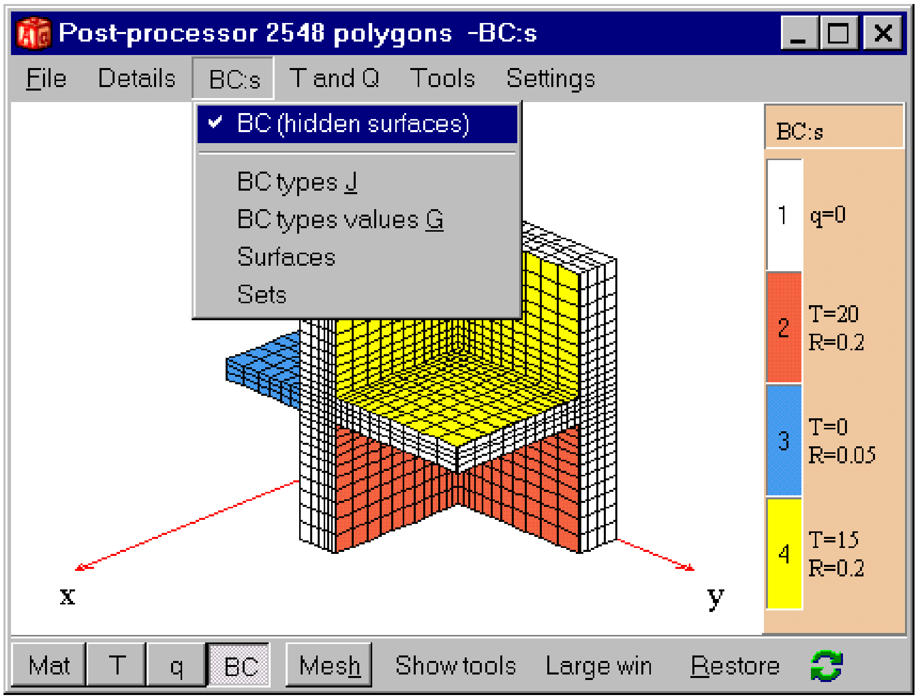
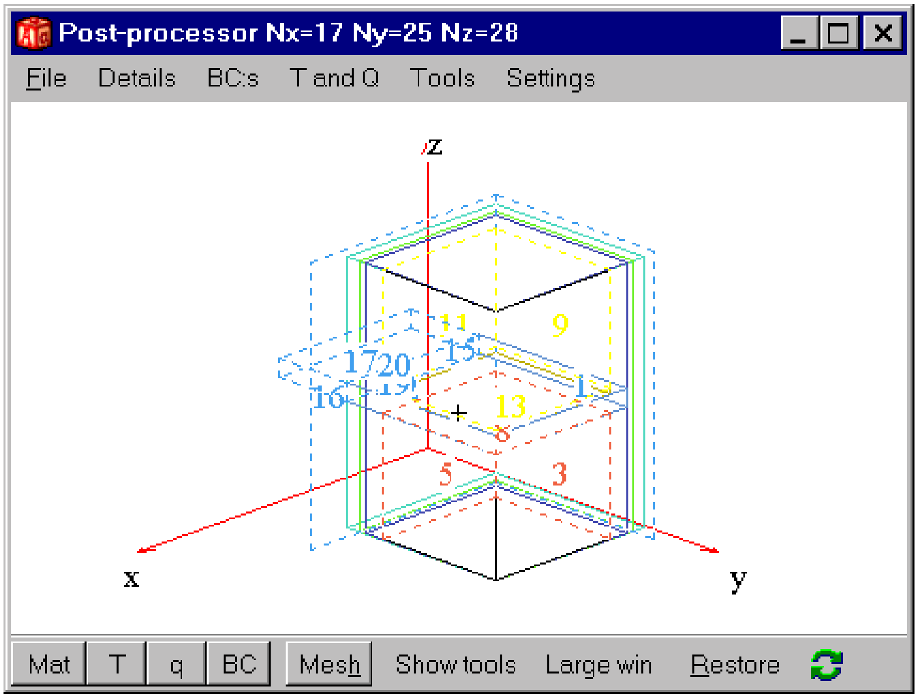


Figure 5.47: Boundary conditions.

Figure 5.48 shows the surface numbers (item *BC:s/Surface*) in 3D and 2D. In the same way, defined sets, and boundary condition values may be drawn.



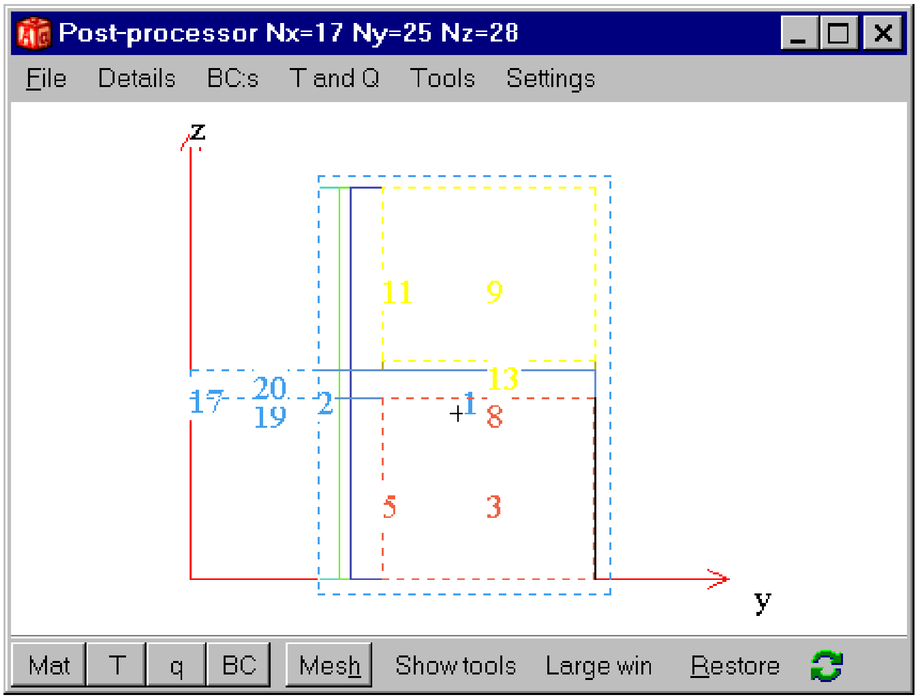


Figure 5.48: Surface numbers (item *BC:s/Surface*) in 3D and 2D.

5.11.6 Viewing temperatures and heat flows

Figure 5.56 shows the *Temperatures* menu. The picture shows the temperatures in gray. Click the scale bar (or *Temperatures/options for scale*) for options, see Figure 5.50.

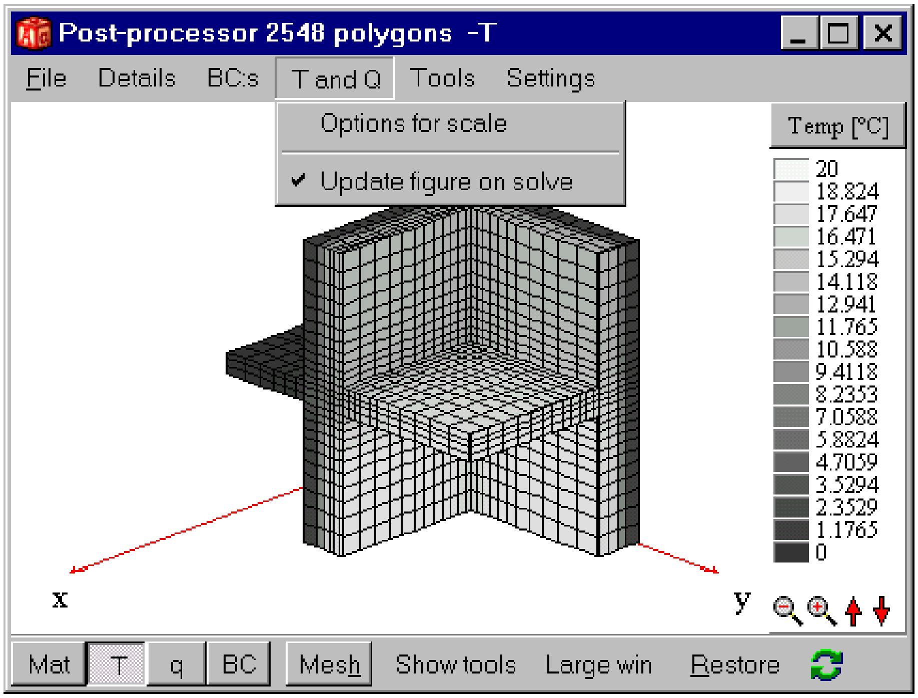
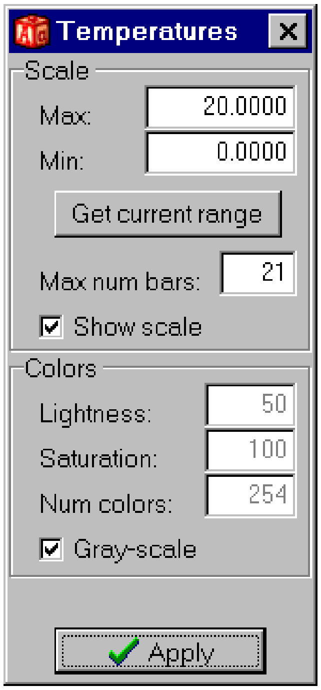


Figure 5.49: Temperatures in gray-scale.



Temperatures

Scale

Max: 20.0000

Min: 0.0000

Max num bars: 21

☑ Show scale

Colors

Lightness: 50

Saturation: 100

Num colors: 254

☑ Gray-scale

Figure 5.50: Options for temperature scale.

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 5.51. 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 in HEAT3 from magenta to blue using a certain number of colors (**Num colors**).

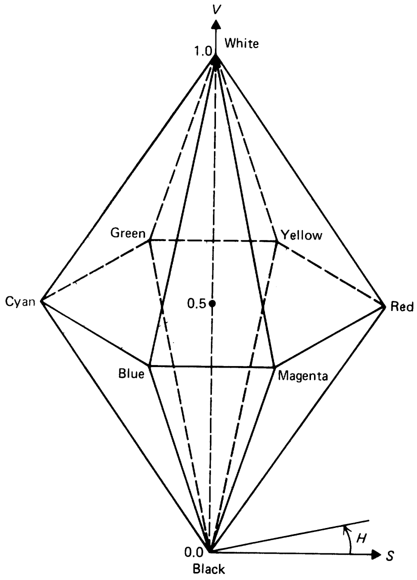


Figure 5.51: The hue-lightness-saturation color model.

The numerical mesh (without temperatures) may be viewed by putting the lightness to 1, see Figure 5.52.

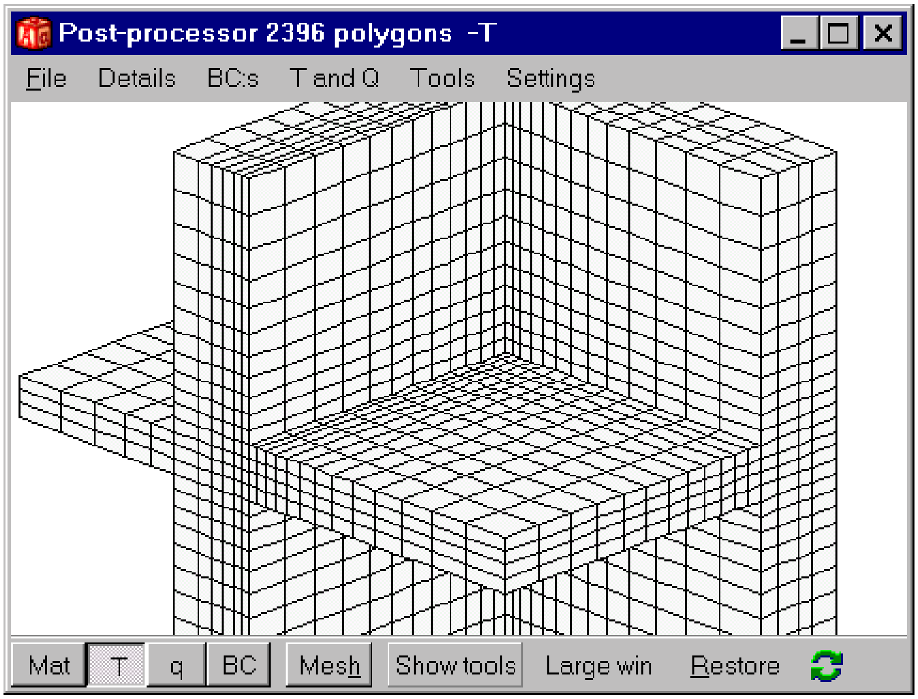


Figure 5.52: Numerical mesh (without colors) when lightness is set to 100 %.

A certain temperature interval may be disabled by clicking the temperature scale. Figure 5.53 shows two disabled temperatures, *T*=18 and *T*=11 ºC. This means that the surfaces with temperatures between 10.5-11.5 and 17.5-18.5 will not be drawn (the temperature step is 1 in this case). A click with the left mouse button on the scale will enable/disable one interval. A click with the *right* mouse button will enable/disable all intervals.

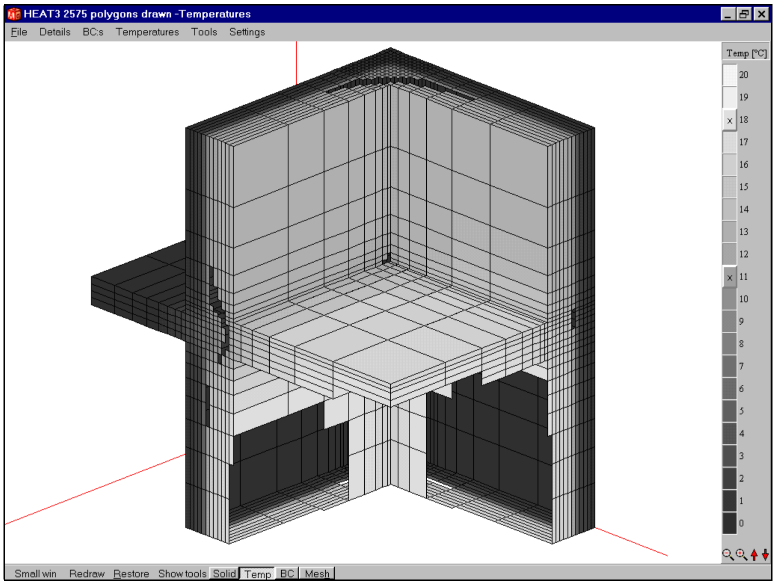
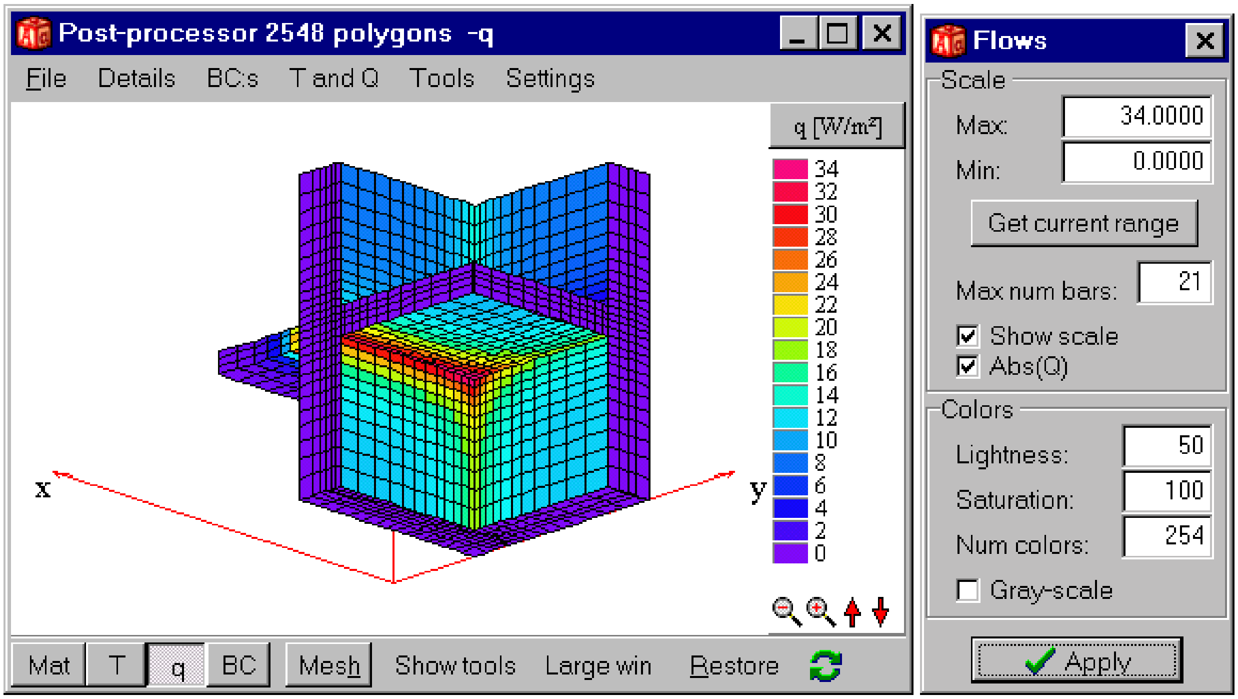


Figure 5.53: Temperature intervals may be hidden by pressing the scale color.

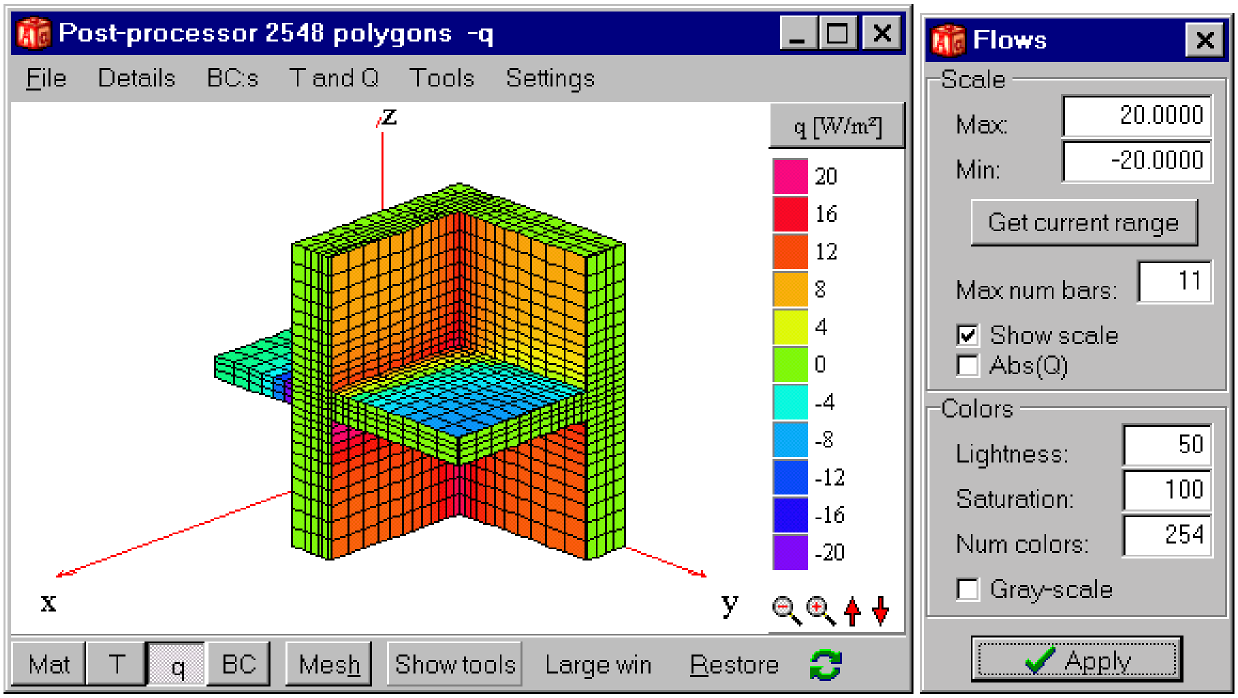
The picture is updated (unless *Temperatures/Update figure on solve* is disabled) during the simulation at time interval given in *Solve/Screen update*.

Figure 5.54 and Figure 5.55 shows pictures of heat flows. Note that the heat flow shown has its direction perpendicular to the cell surface.



Max: 34.0000
Min: 0.0000
Max num bars: 21
☑ Show scale
☑ Abs(Q)
Lightness: 50
Saturation: 100
Num colors: 254
□ Gray-scale

Figure 5.54: Heat flows (tool bar button 'q') shows quickly where leakage occur. In this case the absolute heat flows are shown, see menu option Abs(Q) to the left.



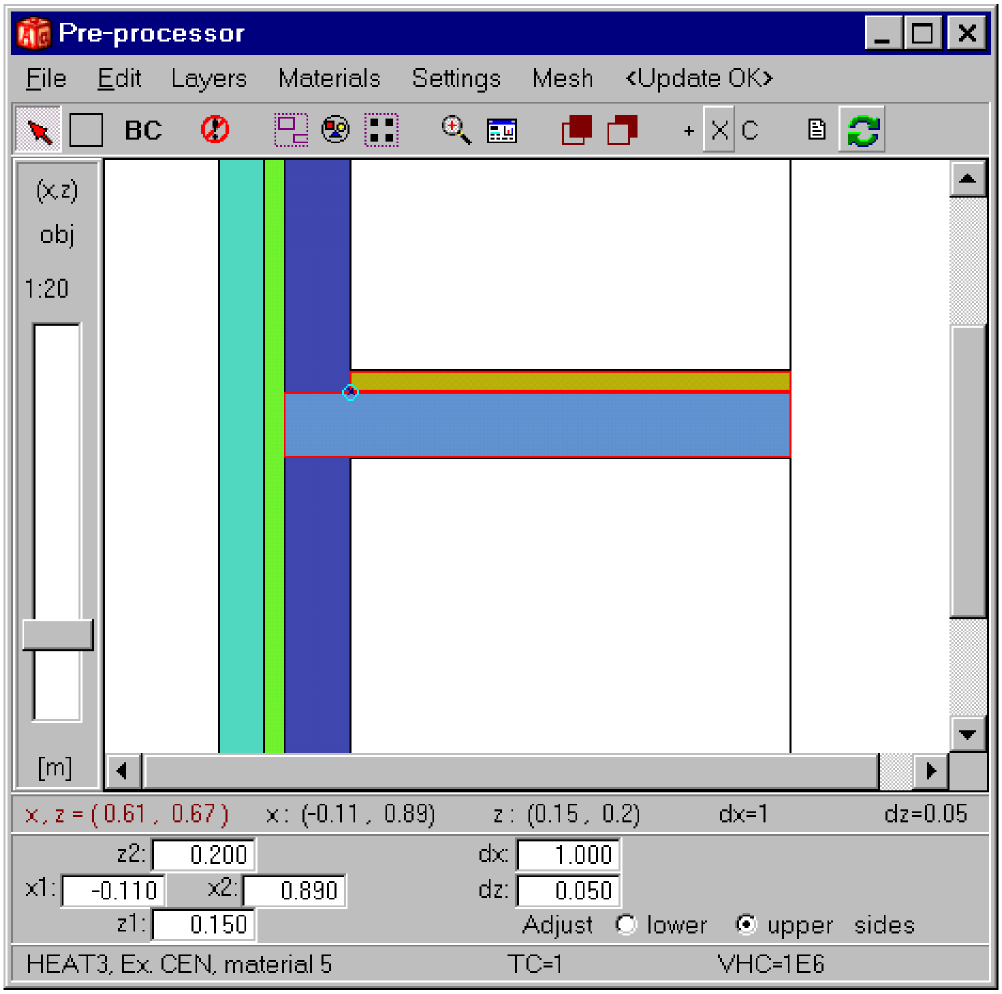
Max: 20.0000
Min: -20.0000
Max num bars: 11
☑ Show scale
□ Abs(Q)
Lightness: 50
Saturation: 100
Num colors: 254
□ Gray-scale

Figure 5.55: Heat flows shown (menu option Abs(Q) disabled to the left). Here all flows entering material are positive (color green to red).

5.11.7 Removing material boxes in the picture

Material boxes can be made invisible in the pre-processor (even though they are accounted for in the calculations). Figure 5.56, left, shows two selected material boxes for the floor that are med invisible using the object inspector (menu tool bar “obj”). The post-processor shows the calculated temperature field. Note that we see the internal temperatures within the material interfaces between visible and invisible boxes.

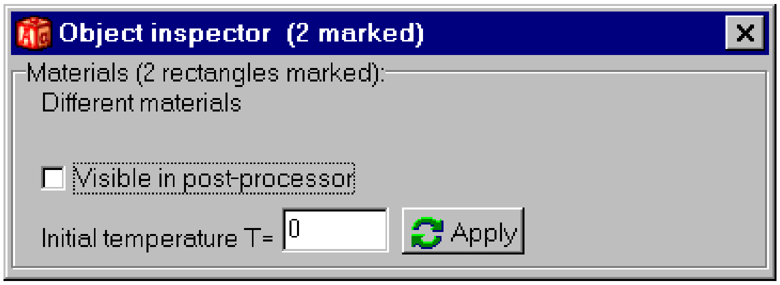
Note for text input using the text editor only: The defined boxes may be coded by a color (an integer 0-10) in the input editor. A box with color 0 will be invisible in the graphic window (even though it is accounted for in the calculations).



x, z = (0.61 , 0.67)   x : (-0.11 , 0.89)   z : (0.15 , 0.2)   dx=1   dz=0.05

z2: 0.200
x1: -0.110   x2: 0.890   dx: 1.000
z1: 0.150   dz: 0.050
Adjust ○ lower ● upper sides

HEAT3, Ex. CEN, material 5   TC=1   VHC=1E6



Materials (2 rectangles marked):
Different materials

□ Visible in post-processor

Initial temperature T= 0

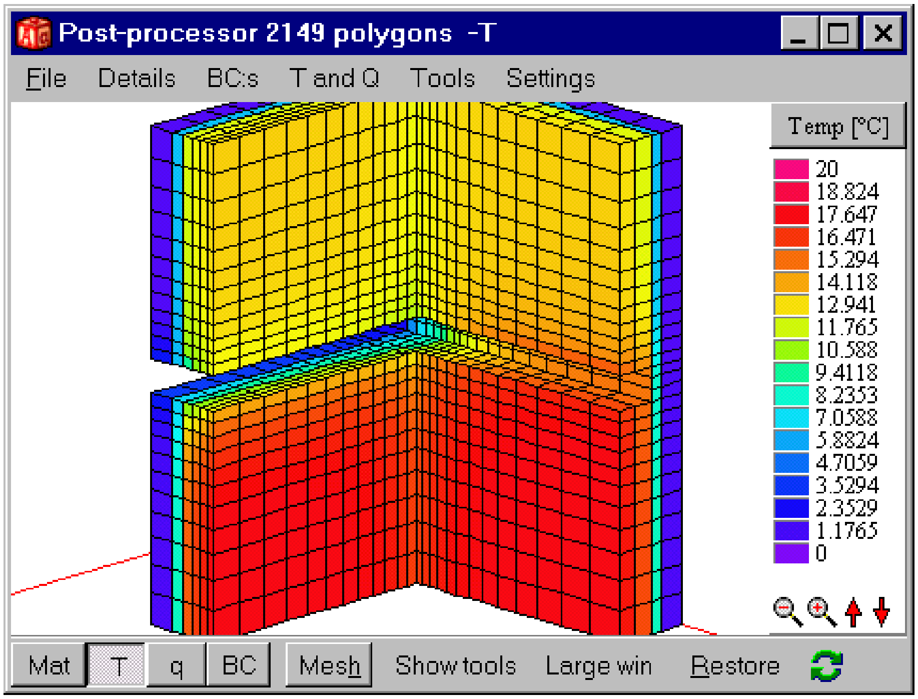
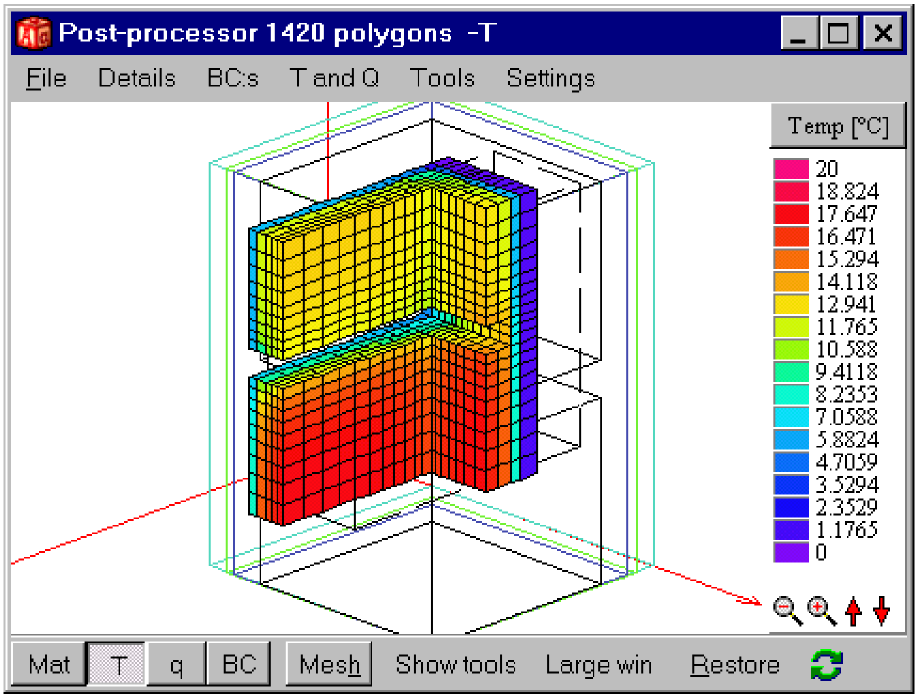
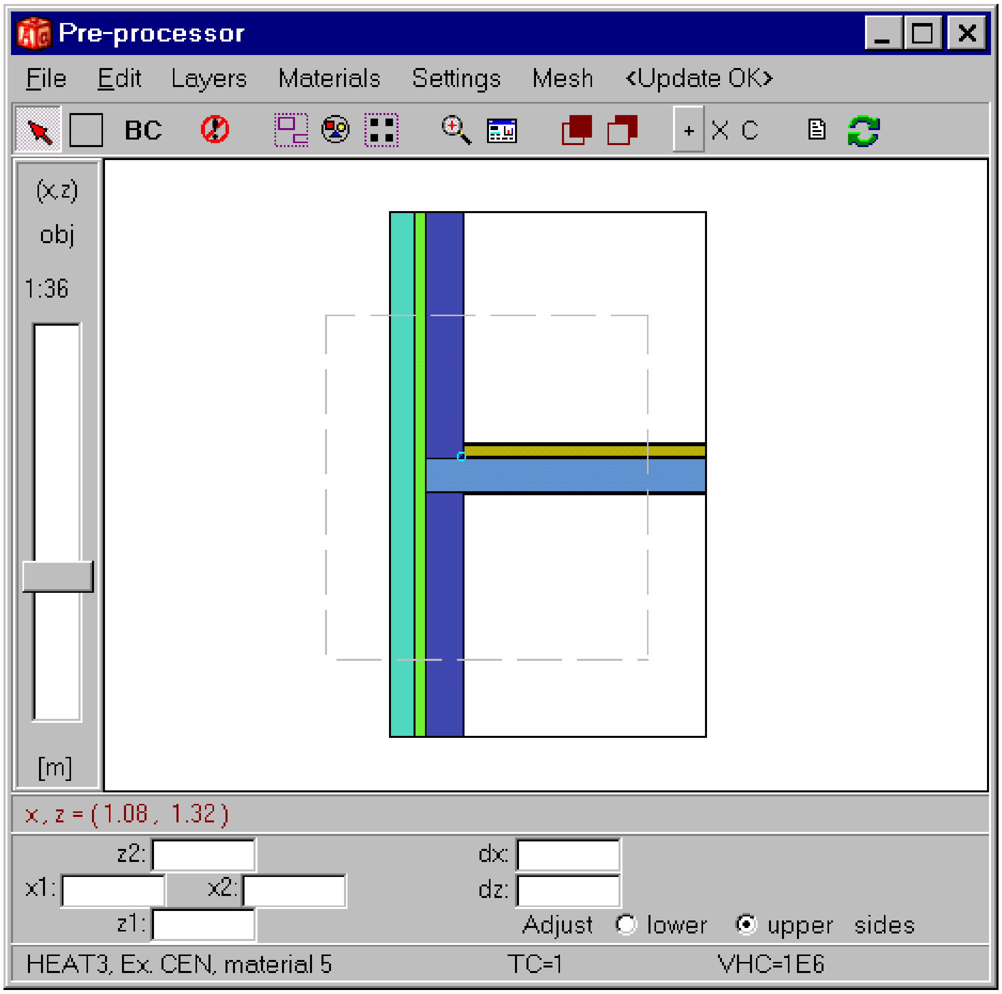


Figure 5.56: Boxes may be graphically discarded (the floor is invisible here).

5.11.8 Cut-out areas

Sub-volumes (boxes) may be cut out by using menu item "C" on the top tool bar in the pre-processor, see Figure 5.57. All material that is inside the box will be drawn.





x, z = (1.08 , 1.32)

Adjust ○ lower ● upper sides

HEAT3, Ex. CEN, material 5 TC=1 VHC=1E6

Figure 5.57: A part of the problem may be viewed by defining a box. The cut-out box is drawn using menu item “C” on the top tool bar in the pre-processor. It is shown as a light gray line in the pre-processor.

5.11.9 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*).

5.11.10 Maximizing scales

It is possible to ’maximize’ the area (item *Tools/maximize*). This means that the scale in each of the three directions will be adjusted to the maximum length in its own direction. As a result, the picture will be stretched in one or two directions. In some cases, this will make it easier to see details without having to enlarge areas, see Figure 5.58. If no ’maximizing’ is chosen, the scale will be adjusted to the largest length of the three directions, see Figure 5.59. In this case, the scale will be the same in all directions.

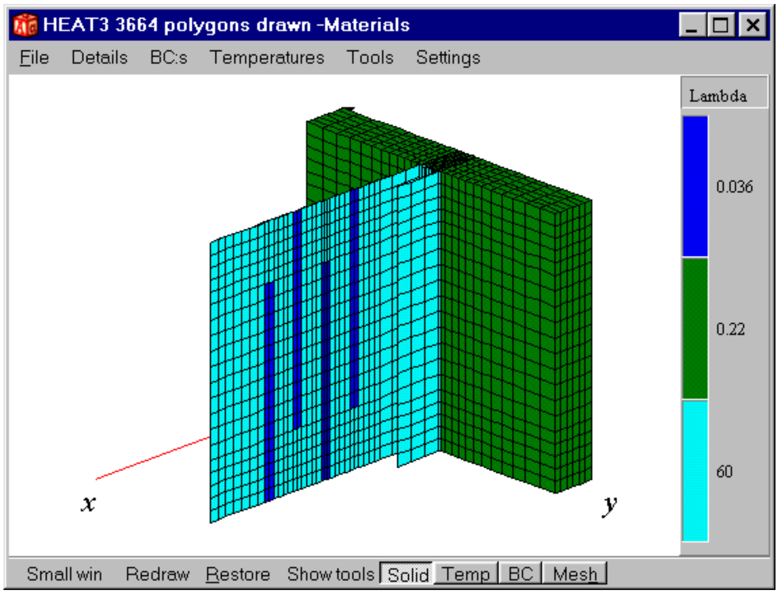


Figure 5.58: The scale in each of the three directions is adjusted to the maximum length in its own direction ('maximized area').

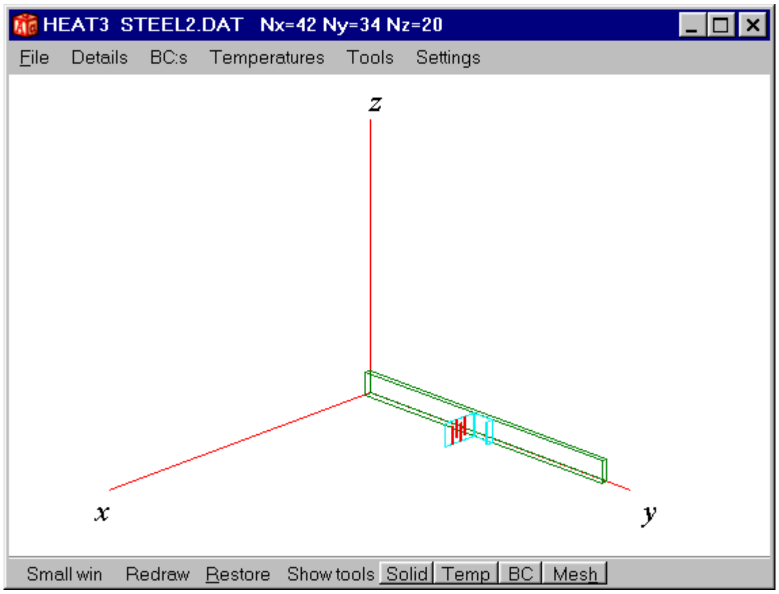
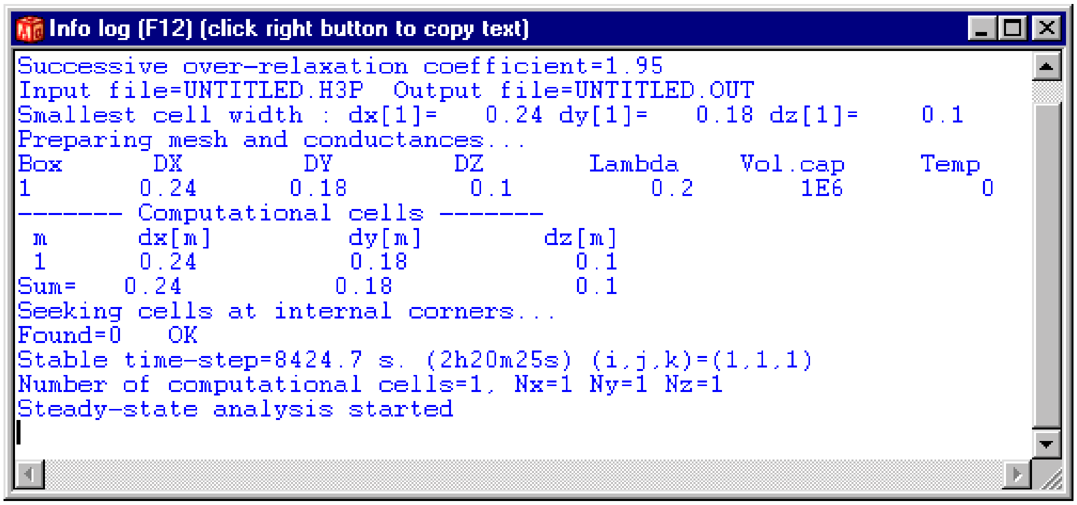


Figure 5.59: The scale is in the normal case adjusted to the largest length of the three directions. The scale will be the same in all directions.

5.12 Info log

The info log (item *Output/Info log*) contains information of input data and generated numerical mesh for the considered problem, see Figure 5.60. 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.



```
Successive over-relaxation coefficient=1.95
Input file=UNTITLED.H3P  Output file=UNTITLED.OUT
Smallest cell width : dx[1]=   0.24 dy[1]=   0.18 dz[1]=    0.1
Preparing mesh and conductances...
Box        DX        DY         DZ        Lambda    Vol.cap     Temp
1         0.24      0.18        0.1         0.2       1E6          0
------- Computational cells -------
 m       dx[m]          dy[m]        dz[m]
 1        0.24           0.18          0.1
Sum=    0.24           0.18            0.1
Seeking cells at internal corners...
Found=0   OK
Stable time-step=8424.7 s. (2h20m25s) (i,j,k)=(1,1,1)
Number of computational cells=1, Nx=1 Ny=1 Nz=1
Steady-state analysis started
```

Figure 5.60: The info log.

6. Numerical performance

6.1 Benchmarks

HEAT3 version 4 is faster than version 3. The improvement is greater for Pentium II, III and IV than for ordinary Pentium (I) processors. Table 6:1 shows benchmarks for some different problems.

HEAT3 version 4 are for the steady-state problems 1.3-3.5 times faster compared with HEAT3 version 3. The transient calculation is as much as 9 times faster.

Pentium III, 500 MHz	**HEAT3 3.0** **CPU-time (sec.)**	**HEAT3 4.0** **CPU-time (sec.)**	**Improvement** **3.0 => 4.0**
Steady-state calculations			
Slab.dat, 2000 iterations	70	35	2
Cen.dat, 2000 iterations	42	12	3.5
Steel1.dat, 10 000 iterations	96	77	1.3
Steel2.dat, 2 000 iterations	45	36	1.3
Cav.dat, 2000 iterations	33	21	1.6
Transient calculations			
CEN.dat, 7756 iterations (1 month)	97	11	9 (!)

Table 6:1: Benchmarks for version 3 and 4.

7. Format for text input data

7.1 Introduction

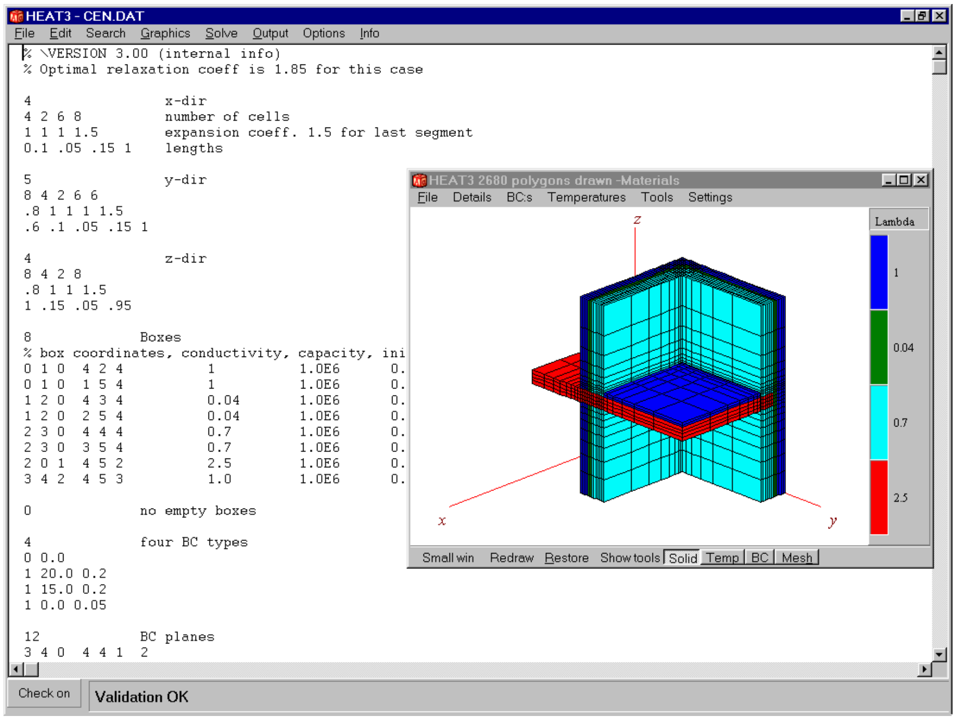
Normally, input data is generated upon changes in the pre-processor. It is however possible to give input data in text mode using the text editor, see Section 5.1.2.

7.2 Editing data

7.2.1 The editor

The editor contains the input data file, see Figure 7.1. The input is validated every time the editor is changed (unless item *Edit/Validate on change* is disabled). Upon an error, the row that contains the error will be marked and a short notice will be displayed on the bottom row.

A graphic window (showing materials, mesh, temperatures, and/or boundary conditions) is updated whenever the input is changed (unless item *Edit/Update graphics on change* is disabled).


HEAT3 - CEN.DAT
File Edit Search Graphics Solve Output Options Info
HEAT3 2680 polygons drawn -Materials
File Details BC:s Temperatures Tools Settings
z
x
y
Lambda
1
0.04
0.7
2.5
Small win Redraw Restore Show tools Solid Temp BC Mesh
Check on

```
% \VERSION 3.00 (internal info)
% Optimal relaxation coeff is 1.85 for this case

4                x-dir
4 2 6 8          number of cells
1 1 1 1.5        expansion coeff. 1.5 for last segment
0.1 .05 .15 1    lengths

5                y-dir
8 4 2 6 6
.8 1 1 1 1.5
.6 .1 .05 .15 1

4                z-dir
8 4 2 8
.8 1 1 1.5
1 .15 .05 .95

8              Boxes
% box coordinates, conductivity, capacity, ini
0 1 0  4 2 4         1          1.0E6      0.
0 1 0  1 5 4         1          1.0E6      0.
1 2 0  4 3 4         0.04       1.0E6      0.
1 2 0  2 5 4         0.04       1.0E6      0.
2 3 0  4 4 4         0.7        1.0E6      0.
2 3 0  3 5 4         0.7        1.0E6      0.
2 0 1  4 5 2         2.5        1.0E6      0.
3 4 2  4 5 3         1.0        1.0E6      0.

0              no empty boxes

4              four BC types
0 0.0
1 20.0 0.2
1 15.0 0.2
1 0.0 0.05

12             BC planes
3 4 0  4 4 1  2
```

Validation OK

Figure 7.1: Input is made in an editor. A graphic window shows materials, mesh, temperatures, and boundary conditions.

A new problem may be given from scratch (item *File/New*) using a empty editor. The help line at the bottom indicates what variables are expected on each row. Comments may be written after the data or after the characters '**%**' or '**;**'.

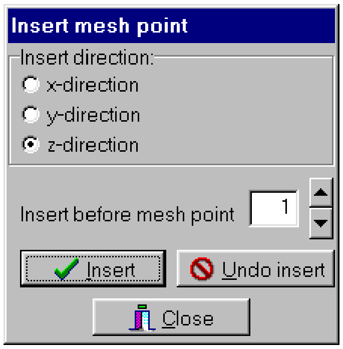
A new problem may also be defined using a template for a simple "cube" problem (item *File/New with template*). A third way is to open and edit an existing file that describes a similar problem. Save the input with a new file name (*File/Save as*). The five data files that were last open will be shown at the end of the file menu.

Standard editing options are available, such as **cut**, **copy**, **paste**, **select all**, **find**, and **replace.** The **undo** and **redo** commands make it possible to regret or remake several hundred changes made in the editor. The font may be changed in *Options/Font.* Backup files (*.BAK) will be created (unless item *Options/Create backup file* is disabled).

7.2.2 Inserting mesh coordinates

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

Input mesh coordinates may be inserted automatically (*Edit/Insert mesh coordinate*, see Figure 7.2). All the data in the editor will be updated for the boxes, surfaces, etc. 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 and maybe the number of cells for the new segments and the expansion coefficients.



Insert mesh point

Insert direction:
○ x-direction
○ y-direction
● z-direction

Insert before mesh point 1

Figure 7.2: Menu for inserting mesh coordinates.

Consider Figure 7.3. The top figure has one segment in each direction. The middle figure shows the coordinates when insertion is made in the *x*-direction before mesh point 1. The first four rows in the input data are changed from

1	segment in the x-direction
10	number of numerical cells
1	expansion coefficient
1	length of segment

to

2	segment in the x-direction
5 5	number of numerical cells
1 1	expansion coefficient
0.5 0.5	length of segment

Two coordinates have now been inserted in all three directions before mesh point 1, see bottom figure. The first four rows are now:

3	segment in the x-direction
2 3 5	number of numerical cells
1 1 1	expansion coefficient
0.25 0.25 0.5	length of segment

The bottom figure shows an added second box with coordinates (1,1,1,2,2,2).

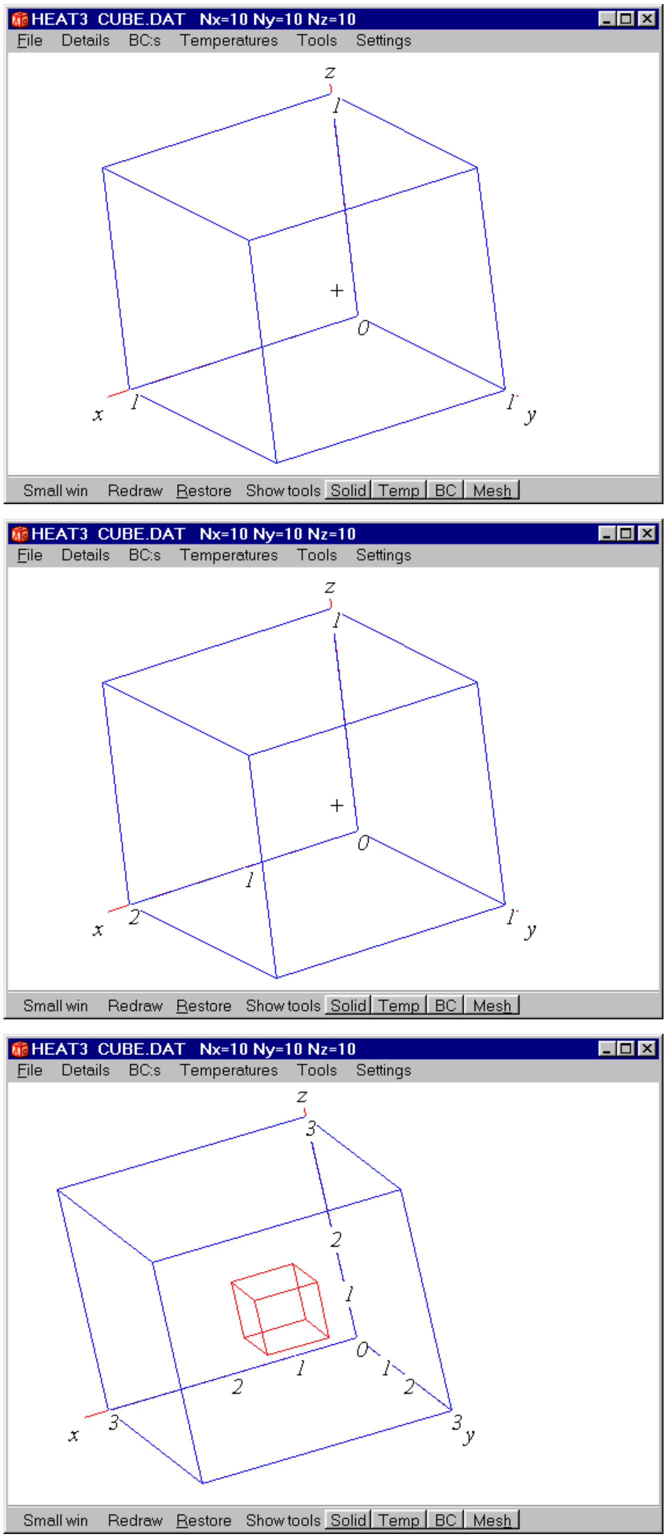

HEAT3 CUBE.DAT Nx=10 Ny=10 Nz=10
File Details BC:s Temperatures Tools Settings
z
1
0
x
1
1
y
Small win Redraw Restore Show tools Solid Temp BC Mesh
HEAT3 CUBE.DAT Nx=10 Ny=10 Nz=10
File Details BC:s Temperatures Tools Settings
z
1
0
1
x
2
1
y
Small win Redraw Restore Show tools Solid Temp BC Mesh
HEAT3 CUBE.DAT Nx=10 Ny=10 Nz=10
File Details BC:s Temperatures Tools Settings
z
3
2
1
0
1
2
x
3
1
2
3
y
Small win Redraw Restore Show tools Solid Temp BC Mesh

Figure 7.3: Mesh coordinates inserted.

Changing the lengths in each direction from

0.25 0.25 0.5 length of segment

to

0.33 0.33 0.33 length of segment

will center the small cube into the large one, see Figure 7.4.

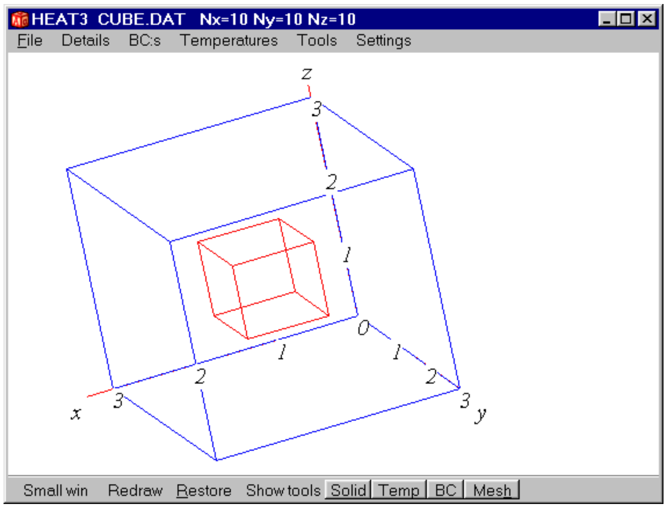


Figure 7.4: Inserting mesh coordinates and changing lengths.

7.3 Input mesh

An *input mesh* facilitates the description of the geometry, the numerical mesh, and the boundary conditions. Consider Figure 7.5. The *x*-axis is divided into a number of segments I. These are enumerated I=1, 2,..., I_{MAX}. The length of segment I is denoted ΔX_I. In the same way, divisions are made in the *y*- and *z*-directions, J=1, 2,..., J_{MAX}, K=1, 2,..., K_{MAX}, respectively. The figure shows one parallelepiped of the input mesh with the side lengths ΔX_I, ΔY_J , and ΔZ_K.

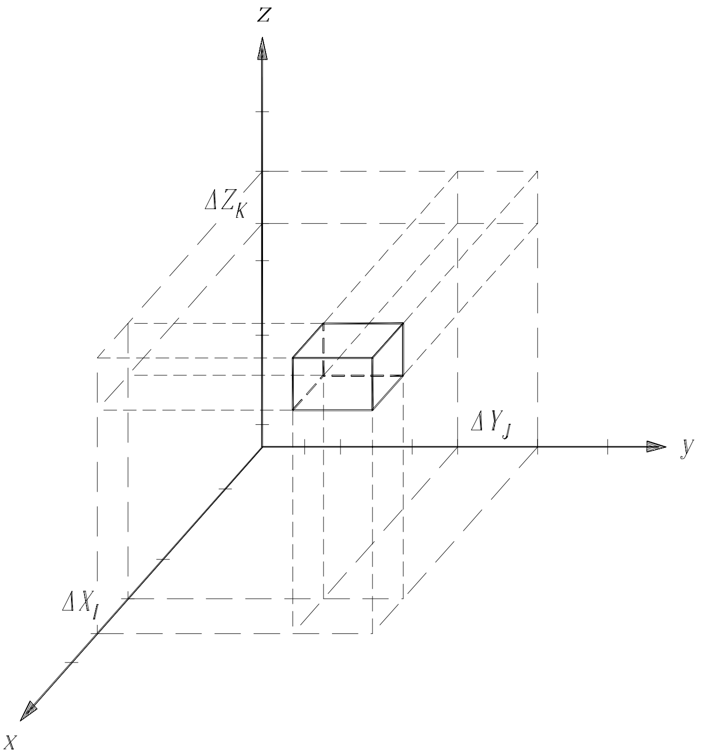


Figure 7.5: A parallelepiped in the input mesh.

The input mesh may represent any parallelepipedical geometrical structure. The structure is built by parallelepipeds defined by six input mesh coordinates, three for the lower inner corner (LI,LJ,LK), and three for the upper outer corner (UI,UJ,UK). Figure 7.6 shows the simplest geometry possible with one parallelepiped. There is one mesh segment in each direction. The coordinates for the parallelepiped are LI=0, LJ=0, LK=0, UI=1, UJ=1, and UK=1. From now on, these are written in a more compact way as (0,0,0,1,1,1).

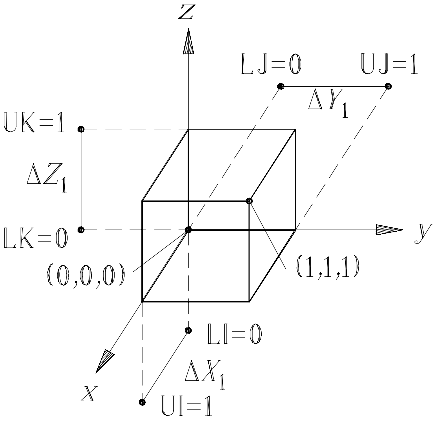


Figure 7.6: One parallelepiped.

Figure 7.7 shows a region consisting of two parallelepipeds. The larger parallelepiped has the coordinates (0,0,0,3,1,1), and the smaller (1,1,0,2,2,1).

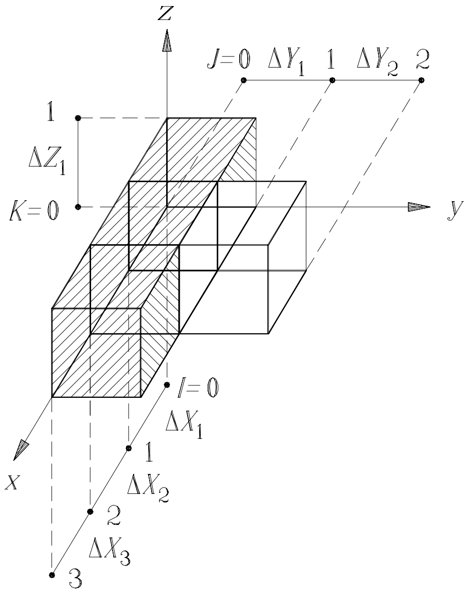


Figure 7.7: Two adjacent parallelepipeds.

As pointed out before, any parallelepipedical structure may be defined by a list of parallelepipeds (from now on also called boxes). Each box may have its own properties regarding thermal conductivity, heat capacity, and even initial temperature.

A box may partly overlap an earlier defined box. In that case, the properties of the new box will prevail. Consider Figure 7.8 with one box overlapping another of different material. The first box has the coordinates (0,0,0,3,1,1), and the second (1,0,0,2,2,1).

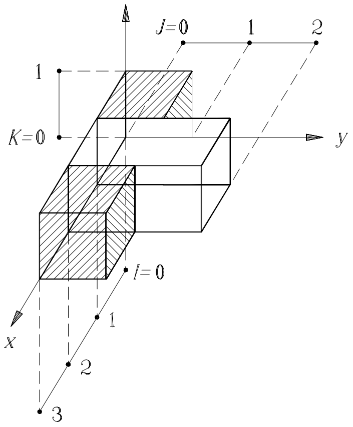


Figure 7.8: One box overlapping another box.

There is also an option for defining ’empty’ boxes (space that is not included in the computational volume). In some cases it may be an efficient way to reduce input data. Consider Figure 7.9 that shows a corner. A typical application would be to calculate the extra heat loss due to the three-dimensional heat flow in the corner, or to estimate the lowest surface temperature. This problem may be described using three boxes with coordinates (0,0,0,2,2,1), (0,0,0,2,1,2), and (0,0,0,1,2,2), respectively. It is also possible to use one box defining a material (0,0,0,2,2,) that is cut out by one empty box (1,1,1,2,2,2).

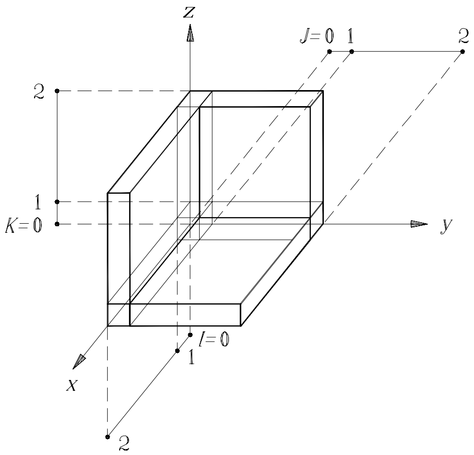


Figure 7.9: This corner may be defined by three boxes, or by one box and one empty box.

7.4 Surfaces defining boundary conditions

Boundary conditions are given in the following way. First, the types of the boundary bonditions are specified in a list. This could for example be the following three types; zero heat flow (adiabatic), temperature on a warm side, and temperature on a cold side. Secondly, surfaces that 'lie' against a boundary of the computational area are defined by six input mesh coordinates. One of the three lower coordinates of the surface will always be the same as one of the three upper ones. Each surface is then coupled to one of the boundary condition types in the list. As an example, the top surface of the larger box in Figure 7.7 has the coordinates (0,0,1,3,1,1). The top surface of the smaller box has the coordinates (1,1,1,2,2,1). Here LK=UK=1. If the boundary condition is the same on both boxes, it is sufficient to use one surface (0,0,1,3,2,1), see Figure 7.10.

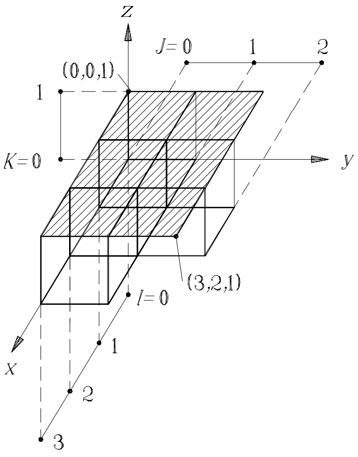


Figure 7.10: Boundary conditions are defined with surfaces, which may extend outside the surfaces of the computational volume.

If any boundary of the computational area is not overlapped by a specified surface, the boundary condition will automatically be the *first* defined type. As an example consider the cube in Figure 7.6. Suppose that the temperature on the upper surface of the cube is 1 ºC, and on the lower side 0 ºC, and that the four other surfaces are adiabatic (zero heat flow). This is done by defining the following three types: Q=0, T=1, and T=0. It is sufficient to define two surfaces, one at the upper side (0,0,1,1,1,1) with the second boundary condition type (T=1), and one at the lower side (0,0,0,1,1,0) with the third boundary condition type (T=0).

Since no surfaces for the other four sides have been defined, the boundary condition will be automatically coupled to the first type (in this case Q=0).

A surface plane may cut through the computational volume without affecting it. Consider as an example Figure 7.7, where a vertical plane between the large and the small box could be defined as (0,1,0,3,1,1). This would induce the same boundary condition on the two boundary areas of the large box (0,1,0,1,1,1) and (2,1,0,3,1,1). If a surface overlaps, partly or completely, previously defined surfaces, the boundary condition for the last mentioned surface applies.

We now introduce *sets of surfaces*. For each of the set (with one or more surfaces), the total heat flow and the minimum and maximum surface temperatures will be presented. Note that if two surfaces in the same set overlap each other, the heat flow will be added twice for the common area (which will influence the presented heat flow for the set).

7.5 Numerical mesh

The number of numerical cells is specified for each segment. As an example, consider segment I with $N_{x,I}$ numerical cells in the x-direction. The cells are normally placed in an equidistant mesh. The length of the numerical cells in the segment is $\Delta X_I/N_{x,I}$. In the same way the lengths in the y- and z-directions are $\Delta Y_I/N_{y,J}$ and $\Delta Z_K/N_{z,K}$. In general, capital letters refer to the input mesh (i.e. ΔX_I, ΔY_J, ΔZ_K), while small letters refer to the numerical mesh (i.e. Δx_i, Δy_j, Δz_k). There must be at least one numerical cell in each input mesh segment.

Figure 7.11 shows the cube with two cells in each of the three directions, i.e. $N_{x,1}$=2, $N_{y,1}$=2, and $N_{z,1}$=2.

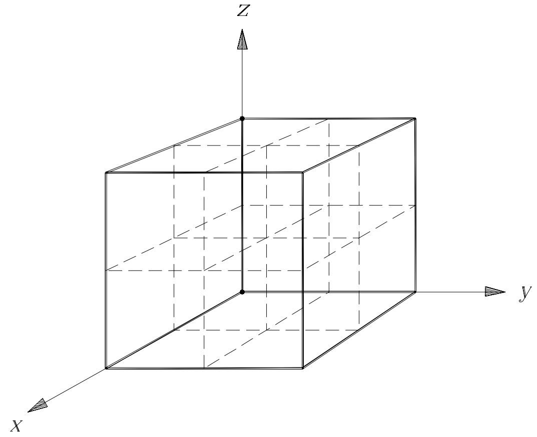


Figure 7.11: The cube with two numerical cells in each direction.

Expansive meshes may be used for concentrating the cells towards areas with large temperature gradients, see the front cover figure where the cells are smaller around the corner. An expansion coefficient ε is given for every segment I,J,K. 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 is 7 m by default. This corresponds to an expansion coefficient of 1. A coefficient ε=2.0 generates the lengths 3, 6 and 12 m (increasing sizes). With ε=0.5, a mesh with the cell lengths 12, 6 and 3 m is obtained (decreasing sizes).

Be aware that an expansive mesh may lead to small computational cells causing the time-step to be very short. A badly chosen computational mesh can increase the computational time dramatically, especially for transient analyses. Do not use extreme expansion coefficients with a large number of cells that would cause very small cells in a segment. ”Normal” values are in most cases those between 0.8-1.2. There are two ways to check the computational mesh. The first way is to look at the mesh graphically. The second way is to look at the cell sizes in the log text.

7.6 Internal resistances

Internal resistances are defined by planes that cut through the computational volume with a resistance *R*, (m²·K/W), between the cells that are divided by the plane, see Eq. (3.2). If a plane overlaps a surface defining a boundary condition of a given temperature with a surface resistance then the total resistance will include both resistances. Consider as an example Figure 7.7. We may define a plane as (0,1,0,3,1,1) with a resistance *R*. In this case, the surface resistance on the boundary areas (0,1,0,1,1,1) and (2,1,0,3,1,1) will be increased by the given value *R*.

7.7 Input data file

The input data file is an ordinary text file. Table 7:1 shows the descriptions and restrictions of the variables.

Variables	Description	Restriction
I_{MAX}, J_{MAX}, K_{MAX}	Number of segments in each direction.	1-50
$N_{x,I}$, $N_{y,J}$, $N_{z,K}$	Number of computational cells in segments.	>0
$\varepsilon_{\mathrm{x,I}}$, $\varepsilon_{\mathrm{y,J}}$, $\varepsilon_{\mathrm{z,K}}$	Expansion factor in segments.	0.5-2.0
$\Delta \mathrm{X}_{\mathrm{I}}$, $\Delta \mathrm{Y}_{\mathrm{J}}$, $\Delta \mathrm{Z}_{\mathrm{K}}$	Length of segments.	>0
LI LJ LK UI UJ UK	Lower and upper input mesh coordinates.	0-50
λ	Thermal conductivity (W/(m·K)).	>0
C	Volumetric heat capacity (J/(m³·K)).	>0
T_{init}	Initial temperature (ºC).	
N_{p}	Number of boxes defining material.	1-250
N_{e}	Number of boxes defining empty space.	0-20
N_{bc}	Number of boundary condition types.	1-10
N_{a}	Number of surfaces for which to specify boundary conditions. All other surfaces will have the first defined boundary condition	1-72
N_{r}	Number of internal resistance planes.	0-10
N_{sets}	Number of sets.	1-100
N_{surf}	Number of surfaces per set.	1-20
N_{h}	Number of heat sources.	0-40

Table 7:1: Description and restrictions of variables. Next page shows the format.

The format of the input data file is shown below. In general, each description is given on a separate row. Empty rows are allowed. Comments may be added on each row after the data. Real values are denoted by *Real1* and *Real2*, and integers by *I*.

I_{MAX}	Number of segments in *x*-direction.
$N_{x,1}$, $N_{x,2}$,..., $N_{x,IMAX}$	Number of cells for each segment.
$\varepsilon_{x,1}$, $\varepsilon_{x,2}$,..., $\varepsilon_{x,IMAX}$	Expansion coefficient for each segment.
ΔX_1, ΔX_2,..., ΔX_{1MAX}	Length for each segment.

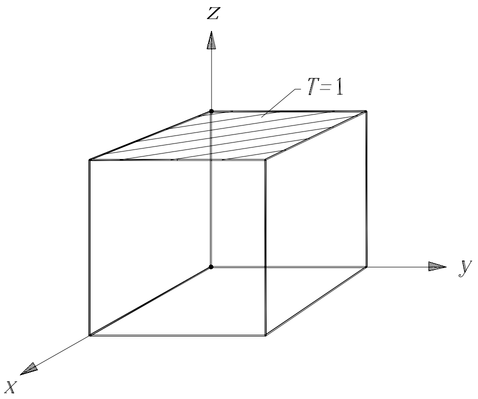
J_{MAX}	Number of segments in *y*-direction.
$N_{y,1}, N_{y,2},..., N_{y,JMAX}$	Number of cells for each segment.
$\varepsilon_{y,1}, \varepsilon_{y,2},..., \varepsilon_{y,JMAX}$	Expansion coefficient for each segment.
$\Delta Y_1, \Delta Y_2,..., \Delta Y_{JMAX}$	Length for each segment.
K_{MAX}	Number of segments in *z*-direction.
$N_{z,1}, N_{z,2},..., N_{z,KMAX}$	Number of cells for each segment.
$\varepsilon_{z,1}, \varepsilon_{z,2},..., \varepsilon_{z,KMAX}$	Expansion coefficient for each segment.
$\Delta Z_1, \Delta Z_2,..., \Delta Z_{KMAX}$	Length for each segment.
N_p	Number of boxes defining a material.
LI LJ LK UI UJ UK λ C T_i I S	Coordinates, thermal properties, initial temperature, optional integer I for color and material name string S
...	
LI LJ LK UI UJ UK λ C T_i I S	... and for last box N_p.
N_e	Number of boxes defining empty space.
LI LJ LK UI UJ UK	Coordinates for first box,
...	
LI LJ LK UI UJ UK	... and coordinates for last box N_e.
N_{bc}	Number of boundary conditions.

I Real1 (Real2)	I=0 for *Q*=Real1 (W/m²).
	I=1 for *T*=Real1 (ºC) and *R*=Real2, (m²·K/W),

...	
I Real1 (Real2)	... and for last boundary condition number N_{bc}.
N_a	Number of surfaces that do not have the first BC
LI LJ LK UI UJ UK I	Coordinates and boundary condition number I (1..N_{bc}),
...	
LI LJ LK UI UJ UK I	... and for last surface plane N_a.
N_r	Number of internal resistance surfaces.
LI LJ LK UI UJ UK Real1	Coordinates and resistance Real1, (m²·K/W),
...	
LI LJ LK UI UJ UK Real1	... and for last plane N_r.
N_{sets}	Number of sets.
N_{surf} I_1 .. I_{Nsurf}	Surfaces in each set. The surface *I1* .. I_{Nsurf} refers to the surfaces given for boundary conditions (1 to N_a),
...	
N_{surf} I_1 .. I_{Nsurf} .	.. and for last set N_{sets}.
N_h	Number of internal heat sources.
LI LJ LK UI UJ UK Real1	Coordinates and effect Real1, (W),
...	
LI LJ LK UI UJ UK Real1	... and for last heat source N_r.

7.8 Example 1

This is an input data example for the cube in Figure 7.6, see *cube.dat*. In this example, the cube has the temperature *T*=1 on the upper side, and *T*=0 on the other five sides, see figure below to the right. The thermal conductivity is 1.0, and the side lengths are 1. We especially want to know the heat flow through the upper side, and through the bottom side. In this case there are six surfaces defining the boundary conditions. There are three sets for which the heat flows will be presented. If there are multiple surfaces in a set, such as the third one, the calculated heat flow for the set will involve all the surfaces. Note that the stop criterion for heat flows *F* is calculated for the sets. This means that all surfaces with a nonzero heat flow must be defined when this criterion is used, and also that each surface belongs to one (and one only) set. Figure 7.12 shows the mesh and the surface temperatures.

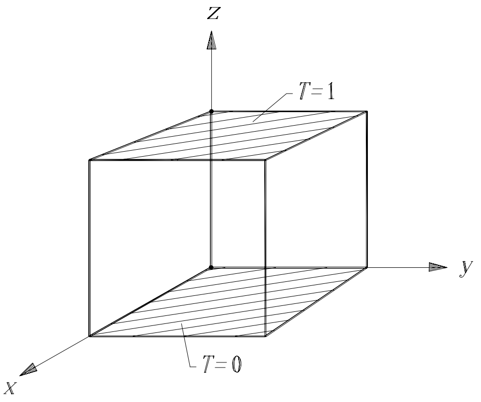


```
1   one segment in the x-direction
10 number of numerical cells
1   expansion coefficient
1   length of segment

1   one segment in the y-direction
10 number of numerical cells
1   number of expansion coefficient
1   length of segment

1   one segment in the z-direction
10 numerical cells
1   expansion coefficient
1   length of segment

1   one box
% box coordinates, conductivity, capacity, initial temp, optional color in graphics
0 0 0  1 1 1        1.0          1.0E6       0.0          1

0   no empty boxes

2           two BC type numbers
1 0.0 0.0           BC number 1, '1' means temperature, temperature 0, surface resistance 0
1 1.0 0.0           BC number 2, '1' means temperature, temperature 1, surface resistance 0

6                   six surfaces
0 0 1  1 1 1  2   coordinates and BC number 2 (temperature is 1) for surface 1
0 0 0  1 1 0  1   coordinates and BC number 1 (temperature is 0) for surface 2
0 0 0  0 1 1  1
0 0 0  1 0 1  1
1 0 0  1 1 1  1
0 1 0  1 1 1  1

0           no internal resistance planes

3                   number of sets to present flows for
1 1                 set 1, 1 surface (surface 1 at the top, the one with coord. 0,0,0,1,1,1)
1 2                 set 2, 1 surface (surface 2 at the bottom, the one with coord. 0,0,0,1,1,0)
4 3 4 5 6           set 3, 4 surfaces (surfaces 3-6)
0           no heat sources
```

7.9 Example 2

Assume that the cube in the previous example has the temperature *T*=1 on the upper side, *T*=0 on the bottom side, and that the four other sides now are adiabatic (zero heat flow), see figure to the right. The heat flow is 1 W through the cube (one-dimensional case). In this case, only two surfaces need to be defined. The other four will automatically be assigned the first boundary condition (zero heat flow).

Note that the initial temperatures and the heat capacities are irrelevant to the solution in the steady-state case. The CPU-time may be somewhat decreased if the initial temperatures are close to the final temperatures (in this case we guess 0.5 ºC). Figure 7.13 shows the mesh and the surface temperatures.



```
1    one segment in the x-direction
10   number of numerical cells
1    expansion coefficient
1    length of segment

1    one segment in the y-direction
10   number of numerical cells
1    number of expansion coefficient
1    length of segment


1    one segment in the z-direction
10   numerical cells
1    expansion coefficient
1    length of segment

1    one box
% box coordinates, conductivity, capacity, initial temp, optional color in graphics
0 0 0  1 1 1       1.0        1E6       0.5        1

0          no empty boxes

3          three BC type numbers
0 0.0               BC number 1, '0' means heat flow, Q=0
1 0.0 0.0           BC number 2, '1' means temperature, temperature 0, surface resistance 0
1 1.0 0.0           BC number 3, '1' means temperature, temperature 1, surface resistance 0

2                   two surfaces
0 0 1  1 1 1  3     coordinates and BC number 3 (temperature is 1) for surface 1
0 0 0  1 1 0  2     coordinates and BC number 2 (temperature is 0) for surface 2

0          no internal resistance planes

2                   number of sets to present flows for
1 1                 set 1, one surface (surface 1, the one with coord. 0,0,0,1,1,1)
1 2                 set 2, one surface (surface 2, the one with coord. 0,0,0,1,1,0)

no heat sources
```

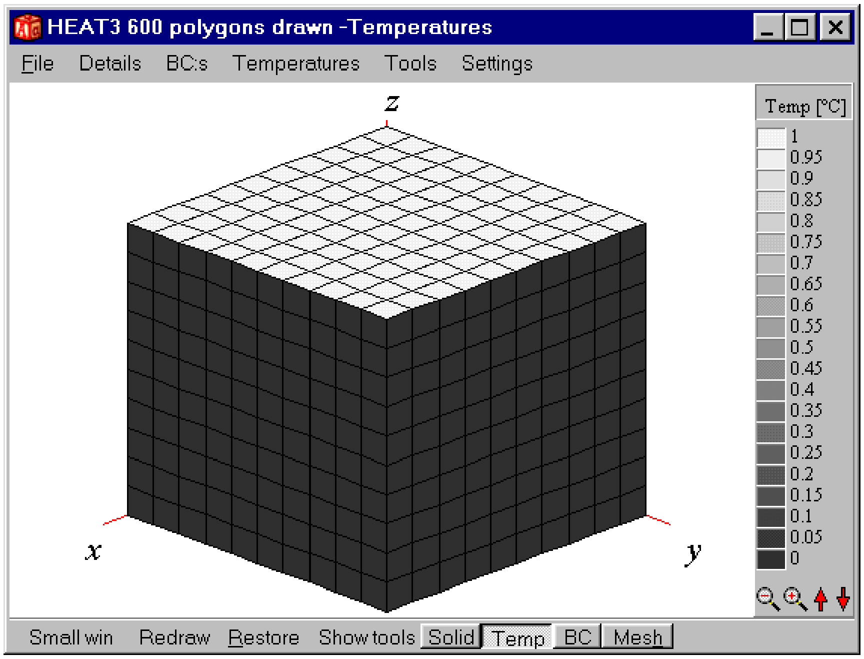


Figure 7.12: Surface temperatures for example 1.

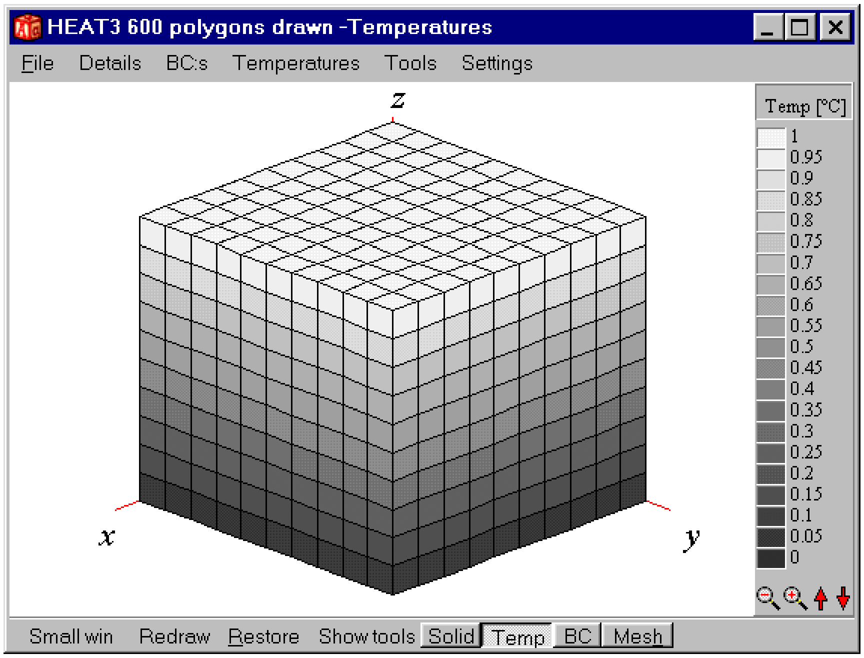


Figure 7.13: Surface temperatures for example 2.

7.10 Example 3

An input data file for the boxes in Figure 7.7 is shown below. The boxes have the temperature *T*=1 on the upper sides, and *T*=0 on the bottom sides. The eight other sides are adiabatic (zero heat flow). There is a thermal resistance of 0.2 $\text{m}^2\cdot\text{K/W}$ in the y-direction (which in this case have no effect since heat flow occurs only in the z-direction for this 1-D problem). Figure 7.14 shows the mesh and the surface temperatures.

```
3               three segments in the x-direction
5 5 5           number of numerical cells (e.g. 5 in each segment)
1 1 1           expansion coefficients (here we use an equidistant mesh)
0.1 0.1 0.1     length of segments (which gives computational cells with dx=0.02)

2               two segments in the y-direction
5 5             number of numerical cells (e.g. 5 in each segment)
1 1             expansion coefficients (here we use an equidistant mesh)
0.1 0.1         length of segments (which gives computational cells with dy=0.02)

1               one segment in the z-direction
10              number of numerical cells
1.1             expansion coefficient (the cell sizes will increase in z-direction)
0.1             length of segment

2               two boxes
% box coordinates, conductivity, capacity, initial temp, optional color in graphics
0 0 0  3 1 1      1.0     1.0E6     0.0     1
1 1 0  2 2 1      1.0     1.0E6     0.0     2

0               no empty boxes

3                     three BC type numbers
0 0.0           BC number 1, '0' means flow, the heat flow is zero (adiabatic)
1 0.0 0.0       BC number 2, '1' means temperature, temperature 0, surface resistance 0
1 1.0 0.0       BC number 3, '1' means temperature, temperature 1, surface resistance 0

2               two surfaces, all other boundaries will be adiabatic
0 0 1  3 2 1  3 BC number 3 (temperature is 1) for surface 1 (upper sides)
0 0 0  3 2 0  2 BC number 2 (temperature is 0) for surface 2 (lower sides)

1               one internal resistance plane
0 1 0  3 1 1 0.2 contact resistance between the small and large box will be 0.2

2               number of sets to present flows for
1 1                   set 1, 1 surface (surface 1, the one with coord. 0,0,1,3,2,1)
1 2                   set 2, 1 surface (surface 2, the one with coord. 0,0,0,3,2,0)

0               no heat sources
```

7.11 Example 4

Here is an input data file for the corner shown in Figure 7.9 (*corner.dat*). The temperature *T* is 20 ºC on the three internal sides, and *T*=0 ºC on the external sides. The other sides are adiabatic. The internal and external surface resistances are 0.13 m²·K/W and 0.04 m²·K/W, respectively. Figure 7.15 shows the mesh and the surface temperatures.

```
2       two segments in the x-direction
5 10            number of numerical cells
1 1.2           expansion (cell sizes will be smaller towards the corner)
0.1 1.0         length of segment

2               two segments in the y-direction
5 10            number of numerical cells
1 1.2           expansion (cell sizes will be smaller towards the corner)
0.1 1           length of segments

2               two segments in the z-direction
5 10            number of numerical cells
1 1.2           expansion (cell sizes will be smaller towards the corner)
0.1 1           length of segments

1
% box coordinates, conductivity, capacity, initial temp, optional color in graphics
0 0 0  2 2 2      1.7      1.0E6     0.0       1

1       one empty box
1 1 1  2 2 2

3               three BC type numbers
0 0.0           BC number 1, '0' means flow, the heat flow is zero (adiabatic)
1 0.0 0.04      BC number 2, '1' means temperature, temperature 0, surface resistance 0.04
1 20.0 0.13     BC number 3, '1' means temperature, temperature 20, surface resistance 0.13

6               six surfaces, all other boundaries will be adiabatic
0 0 0  2 0 2  2 BC number 2 (temperature is 0)  for surface 1   (outer side)
0 0 0  0 2 2  2 BC number 2 (temperature is 0)  for surface 2     ..
0 0 0  2 2 0  2 BC number 2 (temperature is 0)  for surface 3     ..
1 1 1  2 1 2  3 BC number 3 (temperature is 20) for surface 4   (inner side)
1 1 1  1 2 2  3 BC number 3 (temperature is 20) for surface 5     ..
1 1 1  2 2 1  3 BC number 3 (temperature is 20) for surface 6     ..

0       no internal resistance

2               number of sets to present flows for
3 1 2 3
3 4 5 6

0       no heat sources
```

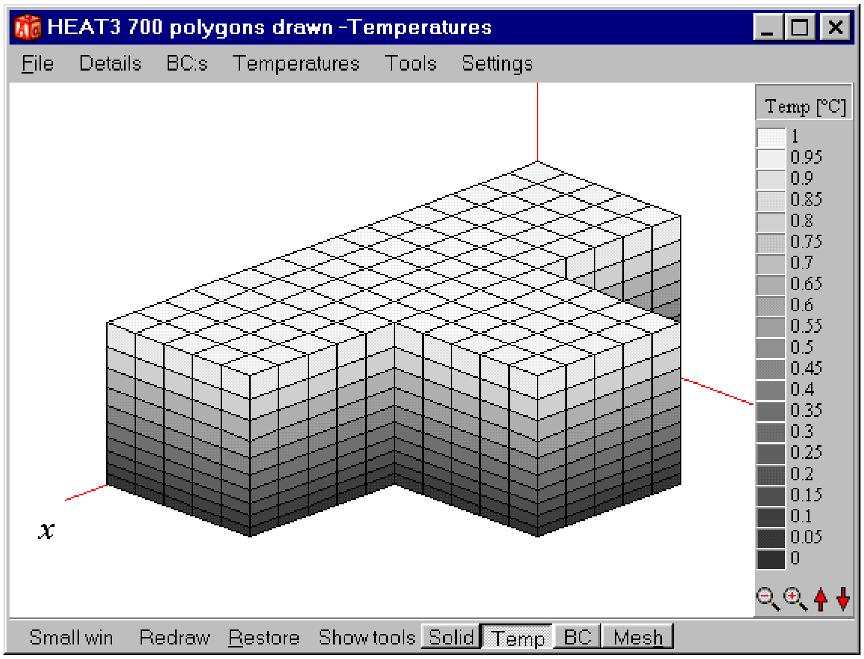


Figure 7.14: Surface temperatures for example 3.

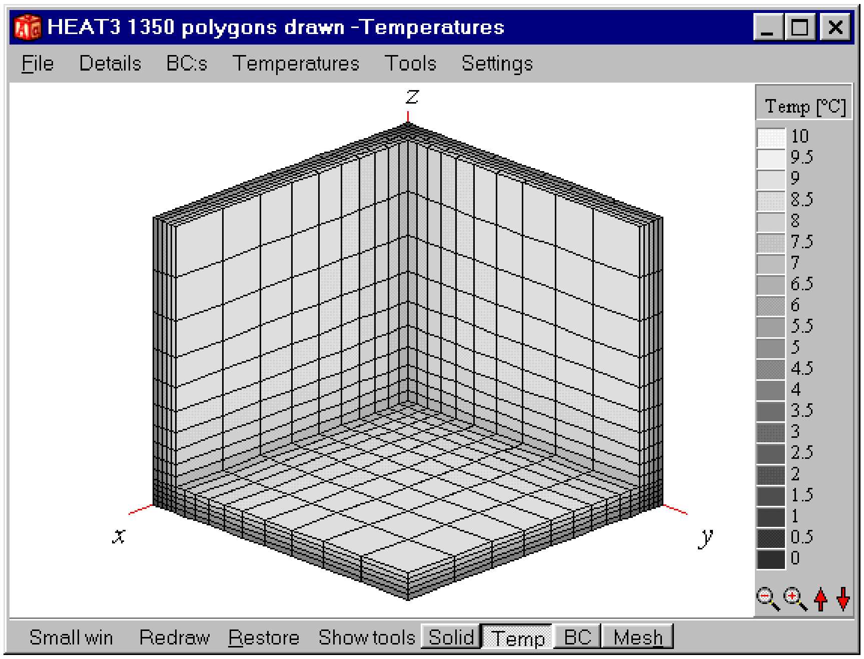


Figure 7.15: Surface temperatures for example 4.

7.12 The info log

The info log (item *Output/View info log*) contains information of input data and generated numerical mesh for the considered problem. It is automatically created when a simulation is started. It may be a good idea to look at the log whenever an expansive mesh is used to see how small the computational cells are. Consider the corner problem, see section 7.11. The sizes for the numerical cells using the expansion coefficient 1.2 are shown below. There is a smooth increase of the sizes for the numerical cells with indices 6 to 15. In this case, the cells are concentrated to the volume where the temperature gradients are largest, i.e. the corner itself.

```
Successive over-relaxation coefficient=1.97
Input file=C:\HEAT3\CORNER.DAT  Output file=CORNER.OUT
Smallest cell width : dx[1]=0.02  dy[1]=0.02  dz[1]=0.02
Preparing mesh and conductances...
Box    DX     DY     DZ    Lambda  Vol.cap  Temp
 1     1.1    1.1    1.1    1.7     1E6      0
------- Computational cells -------
 m   dx[m]    dy[m]    dz[m]
 1    0.02     0.02     0.02
 2    0.02     0.02     0.02
 3    0.02     0.02     0.02
 4    0.02     0.02     0.02
 5    0.02     0.02     0.02
 6  0.0385   0.0385   0.0385
 7  0.0462   0.0462   0.0462
 8  0.0555   0.0555   0.0555
 9  0.0666   0.0666   0.0666
10  0.0799   0.0799   0.0799
11  0.0959   0.0959   0.0959
12   0.115    0.115    0.115
13   0.138    0.138    0.138
14  0.1656   0.1656   0.1656
15  0.1988   0.1988   0.1988
Sum=   1.1      1.1      1.1
Seeking cells at internal corners...
Found=57   OK
Stable time-step=39.177 s.  (i,j,k)=(2,2,2)
Number of computational cells=2375, Nx=15 Ny=15 Nz=15
Area for set 1=3.63 m²
Aarea for set 2=3 m²
Steady-state analysis started.
```

8. More examples using text input

8.1 Introduction to examples

8.1.1 Example 5, metal studs in a wall

This example presents a calculation for a wall with cross-laid steel studs. A few recipes how to choose an appropriate mesh are given. Up to one million computational cells is used.

8.1.2 Example 6, slotted metal studs in a wall

An insulated wall can be supported by slotted steel studs. Calculation of the heat transmittance is a difficult problem numerically due to the high ratio of thermal conductivity between the insulation and the steel.

8.1.3 Example 7, heat flow through a corner

This example of heat flow around a corner is a test reference case in the European Standards.

8.1.4 Example 8, slab on the ground

This example concerns the heat flow from a building to the ground.

8.1.5 Example 9, ground heat storage

Caverns can be used for heat storage. This is exemplified by the underground storage in Skarvik, Sweden (Claesson et al, 1989). The caverns there, used earlier as oil depot, may be filled with water and then used as a heat store. The heat loss from one such cavity is examined here.

8.2 Example 5, metal studs in a wall, steel1.dat

8.2.1 Introduction

Studs and crossbars of steel or aluminum are often used in walls. The thin metal parts create thermal bridges since the thermal conductivity of the metal may be several thousand times higher than that of the insulation material. This results in particular problems for numerical calculation of the heat flow processes involved. A detailed report is given in (Blomberg, 1996).

Figure 8.1 shows a principle sketch of a wall with mineral wool and two cross-laid studs. The thermal conductivity of the steel is about 1700 times greater than that of the insulation material. It is shown that genuinely three-dimensional numerical calculations must be used in order to calculate a proper U-value. Up to 1000000 nodal points are used.

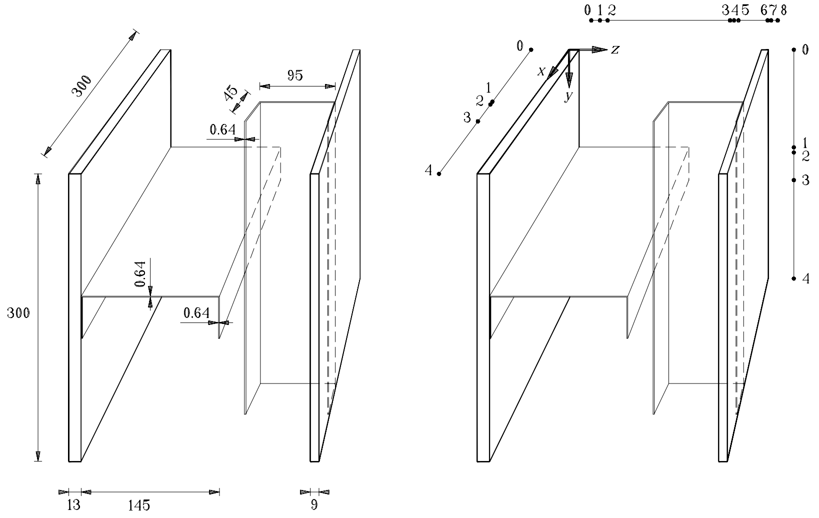


Figure 8.1: Principle sketch of an insulated wall with metal studs connecting two gypsum boards. Input mesh coordinates are shown in the right figure (see also Figure 8.2 and Figure 8.3).

The boundary conditions are as follows. The calculations are made for a unit temperature difference. The air temperature is put to 1 ºC on one side of the wall and 0 ºC on the other side. The indoor and outdoor surface resistances are 0.13 and 0.04 m^2·K/W, respectively. Table 8:1 shows the thermal conductivity of the materials.

The part of the wall that is cut out is considered as representative for the whole wall, even though there is no perfect symmetry. The boundaries of the other four sides through the wall (the boundary areas in the *x,z*- and *y,z*-planes in Figure 8.1) are adiabatic.

Material	Thermal cond. (W/(m^2·K))
Insulation	0.036
Steel	60
Gypsum	0.22

Table 8:1: Thermal conductivities used in the calculations.

A number of segments is specified for each direction in order to define the problem and generate the computational mesh. For the considered wall there are four segments in the *x*-direction, four in the *y*-direction, and eight in the *z*-direction, see also Figure 8.2 and Figure 8.3.

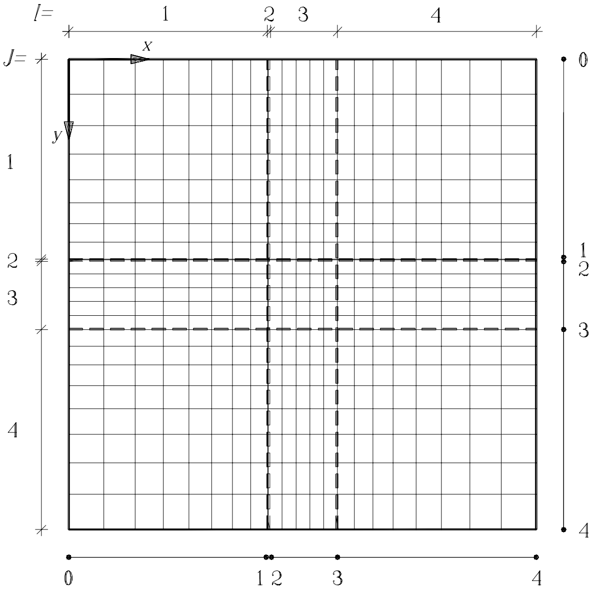


Figure 8.2: Mesh segments and coordinates, and the computational mesh in the *x*,*y*-directions for the case with 11616 numerical cells.

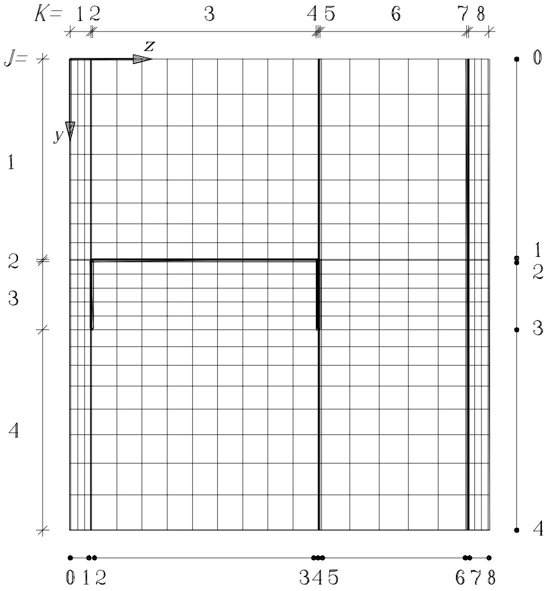


Figure 8.3: Mesh segments and coordinates, and the computational mesh in the *y*,*z*-directions for the case with 11616 numerical cells.

8.2.2 Input file steel1.dat

The input data file of the problem for the case with 11616 computational is shown below. Figure 8.4 shows the calculated temperatures.

```
4                 x-dir
8 1 5 8
0.9 1 1 1.1
0.1275 0.64e-3 44.36e-3 0.1275

4                 y-dir
8 1 5 8
0.9 1 1 1.1
0.1275 0.64e-3 44.36e-3 0.1275

8                 z-dir
3 1 9 1 1 5 1 3
0.8 1 1 1 1 1 1 1.2
13e-3 0.64e-3 .14372 0.64e-3 0.64e-3 93.72e-3 0.64e-3 9e-3

9         boxes
% box coordinates, conductivity, capacity, initial temp, optional color in graphics
0 0 0 4 4 8     0.036       1.0     0.5       0  mineral wool ("0" means that this box is invisible)
0 0 0 4 4 1       0.22      1.0     0.5       2  gypsum
0 0 7 4 4 8       0.22      1.0     0.5       2  gypsum
0 1 1 4 3 2       60.0      1.0     0.5       3  steel flange
0 1 3 4 3 4       60.0      1.0     0.5       3  steel flange
1 0 4 3 4 5       60.0      1.0     0.5       3  steel flange
1 0 6 3 4 7       60.0      1.0     0.5       3  steel flange
0 1 2 4 2 3       60.0      1.0     0.5       4  steel web
1 0 5 2 4 6       60.0      1.0     0.5       4  steel web

0         no empty box

3         BC:s
% type (0=heat flow, 1=temperature and surface resistance)
0 0.0
1 0.0 0.04
1 1.0 0.13

2         surfaces
% number of surfaces in each set, and the surfaces from list above
0 0 0  4 4 0   3
0 0 8  4 4 8   2

0  no internal resistances

2         sets
1 1
1 2

0         no heat sources
```

If a box overlaps earlier defined boxes, the properties of the new box will. This means that it is possible to define the problem using only six boxes as below (steel1b.dat) instead of nine:

```
…
6 boxes
% box coordinates, conductivity, capacity, initial temp, optional color in graphics
0 0 0 4 4 8    0.22  1.0   0.5  2 gypsum
0 0 1 4 4 7   0.036  1.0   0.5  0 mineral wool
0 1 1 4 3 4    60.0  1.0   0.5  4 steel
1 0 4 3 4 7    60.0  1.0   0.5  4 steel
0 2 2 4 3 3   0.036  1.0   0.5  0 mineral wool
2 0 5 3 4 6   0.036  1.0   0.5  0 mineral wool
…
```

8.3 Results

Figure 8.4 shows the calculated temperatures. The mineral wool is excluded by setting the color to zero for the box, see the input data. Figure 8.5 shows the calculated heat flows through the defined sets.

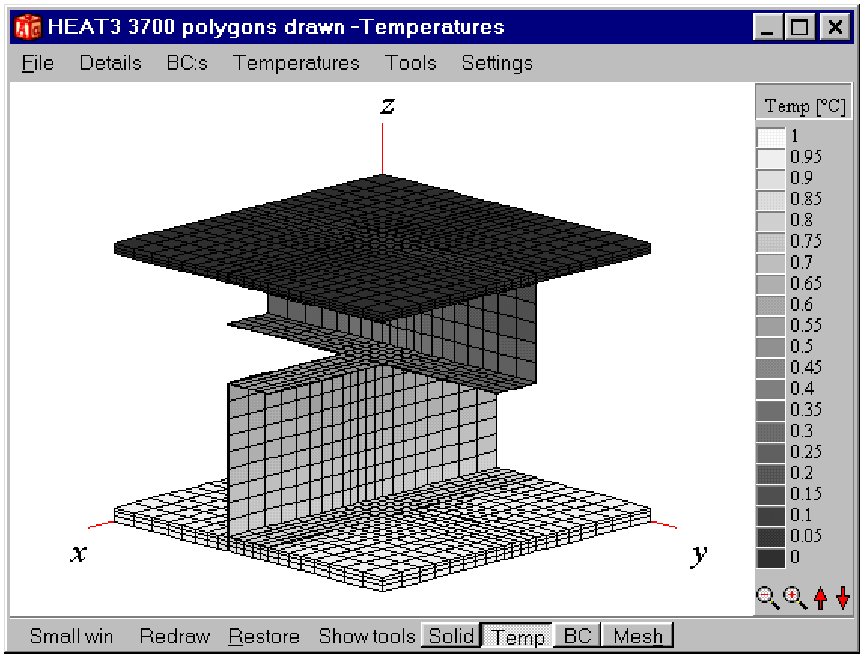
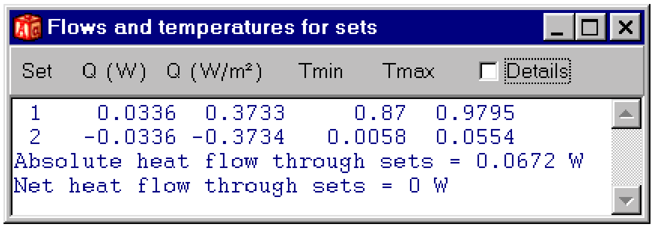


Figure 8.4: Calculated temperatures. The mineral wool is excluded by setting the color to zero for the box.



```
 1     0.0336  0.3733      0.87  0.9795
 2    -0.0336 -0.3734    0.0058  0.0554
Absolute heat flow through sets = 0.0672 W
Net heat flow through sets = 0 W
```

Figure 8.5: Calculated heat flows through the defined sets.

The calculated heat flows through the wall section Q, (W), are shown in Table 8:2 (stop criterion *F*=1·10-3). For an increasing number of cells, the solution converges to the stable flow 0.0342 W. The relative errors compared to the last case with one million cells are given in the third column. The error for the case with 11616 cells is 2 %. The required run-time, labeled as CPU-time, is shown for a Pentium/233.

N	*Q* (W)	Error (%)	CPU-time
128	0.0282	18	
1215	0.0322	6	
2736	0.0329	4	
11616	0.0336	2	7 sec.
193492	0.0341	0.3	3 min.
270400	0.0342	0	4 min.
1000000	0.0342	-	50 min.

Table 8:2: Results for seven different numerical meshes.

The U-value of the wall becomes 0.0342/(0.300·0.300)=0.38 W/(m²·K). The corresponding wall without steel has the U-value 0.144 W/(m²·K). Thus, the wall with steel studs has about 2.6 (0.38/0.144) times larger U-value than a wall without such studs. If the thickness of the steel is doubled from 0.64 mm to 1.28 mm, the U-value becomes 0.513 W/(m²·K), i.e. it increases 35%.

The thermal bridge effect may be reduced significantly by slotting the studs. This is exemplified in the next section.

8.4 *Example 6, slotted metal studs, steel2.dat*

8.4.1 Introduction

Figure 8.6 shows the structure of a wall in which insulation is contained between two gypsum boards of 13 mm thickness each. The distance between the metal studs is denoted by L_g. There is an extra heat loss caused by the metal studs. Slitting the web of the studs perpendicular to the heat flow direction, as shown in Figure 8.6, can reduce this heat loss. The thickness of the stud is denoted *t*.

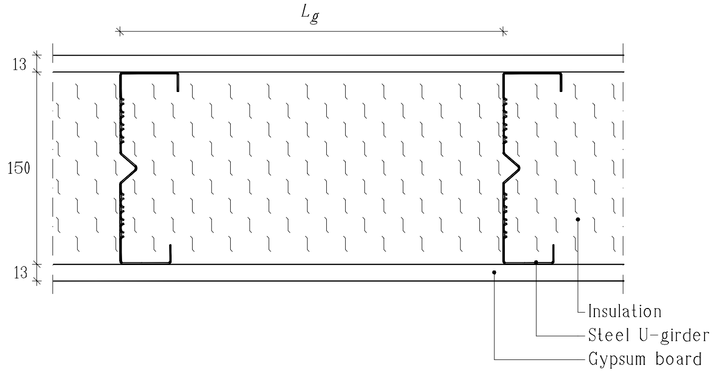


Figure 8.6: Principle sketch of a wall with studs between gypsum boards.

8.4.2 Data for the numerical calculation

The problem is not perfectly symmetric due to different flange lengths (40 and 46 mm, see Figure 8.7). However, this is neglected in the calculations, and the left flange length (46 mm) is used for both sides. The Swedish standards normally prescribe that the sum of the inner and outer surface resistances is 0.17 m²·K/W for U-value calculations. The inner and the outer surface resistances are put to half this value (0.085 m²·K/W) in the calculations to follow. The elevated part in the middle of the web is neglected, and the web is modeled as being straight, as shown in Figure 8.7 (left) by the dashed lines. Numerical tests show that this will give a slightly overestimated value for the heat flow through the wall of 0.2 %. The slitting process causes small elevated rims, as shown in Figure 8.7. These rims are neglected in the calculations, and the effect of this is discussed later.

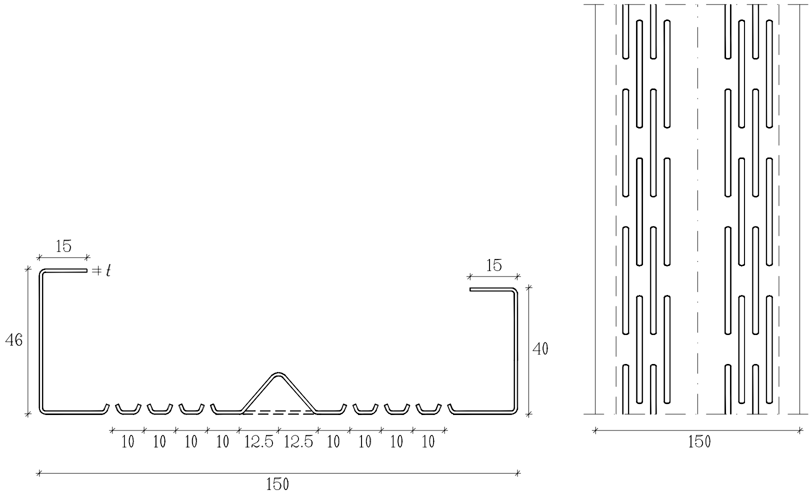


Figure 8.7: The slots (cross-section to the left) decreases the heat conduction.

The calculations have been made for the shaded volume shown in Figure 8.8. The height (perpendicular to the plane in Figure 8.6) is denoted by *s*. The temperature in the air is 0 °C on one side of the wall and 0.5°C at the symmetry line in the middle of the wall. The thermal conductivity is 0.036 W/(m^2·K) and 0.22 W/(m^2·K) for the insulation and the gypsum, respectively.

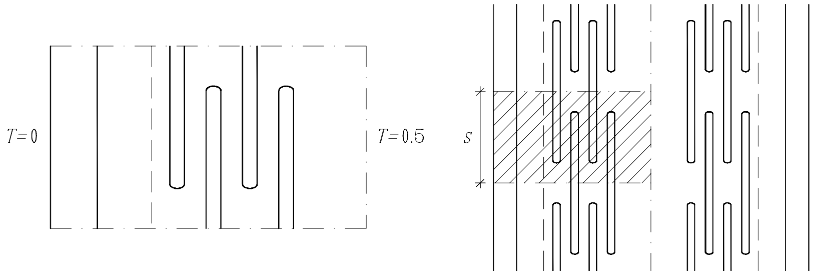


Figure 8.8: The part of studs used in the simulations.

8.4.3 Numerical mesh

Figure 8.9 shows the projection of the numerical mesh on the (*x,y)*-plane and the (*x,z)*-plane in the case involving 30000 cells. The thicker lines show the position of the steel stud. Figure 8.10 shows a part of the mesh and the temperatures in gray-scale. The insulation has been removed in the figure.

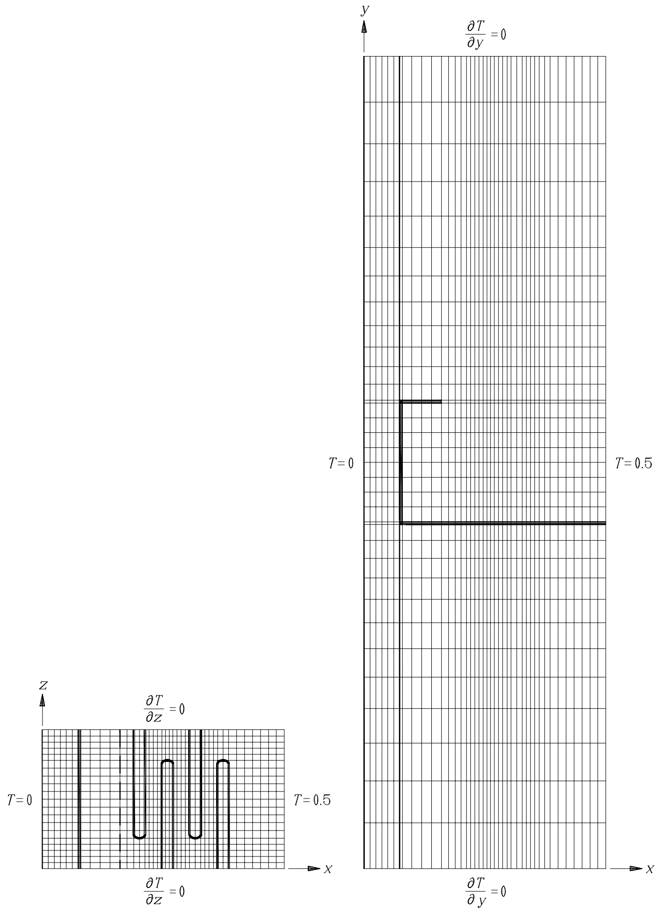

y
$\frac{\partial T}{\partial y}=0$
$T=0$
$T=0.5$
z
$\frac{\partial T}{\partial z}=0$
$T=0$
$T=0.5$
x
$\frac{\partial T}{\partial z}=0$
$\frac{\partial T}{\partial y}=0$
x

Figure 8.9: Projection of the numerical mesh on the (*x*,*z*)-plane and the (*x*,*y*)-plane in the case involving about 30000 computational cells.

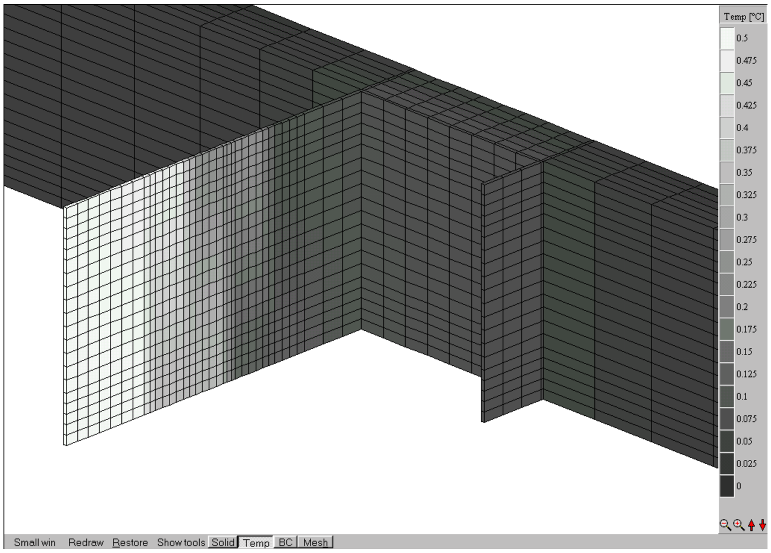


Figure 8.10: Part of the mesh and the temperatures in gray-scale. The insulation has been removed in the figure (only the steel and the gypsum board are shown here).

8.4.4 Input file, *steel2.dat*

```
12
6 1 3 4 2 4 3 4 2 4 2 7
1 1 1 1 1 1 1 1 1 1 1 1
0.013 0.7e-3 14.3e-3 7.5e-3 3.0e-3 7.0e-3 3.0e-3 7.0e-3 3.0e-3 7.0e-3 3.0e-3 19.5e-3

5
12 1 8 1 12
0.9 1 1 1 1.1
0.277 0.7e-3 44.6e-3 0.7e-3 0.277

3
5 10 5
1 1 1
11e-3 28e-3 11e-3

9 boxes
% box coordinates, conductivity, capacity, initial temp, optional color in graphics

0 0 0 12 5 3   0.036 1.0 0.25 0 insulation (0=this box is invisible)
0 0 0  1 5 3      0.22 1.0  0.0 2 gypsum

1 1 0 12 2 3       60 1.0 0.25 3 steel
1 3 0  3 4 3       60 1.0 0.25 3  .
1 1 0  2 4 3       60 1.0 0.25 3  .

4 1 1   5 2 3    0.036 1.0 0.25 4 slots (filled with insulation)
6 1 0   7 2 2    0.036 1.0 0.25 4  .
8 1 1   9 2 3    0.036 1.0 0.25 4  .
10 1 0 11 2 2  0.036 1.0 0.25 4  .

0 empty boxes

3  BC:s
% type (0=heat flow, 1=temperature and surface resistance)
0 0.0
1 0.0 0.085
1 0.5 0.0

2   BC surfaces
0  0 0   0 5 3  2
12 0 0  12 5 3  3

0  int. resistances

2  sets
1 1
1 2

0 no heat sources
```

8.4.5 Results

The calculated heat flows are shown in Table 8:3 for five different meshes. For an increasing number of cells, the solution converges to the stable flow 0.00797 W. The relative errors compared to the last case with one million cells are given in the fifth column. The error for the case with 29000 cells is 1.3 %. The required run-time on a Pentium 233 MHz, labeled as CPU-time, is also shown. The over-relaxation coefficient is here 1.98. This is an optimal value, or close to the optimal value, for the five cases in Table 8:3 (stop criterion is F=10^{-3}).

Cells	Iterations	CPU-time	*Q* (W)	Error (%)
180	203	0s	0.00756	5.4
5400	584	5s	0.00783	1.8
29000	791	37s	0.00787	1.3
120000	1631	6m	0.00793	0.5
1000000	5026	2h 7m	0.00797	-

Table 8:3: Calculated heat flow for five different meshes.

Consider Figure 8.11. The temperature field in the steel plane through the wall is shown for the reference case with slots. The path for the heat flow is increased due to the slots. The figure is drawn in Matlab. The menu item *Output/Write temperature plane* creates the necessary files (the plane cut is made at *y*=0.27735).

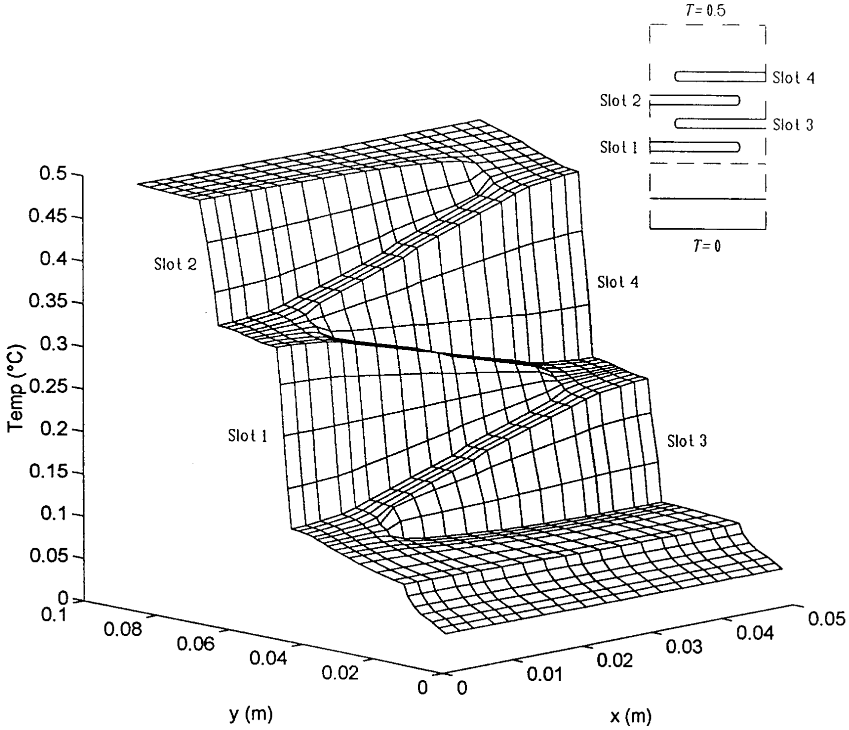


Figure 8.11: Temperature field in the steel plane through the wall.

8.5 Example 7, heat flow through a corner, cen.dat

8.5.1 Introduction

The next example is a test reference case in the European Standards, (CEN, 1995). Figure 8.12 shows the problem from two different viewpoints. There are two walls that meet in a corner, and a floor element. Of interest is the heat loss through the surfaces and the temperatures at the six points U to Z. Figure 8.13 shows the input mesh in the *x,y*-plane and *y,z*-plane, respectively. The boundary conditions are labeled with Greek letters α, β, γ and δ, see Table 8:4. The materials are labeled with numbers 1-5.

Figure 8.14 shows the projection of the numerical mesh on the x,y-plane and the x,z-plane involving 7600 cells as specified in the input data file.

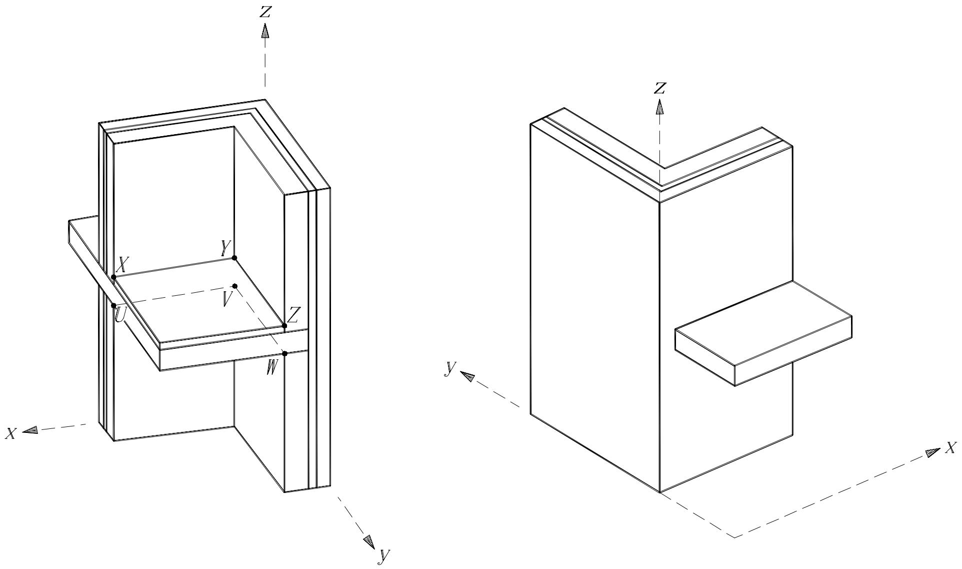


Figure 8.12: Test reference case.

Label	Boundary condition
α	*T*=20 ºC, *R*=0.2 m^2·K/W
β	*T*=15 ºC, *R*=0.2 m^2·K/W
γ	*T*=0 ºC, *R*=0.05 m^2·K/W
δ	*Q*=0 (adiabatic)

Table 8:4: Boundary conditions, see Figure 8.13.

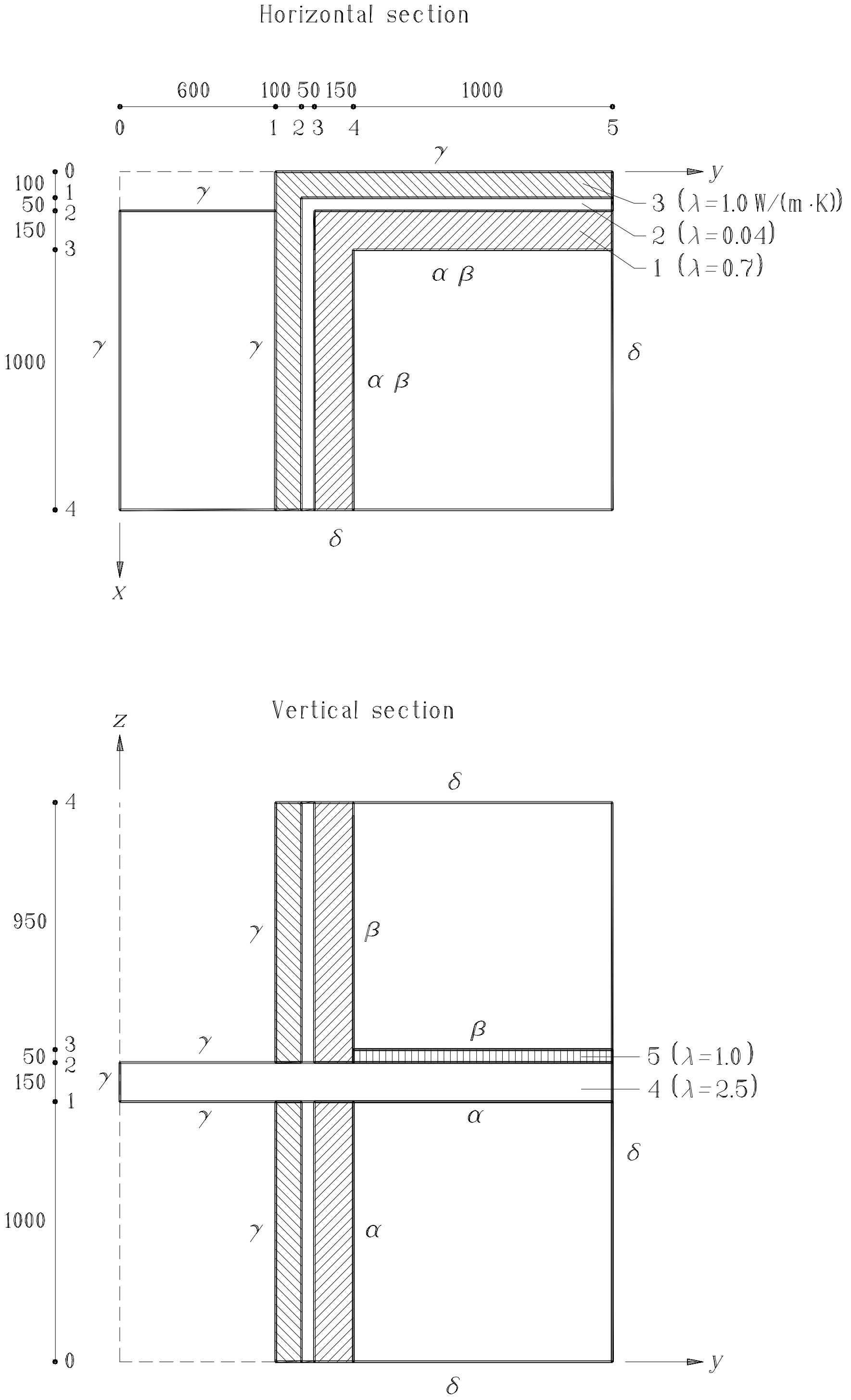


Figure 8.13: Horizontal and vertical sections of the problem. Input mesh coordinates and length of input mesh segments are shown.

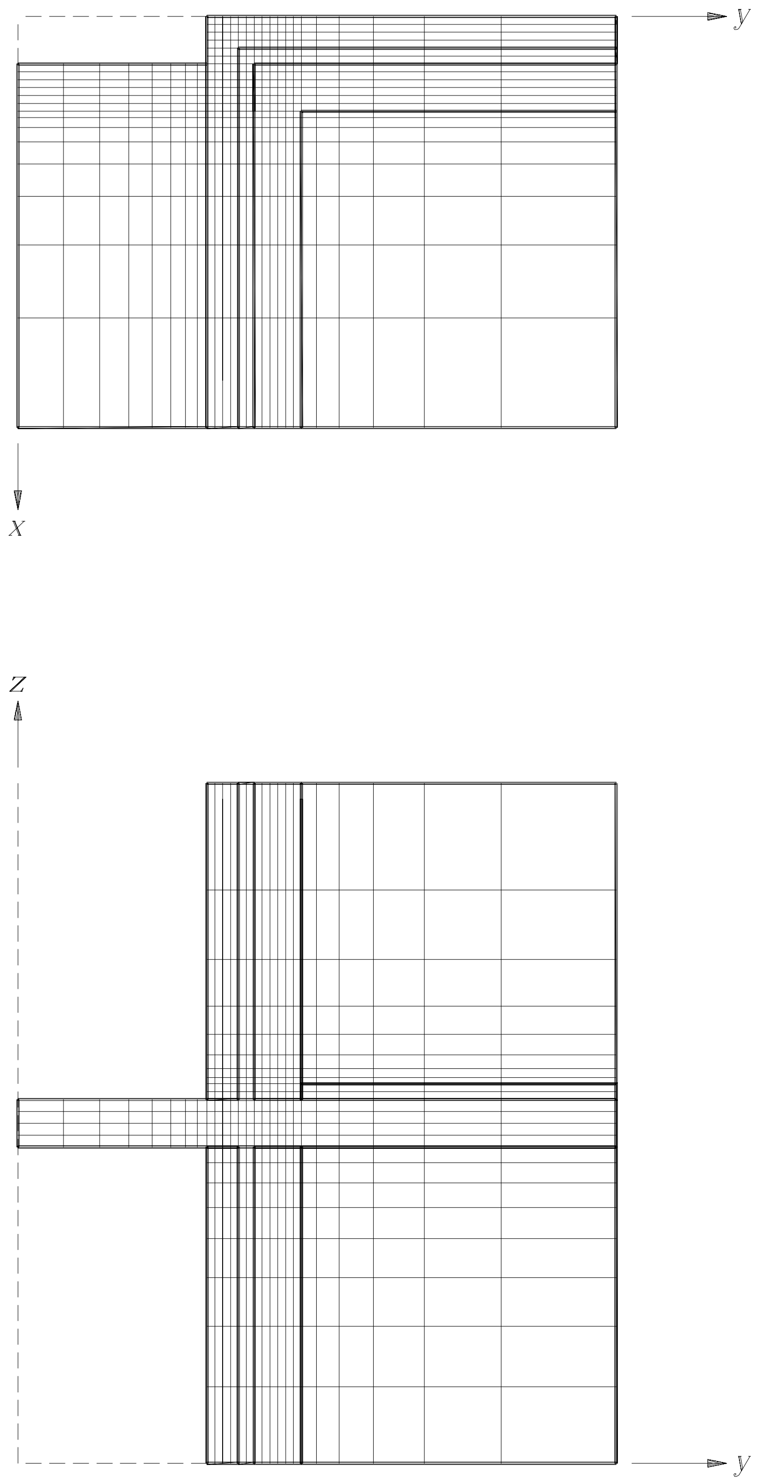


Figure 8.14: Projection of the numerical mesh on the *x*,*y*-plane and the *x*,*z*-plane involving 7600 cells.

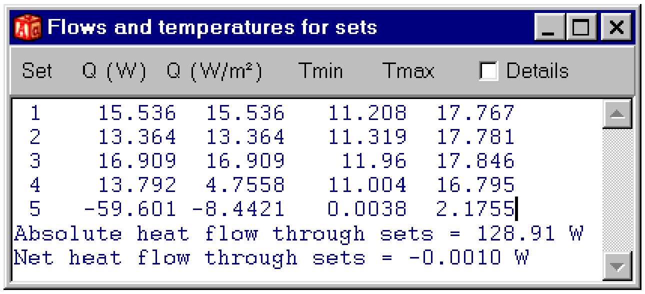
8.5.2 Input file, *cen.dat*

```
4                 x-dir
4 2 6 8           number of cells
1 1 1 1.5         expansion coeff. 1.5 for last segment
0.1 .05 .15 1     lengths

5                 y-dir
8 4 2 6 6
.8 1 1 1 1.5
.6 .1 .05 .15 1

4                 z-dir
8 4 2 8
.8 1 1 1.5
1 .15 .05 .95

8                 Boxes
% box coordinates, conductivity, capacity, initial temp, optional color in graphics
0 1 0  4 2 4          1            1.0E6    0.0      3
0 1 0  1 5 4          1            1.0E6    0.0      3
1 2 0  4 3 4       0.04            1.0E6    0.0      2
1 2 0  2 5 4       0.04            1.0E6    0.0      2
2 3 0  4 4 4        0.7            1.0E6    0.0      1
2 3 0  3 5 4        0.7            1.0E6    0.0      1
2 0 1  4 5 2        2.5            1.0E6    0.0      4
3 4 2  4 5 3        1.0            1.0E6    0.0      5

0         no empty boxes

4         four BC types
0 0.0
1 20.0 0.2
1 15.0 0.2
1 0.0 0.05

12         BC planes
3 4 0  4 4 1  2
3 4 0  3 5 1  2
3 4 1  4 5 1  2
3 4 3  4 4 4  3
3 4 3  3 5 4  3
3 4 3  4 5 3  3
0 1 0  4 1 4  4
0 0 0  0 5 4  4
0 0 0  4 0 4  4
2 0 0  2 1 4  4
0 0 1  4 1 1  4
0 0 2  4 1 2  4
0         no internal resistance
5         number of sets
1  1            ('alpha' surface)
1  2            ('alpha' surface)
1  3            ('alpha' surface)
```

```
3  4 5 6           (all 'beta' surfaces)
6  7 8 9 10 11 12   (all 'gamma' surfaces)
0            no heat sources
```

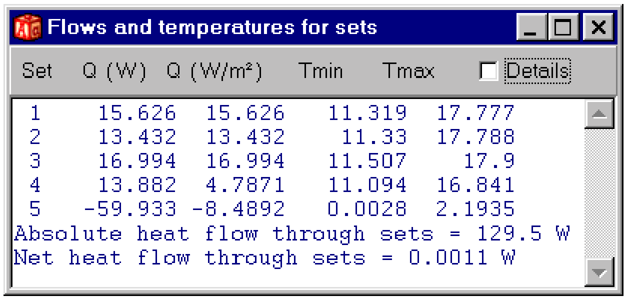
8.5.3 Results

Figure 8.15 and Figure 8.16 show the heat flows using 7600 and 246000 numerical cells, respectively. Consider Figure 8.15. The first three sets show the heat flows for surfaces α. The total heat flow is 15.54+13.36+16.91 = 45.81 W. Set 4 shows the flow through all three β-surfaces (13.79 W), and set 5 shows the flow through all six γ-surfaces (59.6 W).



```
Set   Q (W)  Q (W/m²)    Tmin     Tmax
 1     15.536  15.536   11.208   17.767
 2     13.364  13.364   11.319   17.781
 3     16.909  16.909    11.96   17.846
 4     13.792  4.7558   11.004   16.795
 5    -59.601 -8.4421   0.0038   2.1755
Absolute heat flow through sets = 128.91 W
Net heat flow through sets = -0.0010 W
```

Figure 8.15: Flows through defined sets, *N*=7600.



```
Set   Q (W)  Q (W/m²)    Tmin     Tmax
 1     15.626  15.626   11.319   17.777
 2     13.432  13.432    11.33   17.788
 3     16.994  16.994   11.507     17.9
 4     13.882  4.7871   11.094   16.841
 5    -59.933 -8.4892   0.0028   2.1935
Absolute heat flow through sets = 129.5 W
Net heat flow through sets = 0.0011 W
```

Figure 8.16: Flows through defined sets, *N*=246000.

Table 8:5 shows the calculated heat flows as a function of number of computational nodes, and the values listed in the CEN document for this test reference case. According to the CEN document, the difference between the heat flows calculated by the method being validated, in this case HEAT3, and the listed values should not differ with more than 2 % of the listed values. Consider the case with 7600 cells. The relative error of the heat flows turns out to be 1-2 % (if the listed values from CEN are correct). If one compares the results from HEAT3 with 7600 cells and 720900 cells, the error is less than 1 %. The calculation is shown for a Pentium/233 (the relaxation coefficient is set to ω=1.85, which is an optimal value).

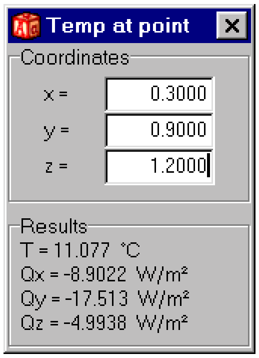
Surfaces	*N*=7600 CPU=6 sec.	*N*=246000 CPU=8 min.	*N*=720900 CPU=44 min.	CEN
α	45.81	46.02	46.07	46.3
β	13.79	13.88	13.89	14.0
γ	-59.60	-59.93	59.96	60.3

Table 8:5: Calculated heat flows (W).

Table 8:6 shows the surface temperatures obtained in HEAT3. According to the CEN document, the difference between the temperatures calculated by the method being validated and the listed values should not exceed 0.1 ºC. Figure 8.17 shows temperatures and heat flows at point *y* (*N*=246000).

Point	(*x*,*y*,*z*)	*N*=7600	*N*=246000	*N*=720900	CEN
U	1.3, 0.9, 1.0	13.0	13.0	12.9	12.9
V	0.3, 0.9, 1.0	11.4	11.4	11.3	11.3
W	0.3, 1.9, 1.0	16.5	16.4	16.4	16.4
X	1.3, 0.9, 1.2	12.5	12.6	12.6	12.6
Y	0.3, 0.9, 1.2	11.0	11.1	11.1	11.1
Z	0.3, 1.9, 1.2	15.2	15.3	15.3	15.3

Table 8:6: Calculated surface temperatures (ºC).



Temp at point

Coordinates

x = 0.3000

y = 0.9000

z = 1.2000

Results

T = 11.077 °C

Qx = -8.9022 W/m²

Qy = -17.513 W/m²

Qz = -4.9938 W/m²

Figure 8.17: Temperatures and heat flows at point *y*, see Figure 8.12, *N*=246000.

The first 12 rows in the input file *cen.dat* are changed as below for the cases with 246000 cells (values to the left) and with 720900 cells (values to the right). Note that only the number of cells and the expansion coefficients have to be changed (marked in bold-face).

4	4
10 5 15 35	**20 10 25 45**
1 1 1 1.05	**1 1 1 1.05**
.1 .05 .15 1	.1 .05 .15 1
5	5
15 10 5 15 30	**20 15 10 20 35**
.85 1 1 1 1.07	**0.9 1 1 1 1.06**
.6 .1 .05 .15 1	.6 .1 .05 .15 1
4	4
28 15 5 27	**35 20 12 33**
.93 1 1 1.07	**.94 1 1 1.06**
1 .15 .05 .95	1 .15 .05 .95

8.6 Example 8, slab on the ground, slab.dat

8.6.1 Introduction

The following example concerns a slab on the ground, see Figure 8.18. The simplification as shown in Figure 8.19 is a typical example how to deal with sloping boundaries. We are interested to know the heat flow from the building to the ground during a year. A steady-state calculation is made with an indoor temperature of 20 ºC, and an average annual outdoor temperature of 9.4 ºC. Table 8:7 shows the material properties. The internal and external surface resistances are 0.13 m²·K/W and 0.04 m²·K/W, respectively. A study dealing with frost penetration under a building is presented in (Harderup et al, 1994).

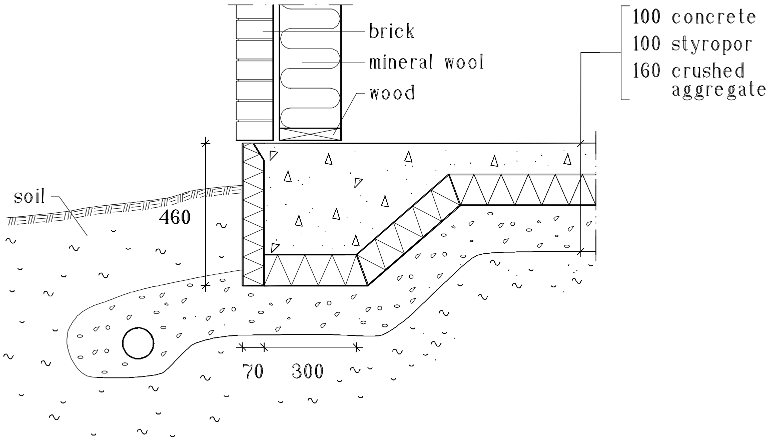


Figure 8.18: Slab on the ground.

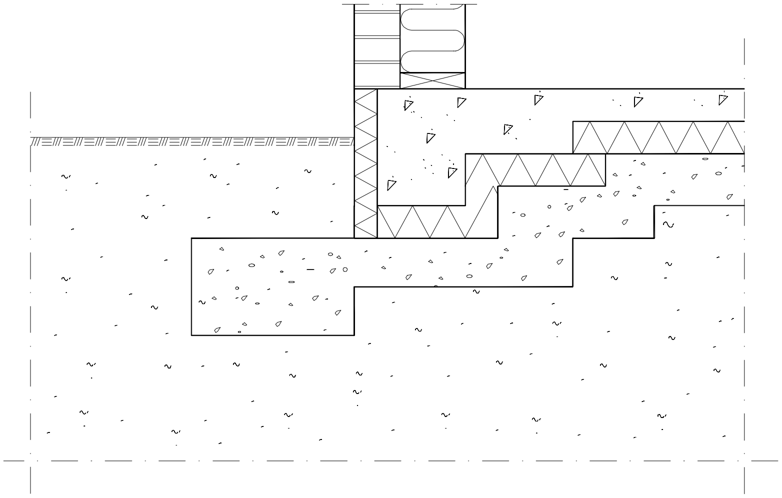


Figure 8.19: Simplified geometry for the calculations.

Material	Thermal conductivity (W/(m·K))
Soil	2.2
Crushed aggregate	2.0
Brick	0.44 (0.60)
Wood	0.14
Concrete	1.7
Styropor	0.039
Mineral wool	0.036

Table 8:7: Thermal conductivities.

The resistance for the air gap in the wall is taken into account in such way that an effective thermal conductivity 0.44 W/(m·K) for the brick is used. The size of the house, counted between the interior sides of the walls, is 10·8 m². Due to symmetry, only one quarter of the house is considered. see Figure 8.20 shows the input mesh coordinates and the mesh lengths. Due to symmetry the description in the x- and y-directions is equal (except for the last segment 10 where the length in the y-direction is 3.42 m, see Figure 8.20).

Each input parallelepiped in Figure 8.20 has a number in a circle referring to a parallelepiped in the input data file, see the next section. Two numbers together means that there are two boxes extending in the *x*- and *y*-directions, respectively. Note that for overlapping boxes, the last mentioned prevails. Consider as an example box 1, 2, and 3. Box 1 (soil) has the coordinates (0,0,0,10,10,7). The overlapping box 2 (crushed aggregate) has the coordinates (1,1,1,10,10,3). Box 3 (soil) (7,7,2,10,10,3) will overlap box 2, etc. Box 6 and 7 have the coordinates (6,6,3,9,10,5) in the y-direction, and (6,6,3,10,9,5) in the x-direction (note the symmetry). This provides a rather compact way to describe the geometry.

The input in a complicated case as this one is quite tricky, but it is fairly simple to graphically check that the input is correct. Figure 8.22 shows the numerical mesh near the house.

The calculations are made for an area extending 20 m outwards and 20 m downwards from the house. A rule of thumb is to extend at least twice the width of the house, in this case about 20 meters. If 40 m (65426 cells) would be used instead of 20 m (25221 cells), the results would differ with less than 0.1% for this particular case.

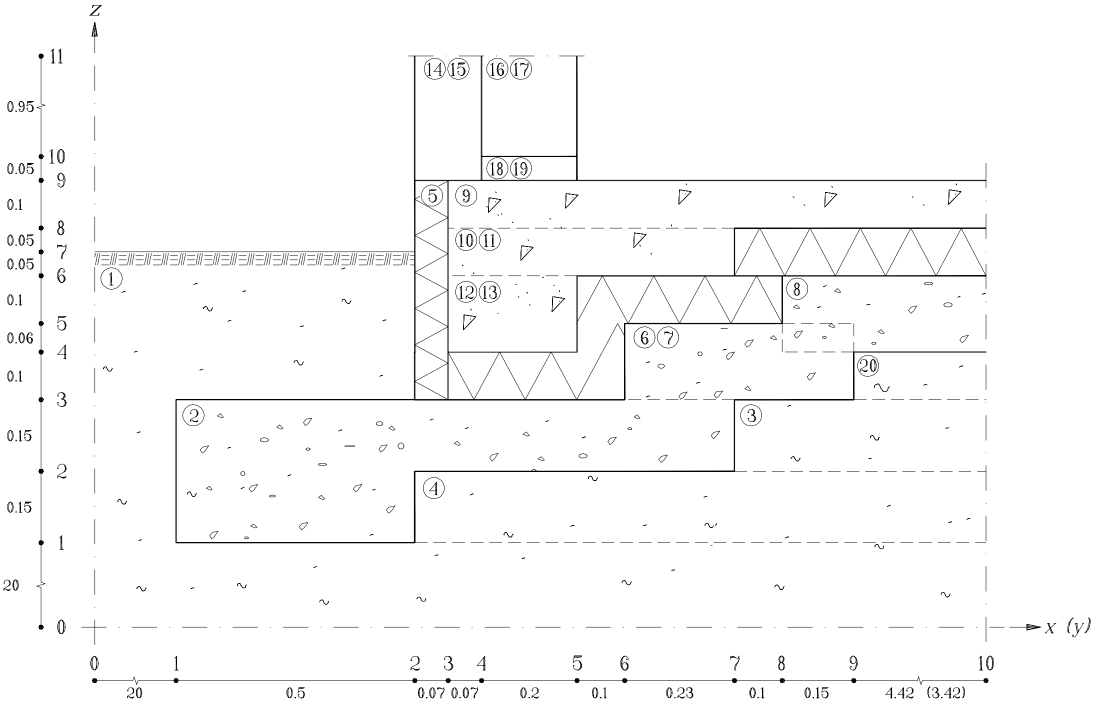


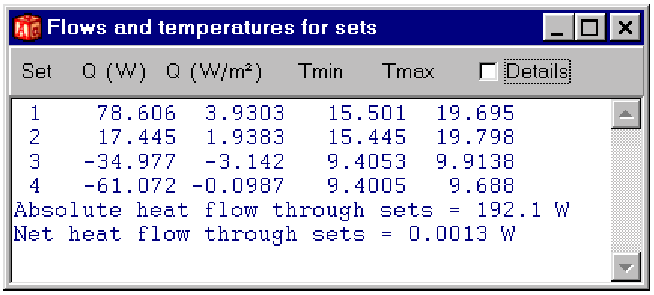
Figure 8.20: Input mesh coordinates and length of input mesh segments are shown. Each parallelepiped has a number in a circle referring to a parallelepiped in the input data file.

8.7 Input file, slab.dat

```
10  x-dir
11 5 1 1 2 1 2 1 1 10
0.7 1 1 1 1 1 1 1 1 1.2
20 0.5 .07 .07 .2 .1 .23 .10 .15 4.42
10  y-dir
11 5 1 1 2 1 2 1 1 10
0.7 1 1 1 1 1 1 1 1 1.2
20 0.5 .07 .07 .2 .1 .23 .10 .15 3.42
11  z-dir
11 2 2 1 1 1 1 1 1 1 8
.68 1 1 1 1 1 1 1 1 1 1.05
20 .15 .15 .1 .06 .1 .05 .05 .1 .05 .95

20
% box coordinates, conductivity, capacity, initial temp, optional color in graphics
%               lambda  C  T    color in graphics               (see Figure 8.20)
0 0 0 10 10 7    2.2    1 9.4  1  soil                         circle 1
1 1 1 10 10 3    2.0    1 9.4  2  crushed aggregate            circle 2
7 7 2 10 10 3    2.2    1 9.4  1  soil                         circle 3
2 2 1 10 10 2    2.2    1 9.4  1  soil                         circle 4
2 2 3 10 10 9  0.039  1 9.4  3  styropor                     circle 5
6 6 3 9 10 5     2.0    1 9.4  2  crushed aggregate            circle 6
6 6 3 10 9 5     2.0    1 9.4  2  crushed aggregate            circle 7
8 8 4 10 10 6    2.0    1 9.4  2  crushed aggregate            circle 8
3 3 8 10 10 9    1.7    1 9.4  4  concrete                     circle 9
3 3 6 7 10 8     1.7    1 9.4  4  concrete                     circle 10
3 3 6 10 7 8     1.7    1 9.4  4  concrete                     circle 11
3 3 4 5 10 6     1.7    1 9.4  4  concrete                     circle 12
3 3 4 10 5 6     1.7    1 9.4  4  concrete                     circle 13
2 2 9 4 10 11   0.44   1 9.4  5  brick                        circle 14
2 2 9 10 4 11   0.44   1 9.4  5  brick                        circle 15
4 4 10 5 10 11 0.036 1 9.4  6  mineral wool                 circle 16
4 4 10 10 5 11 0.036 1 9.4  6  mineral wool                 circle 17
4 4 9 5 10 10   0.14   1 9.4  7  wood                         circle 18
4 4 9 10 5 10   0.14   1 9.4  7  wood                         circle 19
9 9 3 10 10 4    2.2    1 9.4  1  soil                         circle 20

0   empty box

3   three BC type numbers
0 0.0
1 9.4  0.04
1 20.0 0.13

7   seven surfaces, all other boundaries will be adiabatic
5 5 9 10 10 9  3 floor
5 5 9 5 10 11  3 wall interior side
5 5 9 10 5 11  3 wall interior side
2 2 7 2 10 11  2 wall exterior side
2 2 7 10 2 11  2 wall exterior side
0 0 7 2 10 7    2 ground surface
2 0 7 10 2 7    2 ground surface

0   no internal resistance
4   number of sets for which to present flows
1 1       floor
2 2 3    internal sides of walls
2 4 5    external sides of walls
2 6 7    ground
0  no heat sources
```

8.8 Results

Figure 8.21 shows the heat flow through the floor (set 1), through the walls (sets 2 and 3), to the ground (set 4), and the minimum and maximum surface temperatures. Using the criterion F=$1\cdot10^{-3}$ instead of $1\cdot10^{-5}$ would in this case give heat flows that differ by less than 1 %, but this calculation takes 25 seconds instead of 49 seconds (Pentium/233). The heat flow to the ground from the house during one year becomes 61.1·4·24·365=2.1 MWh. Note that the calculated heat flows represent one quarter of the house, so we multiply by 4. Figure 8.22 shows the surface temperatures in gray-scale near the house. Figure 8.23 shows the temperatures on the concrete surface shown from below. The soil and the crushed aggregate are invisible here (color is zero for the boxes defined in the input). Figure 8.24 shows the temperatures on the soil surface. Everything but the soil is invisible here (color is zero for the boxes defined in the input).



Flows and temperatures for sets

Set Q (W) Q (W/m²) Tmin Tmax □ Details

```
 1     78.606  3.9303   15.501  19.695
 2     17.445  1.9383   15.445  19.798
 3    -34.977  -3.142   9.4053  9.9138
 4    -61.072 -0.0987   9.4005   9.688
Absolute heat flow through sets = 192.1 W
Net heat flow through sets = 0.0013 W
```

Figure 8.21: Calculated heat flows and surface temperatures, (N=25221).

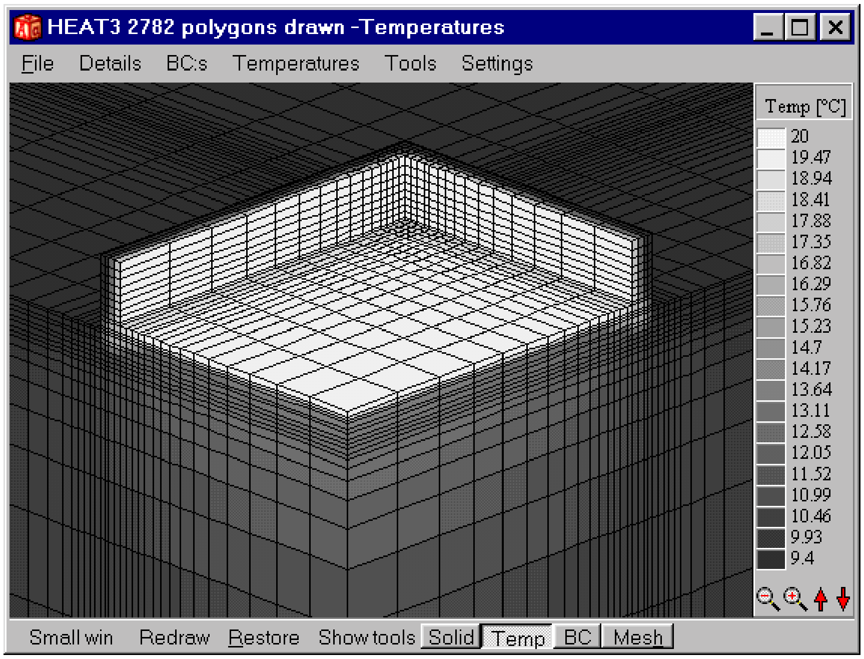


Figure 8.22: Surface temperatures in gray-scale and the mesh around the house.

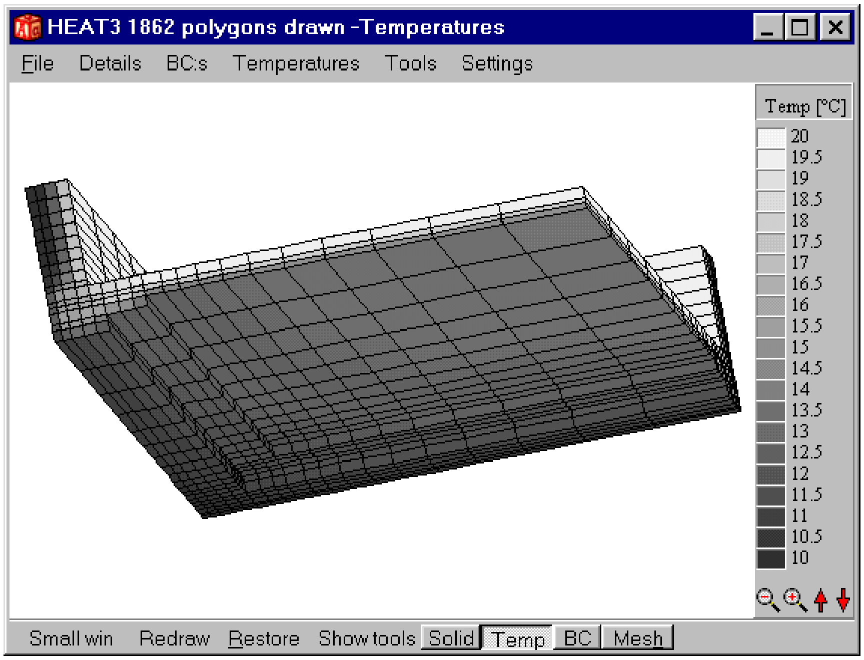


Figure 8.23: Temperatures on the concrete surface shown from below. The soil and the crushed aggregate are invisible here (color is zero for the boxes defined in the input).

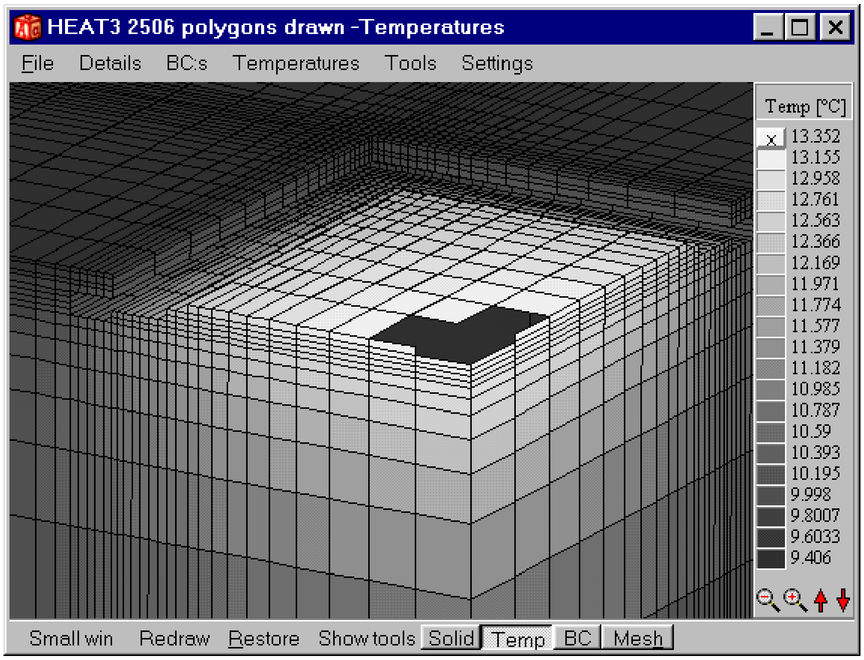


Figure 8.24: Temperatures on the soil surface. Everything but the soil is invisible here (color is zero for the boxes defined in the input). The maximum temperature in the soil is 13.4 ºC (these surfaces are made invisible by disabling the interval on the scale to the right).

8.9 Example 9, ground heat storage, cav.dat

8.9.1 Introduction

Rock caverns may be used for heat storage, here exemplified by the underground storage in Skarvik, Sweden. A comprehensive analysis is presented in (Claesson et al, 1989). The caverns there, used earlier as oil depot, may be filled with water and then used as a heat store. The heat loss from one such cavern is examined here. A more detailed study of the problem is given in (Blomberg, 1996).

Consider Figure 8.25 showing an underground cavern. The length, height and width of the cavern are L=180 m, H=30 m, and B=18 m, respectively. The distance from the upper surface in the cavern to the ground surface is 30 m. The thermal conductivity of the ground is 3.5 W/(m·K).

There are two symmetry planes, the *x*,*z*-plane, and the *y*,*z*-plane, see Figure 8.26. The calculations are made for one quarter of the entire area that extends 500 m outwards and 500 m downwards from the cavern. The temperature on the surfaces in the cavern is put to 1 ºC, and the temperature at the ground surface is put to 0 ºC. The input data file in the next section gives a numerical mesh with 15968 cells. Figure 8.26 shows the input mesh. Figure 8.27 shows the numerical mesh with 15968 cells (projected on the *x*,*z*-plane, *x*,*y*-plane, and *y*,*z*-planes).

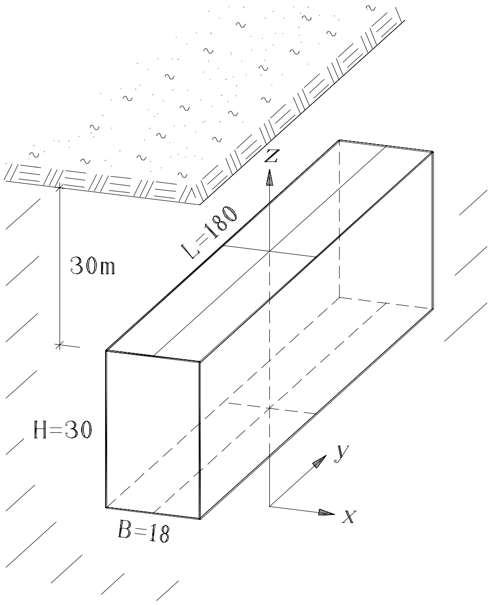


Figure 8.25: Underground cavern used for heat storage.

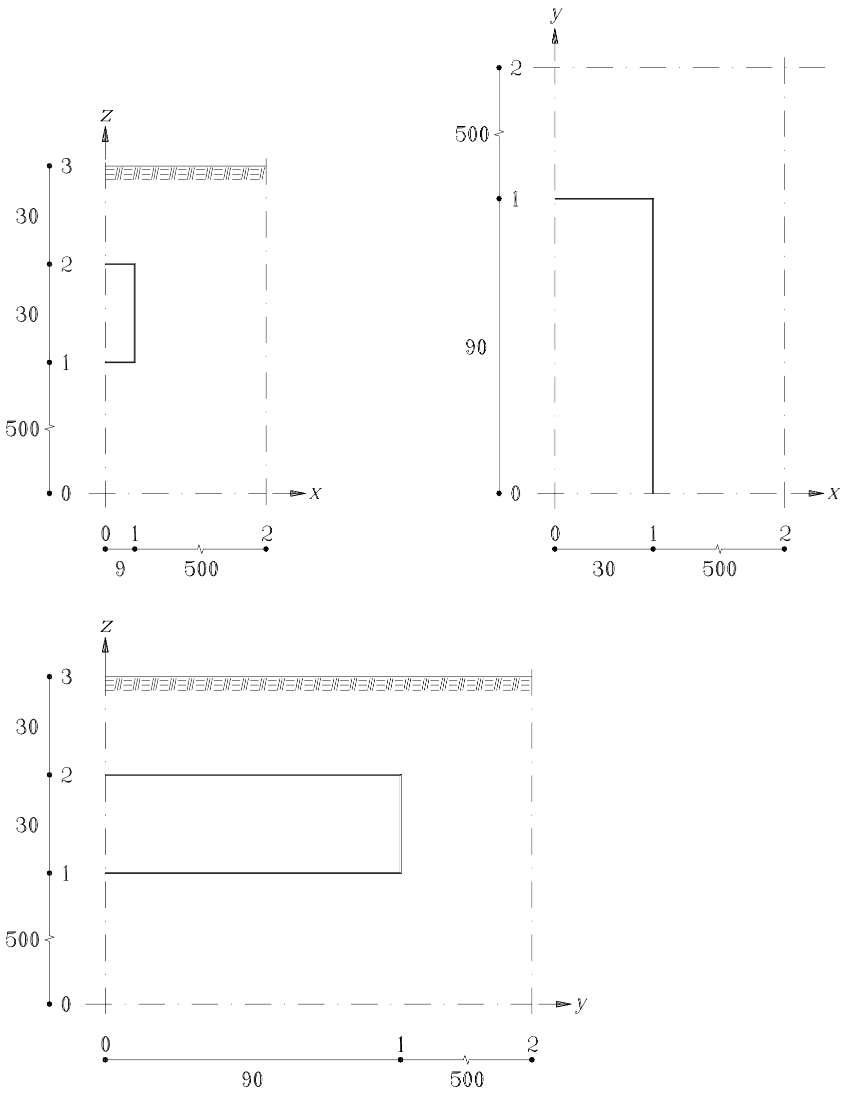


Figure 8.26: Input mesh coordinates and lengths (m).

8.9.2 Input file, cav.dat

```
2       x-dir
2 20
0.5 1.15
9 500

2       y-dir
6 20
0.7 1.15
90 500

3       z-dir
20 4 4
0.9 1 1.2
500 30 30

1       one box
% box coordinates, conductivity, capacity, initial temp, optional color in graphics
0 0 0  2 2 3           3.5        2.19E6      0.0           1

1       one empty box (the cavern)
0 0 1  1 1 2

3       three BC type numbers
0 0       (adiabatic)
1 1.0 0.0               (T=1, R=0)
1 0.0 0.0               (T=0, R=0)

5       five surfaces
1 0 1  1 1 2  2  plane with normal in x-direction in cavity
0 1 1  1 1 2  2  plane with normal in y-direction in cavity
0 0 1  1 1 1  2  bottom of cavern
0 0 2  1 1 2  2  top of cavern
0 0 3  2 2 3  3  ground surface

0       no internal resistance planes

2       two sets
4 1 2 3 4    all surfaces in the cavern
1 5          ground surface

0      no heat sources
```

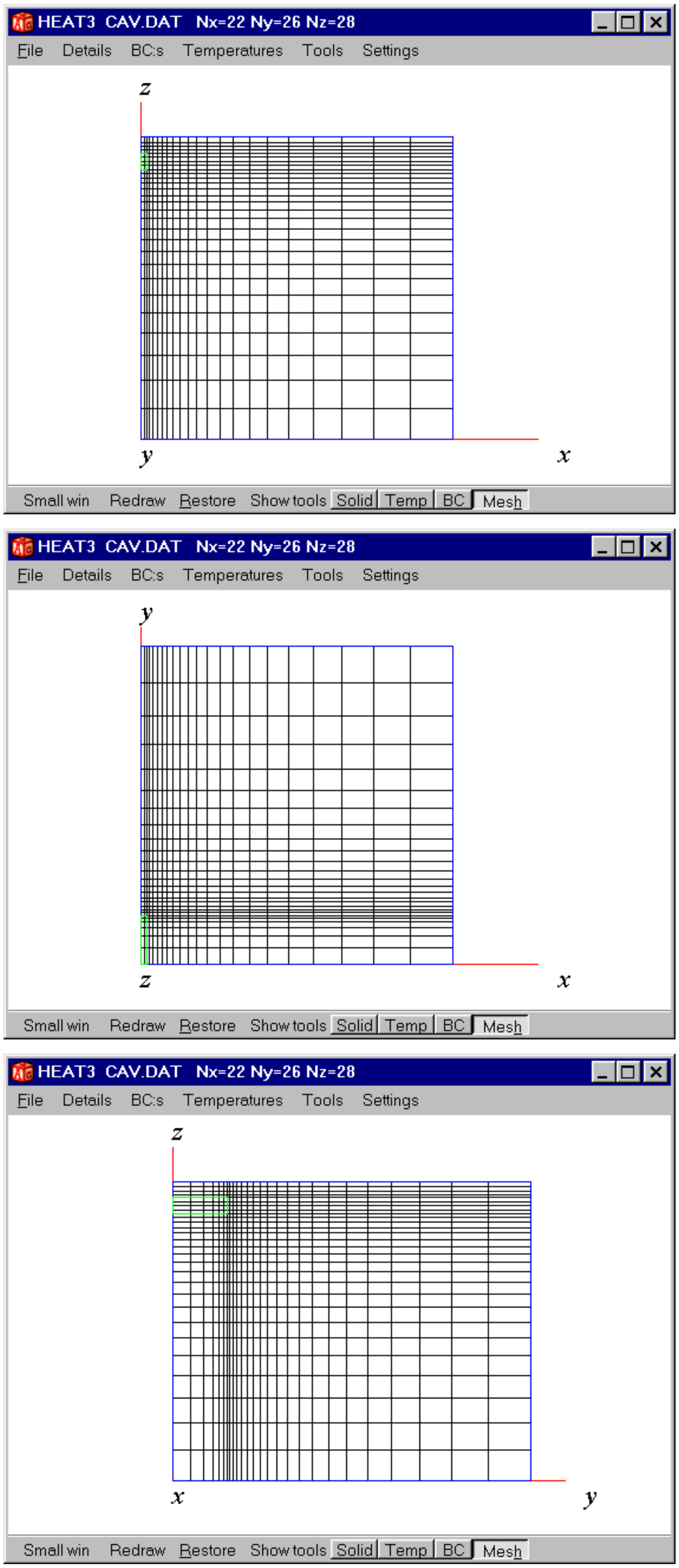
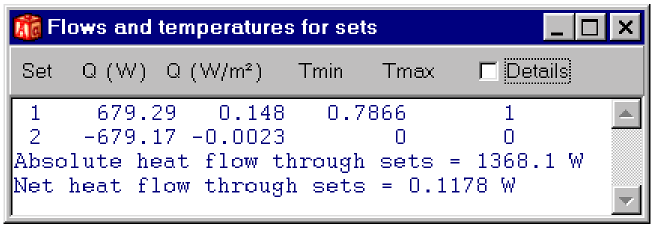

HEAT3 CAV.DAT Nx=22 Ny=26 Nz=28
File Details BC:s Temperatures Tools Settings
z
y
x
Small win Redraw Restore Show tools Solid Temp BC Mesh
HEAT3 CAV.DAT Nx=22 Ny=26 Nz=28
File Details BC:s Temperatures Tools Settings
y
z
x
Small win Redraw Restore Show tools Solid Temp BC Mesh
HEAT3 CAV.DAT Nx=22 Ny=26 Nz=28
File Details BC:s Temperatures Tools Settings
z
x
y
Small win Redraw Restore Show tools Solid Temp BC Mesh

Figure 8.27: Numerical mesh with 15968 cells (mesh projected on the x,z-plane, x,y-plane, and y,z-plane, respectively).

8.9.3 Results

The heat flow from the cavern is 679·4=2716 W, see Figure 8.28. Note that the calculated heat flows represent one quarter of the cavern, so we multiply by 4. The calculation took 19 sec. on a Pentium/233. The error compared to a calculation with 135000 cells (Q=2800) is 3 %. This calculation took about 13 min. Figure 8.29 shows the temperatures and meshes with *N*=15968 (above) and *N*=135250 (below).



```
Set   Q (W)  Q (W/m²)    Tmin     Tmax
 1     679.29    0.148    0.7866        1
 2    -679.17 -0.0023          0        0
Absolute heat flow through sets = 1368.1 W
Net heat flow through sets = 0.1178 W
```

Figure 8.28: Heat flows for the two sets, *N*=15968.

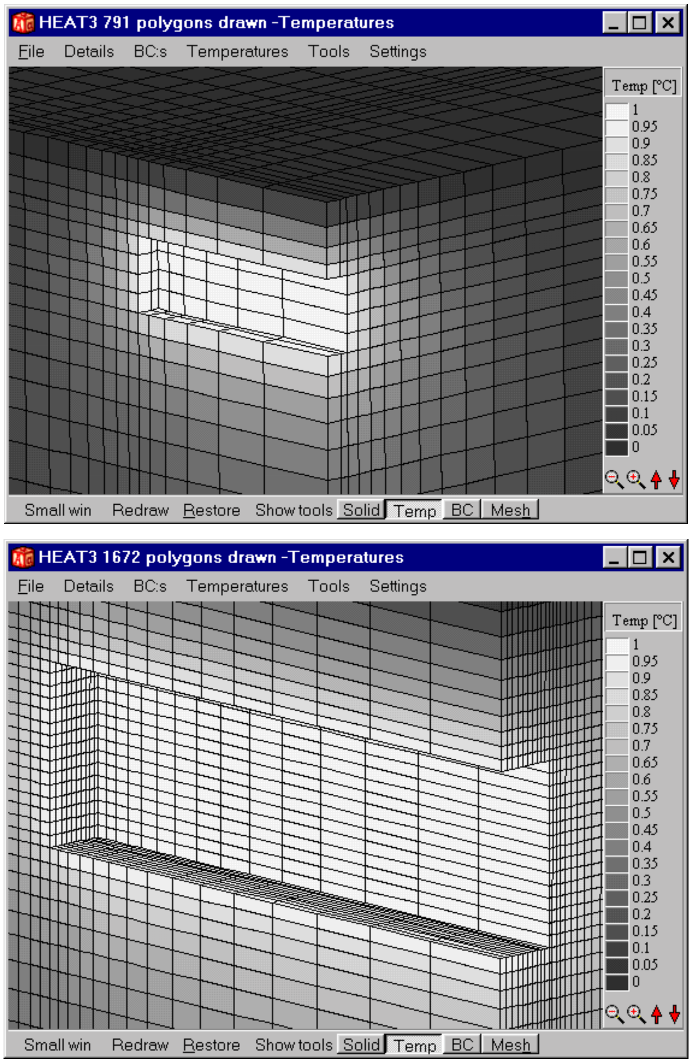


Figure 8.29: Temperatures and mesh near the cavern with *N*=15968 (above) and *N*=135250 (below).

9. A few tips

9.1 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.

9.2 Numerical aspects

9.2.1 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 2*n* cells. The relative difference between the flows must be smaller than 2%. Further mesh division is required if the difference is larger.

9.2.2 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.

9.2.3 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.

9.2.4 Symmetric cases

Use symmetry whenever possible. A slab on the ground is one example when it often is sufficient to consider a quarter of the problem.

9.2.5 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. It should be sufficient to use an extension twice the width of the house for a three-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).

9.2.6 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. 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. 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 including the initial temperatures (the mean annual air temperature is a good choice). Do not forget the 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 consequent years).
5. A sudden cold spell (or several) 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.

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. 1996. Thesis. H*eat 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.

Blomberg, T. 1997. *HEAT2 - A two-dimensional heat transfer computer program. Manual for HEAT2*. Department of Building Physics, Lund University. P.O. Box 118, S-221 00 Lund, Sweden. CODEN:LUTVDG/(TVBH-7198).

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.

Claesson, J., T. Blomberg, P. Wallentén. 1989. *Korttidsvärmelagring i bergrum*. Department of Building Physics, Lund University. P.O. Box 118, S-221 00 Lund, Sweden.

Claesson, J. C-E. Hagentoft. 1994. *Basic Building Physics.* Department of Building Physics, Lund University. P.O. Box 118, S-221 00 Lund, Sweden.

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

Foley J.D., Van Dam A. 1984. *Fundamentals of interactive computer graphics.* Addison-Wesley Publishing Company, Reading, Massachusetts. ISBN 0-201-14468-9.

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.

Harderup L-E, Blomberg T, Elmroth A. 1994. *Die Frosteindringung bei der Grundlegung für Gebäude ohne Unterkellerung* (Frost penetration under the foundation of a building without cellar, written in German). Department of Building Physics, Lund University. P.O. Box 118, S-221 00 Lund, Sweden. Report TVBH-7174.

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

MathWorks Inc. 1992. *Matlab*. Cochituate Place, 24 Prime Park Way, Natick, Massachusetts 01760, USA.

Appendix A. Thermal data

The values below are taken from Catalogue of Material Properties, International Energy Agency, report Annex XIV, Volume 3. Average data is taken for density, specific heat capacity, and thermal conductivity.

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

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. $C = \rho \cdot c_p$.

IEA Report Annex XIV

Material	C, (J/(m³·K))			λ, (W/(m·K))
Brick	1.49	.45		
Concrete		1.83	2.7	
Concrete cellular	.53	.52		
Concrete lightweight	1.83	.49		
Cork	.21	.042		
Glass (Window)	2.3	1.0		
Glass-wool	.062	.033		
Gypsum board	.88	.1		
Masonry	.88	.1		
Mortar		1.21	.53	
Plywood	1.17	.125		
Polystyrene expanded	.039	.033		
Polystyrene extruded	.059	.024		
Rock-wool	.094	.037		
Steel	3.7	60		
Wood (Oak)		1.7	.19	
Woodwool cement	.64	.071		

CEN (window materials, no capacities)

Material	C	λ
METALS		
Aluminum alloy	1	160
Steel	1	50
Stainless steel	1	17
WOODS		
Hardwood	1	.2
Softwood	1	.14
Plywood	1	.15
Redwood/cedar	1	.11
GLAZING MATERIALS		
Float Glass	1	1
Plexiglass	1	.2
FRAME MATERIALS		

Glass fibre resin	1	.3
PVC/Vinyl (rigid)	1	.17
Wood chip board	1	.13
Porous wood fibre panel	1	.08

SPACER MATERIALS

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

THERMAL BREAK MATERIALS

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

SOLID PLASTICS

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

SEALANTS

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

WEATHERSTRIPPING

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

Appendix B. Matlab examples

Here are some examples of the Matlab syntax. The following command that will change the view and label the axes for a cut-out temperature plane.

```
view([-1 1 1])
xlabel(’x (m)’)
ylabel(’y (m)’)
zlabel(’Temp ( C)’)
```

The following commands will change the view and label the axes for a 3-D figure.

```
view([-1 -1 1])
xlabel(’x (m)’)
ylabel(’y (m)’)
zlabel(’z (m)’)
```

The following script generates a contour plot. The first row ’test’ (an ’m-file’ created by HEAT3) will initialize the arrays z, x, and y. The second row gives a vector with temperatures where we want isotherms to be drawn. The third row draws contours at the values specified in vector V, where ’c’ is a handle to the figure (used in the last row). The fourth row sets the aspect ratio and the axis limits so the image has square pixels. The fifth row allows manual zooming using the mouse. The sixth row turns the zoom off. The two next rows label the axes. The last row places contour labels at the locations clicked on with the mouse. Pressing the return key terminates labeling.

```
test
v=([1 2 3 4 5 6 7 8]);
c=contour(x,y,z,v);
axis(’image’);
zoom;
zoom off;
xlabel(’y (m)’);
ylabel(’z (m)’);
clabel(c,’manual’);
```

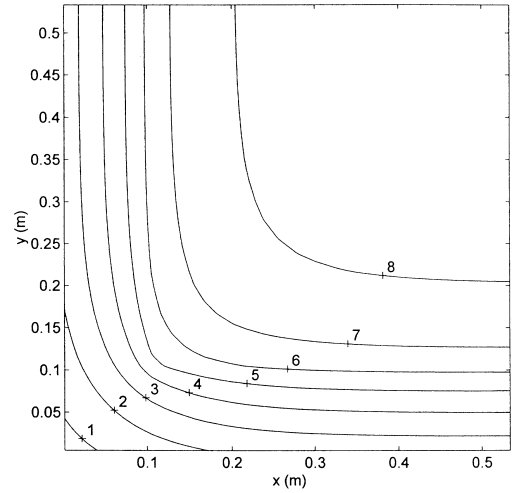


Figure B.9.1: Contour plot of a temperature field for a plane that cuts through the floor, see the corner problem in Section 7.11.

Appendix C. Important news for users of version 3

DAT-files and H3P-files

As long as input is made using the pre-processor (and the text input editor is not used) all relevant input data will be saved in a H3P-file. There is no need to save data in a DAT file since the appropriate input data is created when a H3P-file is loaded.

When input is made in text mode as in HEAT3 version 3 you should save the DAT file. The so-called option file .OPT will also be saved along with the data file as usual.

Material string in text editor

An optional material string may be given for each material box using in the text input editor, see Section 7.7:

```
2 material boxes
0 1 0   2 3 1   0.2 1E6     0   1 Mymaterial name
1 0 0   3 2 1   2.7 1.83E6  0   2 concrete, IEA
```

The post-processor figures showing material names, thermal conductivity and volumetric heat capacity are listed in the optional color (1 and 2 in the example above) that is specified before the material name string.

Using sets

When input using the pre-processor is made, each new defined boundary surface will be the same as a new set (in version 3 a set could be a collection of boundary surfaces). It is however still possible to collect multiple boundary surfaces for boundary conditions into sets in text input mode.

Stop criteria

The default stop criteria is changed to be the sum of all internal flows, i.e. the sum of all net flows flowing to each cell. The reason is that the default boundary condition may be non-adiabatic (e.g. a given temperature). Boundary heat flows for stop criterion type 1 (boundary heat flows) are only calculated for defined boundaries (not belonging to the default type) and will in that case never be zero.

However, if the default boundary condition is non-adiabatic it is fine to use stop criterion type 1 (boundary heat flows) instead.

The default stop criteria (sum of all internal flows) works well in most cases. However, it may be a good idea to change the limit (default is 0.0010 W) to suit the problem.