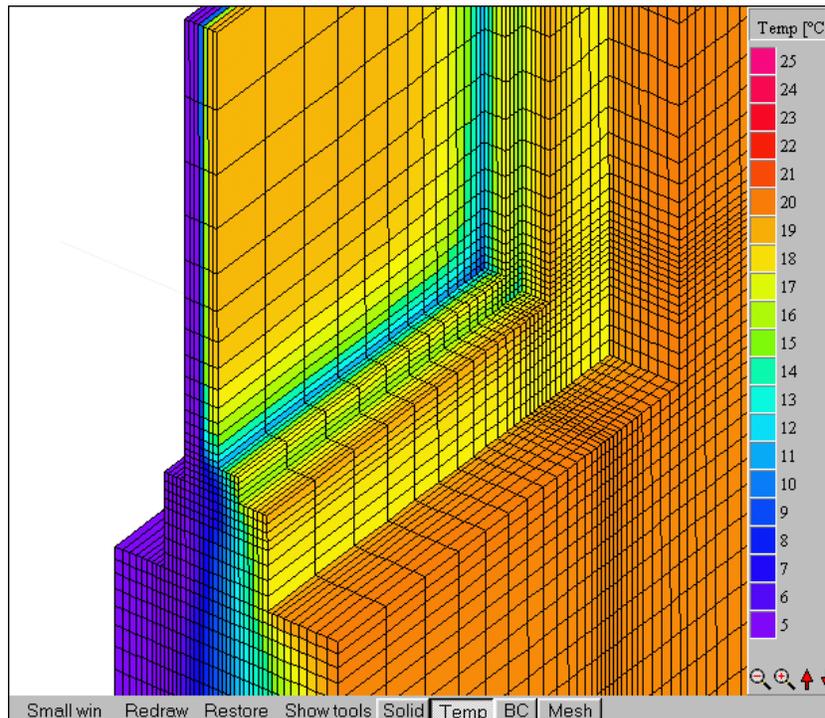


HEAT3

A PC-program for heat transfer in three dimensions.

Manual with brief theory and examples.

HEAT3 version 3.0



March 23, 1998

(Revised February 10, 1999)

Thomas Blomberg

Contents

1. INTRODUCTION	7
1.1 HOW TO AVOID READING THIS MANUAL	7
1.2 OVERVIEW	7
1.3 HEAT3 COMPARED TO HEAT2	7
1.4 VERSIONS OF HEAT3	7
1.5 INSTALLATION AND REQUIREMENTS	8
1.6 TECHNICAL SUPPORT	8
1.7 BENCHMARKS	8
2. MATHEMATICAL DESCRIPTION	9
2.1 GOVERNING DIFFERENTIAL EQUATIONS	9
2.2 BOUNDARY CONDITIONS	9
2.3 INITIAL CONDITIONS	10
3. NUMERICAL FORMULATION	11
3.1 INTRODUCTION	11
3.2 THERMAL CONDUCTANCES	11
3.3 HEAT FLOWS	12
3.4 NEW TEMPERATURES	13
3.5 CHOICE OF TIME-STEP	14
3.6 ITERATIVE CALCULATION	14
3.7 STEADY-STATE CASE	14
4. OVERVIEW OF INPUT	15
4.1 INPUT MESH	15
4.2 SURFACES DEFINING BOUNDARY CONDITIONS	17
4.3 NUMERICAL MESH	18
4.4 INTERNAL RESISTANCES	19
5. WORKING WITH HEAT3	21
5.1 EDITING DATA	21
5.1.1 <i>The editor</i>	21
5.1.2 <i>Inserting mesh coordinates</i>	22
5.2 SOLVING THE PROBLEM	24
5.2.1 <i>Steady-state stop criteria</i>	24
5.2.2 <i>Successive over-relaxation used for the steady-state calculations</i>	25
5.2.3 <i>Simulation time for transient calculation</i>	26
5.2.4 <i>Functions</i>	27
5.2.4.1 <i>Example 1 - sinusoidal function with a time period of one year</i>	29
5.2.4.2 <i>Example 2 - sinusoidal function with a time period of one day</i>	30
5.2.4.3 <i>Example 3 - step-wise constant function</i>	30
5.2.4.4 <i>Example 4 - step-wise linear function</i>	31
5.2.5 <i>Simulation window</i>	31
5.2.6 <i>Screen update</i>	32
5.2.7 <i>Output file options</i>	32
5.2.8 <i>Reset</i>	32
5.3 OUTPUT DATA	32
5.3.1 <i>Flows and temperatures for sets</i>	32
5.3.2 <i>Temperature and heat flows at a given point</i>	33
5.3.3 <i>Output files with temperature planes</i>	33

5.3.4	<i>Output files for Matlab 3D figures</i>	35
5.4	GRAPHICS	36
5.4.1	<i>Introduction</i>	36
5.4.2	<i>Basic commands</i>	37
5.4.3	<i>Menu item Details</i>	37
5.4.4	<i>Viewing thermal conductivities and heat capacities</i>	38
5.4.5	<i>Viewing boundary conditions</i>	39
5.4.6	<i>Viewing temperatures</i>	40
5.4.7	<i>Removing boxes in the picture</i>	43
5.4.8	<i>Cut-out areas</i>	43
5.4.9	<i>Saving images</i>	43
5.4.10	<i>Maximizing scales</i>	44
5.5	THE OPTIONS FILE	45
5.6	SAVING TEMPERATURE FIELD TO FILE	45
5.7	SAVING CONDUCTANCES, CAPACITIES, AND TEMPERATURES TO FILE	45
6.	FORMAT FOR INPUT DATA	47
6.1	INPUT DATA FILE	47
6.2	EXAMPLE 1	49
6.3	EXAMPLE 2	50
6.4	EXAMPLE 3	52
6.5	EXAMPLE 4	53
6.6	THE INFO LOG	55
7.	MORE EXAMPLES	57
7.1	INTRODUCTION TO EXAMPLES	57
7.1.1	<i>Example 5, metal studs in a wall</i>	57
7.1.2	<i>Example 6, slotted metal studs in a wall</i>	57
7.1.3	<i>Example 7, heat flow through a corner</i>	57
7.1.4	<i>Example 8, slab on the ground</i>	57
7.1.5	<i>Example 9, ground heat storage</i>	57
7.2	EXAMPLE 5, METAL STUDS IN A WALL, STEEL1.DAT	58
7.2.1	<i>Introduction</i>	58
7.2.2	<i>Input file steel1.dat</i>	60
7.3	RESULTS	61
7.4	EXAMPLE 6, SLOTTED METAL STUDS, STEEL2.DAT	63
7.4.1	<i>Introduction</i>	63
7.4.2	<i>Data for the numerical calculation</i>	63
7.4.3	<i>Numerical mesh</i>	64
7.4.4	<i>Input file, steel2.dat</i>	66
7.4.5	<i>Results</i>	67
7.5	EXAMPLE 7, HEAT FLOW THROUGH A CORNER, CEN.DAT	68
7.5.1	<i>Introduction</i>	68
7.5.2	<i>Input file, cen.dat</i>	71
7.5.3	<i>Results</i>	72
7.6	EXAMPLE 8, SLAB ON THE GROUND, SLAB.DAT	74
7.6.1	<i>Introduction</i>	74
7.7	INPUT FILE, SLAB.DAT	77
7.8	RESULTS	78
7.9	EXAMPLE 9, GROUND HEAT STORAGE, CAV.DAT	80
7.9.1	<i>Introduction</i>	80
7.9.2	<i>Input file, cav.dat</i>	82
7.9.3	<i>Results</i>	84
8.	A FEW TIPS	85

8.1	INTRODUCTION.....	85
8.2	NUMERICAL ASPECTS	85
8.2.1	<i>Numerical mesh</i>	85
8.2.2	<i>Expansive meshes</i>	85
8.2.3	<i>Steady-state calculations</i>	85
8.2.4	<i>Symmetric cases</i>	85
8.2.5	<i>Computational area for calculations of heat losses to the ground</i>	85
8.2.6	<i>Transient calculations for heat losses to the ground</i>	85
	REFERENCES	87
	APPENDIX A. THERMAL DATA.....	89
	APPENDIX B. MATLAB EXAMPLES	91

1. Introduction

1.1 How to avoid reading this manual

For a quick start read Chapter 4 (*Overview of input*), and follow one or two of the first examples given in Chapter 6. It is not necessary to read chapter 2 and 3 (mathematical description and numerical formulation) to use the program.

1.2 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. The steady-state problem shown on the front cover (7600 numerical cells) took 6 seconds to solve on a Pentium/233.

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.

1.3 HEAT3 compared to HEAT2

There are many similarities between HEAT2 (Blomberg, 1997), HEAT2R (Blomberg, 1994) and HEAT3. These two 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. Both of them use an *input mesh* to describe the geometry, the numerical mesh, and the boundary conditions.

1.4 Versions of HEAT3

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 available. One million require about 34 MB RAM, and 125000 cells can be addressed with 6 MB. Versions for other memory configurations are available upon request.

When using the larger version with one million cells, try to close other programs (especially those that require much memory).

Version	Maximum cells	Required memory	Recommended memory in PC
HEAT3-50	50·50·50=125000	6 MB	16 MB
HEAT3-C	100·100·100=1000000	34 MB	48 MB

Table 1.1: Versions of HEAT3.

1.5 Installation and requirements

Before using the installation diskette, it is recommended to make a backup copy. The following steps describe the installation process in detail:

1. Start Windows 95 or NT.
2. Insert the installation diskette.
3. Run "A: Setup".
4. Follow the instructions on the screen.
5. Read README.TXT (if present) for update information.

The following files are used:

- HEAT3-50.EXE and HEAT3-C.EXE are the main programs.
- Files with extension DAT are input data files.
- Files with extension OPT are files with settings for each input data file.
- Files with extension FUN are files with function values.
- Files with extension IJK contain the temperature field.

A PC with 16 MB RAM running Windows 95 or Windows NT is required, see Table 1.1. Note that the temperature pictures will look better if you use at least 16-bit colors ("High Color").

1.6 Technical support

News are regularly updated on the WWW at <http://www.blocon.se>. Questions and comments may be sent by email to info@blocon.se or by fax to +46 46 136264.

1.7 Benchmarks

It might be of interest to compare the numeric performance for different computers. Table 1.2 shows the calculation speed relative a Pentium 120 using HEAT3. The Pentium Pro 200 is surprisingly fast. The Pentium 233 MMX is here surprisingly slow relative to a Pentium 120. The results depend of course also on motherboard performance, cache memory, etc.

Processor	Speed relative a Pentium 120 MHz
Pentium 120	1.0
Pentium 233 MMX	1.4
Pentium Pro 200	3.4
Pentium II 333	3.8

Table 1.2: Numerical performance for four different computers using HEAT3.

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

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

Here I , (W/m^3), is the rate of internal heat generation. The thermal conductivity is denoted by λ , ($\text{W}/(\text{m}\cdot\text{K})$). The volumetric heat capacity is denoted by C , ($\text{J}/(\text{m}^3\cdot\text{K})$), which is the density ρ , (kg/m^3), times the specific heat capacity c_p , ($\text{J}/(\text{kg}\cdot\text{K})$), i.e. $C = \rho \cdot c_p$. The 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 , ($\text{m}^2\cdot\text{K}/\text{W}$):

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

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

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

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

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

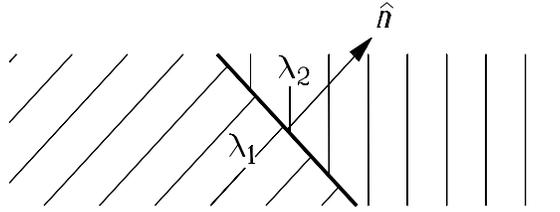


Figure 2.1: The normal \hat{n} at an internal boundary.

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

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

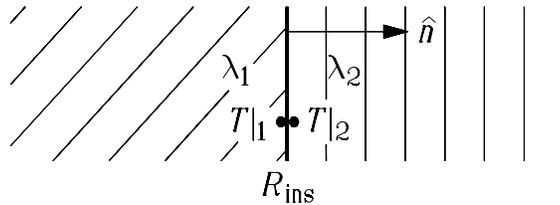


Figure 2.2: Case involving internal insulation.

2.3 Initial conditions

The initial temperature distribution at 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:

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

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.

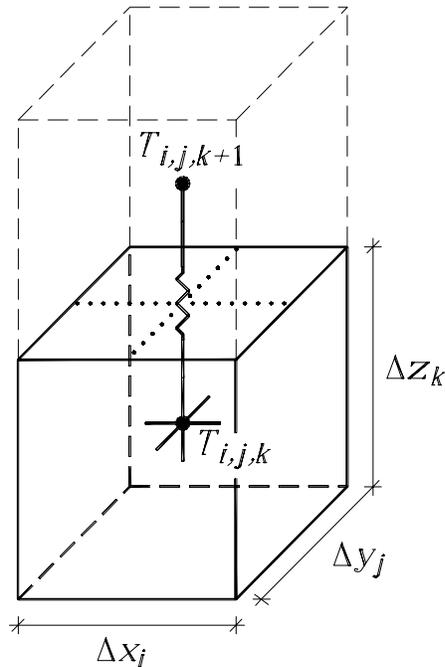


Figure 3.1: Computational cells (i,j,k) and $(i,j,k+1)$.

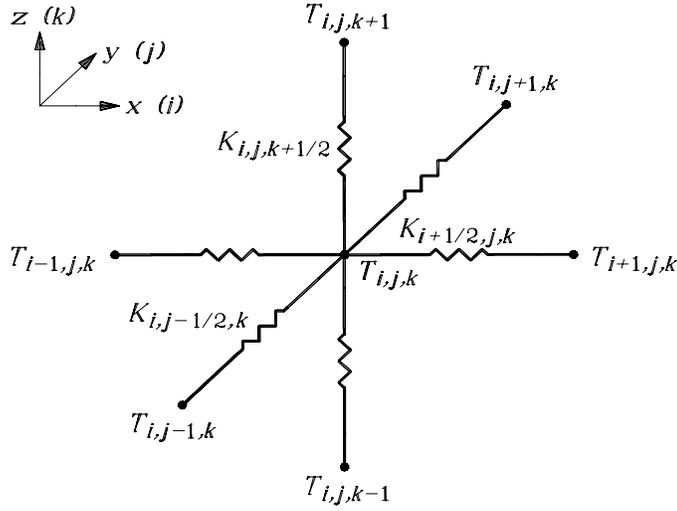


Figure 3.2: Thermal conductances connected to cell (i,j,k) .

Fig. 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):

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

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 $R_{i,j,k+1/2}$, ($\text{m}^2 \cdot \text{K/W}$), 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

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

Here, $R_{1/2,j,k}$, ($\text{m}^2 \cdot \text{K/W}$), 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),

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

Changes in the energy of a cell due to heat sources/sinks I , (W/m^3), 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:

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

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

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

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

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

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

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

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

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

3.5 Choice of time-step

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

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

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 (Efring, 1990).

3.6 Iterative calculation

Equation (3.5) gives the new temperature based on the change in energy during the time-step Δt . This calculation is made for all cells. The updated temperatures give the new heat 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 ω may give calculation times between 1/70th and 1/10th of that required for a calculation not using over-relaxation ($\omega=1.0$), see Section 5.2.2. The optimized ω typically lies in the range 1.8-2. In HEAT2 (and HEAT2R, and HEAT3) this factor is initially set to 1.95. Equation (3.5) is modified to

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

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 Δt equal to the right-hand side in (3.6). The chosen capacities are then

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

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

4. Overview of input

4.1 Input mesh

An *input mesh* facilitates the description of the geometry, the numerical mesh, and the boundary conditions. Consider Fig. 4.1. The x -axis is divided into a number of segments I . These are enumerated $I=1, 2, \dots, I_{MAX}$. The length of segment I is denoted ΔX_I . In the same way, divisions are made in the y - and z -directions, $J=1, 2, \dots, J_{MAX}$, $K=1, 2, \dots, K_{MAX}$, respectively. The figure shows one parallelepiped of the input mesh with the side lengths ΔX_I , ΔY_J , and ΔZ_K .

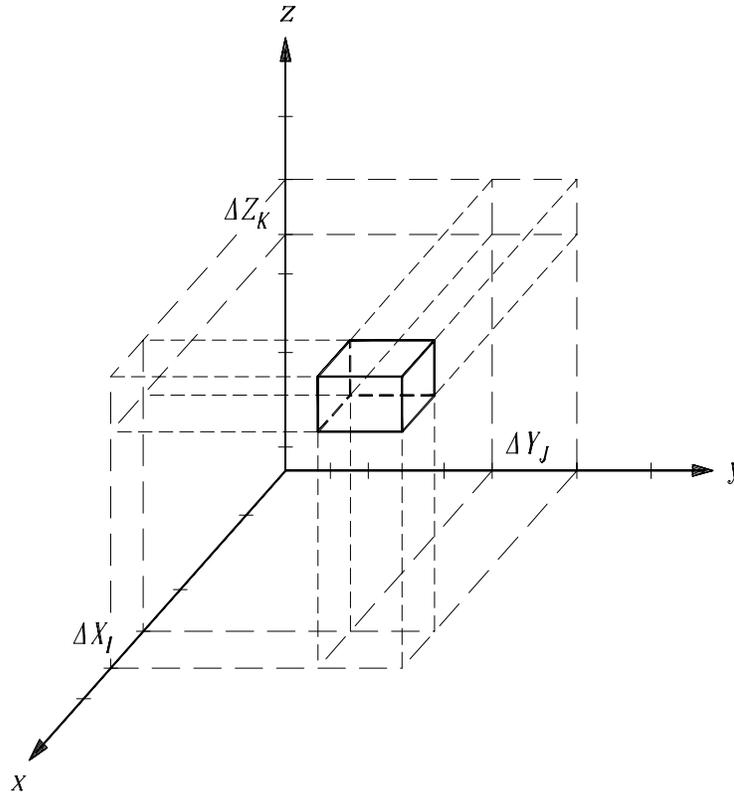


Figure 4.1: A parallelepiped in the input mesh.

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 4.2 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 $(0,0,0,1,1,1)$.

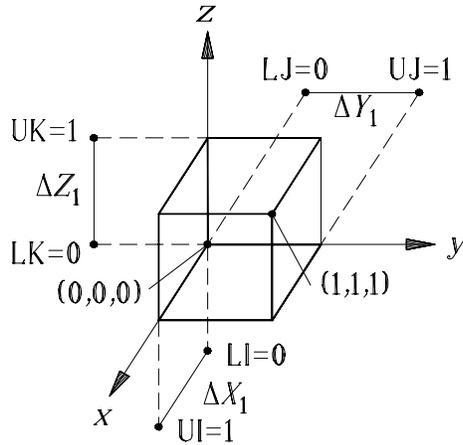


Figure 4.2: One parallelepiped.

Figure 4.3 shows a region consisting of two parallelepipeds. The larger parallelepiped has the coordinates $(0,0,0,3,1,1)$, and the smaller $(1,1,0,2,2,1)$.

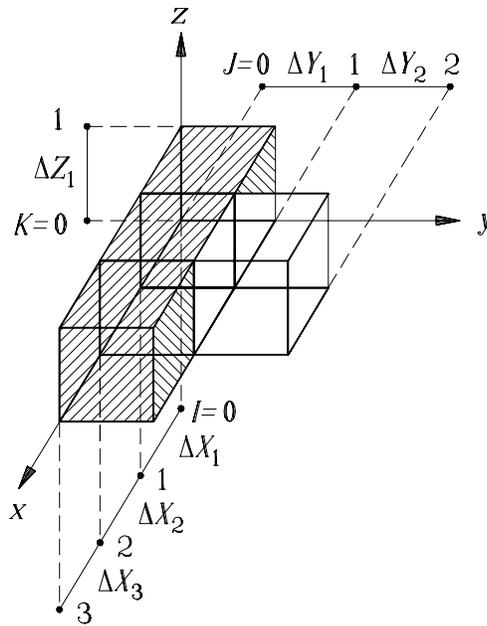


Figure 4.3: Two adjacent parallelepipeds.

As pointed out before, any parallelepipedal 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 Fig. 4.4 with one box overlapping another of different material. The first box has the coordinates $(0,0,0,3,1,1)$, and the second $(1,0,0,2,2,1)$.

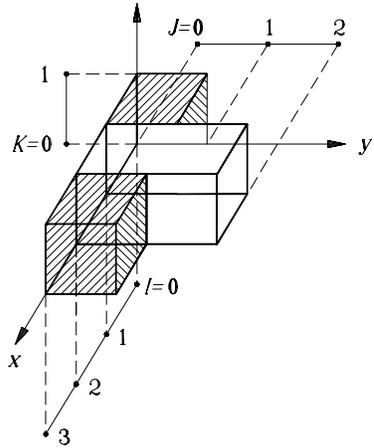


Figure 4.4: One box overlapping another box.

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 Fig. 4.5 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 $(0,0,0,2,2,2)$ that is cut out by one empty box $(1,1,1,2,2,2)$.

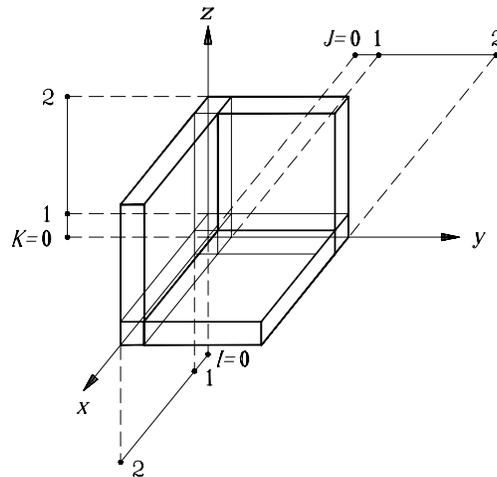


Figure 4.5: This corner may be defined by three boxes, or by one box and one empty box.

4.2 Surfaces defining boundary conditions

Boundary conditions are given in the following way. First, the types of the boundary conditions 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 Fig. 4.3 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 Fig. 4.6.

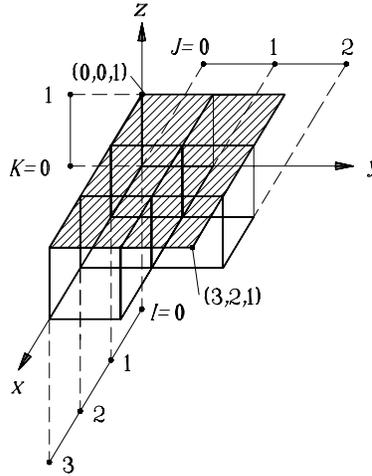


Figure 4.6: Boundary conditions are defined with surfaces, which may extend outside the surfaces of the computational volume.

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 4.2. Suppose that the temperature on the upper surface of the cube is 1 °C, and on the lower side 0 °C, 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 Fig. 4.3, 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).

4.3 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 $\Delta Y_I/N_{y,I}$ and $\Delta Z_I/N_{z,I}$. In general, capital letters refer to the input mesh (i.e. ΔX_I , ΔY_I , ΔZ_I), 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 4.7 shows the cube with two cells in each of the three directions, i.e. $N_{x,I}=2$, $N_{y,I}=2$, and $N_{z,I}=2$.

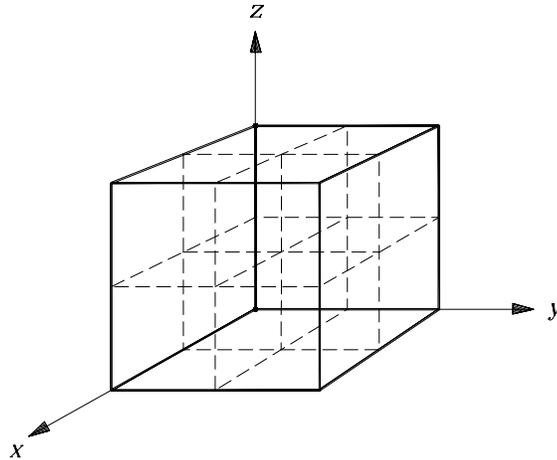


Figure 4.7: The cube with two numerical cells in each direction.

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 $\epsilon=2.0$ generates the lengths 3, 6 and 12 m (increasing sizes). With $\epsilon=0.5$, 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.

4.4 Internal resistances

Internal resistances are defined by planes that cut through the computational volume with a resistance R , ($\text{m}^2 \cdot \text{K}/\text{W}$), 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, see Section 4.2, then the total resistance will include both resistances. Consider as an example Fig. 4.3. 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 .

5. Working with HEAT3

5.1 Editing data

5.1.1 The editor

The editor contains the input data file, see Fig. 5.1. 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).

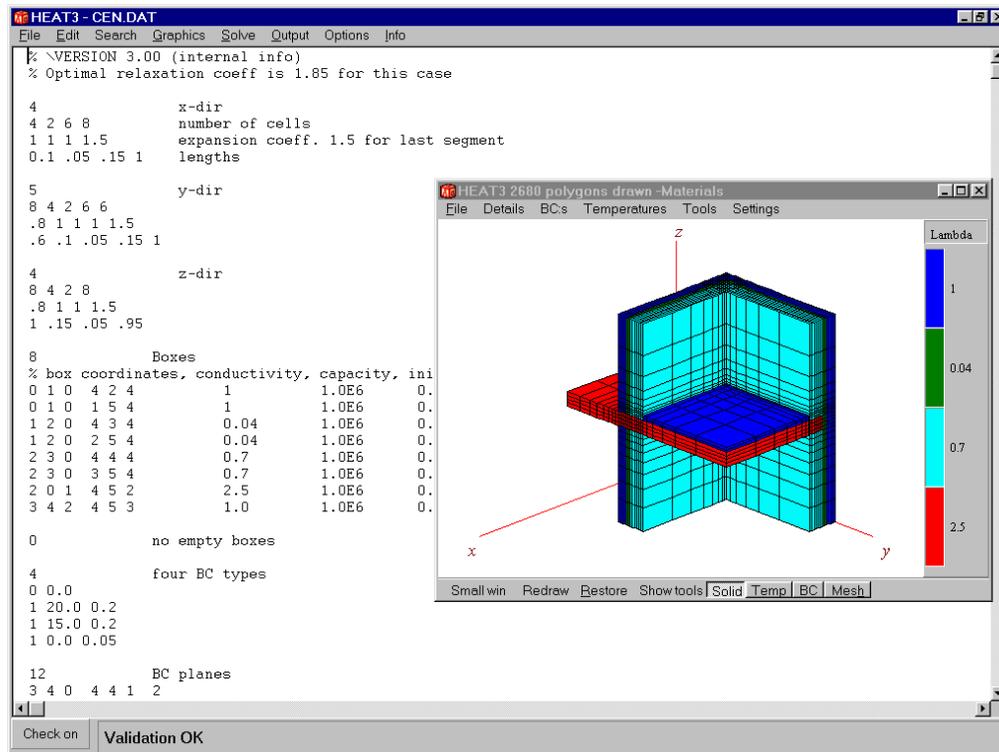


Figure 5.1: Input is made in an editor. A graphic window shows materials, mesh, temperatures, and boundary conditions.

A new problem may be given from scratch (item *File/New*) using an 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).

5.1.2 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 Fig. 5.2). 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.

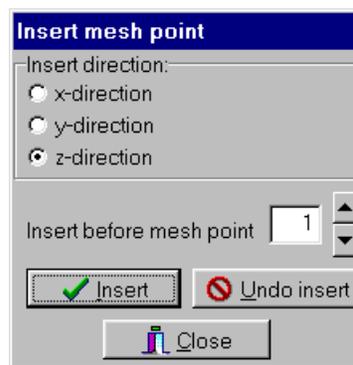


Figure 5.2: Menu for inserting mesh coordinates.

Consider Fig. 5.3. 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).

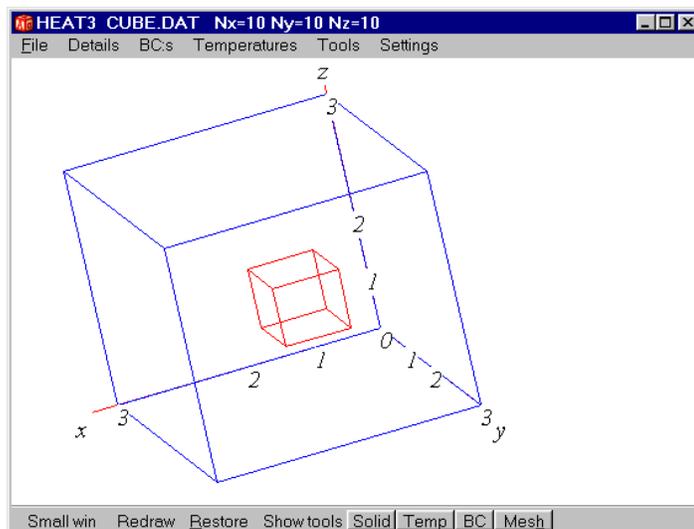
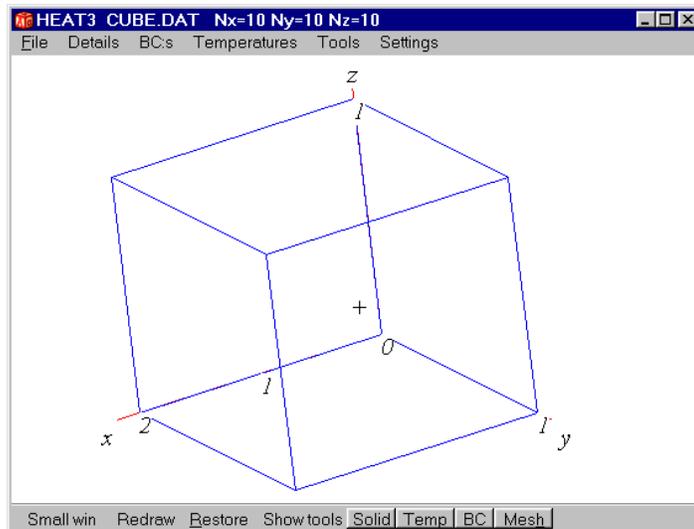
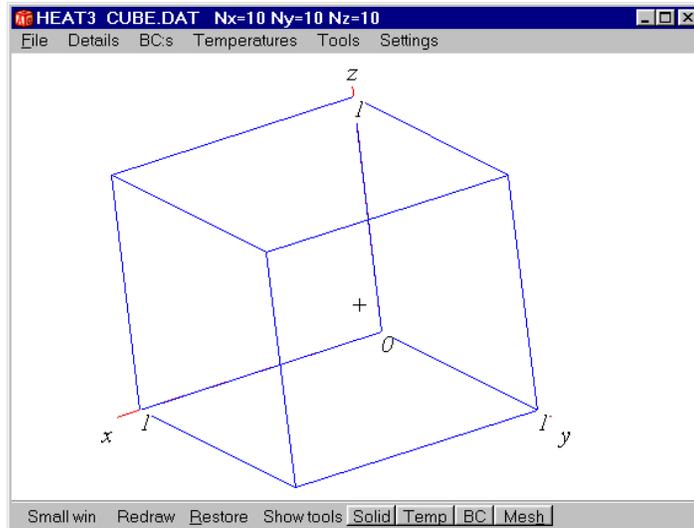


Figure 5.3: Mesh coordinates inserted.

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 Fig. 5.4.

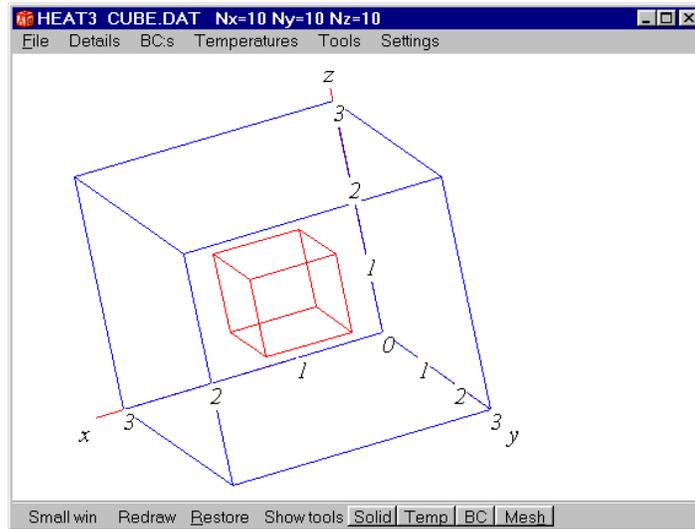


Figure 5.4: Inserting mesh coordinates and changing lengths.

5.2 Solving the problem

5.2.1 Steady-state stop criteria

There are three ways to give a stop criterion for a steady-state simulation, see Fig. 5.5. 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).

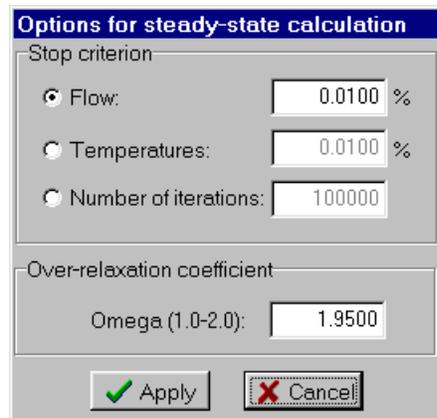


Figure 5.5: Menu for steady-state stop criterion.

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

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 , see Eq. (5.2).

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

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, see Section 7.5. Table 5.1 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 5.1: 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 in HEAT3 by letting the criterion be valid at 10 different time-steps before the simulation stops.

The criterion F is calculated for all defined sets, see Section 4.2. 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 simply to give a maximum number of iterations.

5.2.2 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), see Section 3.7.

Consider the five examples in Chapter 7. The calculation time on a Pentium/233 as a function of the relaxation coefficient ω is shown in Table 5.2. The number of computational cells is indicated. The stop criterion $F=10^{-5}$, see Section 5.2.1, is used.

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

Table 5.2: 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 here gives calculation times between 1/73 and 1/12 of that required for a calculation not using over-relaxation ($\omega=1.0$). The optimal ω lies typically in the range 1.8-2. The default value in HEAT3 is $\omega=1.95$. In relation to the optimal value for the examples in Table 5.2, the default value gives no more than about twice the CPU-time for the examples in Table 5.2.

5.2.3 Simulation time for transient calculation

Simulation stop time is given in *Solve/Options for transient*, see Fig. 5.6. The start time, denoted by current time, may be arbitrarily chosen.

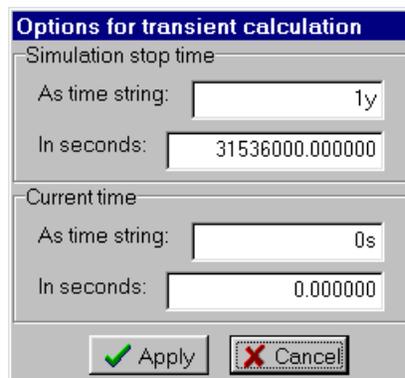


Figure 5.6: Menu for simulation time.

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

5.2.4 Functions

The boundary temperatures may be constant in time, or time-dependent using a sinusoidal, a step-wise constant, or a step-wise linear function. These are defined as:

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

The function is chosen in item *Solve/Function*, see Fig. 5.7. Give f_1, f_2, t_0 , and t_p for the sinusoidal function. The phase and period time may be given in seconds or using a time-string, see section 5.2.3.

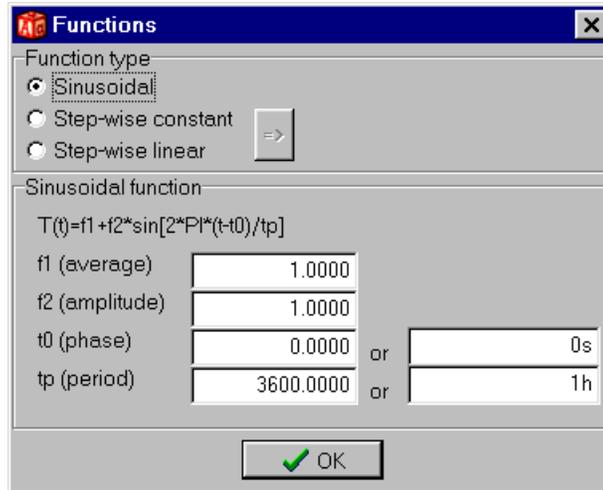


Figure 5.7: Functions are given in item *Solve/Function*.

The step-wise constant and step-wise linear values are given in an editor, see Fig. 5.8. 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 Fig. 5.8 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 Figs 5.12 and 5.13. The maximum number of values is constrained by the 16 MB of text that the editor can hold. Using e.g. 16 characters per row would give a maximum number of one million function values.

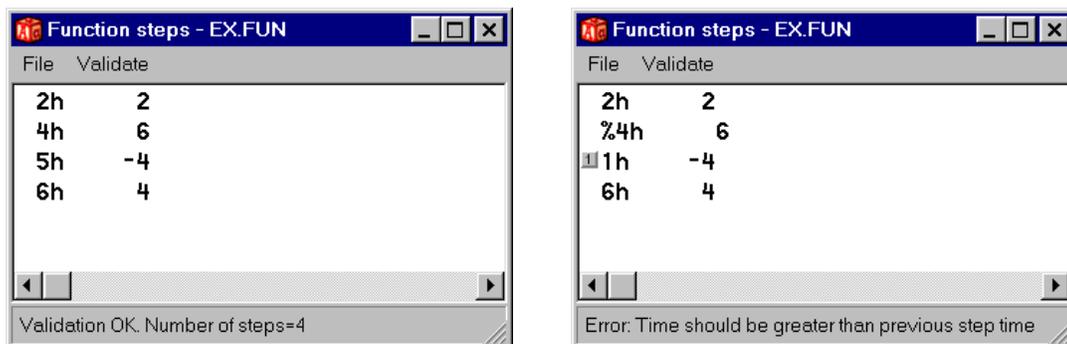


Figure 5.8: Data for function steps are given in an editor. The input is validated and a marker will be shown if there is an error.

The function values may be generated in other programs and pasted into the function editor. Fig 5.9 shows the function $\ln(x)$ created in Excel for a few values that have been pasted into the function editor in HEAT3.

The data for the function steps may be saved (*.FUN) and later be used in other problems.

During the simulation, the function value will be shown, see Fig. 5.14 (right).

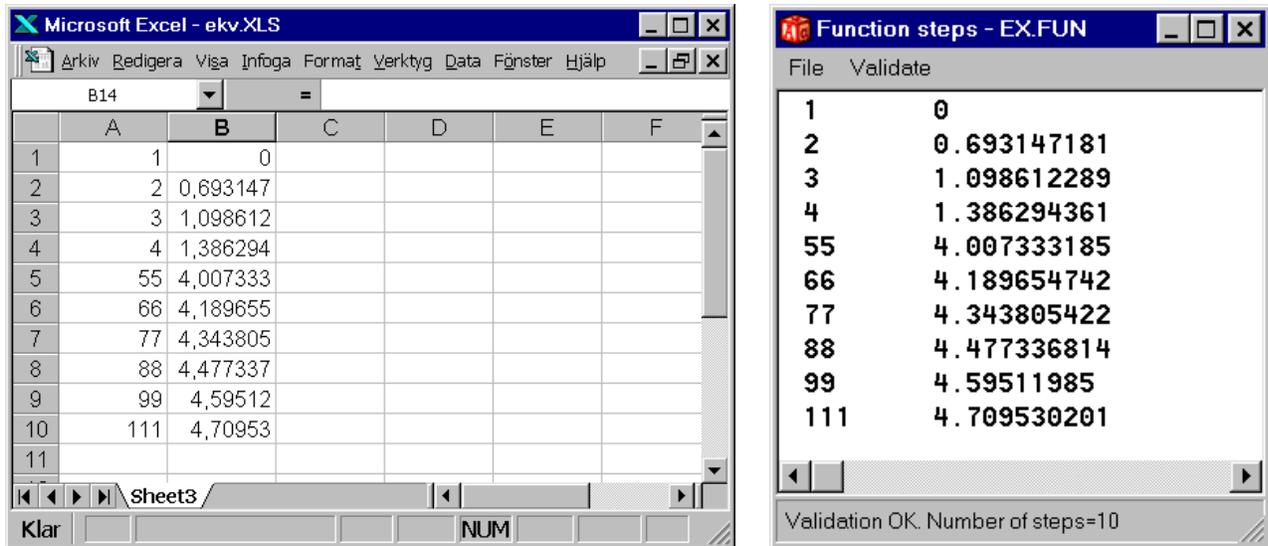
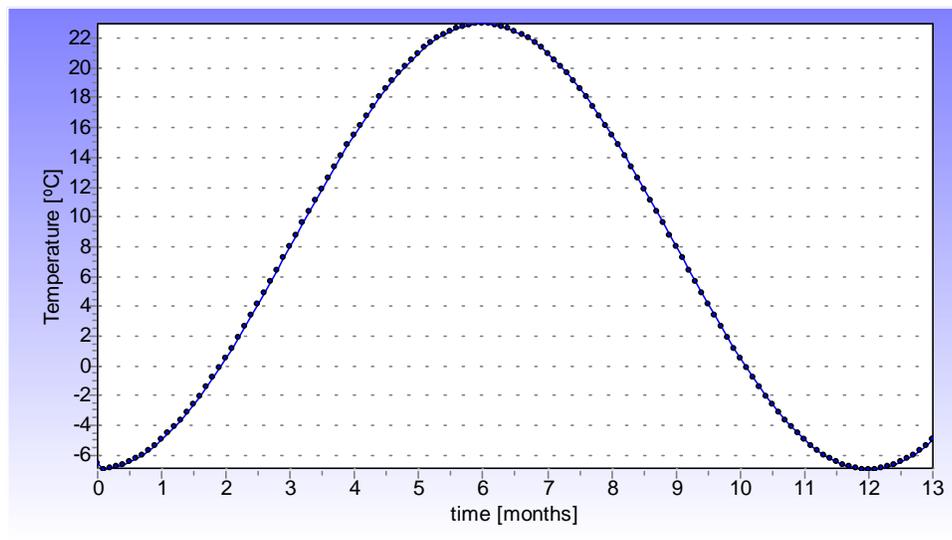


Figure 5.9: Arbitrary function step values may be pasted from other programs. Here is the function $\ln(x)$ created in Microsoft Excel for a few values that have been pasted into the function editor in HEAT3.

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

Figure 5.10 shows a sinusoidal function with the average value $8\text{ }^\circ\text{C}$, the amplitude $15\text{ }^\circ\text{C}$, the time phase three months ($-3q$), and the time period one year ($1y$). With this time phase ($-3q$ or $+9q$), the coolest temperature ($-7\text{ }^\circ\text{C}$) will occur after $0, 12, \dots, 12n$ ($0 \leq n \leq \infty$) months, i.e. on the 1:st of January every year. The warmest temperature ($23\text{ }^\circ\text{C}$) will occur after $6, 18, \dots$, months, i.e. on the 1:st of July every year¹.



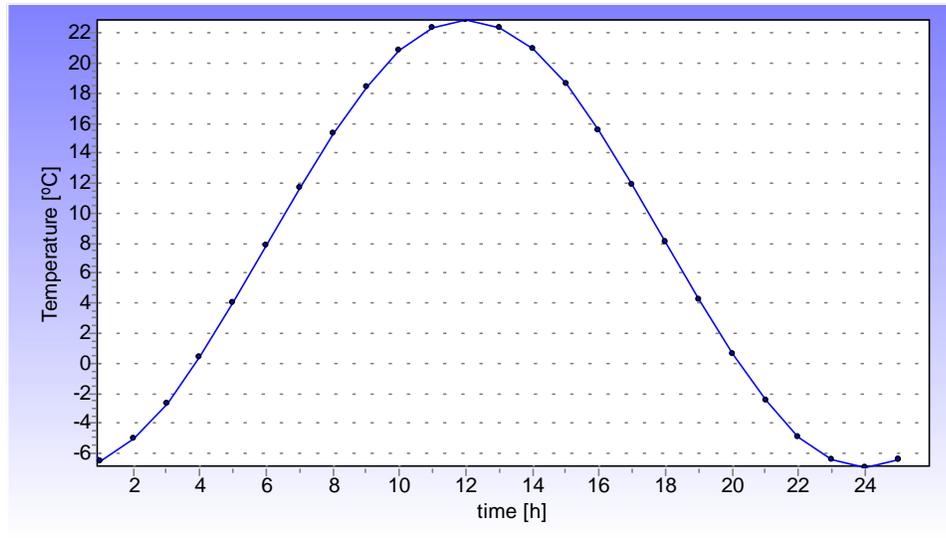
$$f(t) = 8 + 15 \cdot \sin\left(\frac{2\pi(t - 3q)}{1y}\right)$$

Figure 5.10: Sinusoidal function with $f_i=8, f_2=15, t_o=3q$, and $t_p=1y$.

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

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

Figure 5.11 shows a sinusoidal function with the average value 8 °C, the amplitude 15 °C, the time phase six hours (-6h), and the time period one day (1d). With this time phase (-6h or +18h), the coolest temperature (-7 °C) will occur every day at midnight, and the warmest temperature (23 °C) will occur at noon.



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

Figure 5.11: Sinusoidal function with $f_1=8$, $f_2=15$, $t_o=6h$, and $t_p=1d$.

5.2.4.3 Example 3 - step-wise constant function

Figure 5.12 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.

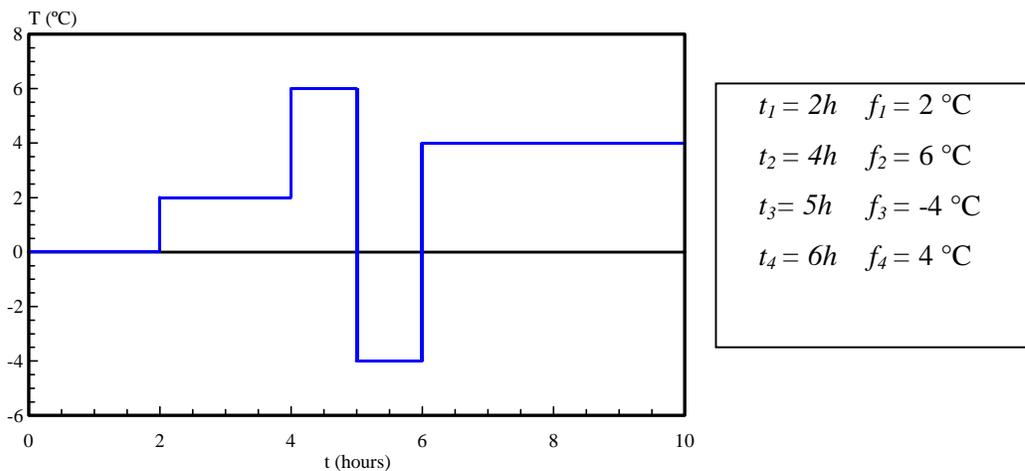


Figure 5.12: Step-wise constant function for the four steps given to the right.

5.2.4.4 Example 4 - step-wise linear function

Figure 5.13 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 $t_i=0$). The function value for the last step will be sustained.

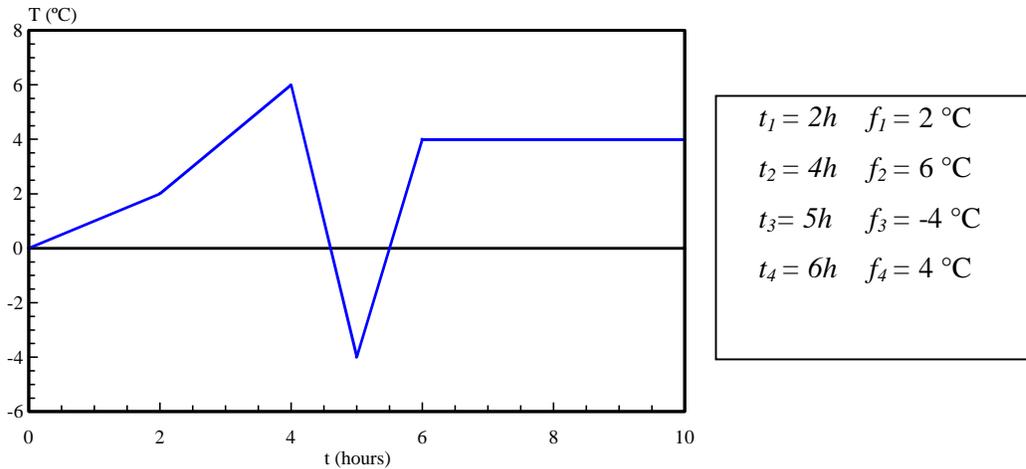


Figure 5.13: Step-wise linear function for the four steps given to the right.

5.2.5 Simulation window

An information window is shown during the simulation. Figure 5.14, 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* (see section 5.3.1), *Stop criterion* (see sections 5.2.1 and 5.2.3), *Screen update* (see section 5.2.6). If a function is used, the current function value will be shown with a short-cut (see section 5.2.4).

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

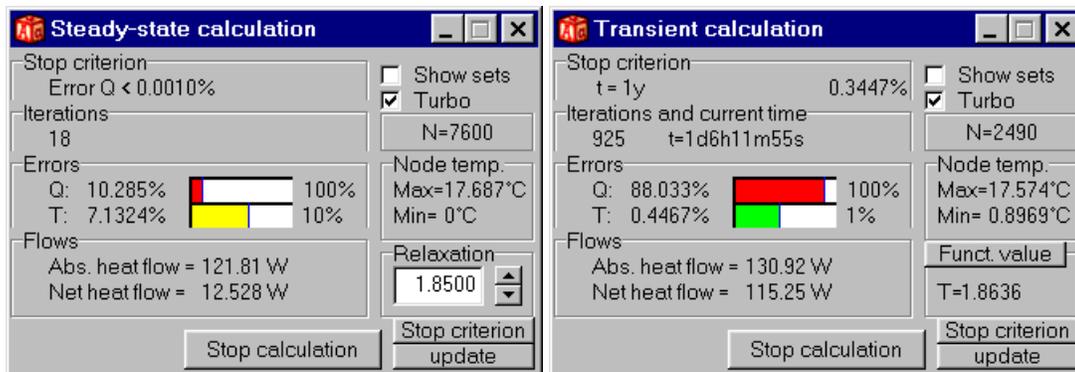


Figure 5.14: Windows shown during steady-state and the transient simulations.

5.2.6 Screen update

Data shown during the calculation are updated as defined in item *Solve/Screen update*, see Fig. 5.15. The update may be an interval depending on a number of iteration, or on the CPU-time. The default value for the screen update is set to every 5:th second.

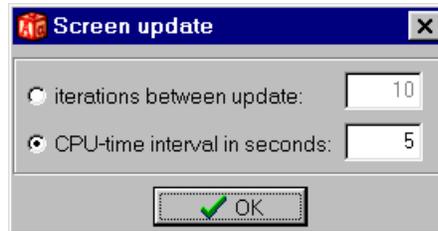


Figure 5.15: Data shown during the calculation are updated as defined here.

5.2.7 Output file options

During the simulation heat flows may be written to an output file, see item *Solve/Output file options* and Fig. 5.16.

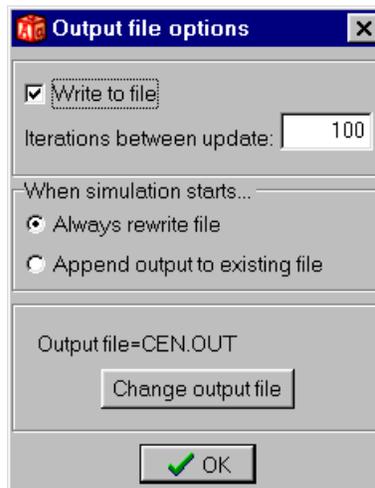


Figure 5.16: Output file options.

5.2.8 Reset

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

5.3 Output data

5.3.1 Flows and temperatures for sets

Item *Output/View sets* gives the heat flows for all sets, see Fig. 5.17. If “**Details**” is checked, the position for the maximum and minimum temperature will also be indicated.

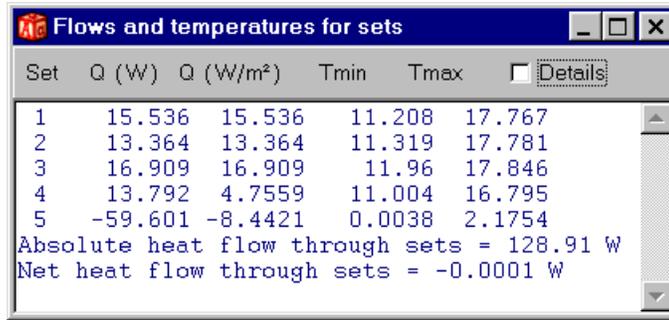


Figure 5.17: Flows and temperatures for the sets.

5.3.2 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 Fig. 5.18. The data are updated during the simulation, see Section 5.2.6, *Screen update*.

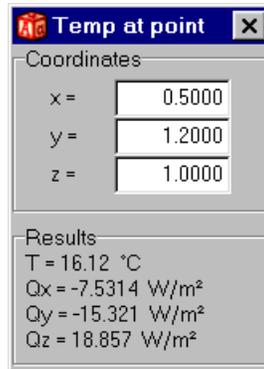


Figure 5.18: Temperature and heat flows shown for a given point.

5.3.3 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 Fig 5.19. 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 5.3 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 ($x_1..x_{N_x}$) are written to *filename.x*. The distances in the y -direction ($y_1..y_{N_y}$) are written to *filename.y*.

Plane with normal in	Data in file .x	Data in file .y	Data in file .z
<i>x-dir</i>	$y_1..y_{N_y}$	$z_1..z_{N_z}$	$T(D,y,z)$
<i>y-dir</i>	$z_1..z_{N_z}$	$x_1..x_{N_x}$	$T(x,D,z)$
<i>z-dir</i>	$x_1..x_{N_x}$	$y_1..y_{N_y}$	$T(x,y,D)$

Table 5.3: 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.

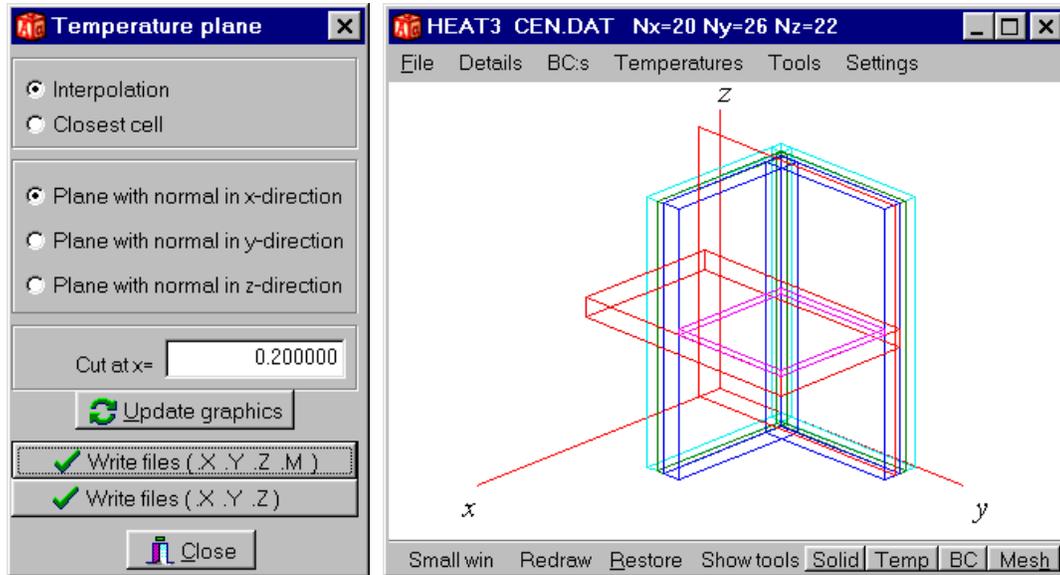


Figure 5.19: Temperature field for any plane may be written to a file. The plane (leveled at $x=0.2$) is indicated in the right figure.

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 Fig. 5.20.

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

The axes in Fig 5.20 are labeled in Matlab using the following command:

```
xlabel('y (m)')
ylabel('z (m)')
zlabel('T (°C)')
```

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.

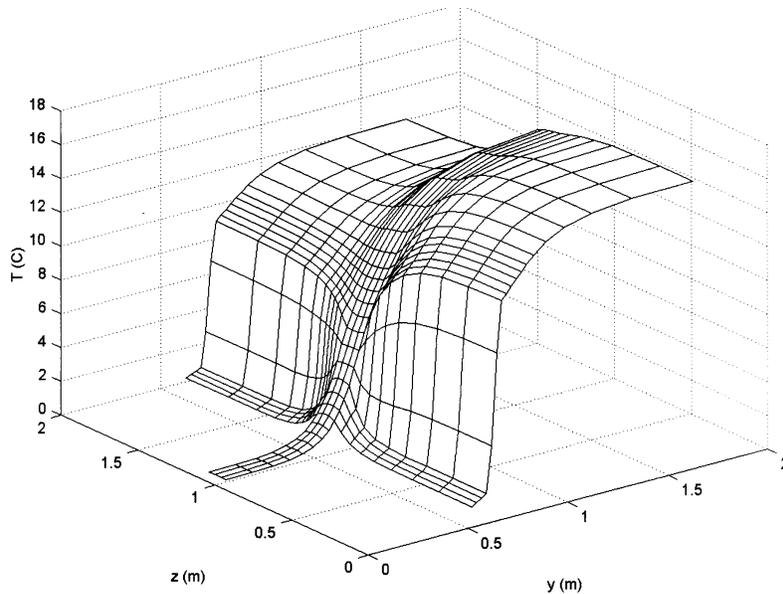


Figure 5.20: Surface plot of a temperature plane.

5.3.4 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 5.21 shows the options.

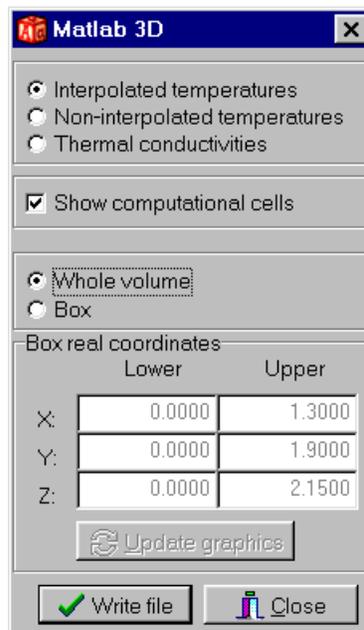


Figure 5.21: Menu for Matlab 3D figures.

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 between 4.2 and 5.1 (ver 5.1 was the present version when this was written) is 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).

5.4 Graphics

5.4.1 Introduction

The "graphic window", see Fig 5.22, shows surface temperatures 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, see section 5.1. The temperatures may be updated during the simulation (item *Temperatures/Update figure on solve*). Colors and fonts may be changed in item *Settings*.

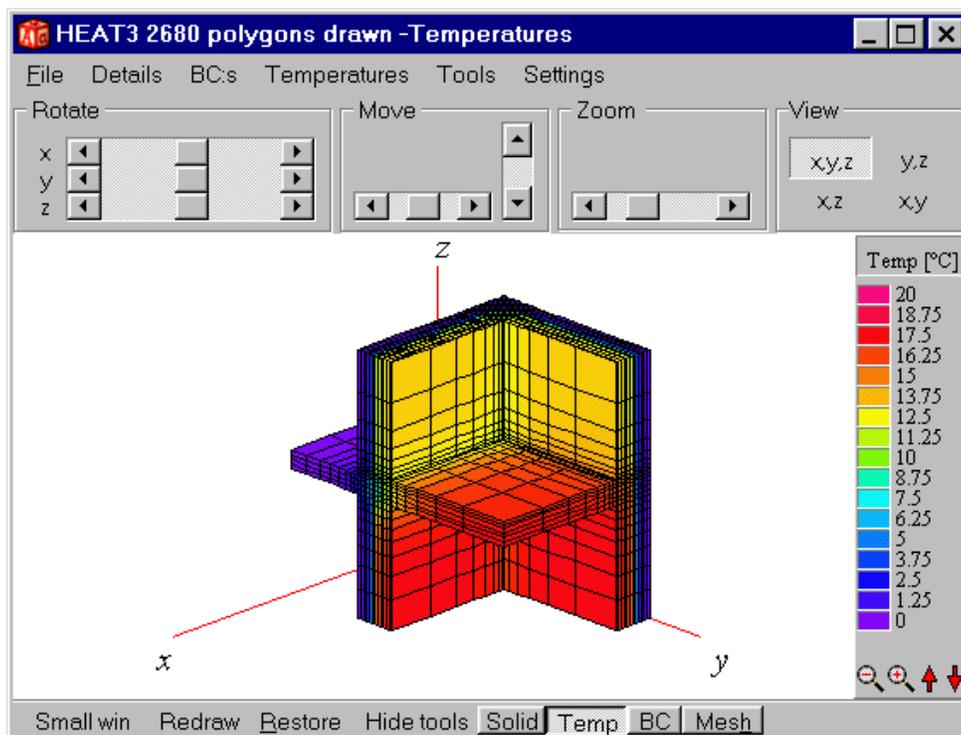


Figure 5.22: The graphic window.

5.4.2 Basic commands

The picture may be rotated, moved, or zoomed using the top tool-bar, the mouse, or directly from the keyboard, see Fig. 5.23 for mouse and keyboard commands.

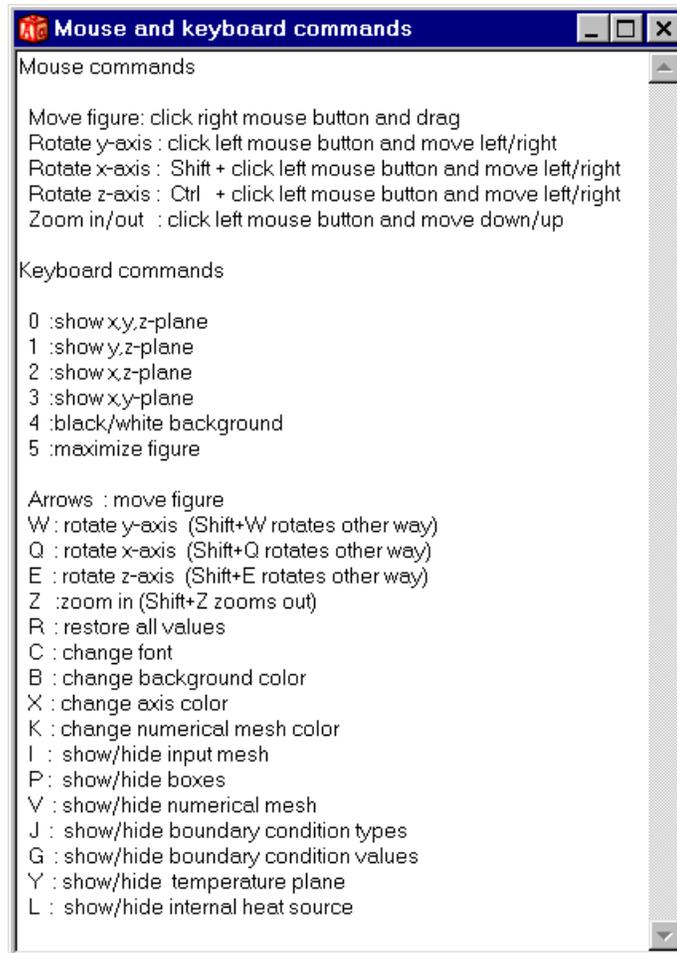


Figure 5.23: Shortcuts for basic commands.

The list in Fig. 5.23 is displayed by item *Tools/Help shortcuts*.

5.4.3 Menu item *Details*

Figure 5.24 shows the menu item *Details* with its option. As an example, input mesh coordinates are marked in the picture.

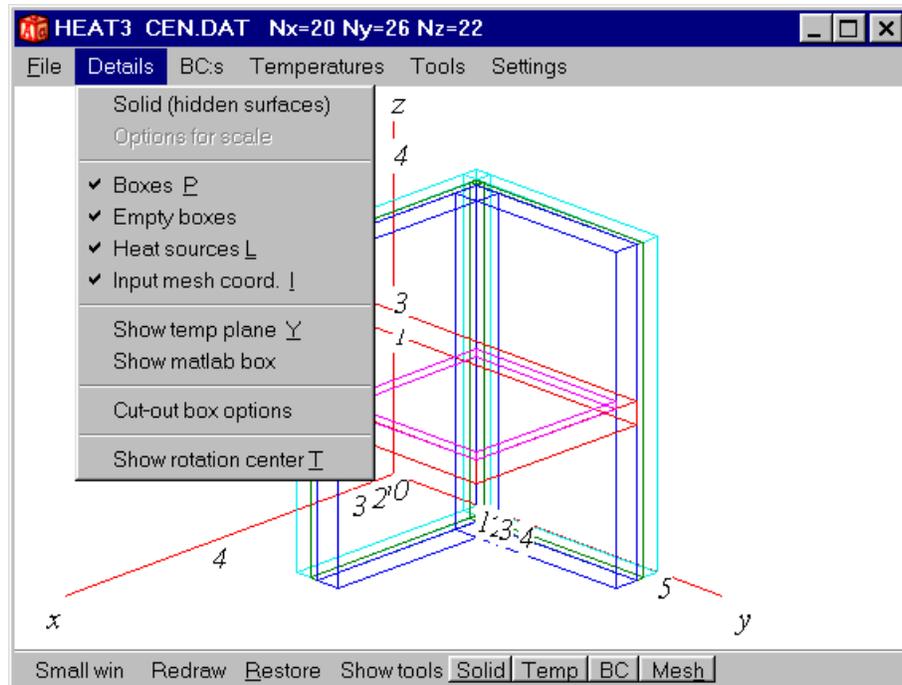


Figure 5.24: Input mesh coordinates are marked in this picture.

5.4.4 Viewing thermal conductivities and heat capacities

Click “Solid” on the bottom tool bar (or check item *Details/Solid*) to view thermal conductivities, see Figure 5.25, left. The color and the scale gives the thermal conductivity.

Clicking the scale area to the right (or *Details/Options for Scale*) brings up the window shown in Fig 5.26. Figure 5.25, right, shows the box colors as given in the input data editor.

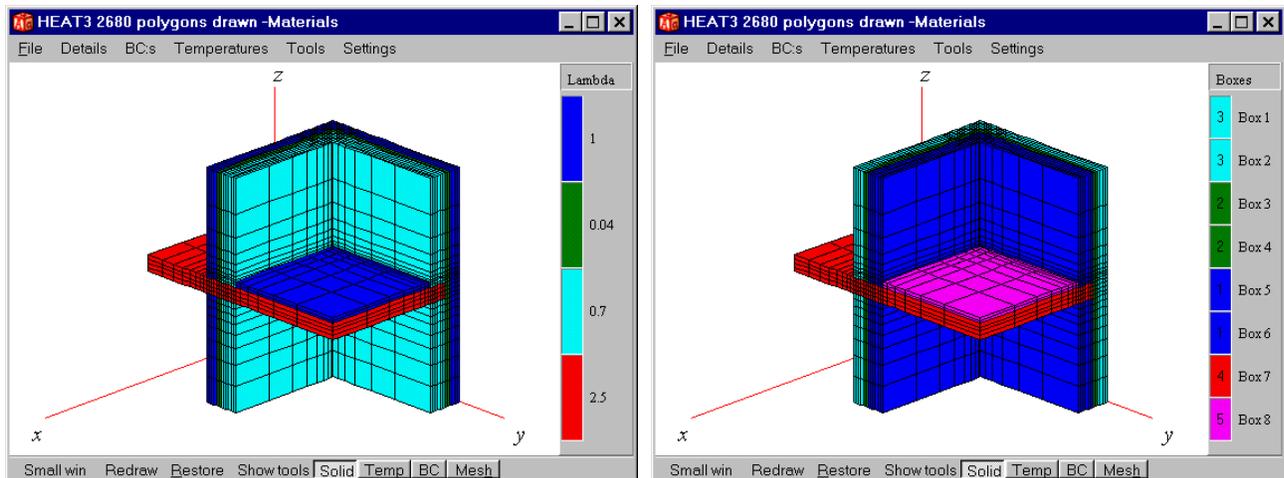


Figure 5.25: Thermal conductivities (left) and boxes (right).

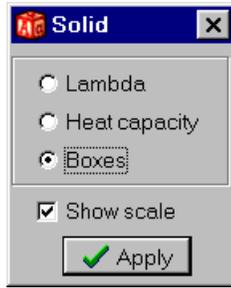


Figure 5.26: Click the scale bar (or *Details/Options for Scale*) for options.

5.4.5 Viewing boundary conditions

Figure 5.27 shows the menu item BC:s with its option. The picture and the scale indicate the specified boundary conditions.

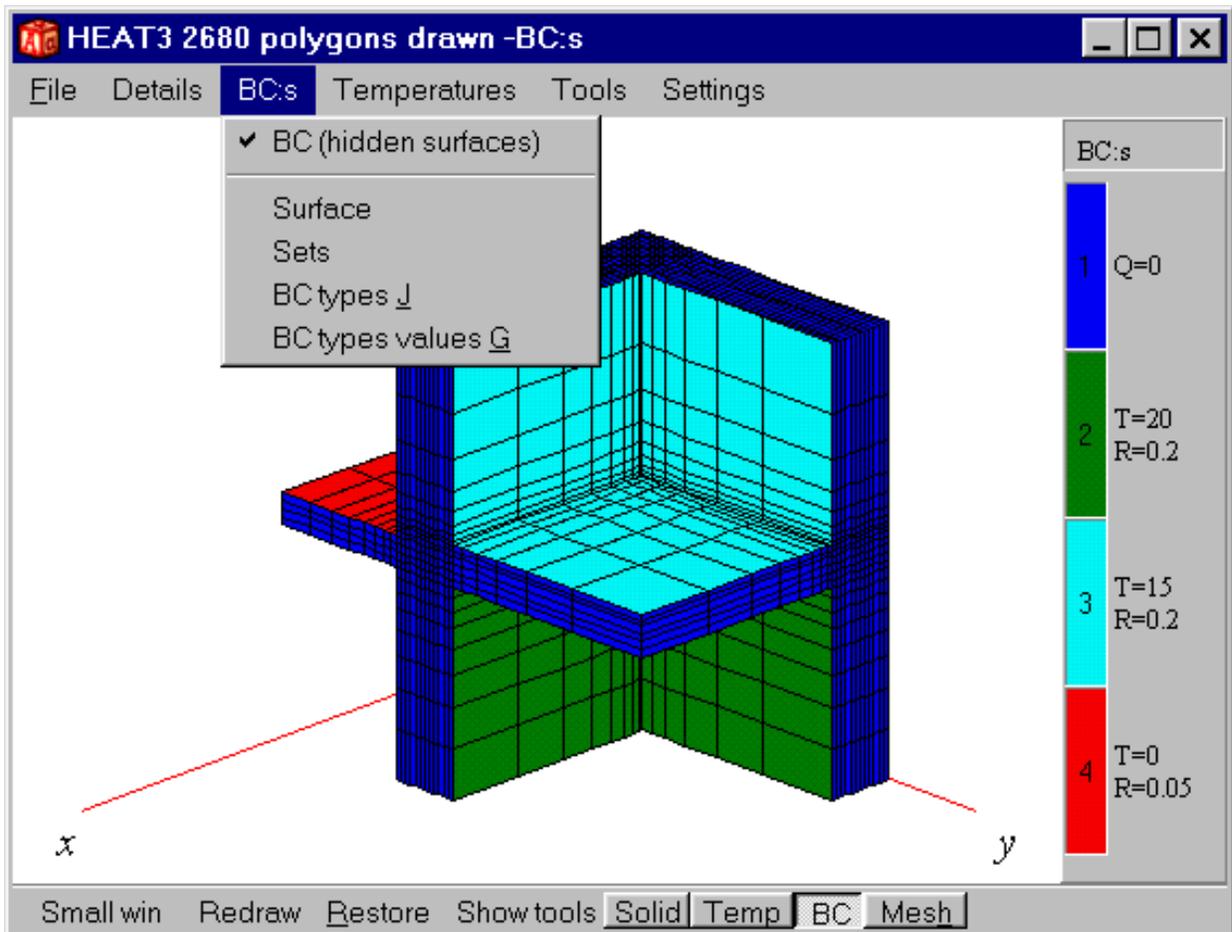


Figure 5.27: Boundary conditions.

Figure 5.28 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.

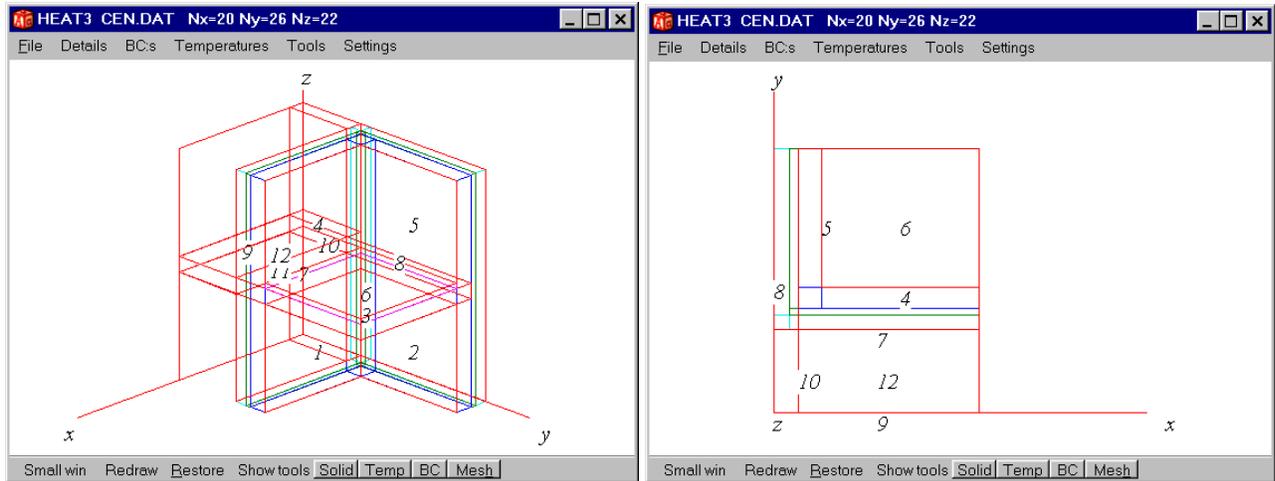


Figure 5.28: Surface numbers (item *BC:s/Surface*) in 3D and 2D.

5.4.6 Viewing temperatures

Fig 5.29 shows the *Temperatures* menu. The picture shows the temperatures in gray-scale from above (2D). Click the scale bar (or *Temperatures/options for scale*) for options, see Fig. 5.30.

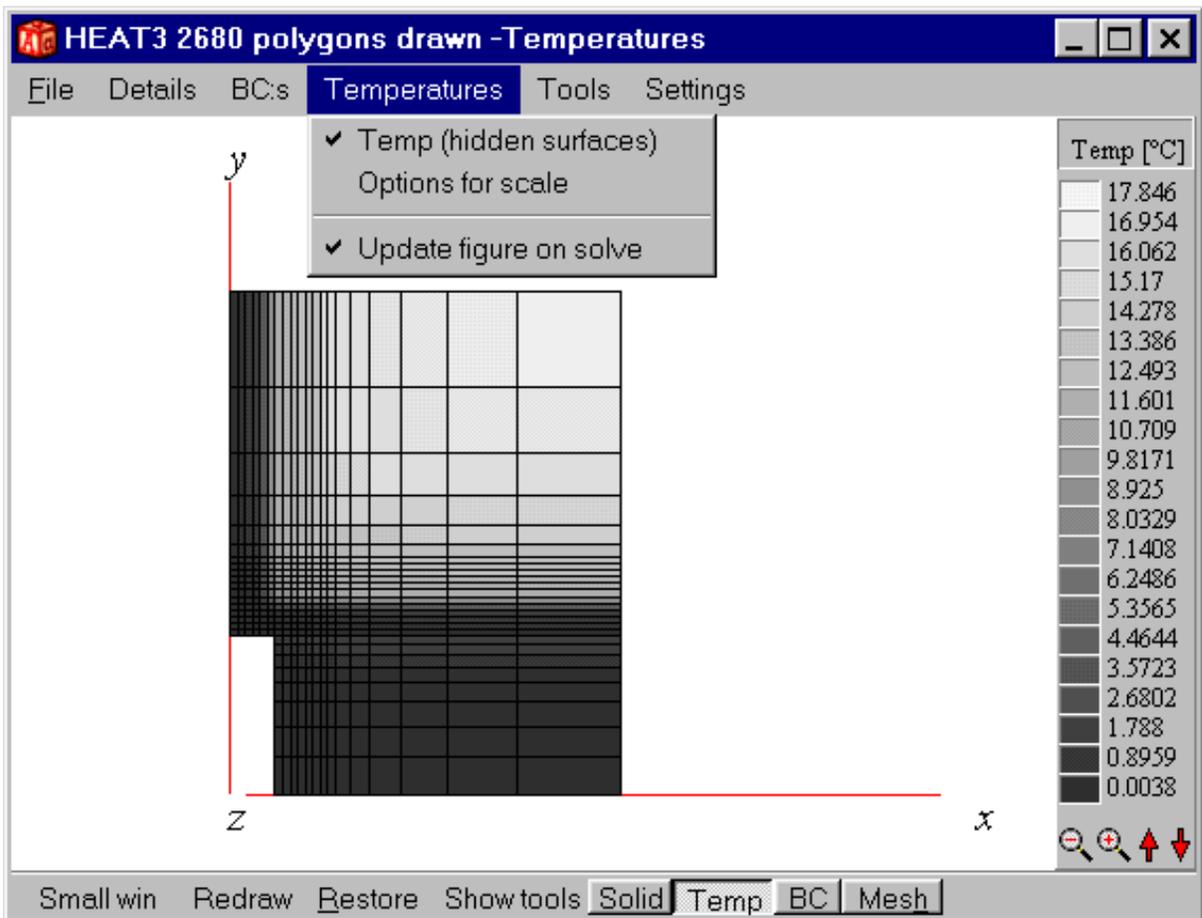


Figure 5.29: Temperatures in gray-scale (2D).

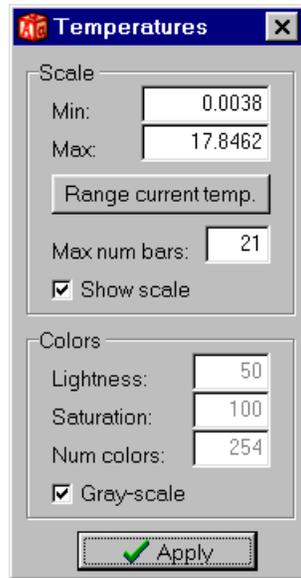


Figure 5.30: Options for temperature scale.

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 Fig. 5.31. 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**).

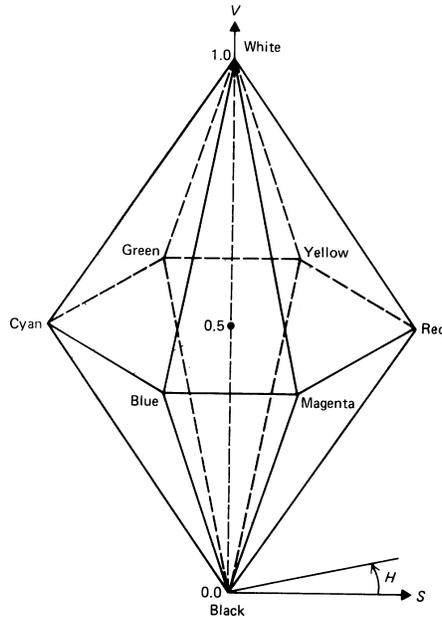


Figure 5.31: The hue-lightness-saturation color model.

The numerical mesh (without temperatures) may be viewed by putting the lightness to 1, see Fig. 5.32.

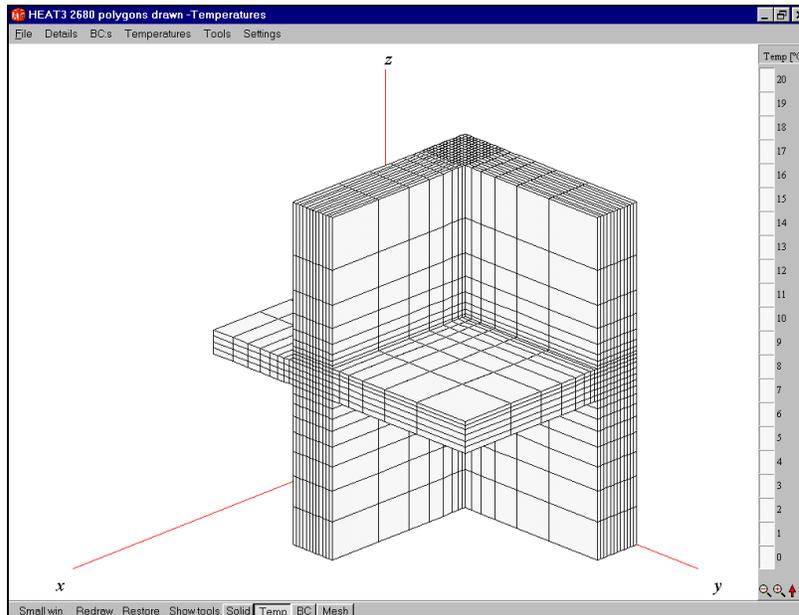


Figure 5.32: Numerical mesh (without colors) when lightness is set to 100 %.

A certain temperature interval may be disabled by clicking the temperature scale. Fig 5.33 shows two disabled temperatures, $T=18$ and $T=11$ °C. 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.

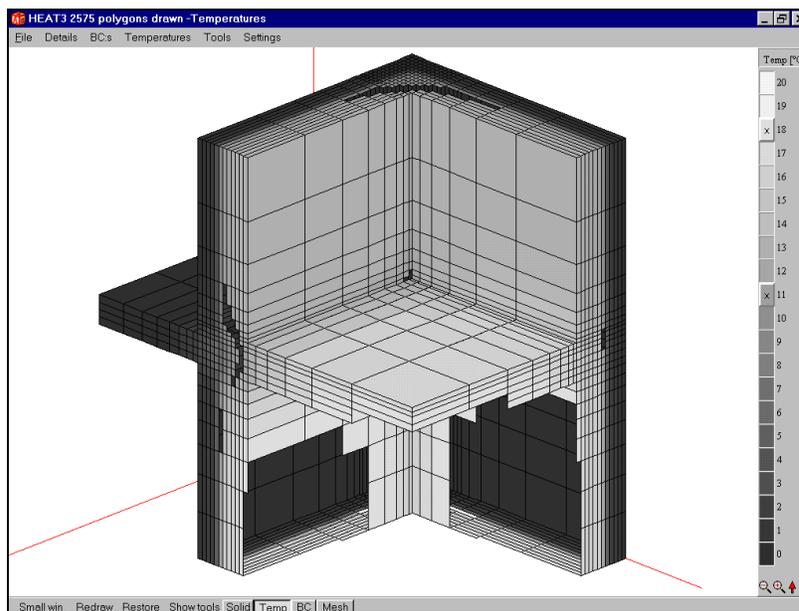


Figure 5.33: Temperature intervals may be hidden by pressing the scale color.

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

5.4.7 Removing boxes in the picture

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). Figure 5.34, left, shows the boxes with their defined colors. Here, two boxes defining the floor are invisible (color 0). The right picture shows the temperature field. Note that we see the internal temperatures at the interfaces to the invisible boxes.

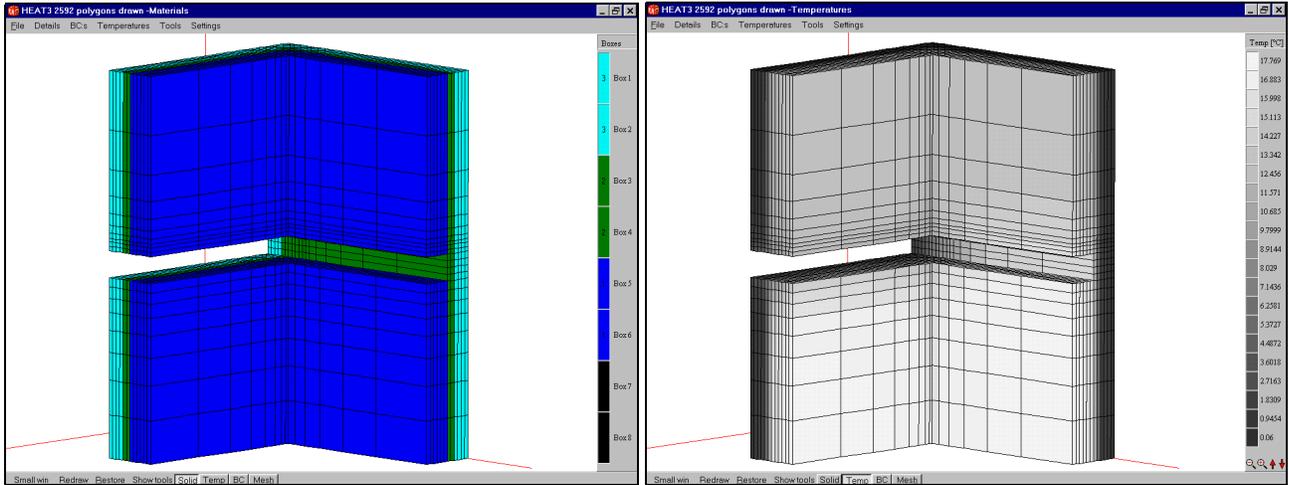


Figure 5.34: Boxes may be graphically discarded (the floor is invisible here).

5.4.8 Cut-out areas

Sub-volumes (boxes) may be cut out by defining a box in item *Details/Cut-out box* option, see Fig. 5.35, left. All surfaces that is inside the box will be drawn. Figure 5.35, right, shows the temperatures.

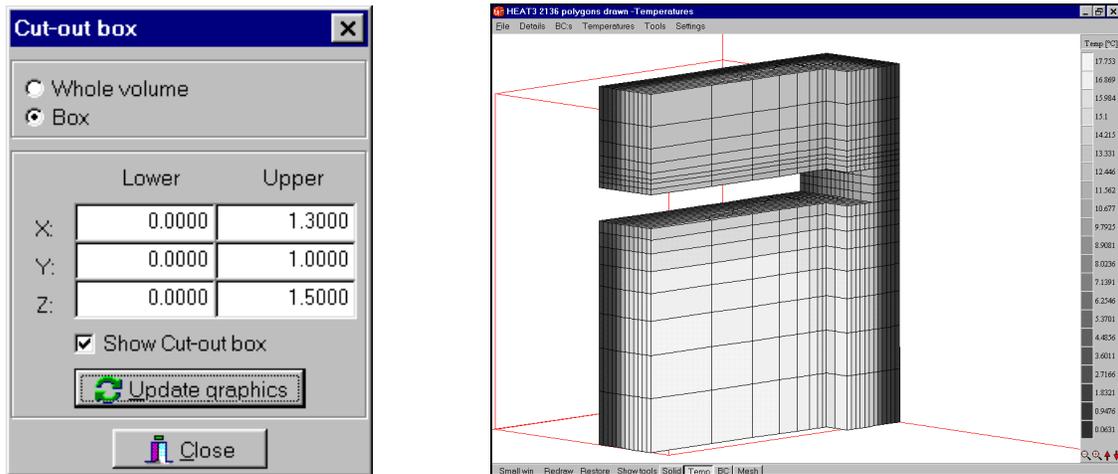


Figure 5.35: A part of the problem may be viewed by defining a box.

5.4.9 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*).

5.4.10 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 Fig 5.36 (the example is given in Section 7.4). If no 'maximizing' is chosen, the scale will be adjusted to the largest length of the three directions, see Fig. 5.37. In this case, the scale will be the same in all directions.

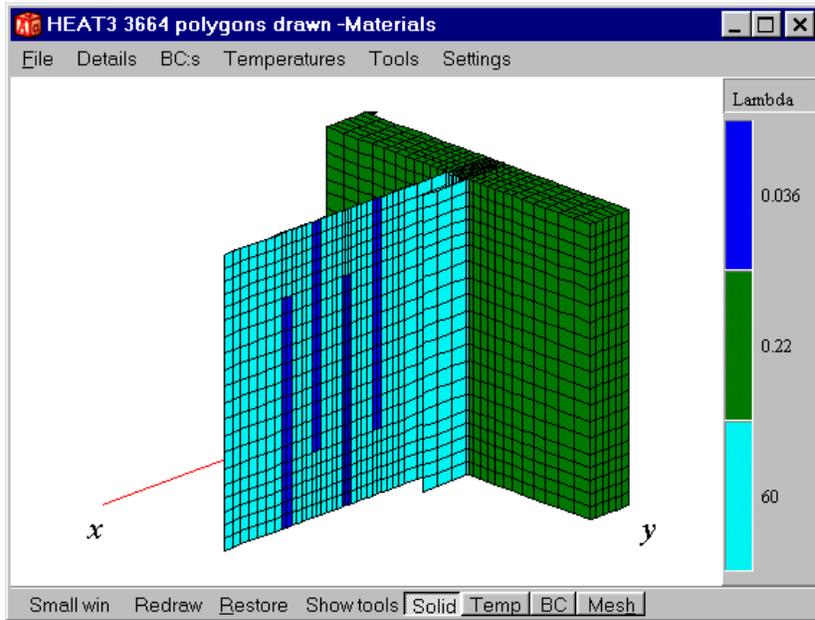


Figure 5.36: The scale in each of the three directions is adjusted to the maximum length in its own direction ('maximized area').

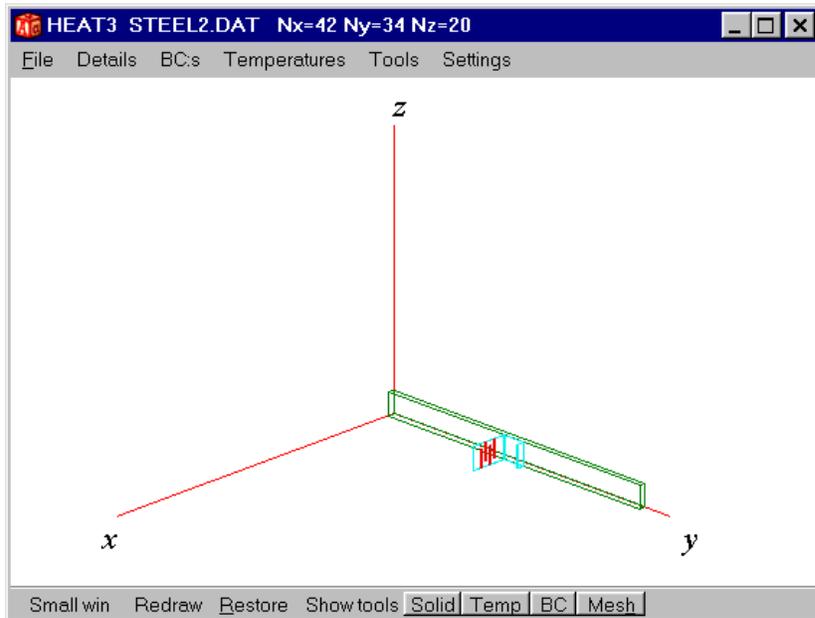
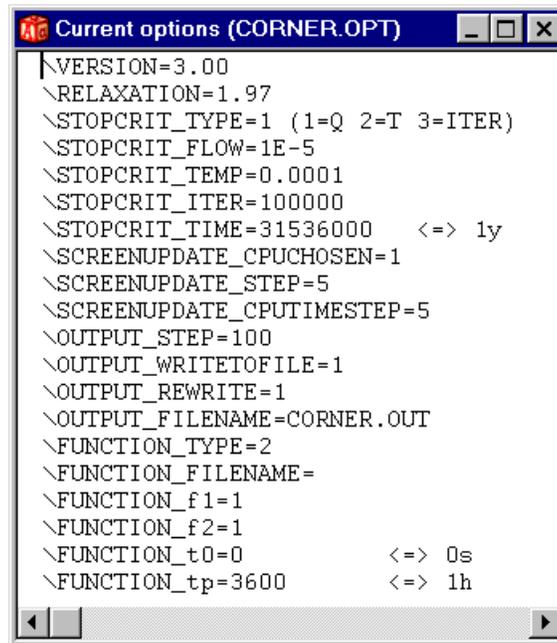


Figure 5.37: The scale is in the normal case adjusted to the largest length of the three directions. The scale will be the same in all directions.

5.5 The options file

Every time the input data file (*.DAT) is saved, an option file will also be saved with the same filename but with the extension (*.OPT). The option file contains most of the desktop settings in HEAT3. Figure 5.38 shows the options file for the corner problem *corner.opt*, see item *Options/Show options file*.

If an input data file is read and the options file is missing, a message will appear and a new will be created with default values for the desktop.



```
Current options (CORNER.OPT)
\VERSION=3.00
\RELAXATION=1.97
\STOPCRIT_TYPE=1 (1=Q 2=T 3=ITER)
\STOPCRIT_FLOW=1E-5
\STOPCRIT_TEMP=0.0001
\STOPCRIT_ITER=100000
\STOPCRIT_TIME=31536000 <=> 1y
\SCREENUPDATE_CPUCHOSEN=1
\SCREENUPDATE_STEP=5
\SCREENUPDATE_CPUTIMESTEP=5
\OUTPUT_STEP=100
\OUTPUT_WRITETOFILE=1
\OUTPUT_REWRITE=1
\OUTPUT_FILENAME=CORNER.OUT
\FUNCTION_TYPE=2
\FUNCTION_FILENAME=
\FUNCTION_f1=1
\FUNCTION_f2=1
\FUNCTION_t0=0 <=> 0s
\FUNCTION_tp=3600 <=> 1h
```

Figure 5.38: Settings for the desktop that is saved to an option file (*.OPT) for each input data file (*.DAT), in this case *corner.opt*.

5.6 Saving temperature field to file

It is possible to stop a simulation (steady-state or transient) and quit HEAT3, and later revert the simulation from where it was interrupted. To do this, save the temperatures to disk, item *File/Write temperature field*, for the problem before quitting HEAT3. The default extension for the binary file is (*.IJK). When restarting HEAT3 later, open the temperature file and continue the simulation. The IJK-file does *not* contain input data, only the temperature field, and the point of time when the simulation was interrupted.

Note that the description of the computational volume must be the same for the current input data when the temperature file was written, i.e. the first 12 rows in the input data files, and the coordinates describing the parallelepipeds should be the same. Boundary conditions and materials (thermal conductivities and heat capacities) of the given parallelepipeds may however be changed.

If there is a temperature file (*.IJK) with the same name when an input data file is loaded (*.DAT), a question will appear whether to load the temperature file or not.

5.7 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 Fig. 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

```

6. Format for input data

6.1 Input data file

The input data file is an ordinary text file. Table 6.1 shows the descriptions and restrictions of the variables.

Variables	Description	Restriction
$I_{MAX}, J_{MAX}, K_{MAX}$	Number of segments in each direction.	1-40
$N_{x,I}, N_{y,J}, N_{z,K}$	Number of computational cells in segments.	>0
$\epsilon_{x,I}, \epsilon_{y,J}, \epsilon_{z,K}$	Expansion factor in segments.	0.5-2.0
$\Delta X_I, \Delta Y_J, \Delta Z_K$	Length of segments.	>0
LI LJ LK UI UJ UK	Lower and upper input mesh coordinates.	0-40
λ	Thermal conductivity (W/(m·K)).	>0
C	Volumetric heat capacity (J/(m ³ ·K)).	>0
T_{init}	Initial temperature (°C).	
N_p	Number of boxes defining material.	1-100
N_e	Number of boxes defining empty space.	0-20
N_{bc}	Number of boundary condition types.	1-20
N_a	Number of surfaces for which to specify boundary conditions. All other surfaces will have the first defined boundary condition	1-50
N_r	Number of internal resistance planes.	0-10
N_{sets}	Number of sets.	1-20
N_{surf}	Number of surfaces per set.	1-20
N_h	Number of heat sources.	0-40

Table 6.1: 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.
$N_{x,1}, N_{x,2}, \dots, N_{x,IMAX}$	Number of cells for each segment.
$\epsilon_{x,1}, \epsilon_{x,2}, \dots, \epsilon_{x,IMAX}$	Expansion coefficient for each segment.
$\Delta X_1, \Delta X_2, \dots, \Delta X_{IMAX}$	Length for each segment.
J_{MAX}	Number of segments in y -direction.
$N_{y,1}, N_{y,2}, \dots, N_{y,JMAX}$	Number of cells for each segment.
$\epsilon_{y,1}, \epsilon_{y,2}, \dots, \epsilon_{y,JMAX}$	Expansion coefficient for each segment.
$\Delta Y_1, \Delta Y_2, \dots, \Delta Y_{JMAX}$	Length for each segment.
K_{MAX}	Number of segments in z -direction.
$N_{z,1}, N_{z,2}, \dots, N_{z,KMAX}$	Number of cells for each segment.
$\epsilon_{z,1}, \epsilon_{z,2}, \dots, \epsilon_{z,KMAX}$	Expansion coefficient for each segment.
$\Delta Z_1, \Delta Z_2, \dots, \Delta Z_{KMAX}$	Length for each segment.
N_p	Number of boxes defining a material.
LI LJ LK UI UJ UK $\lambda C T_{init} I$	Coordinates, thermal properties, initial temperature,
...	and optional integer I for color
LI LJ LK UI UJ UK $\lambda C T_{init} I$... 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/m ²).
...	I=1 for T =Real1 (°C) and R =Real2, (m ² ·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, (m ² ·K/W),
...	
LI LJ LK UI UJ UK Real1	... and for last plane N_r .
N_{sets}	Number of sets.
$N_{surf} I_1 .. I_{N_{surf}}$	Surfaces in each set. The surface $II .. I_{N_{surf}}$ refers to
...	the surfaces given for boundary conditions (1 to N_a),
$N_{surf} I_1 .. I_{N_{surf}}$.. 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_h .

6.2 Example 1

This is an input data example for the cube in Figure 4.2, 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 , see Section 5.2.1, 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 6.1 shows the mesh and the surface temperatures.

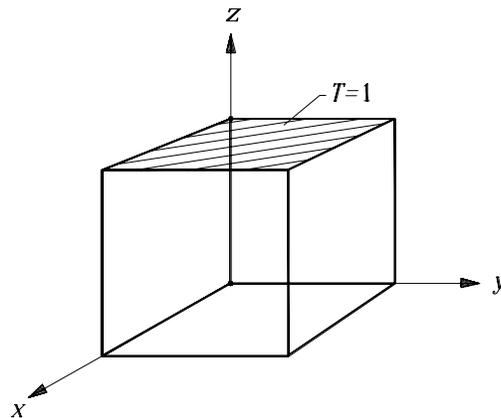
```

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)

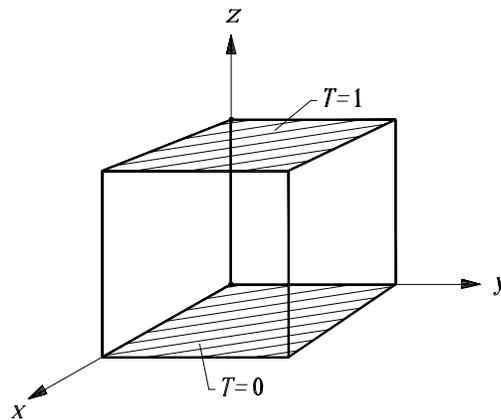
0 no heat sources

```

6.3 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 °C). Figure 6.2 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

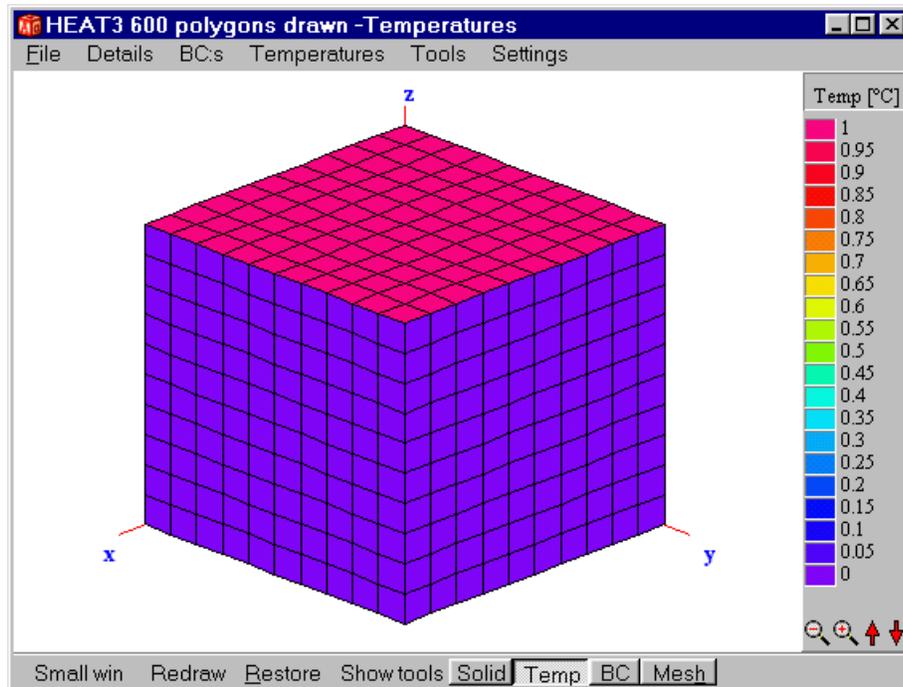


Figure 6.1: Surface temperatures for example 1.

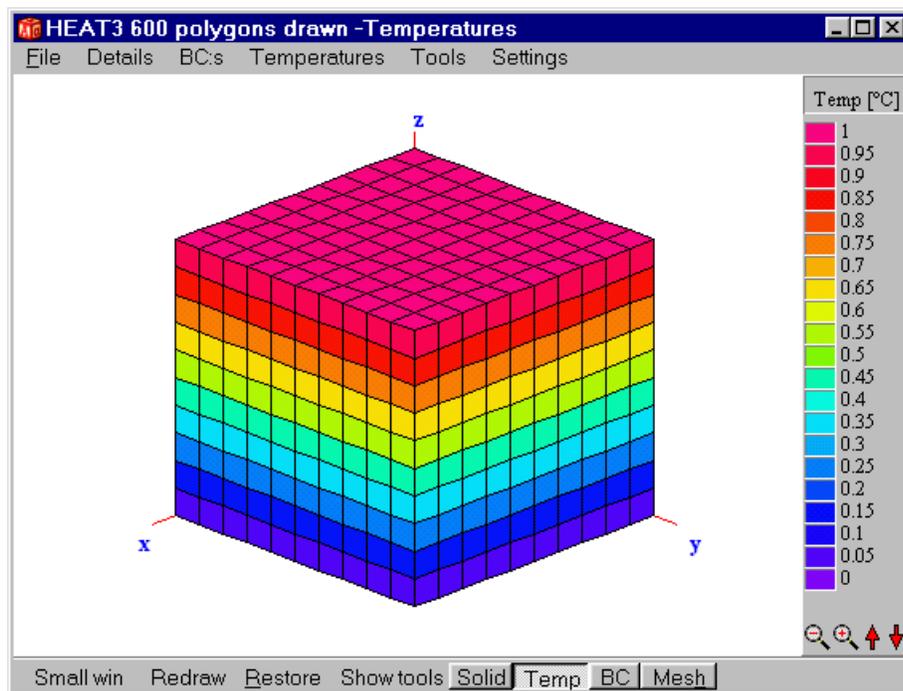


Figure 6.2: Surface temperatures for example 2.

6.4 Example 3

An input data file for the boxes in Figure 4.3 is shown below (*2boxes.dat*). 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 $0.2 \text{ m}^2\cdot\text{K}/\text{W}$ 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 6.3 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

```

6.5 Example 4

Here is an input data file for the corner shown in Fig. 4.5 (*corner.dat*). The temperature T is 20 °C on the three internal sides, and $T=0$ °C 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 6.4 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
```

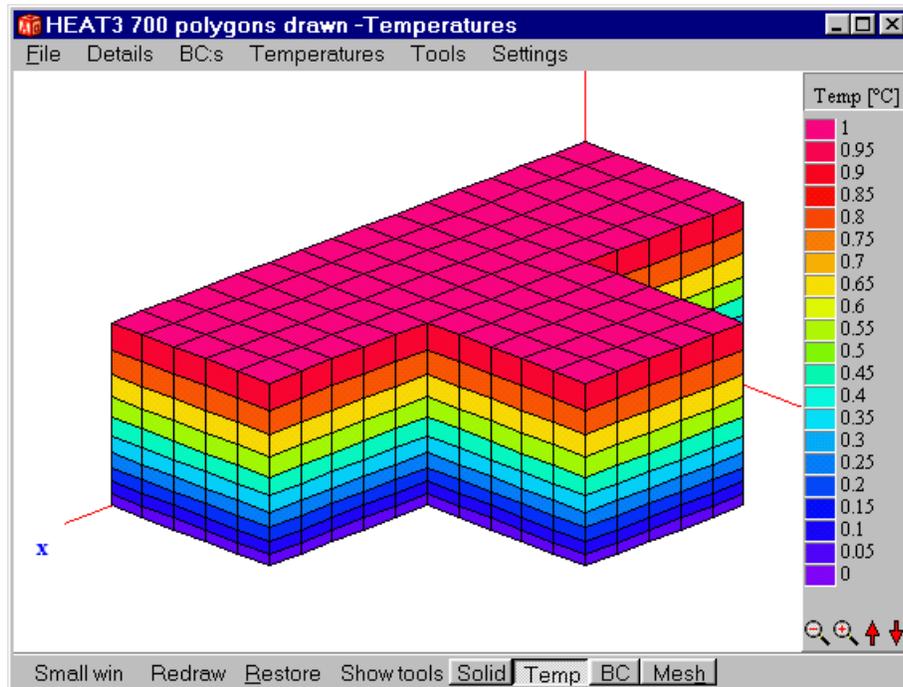


Figure 6.3: Surface temperatures for example 3.

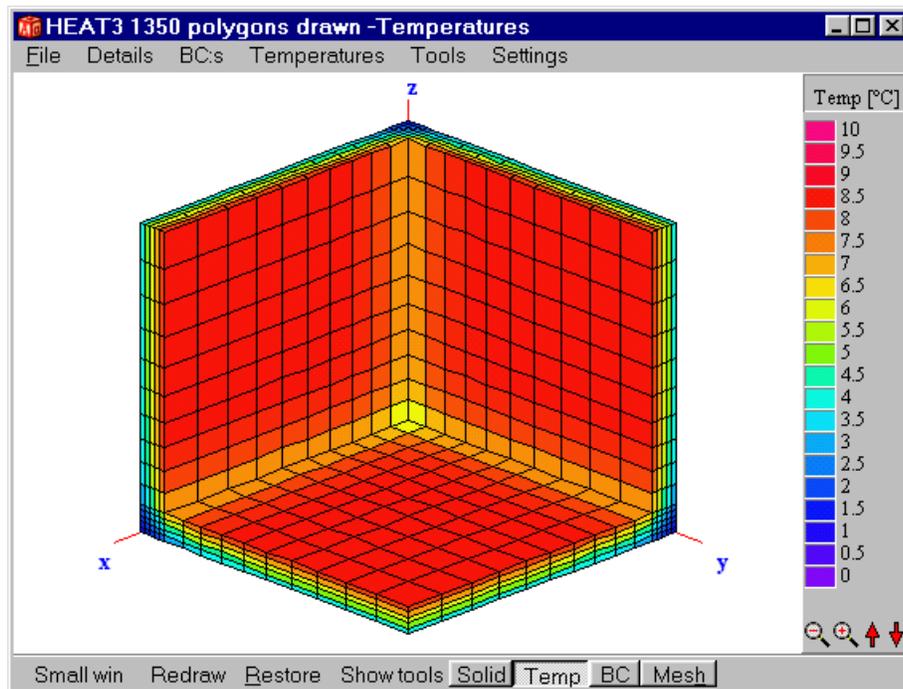


Figure 6.4: Surface temperatures for example 4.

6.6 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 6.5. 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.
```


7. More examples

7.1 Introduction to examples

7.1.1 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.

7.1.2 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.

7.1.3 Example 7, heat flow through a corner

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

7.1.4 Example 8, slab on the ground

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

7.1.5 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.

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

7.2.1 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 7.1 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.

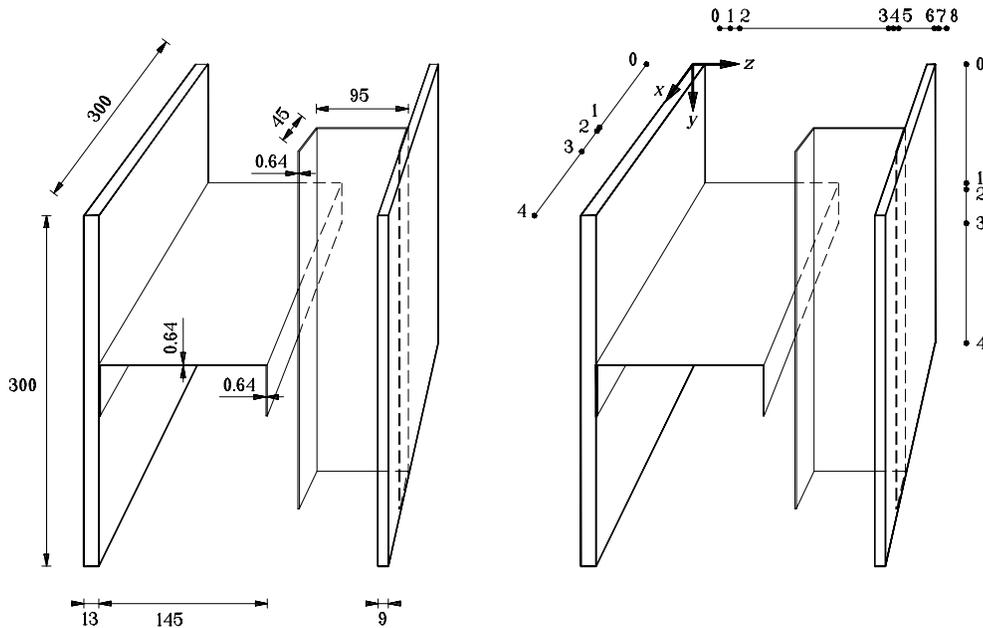


Figure 7.1: Principle sketch of an insulated wall with metal studs connecting two gypsum boards. Input mesh coordinates are shown in the right figure (see also Figs 7.2 and 7.3).

The boundary conditions are as follows. The calculations are made for a unit temperature difference. The air temperature is put to 1 °C on one side of the wall and 0 °C on the other side. The indoor and outdoor surface resistances are 0.13 and 0.04 m²·K/W, respectively. Table 7.1 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 Fig. 7.1) are adiabatic.

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

Table 7.1: 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 Figs 7.2 and 7.3.

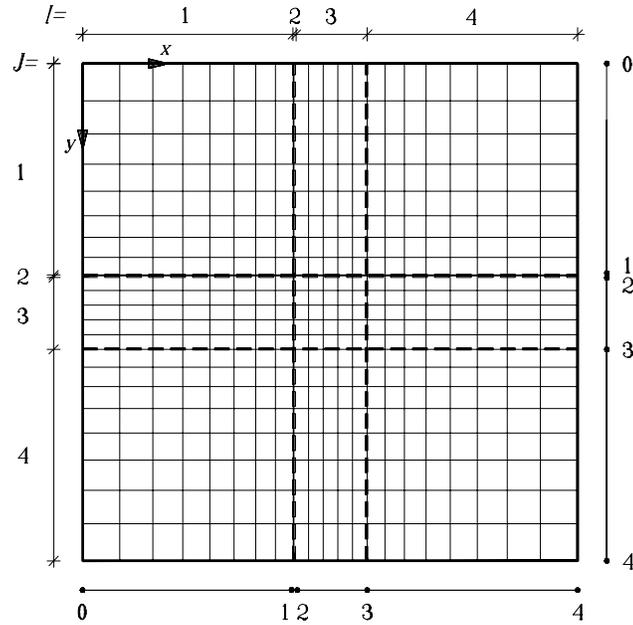


Figure 7.2: Mesh segments and coordinates, and the computational mesh in the x,y -directions for the case with 11616 numerical cells.

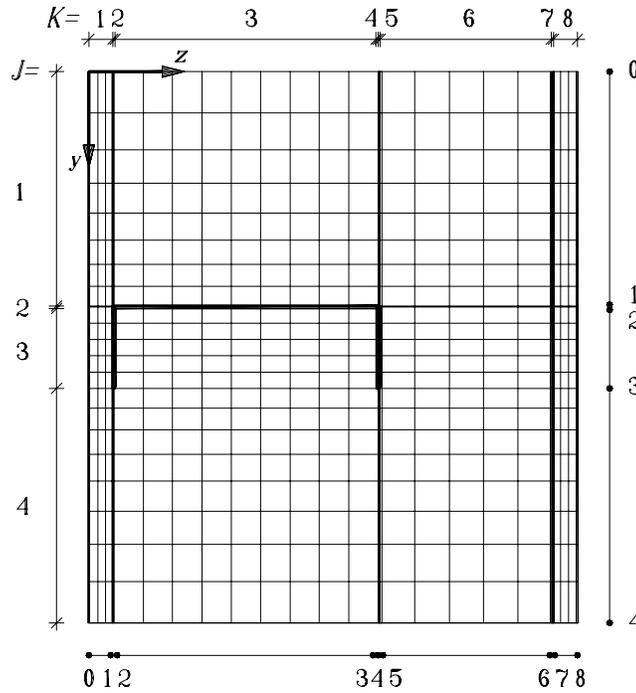


Figure 7.3: Mesh segments and coordinates, and the computational mesh in the y,z -directions for the case with 11616 numerical cells.

7.2.2 Input file steel1.dat

The input data file of the problem for the case with 11616 computational is shown below. Figure 7.4 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 apply as pointed out before in Section 4.1. 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 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
...

```

7.3 Results

Figure 7.4 shows the calculated temperatures. The mineral wool is excluded by setting the color to zero for the box, see the input data. Figure 7.5 shows the calculated heat flows through the defined sets.

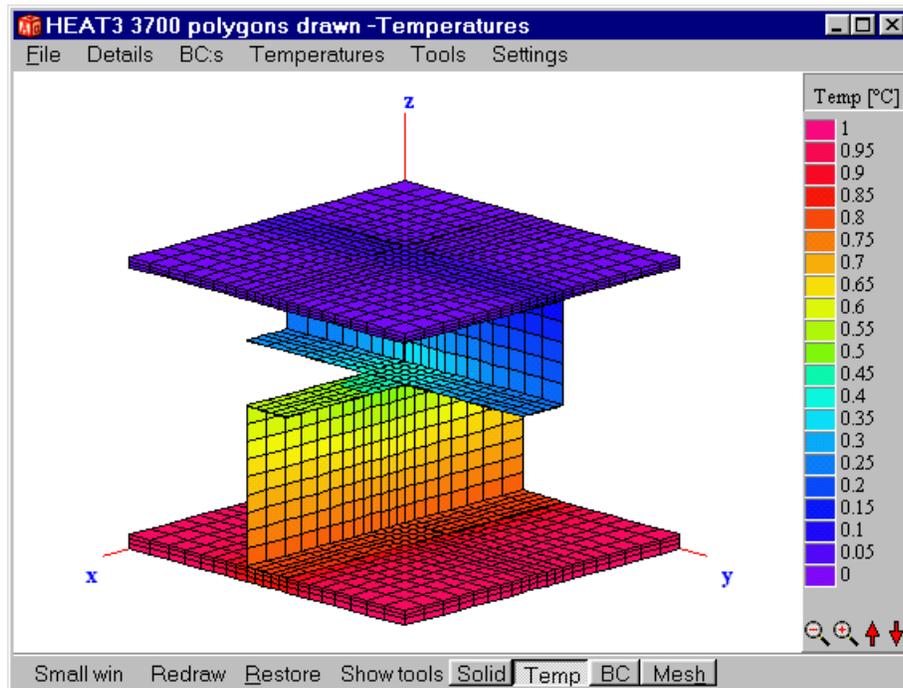


Figure 7.4: Calculated temperatures. The mineral wool is excluded by setting the color to zero for the box.

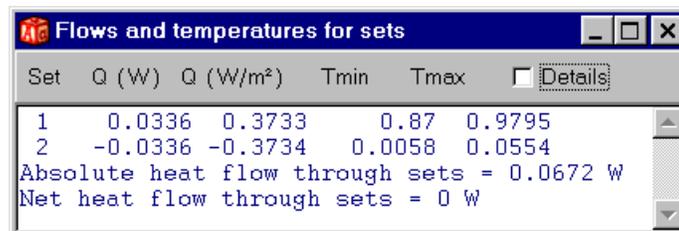


Figure 7.5: Calculated heat flows through the defined sets.

The calculated heat flows through the wall section Q , (W), are shown in Table 7.1 (stop criterion $F=1 \cdot 10^{-3}$). 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 7.1: Results for seven different numerical meshes.

The U-value of the wall becomes $0.0342 / (0.300 \cdot 0.300) = 0.38$ 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.

7.4 Example 6, slotted metal studs, steel2.dat

7.4.1 Introduction

Figure 7.6 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 Fig. 7.7, can reduce this heat loss. The thickness of the stud is denoted t .

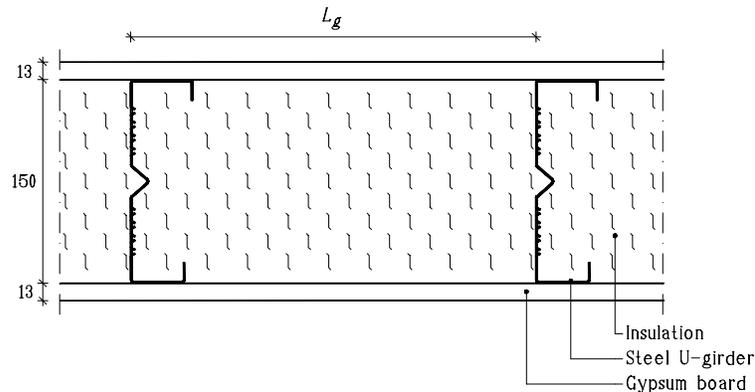


Figure 7.6: Principle sketch of a wall with studs between gypsum boards.

7.4.2 Data for the numerical calculation

The problem is not perfectly symmetric due to different flange lengths (40 and 46 mm, see Fig. 7.7). 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 $0.17 \text{ m}^2\cdot\text{K}/\text{W}$ for U-value calculations. The inner and the outer surface resistances are put to half this value ($0.085 \text{ m}^2\cdot\text{K}/\text{W}$) 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 Fig. 7.7 (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 Fig. 7.7. These rims are neglected in the calculations, and the effect of this is discussed later.

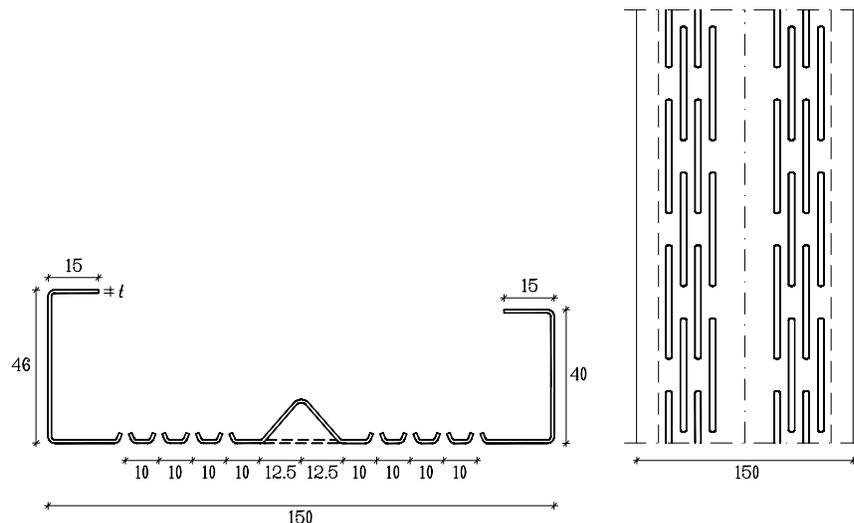


Figure 7.7: The slots (cross-section to the left) decreases the heat conduction.

The calculations have been made for the shaded volume shown in Fig. 7.8. The height (perpendicular to the plane in Fig. 7.6) is denoted by s . The temperature in the air is $0\text{ }^{\circ}\text{C}$ on one side of the wall and $0.5\text{ }^{\circ}\text{C}$ at the symmetry line in the middle of the wall. The thermal conductivity is $0.036\text{ W}/(\text{m}^2\cdot\text{K})$ and $0.22\text{ W}/(\text{m}^2\cdot\text{K})$ for the insulation and the gypsum, respectively.

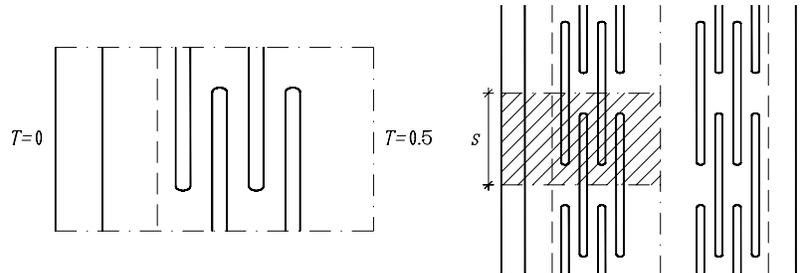


Figure 7.8: The part of studs used in the simulations.

7.4.3 Numerical mesh

Figure 7.9 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 7.10 shows a part of the mesh and the temperatures in gray-scale. The insulation has been removed in the figure.

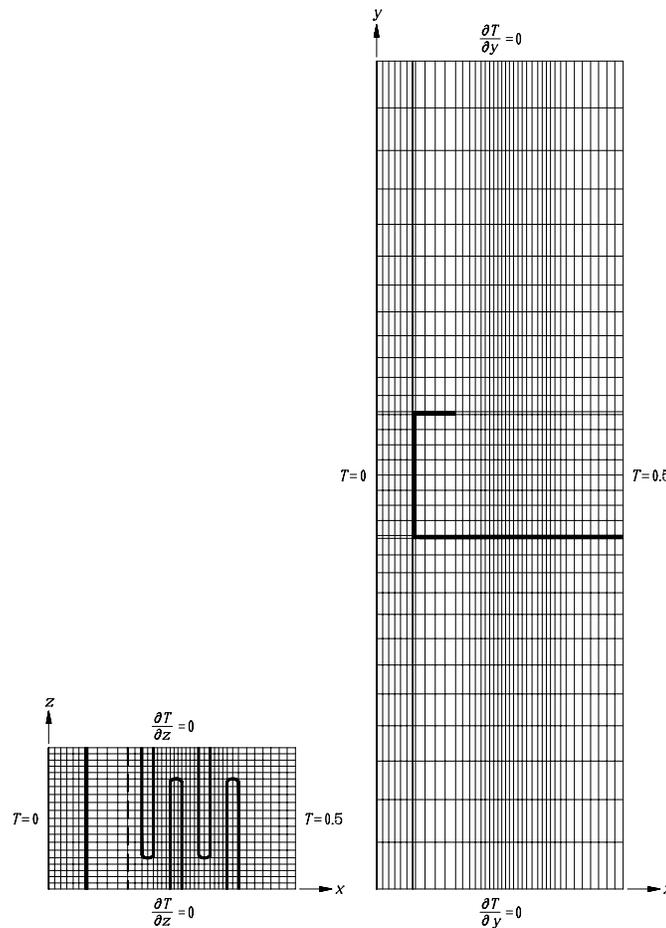


Figure 7.9: Projection of the numerical mesh on the (x,z) -plane and the (x,y) -plane in the case involving about 30000 computational cells.

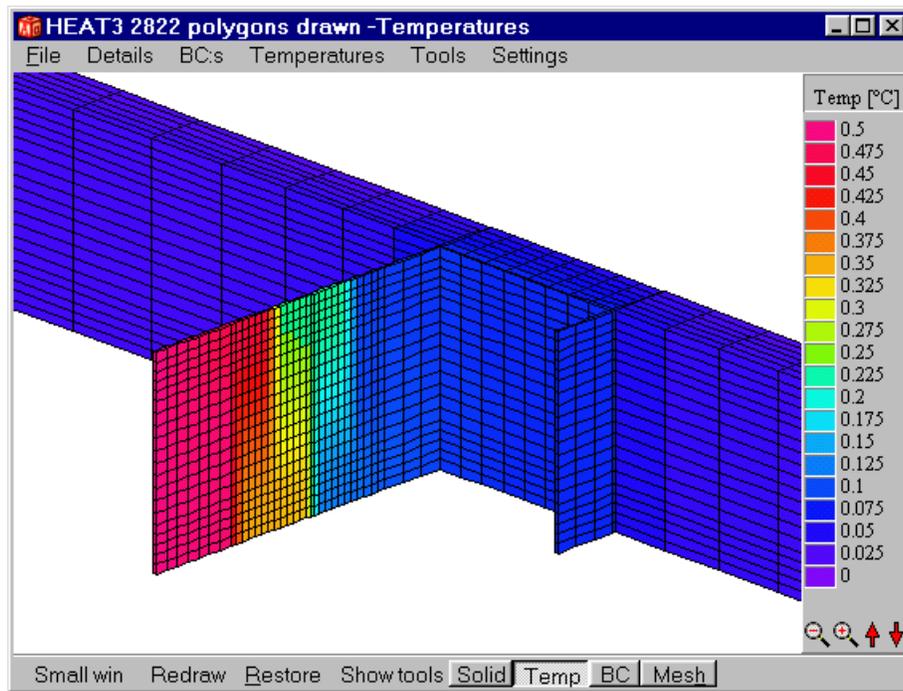


Figure 7.10: Part of the mesh and the temperatures in gray-scale. The insulation has been removed in the figure (only the steel and the gypsum board are shown here).

7.4.4 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
```

7.4.5 Results

The calculated heat flows are shown in Table 7.2 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 7.2 (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 7.2: Calculated heat flow for five different meshes.

Consider Fig. 7.11. 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 $y=0.27735$).

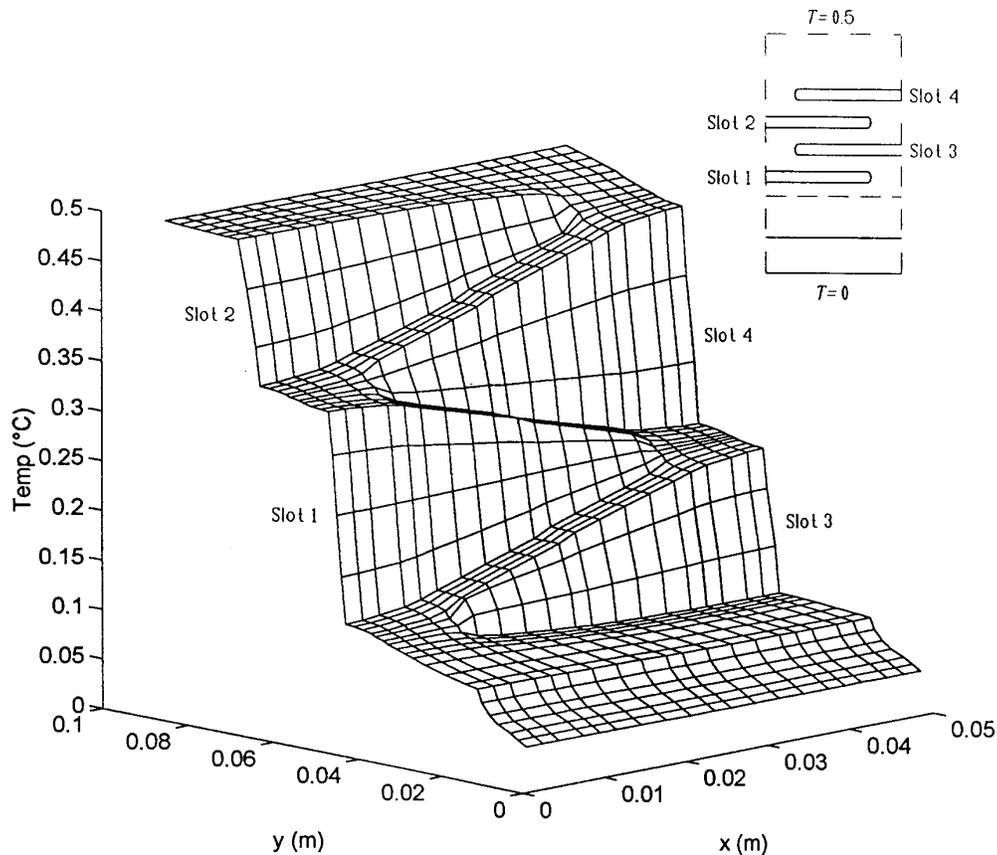


Figure 7.11: Temperature field in the steel plane through the wall.

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

7.5.1 Introduction

The next example is a test reference case in the European Standards, (CEN, 1995). Figure 7.12 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 7.13 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 7.3. The materials are labeled with numbers 1-5.

Figure 7.14 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.

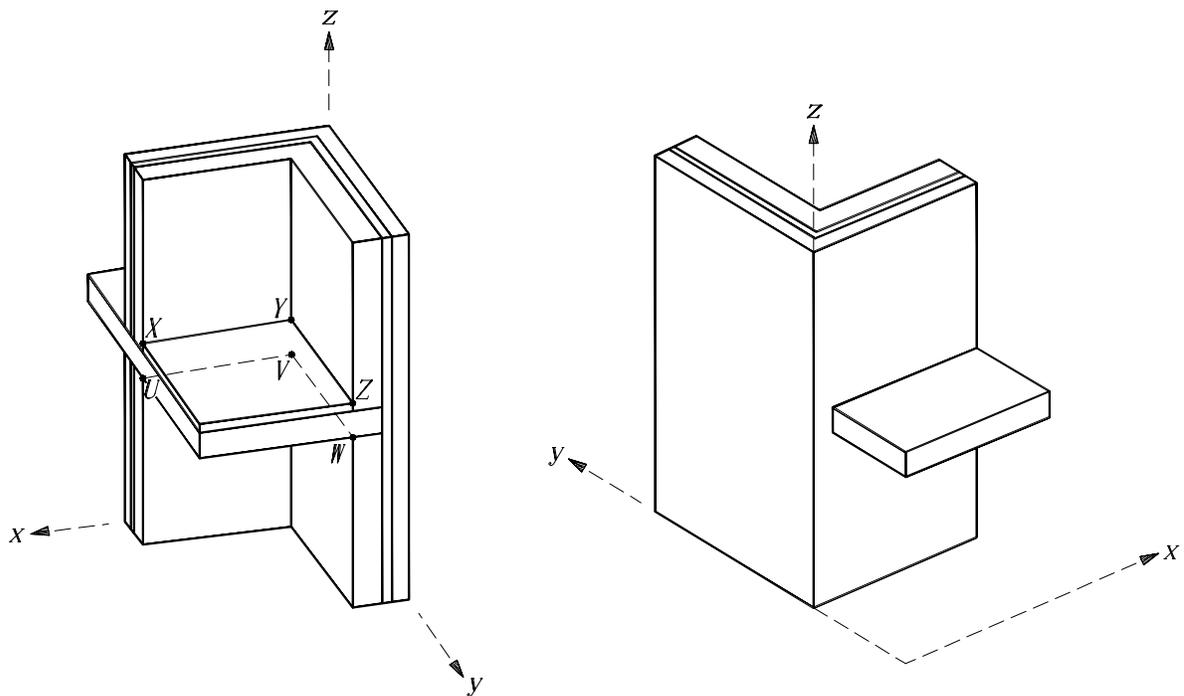


Figure 7.12: Test reference case.

Label	Boundary condition
α	$T=20\text{ }^{\circ}\text{C}$, $R=0.2\text{ m}^2\cdot\text{K}/\text{W}$
β	$T=15\text{ }^{\circ}\text{C}$, $R=0.2\text{ m}^2\cdot\text{K}/\text{W}$
γ	$T=0\text{ }^{\circ}\text{C}$, $R=0.05\text{ m}^2\cdot\text{K}/\text{W}$
δ	$Q=0$ (adiabatic)

Table 7.3: Boundary conditions, see Fig. 7.13.

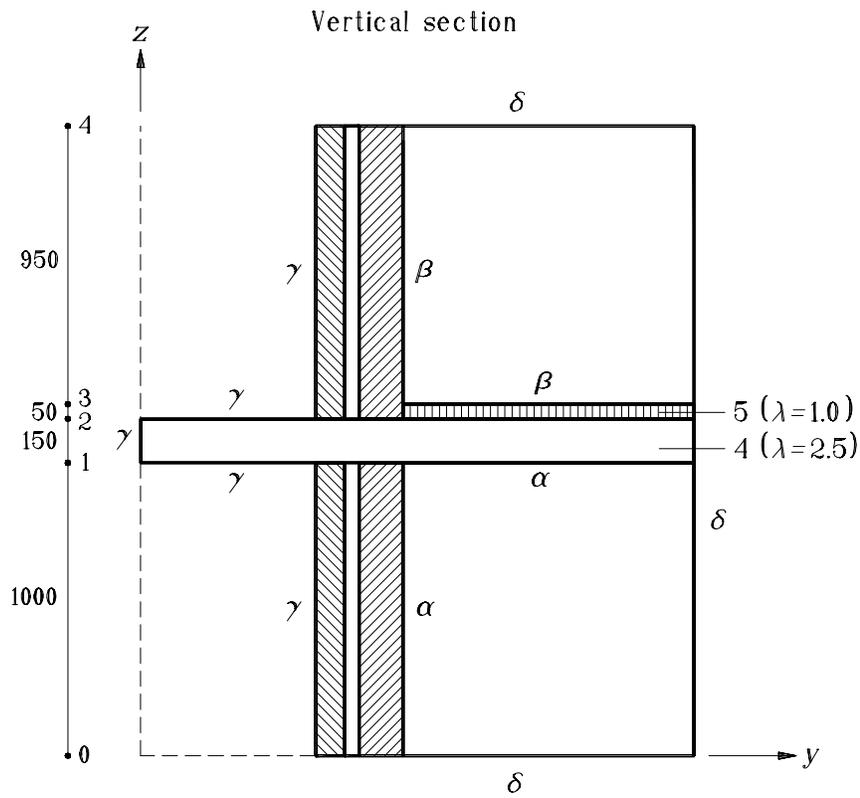
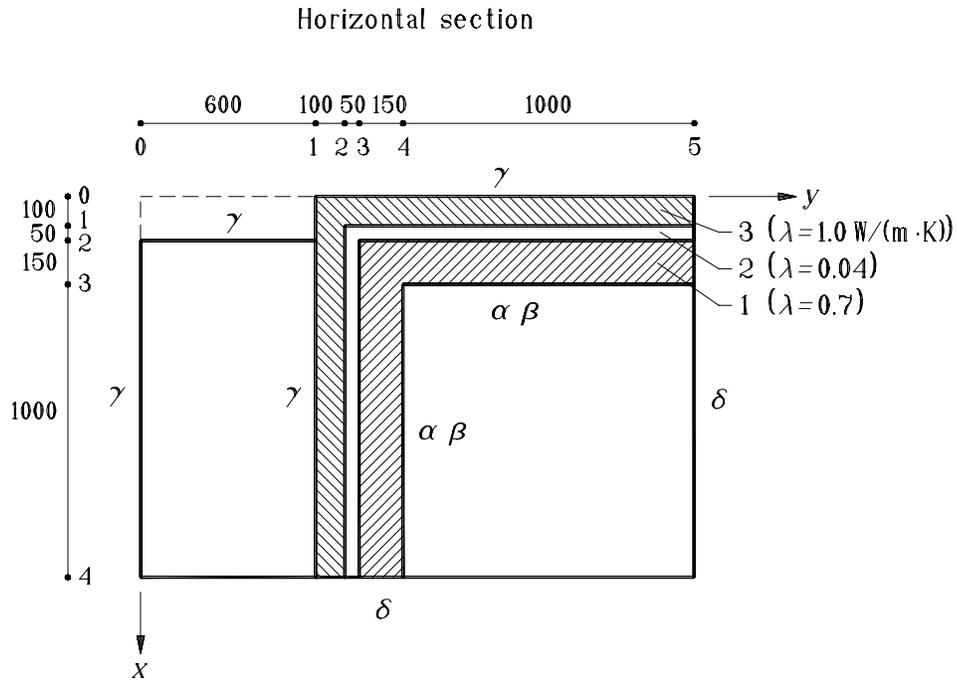


Figure 7.13: Horizontal and vertical sections of the problem. Input mesh coordinates and length of input mesh segments are shown.

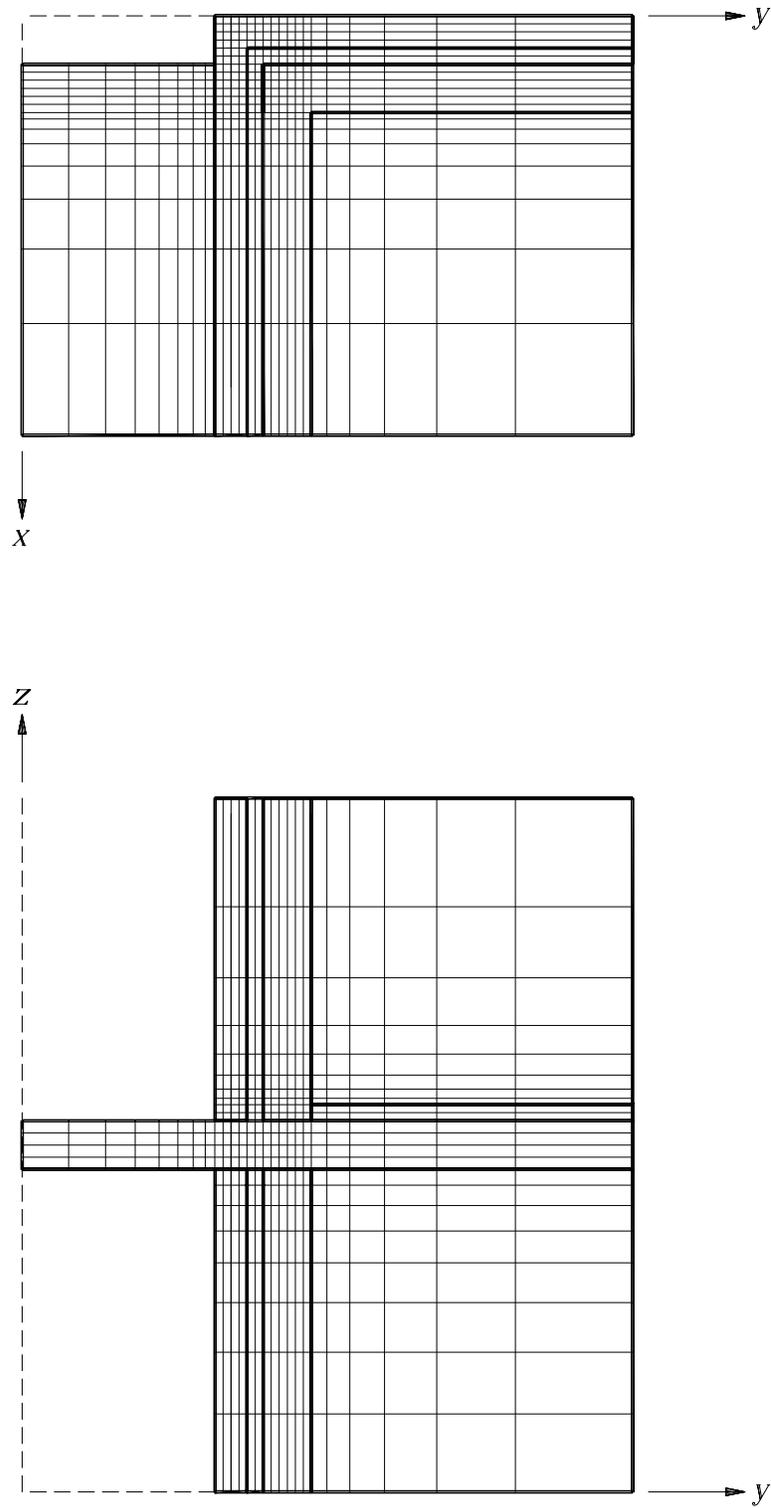


Figure 7.14: Projection of the numerical mesh on the x,y -plane and the x,z -plane involving 7600 cells.

7.5.2 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
```

7.5.3 Results

Figures 7.15 and 7.16 show the heat flows using 7600 and 246000 numerical cells, respectively. Consider figure 7.15. 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/m ²)	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 7.15: Flows through defined sets, $N=7600$.

Set	Q (W)	Q (W/m ²)	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 7.16: Flows through defined sets, $N=246000$.

Table 7.4 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 $\omega=1.85$, 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 7.4: Calculated heat flows (W).

Table 7.5 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 °C. Figure 7.17 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 7.5: Calculated surface temperatures (°C).

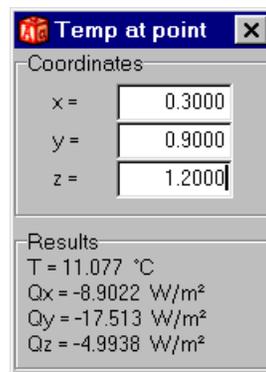


Figure 7.17: Temperatures and heat flows at point y, see Fig. 7.12, $N=246000$.

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

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

7.6.1 Introduction

The following example concerns a slab on the ground, see Figure 7.18. The simplification as shown in Fig. 7.19 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 °C, and an average annual outdoor temperature of 9.4 °C. Table 7.6 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).

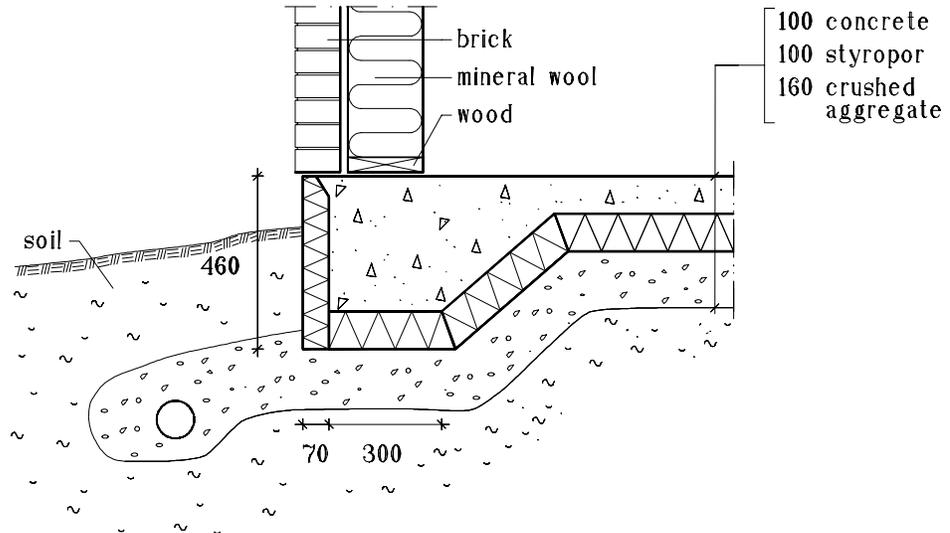


Figure 7.18: Slab on the ground.

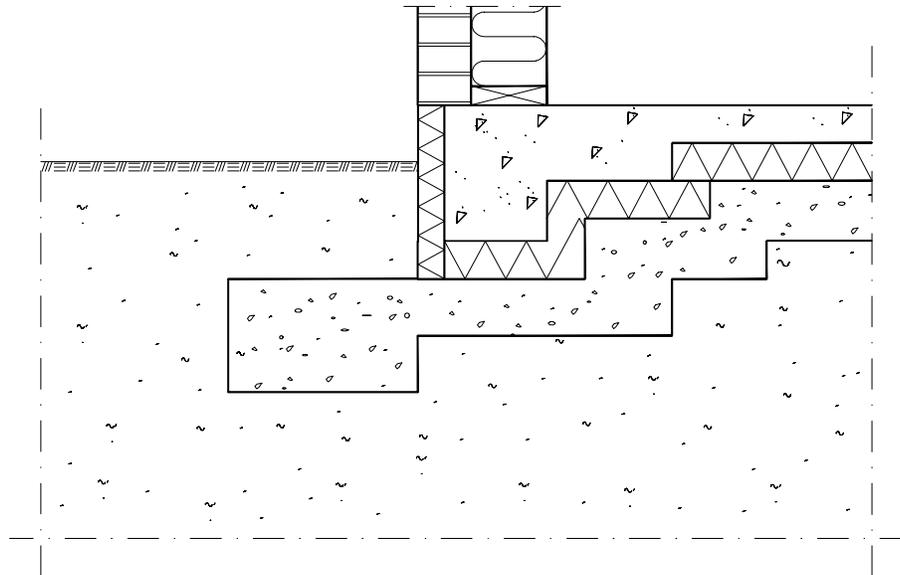


Figure 7.19: Simplified geometry for the calculations.

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

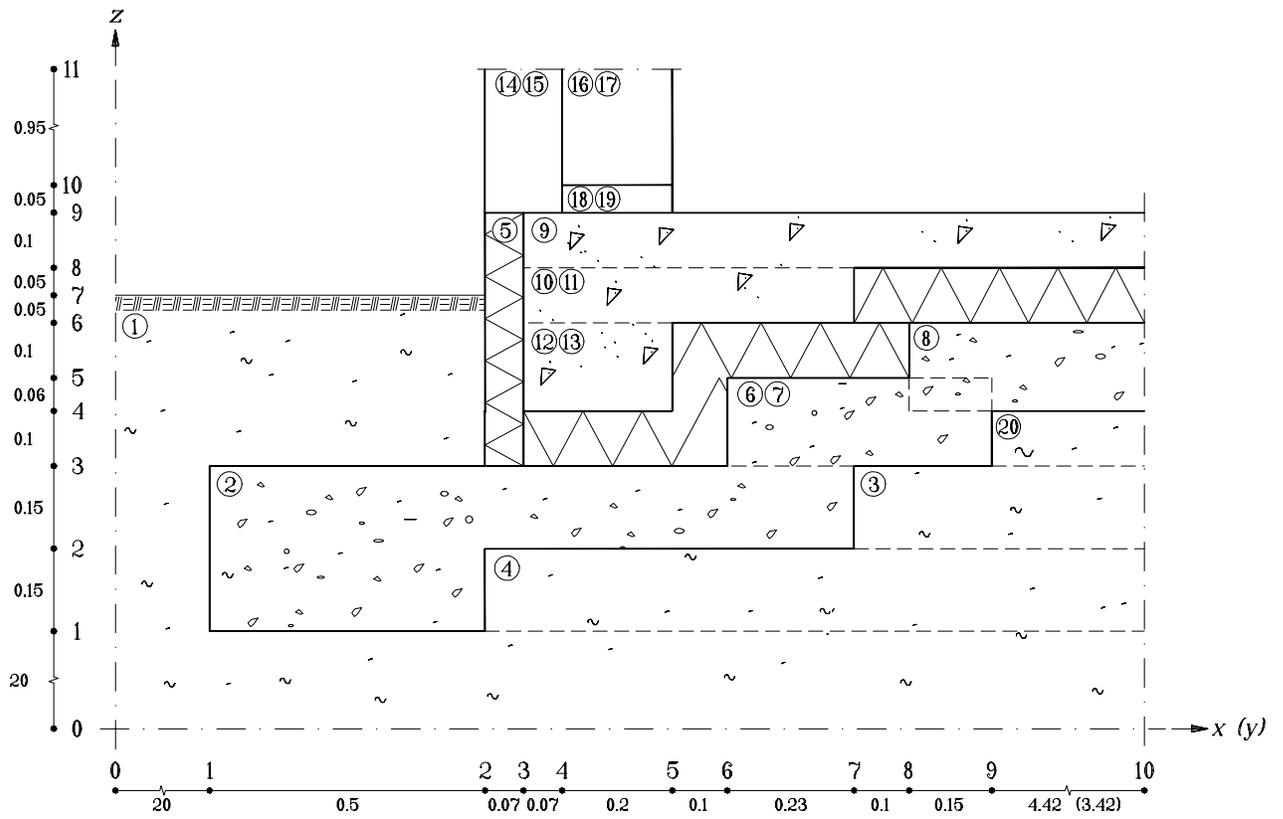
Table 7.6: Thermal conductivities.

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. Figure 7.20 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 Fig. 7.20).

Each input parallelepiped in Fig. 7.20 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, see section 4.1. 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 7.22 shows the 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.



Figur 7.20: 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.

7.7 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 fig. 7.20)
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

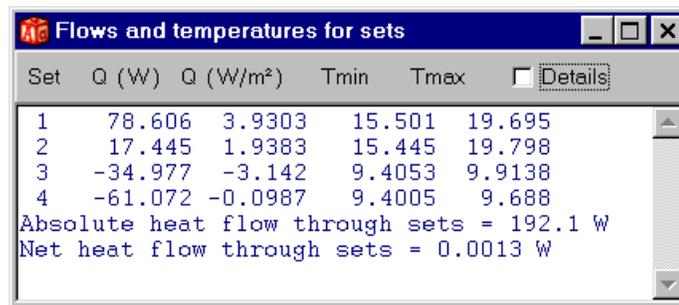
0 no internal resistance

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
```

7.8 Results

Figure 7.21 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 $F=1 \cdot 10^{-3}$ instead of $1 \cdot 10^{-5}$ 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 $61.1 \cdot 4 \cdot 24 \cdot 365 = 2.1$ MWh. Note that the calculated heat flows represent one quarter of the house, so we multiply by 4. Figure 7.22 shows the surface temperatures in gray-scale near the house. Figure 7.23 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 7.24 shows the temperatures on the soil surface. Everything but the soil is invisible here (color is zero for the boxes defined in the input).



Set	Q (W)	Q (W/m ²)	Tmin	Tmax
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 7.21: Calculated heat flows and surface temperatures, ($N=25221$).

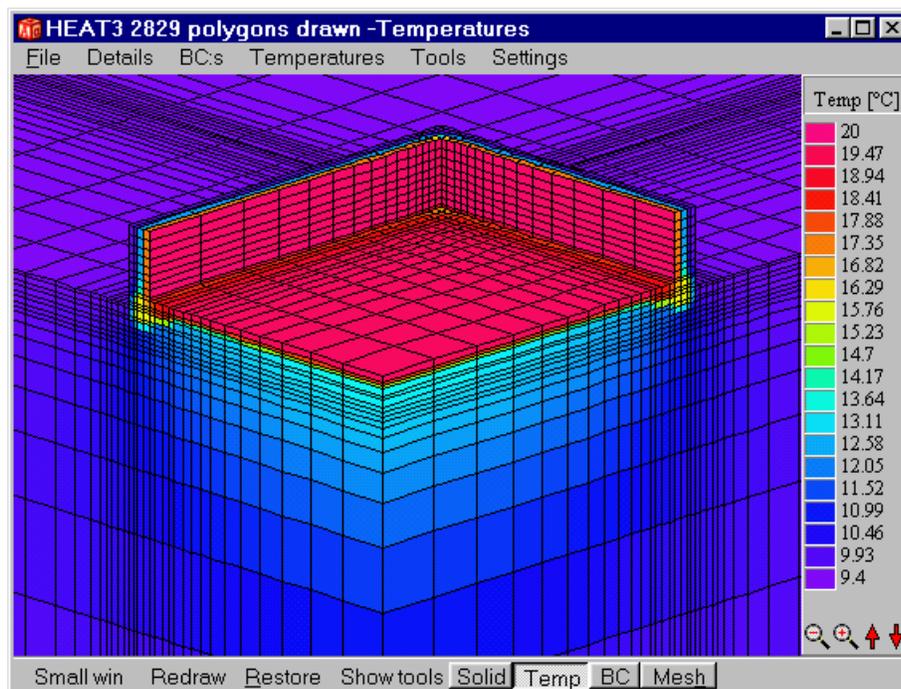


Figure 7.22: Surface temperatures in gray-scale and the mesh around the house.

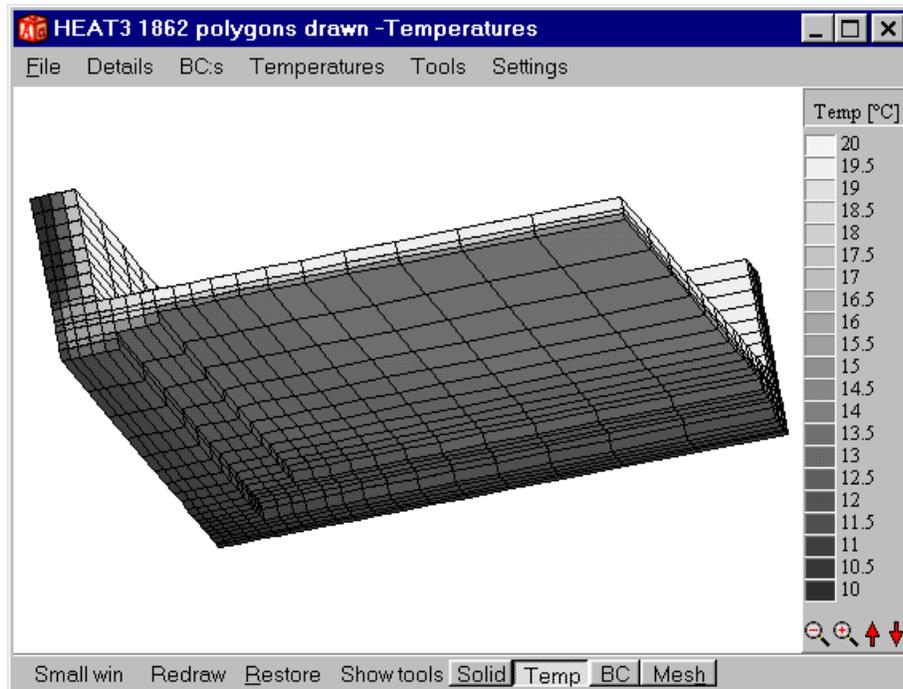


Figure 7.23: Temperatures on the concrete surface shown from below (gray-scale). The soil and the crushed aggregate are invisible here (color is zero for the boxes defined in the input).

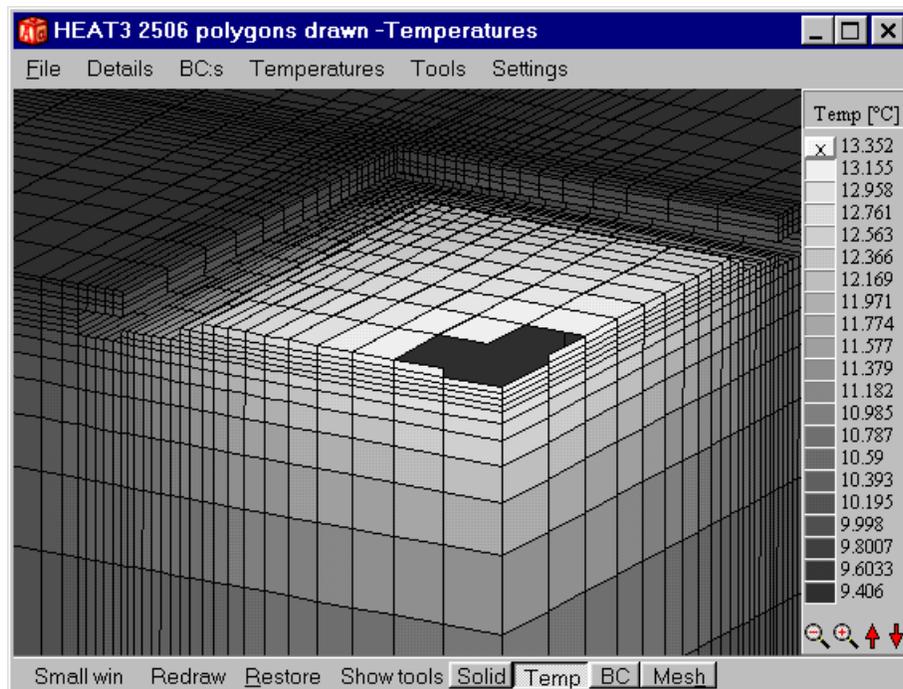


Figure 7.24: Temperatures on the soil surface (gray-scale). Everything but the soil is invisible here (color is zero for the boxes defined in the input). The maximum temperature in the soil is 13.4 °C (these surfaces are made invisible by disabling the interval on the scale to the right).

7.9 Example 9, ground heat storage, cav.dat

7.9.1 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 Fig. 7.25 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 Fig. 7.18. 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 °C, and the temperature at the ground surface is put to 0 °C. The input data file in the next section gives a numerical mesh with 15968 cells. Figure 7.26 shows the input mesh. Figure 7.27 shows the numerical mesh with 15968 cells (projected on the x,z -plane, x,y -plane, and y,z -planes).

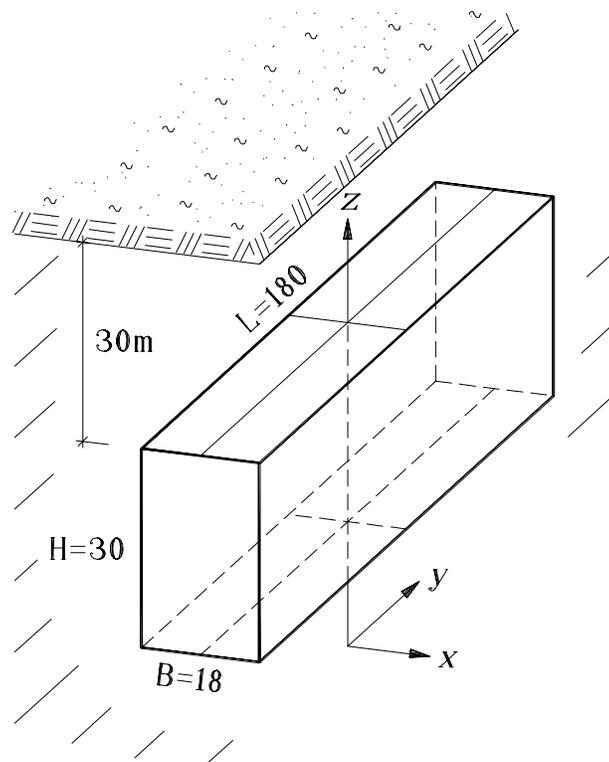


Figure 7.25: Underground cavern used for heat storage.

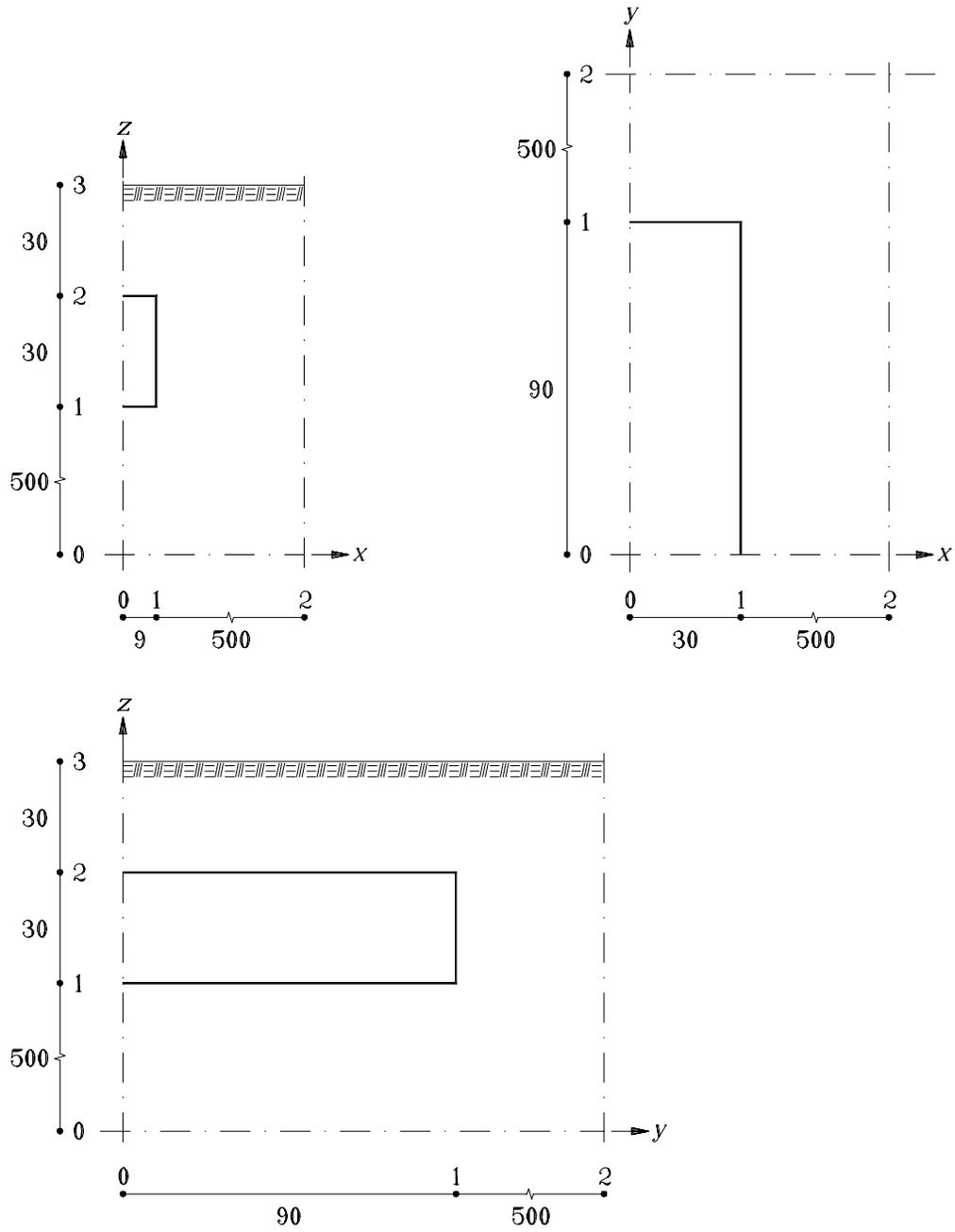


Figure 7.26: Input mesh coordinates and lengths (m).

7.9.2 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
```

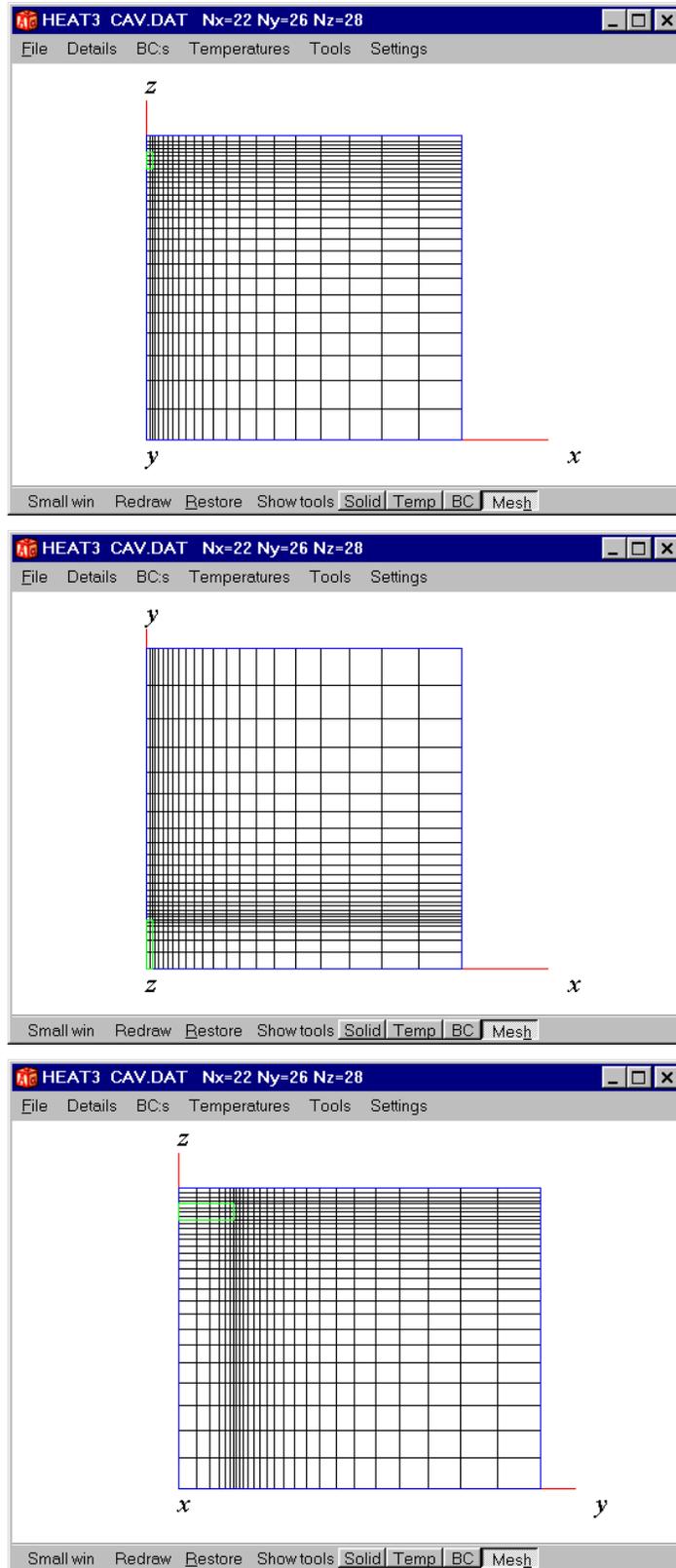
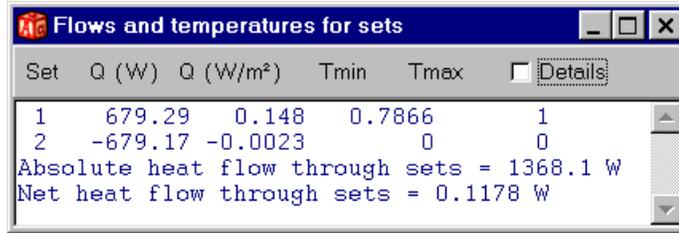


Figure 7.27: Numerical mesh with 15968 cells (mesh projected on the x,z-plane, x,y-plane, and y,z-plane, respectively).

7.9.3 Results

The heat flow from the cavern is $679.4 \times 4 = 2716$ W, see Fig. 7.28. 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 7.29 shows the temperatures and meshes with $N=15968$ (above) and $N=135250$ (below).



Set	Q (W)	Q (W/m ²)	Tmin	Tmax	<input type="checkbox"/> Details
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 7.28: Heat flows for the two sets, $N=15968$.

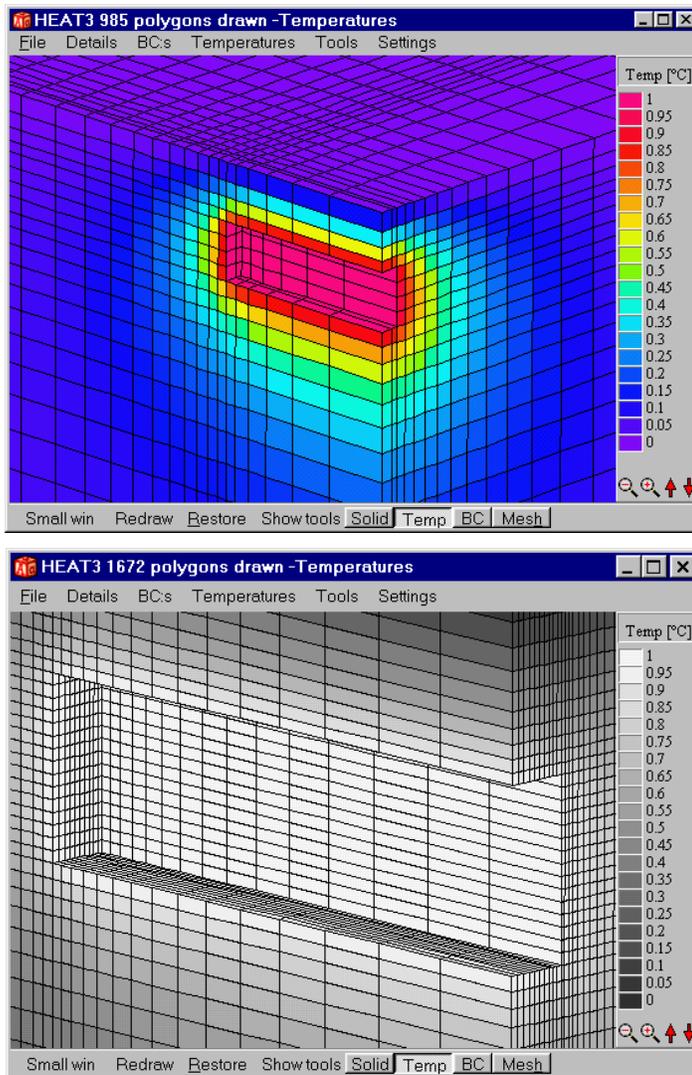


Figure 7.29: Temperatures and mesh near the cavern with $N=15968$ (above) and $N=135250$ (below in gray-scale).

8. A few tips

8.1 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.

8.2 Numerical aspects

8.2.1 Numerical mesh

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

8.2.2 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.

8.2.3 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.

8.2.4 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.

8.2.5 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).

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

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)*. Byggeforskningsrådet. 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/(m ³ ·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, see Section 5.6. 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');
```

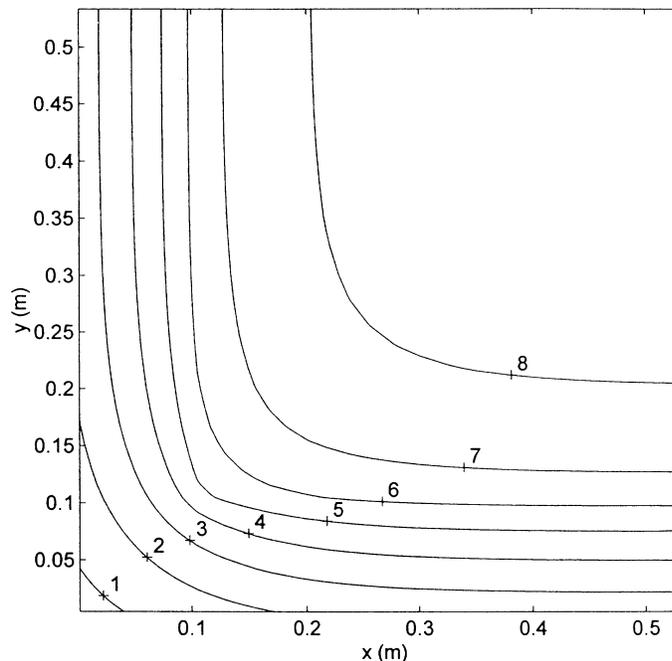


Figure B.8.1: Contour plot of a temperature field for a plane that cuts through the floor, see the corner problem in Section 6.5.