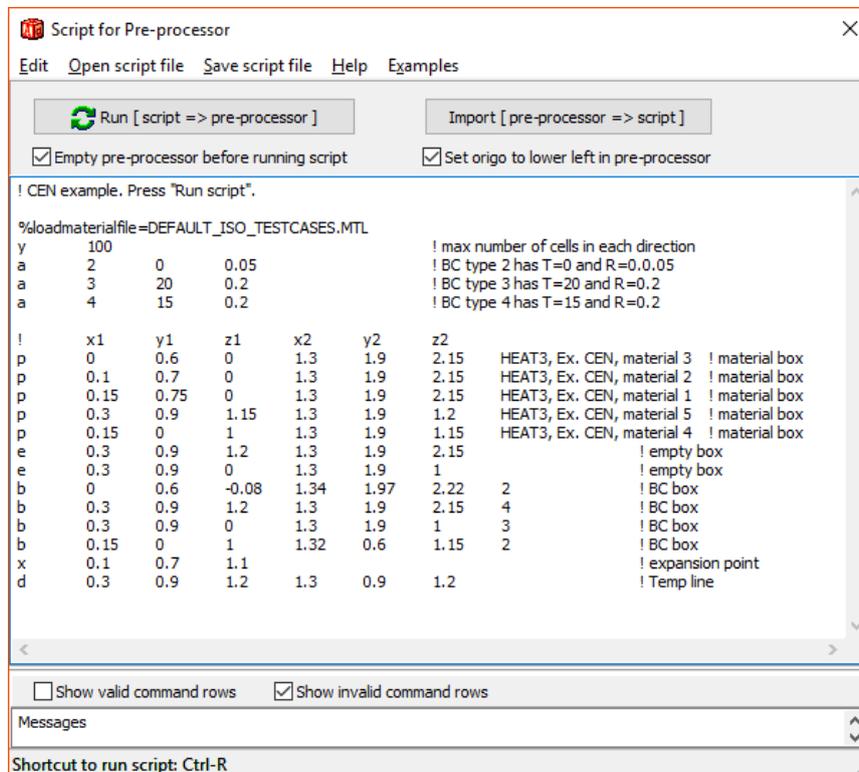


HEAT3 version 8

A PC-program for heat transfer in three dimensions

Update manual



March 23, 2017

BLOCON

www.buildingphysics.com

Contents

1. WHAT'S NEW IN HEAT3 VERSION 7.....	3
1.1 SCRIPT FOR PRE-PROCESSOR	3
1.2 MISCELLANEOUS	3
1.3 INCREASED SIZES FOR DATA FIELDS	3
1.4 REQUIREMENTS	4
1.5 LIGHT VERSION.....	4
2. SCRIPT FOR PRE-PROCESSOR	5
2.1 INTRODUCTION	5
2.2 SCRIPT COMMANDS	6
2.3 EXECUTING THE SCRIPT – AN EXAMPLE	10
2.4 IMPORTING GEOMETRY FROM PRE-PROCESSOR TO A SCRIPT.....	13
2.5 ADJUSTMENT OF THE SCALE-FACTOR.....	14
2.6 EXAMPLE CEN	16
2.7 EXAMPLE SLAB.....	22
2.8 RESULTS	26
2.9 ESTIMATION OF NUMERICAL ERROR.....	29
2.10 IMPROVING THE MESH (“ADVANCED USERS”)	31
2.11 SAVE/OPEN SCRIPT FILES	35
3. A FEW TIPS	36
3.1 INTRODUCTION	36
3.2 NUMERICAL ASPECTS	36
3.2.1 <i>Numerical mesh</i>	36
3.2.2 <i>Expansive meshes</i>	36
3.2.3 <i>Steady-state calculations</i>	36
3.2.4 <i>Symmetric cases</i>	36
3.2.5 <i>Computational area for calculations of heat losses to the ground</i>	36
3.2.6 <i>Transient calculations for heat losses to the ground</i>	36
4. MISCELLANEOUS	38
4.1 ADAPTED FOR WINDOWS 7 AND 8.....	38
4.2 IMPROVEMENTS FOR “TEXT EDITOR”	39

1. What's new in HEAT3 version 7

Blocon is proud to present a new version of HEAT3. Many new important features have been added. Up-to-date information is given on www.buildingphysics.com.

This update manual covers the new features that have been added since version 7. New users should also read the update manuals for version 7, 6, 5, and the full manual for version 4 at

http://www.buildingphysics.com/manuals/HEAT3_7_update.pdf

http://www.buildingphysics.com/manuals/HEAT3_6_update.pdf

http://www.buildingphysics.com/manuals/HEAT3_5_update_manual.pdf

http://www.buildingphysics.com/manuals/HEAT3_4.pdf

Tips for reading for beginners:

See examples in section 2.6 and 2.7.

For a quick start read Chapter 4 in Manual [HEAT3 4.0](#). The examples in Chapter 5 would also be useful. After this, look at [Update manual for HEAT3 5.0](#) and [Update manual for HEAT3 6.0](#). Also see the examples for the test cases: [ISO 10211 validation test cases](#).

1.1 Script for pre-processor

The complete geometry can now efficiently be described by text. It is also possible to vice versa convert already drawn geometry in the pre-processor to a script. See Chapter 2.

1.2 Miscellaneous

Many minor improvements are made since version 7. The most important are shown below:

- Better adaption to Windows 7, 8 and 10.
- Single-core calculation is always used by default if the number of cells is less than 500,000 (v7 used 100,000 in some cases).
- Improvements in “Text editor”, see section 4.2.

1.3 Increased sizes for data fields

Some data fields have been increased in sizes, see below.

Maximum number of	HEAT3 version 7	HEAT3 version 8
Material boxes	500	1,000
Boundary condition boxes	30	50

1.4 Requirements

HEAT3 runs on both 32-bit and 64-bit Windows 7, 8, and 10, and compatibles. Only about 15 MB hard disk space is needed.

HEAT3 needs about 1.7 GB RAM allocated space when all nodes (50 million) are used. A PC with 3 GB RAM is recommended in that case.

Version	Max number of nodes	Required memory RAM	Recommended RAM for PC
HEAT3_v8.exe	50,653,000 (370·370·370)	1.7 GB	3 GB

A good tip is to close down other applications to increase the available memory (if insufficient). If HEAT3 is swapping data to/from the hard disk it is a sign that virtual memory beyond the installed RAM is used. This will happen if you try to run large cases on a PC with too little memory. This can cause very long calculation time since the hard disk will be used for some of the memory for the allocated matrices. It should not be any problem unless the PC has less than 3GB RAM.

1.5 Light version

The light (demo) version has the following restrictions:

- Data cannot be saved.
- Materials cannot be added or changed.
- Max 1,000,000 (100·100·100) nodes.
- Results for heat flows are hidden for larger problems with more than 25,000 nodes and with more than 30 nodes in each direction.
- Thermal coupling coefficients (L2D) will not be calculated.
- Report preview is disabled.
- Some other restrictions apply.

2. Script for Pre-processor

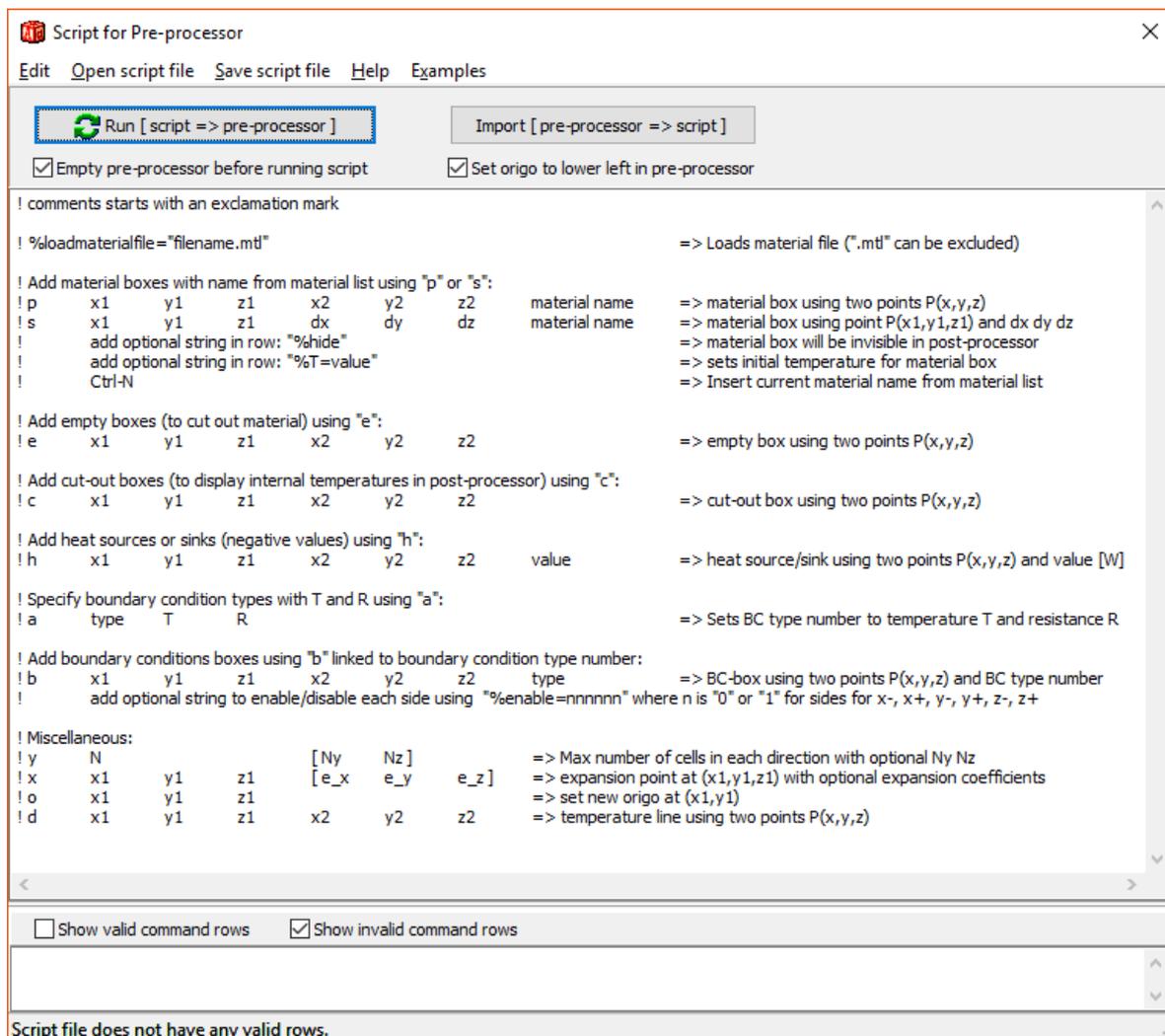
2.1 Introduction

Geometrical input can now be given using text scripts just like in HEAT2 v10. It is possible to run different scripts in order to build a complete geometry. E.g. the user can have different scripts describing e.g. wall types and combine those with different scripts describing e.g. window frames.

It is also possible to vice versa generate a script from an already drawn geometry in the pre-processor. This means that a user can “export” already drawn geometries from the pre-processor and save them in a library for future use.

It is also possible to give simple boundary conditions and numerical mesh which means that a complete case often can be described using a single script. So, for many cases it will not be necessary to load/save a project-file (.h3p). You can just start HEAT3, open a script file and run it. For some cases, e.g. if a boundary condition with a temperature function is used, or data for the recorder is given, etc., the h3p-file needs however to be saved/loaded in order to keep these settings.

Input data is given in SI-units (e.g. lengths are given in meters).



Press "Help" to show this text.

2.2 Script commands

Comments starts with an exclamation mark. The following script commands are possible.

%loadmaterialfile="filename.mtl" => Loads material file (.mtl can be excluded)

Material boxes can be specified using one of the below options:

p x1 y1 z1 x2 y2 z2 material name => box using two points P(x,y,z)

s x1 y1 z1 dx dy dz material name => box using point P(x1,y1,z1) and dx, dy,dz

Example:

```
p 0 0 0 0.1 0.4 0.5 concrete, IEA ! material box
```

The material name "*concrete, IEA*" should be a string from the material list. Make sure that the correct material file is loaded, either in the current session of HEAT3 or by using the command %loadmaterialfile.

There is a short-cut to insert the current material name text string from item shown in the material list: **Ctrl-N** (or menu item "Edit/Insert material name").

The optional string "%hide" will make the material box invisible in the post-processor (allowing internal temperatures between materials to be viewed). Example:

```
p 0 0 0 0.1 0.4 0.5 concrete, IEA %hide
```

The optional string "%T=value" will set an initial temperature (°C) for the material box. Example:

```
s 0 0 0 0.1 0.4 0.5 concrete, IEA %T=10
```

A tip: comment out objects if you do not want to use them, such as:

```
!p 0 0 0 0.1 0.4 0.5 concrete, IEA
```

Add **empty boxes** (to cut out material) using "e":

```
e x1 y1 z1 x2 y2 z2 => empty box using two points P(x,y,z)
```

Add **cut-out boxes** (to display internal temperatures in post-processor) using "c":

```
c x1 y1 z1 x2 y2 z2 => cut-out box using two points P(x,y,z)
```

Add **heat sources or sinks** (negative values) using "h":

```
h x1 y1 z1 x2 y2 z2 value (negative value means heat sink)
```

=> heat source/sink using two points P(x,y,z) and value

It is possible to specify simple **boundary conditions** with temperature T and resistance R using "a":

```
a type T R => Sets BC type number to T and R
```

The type number must be greater than "1" since the first BC type is reserved to be adiabatic (Q=0).

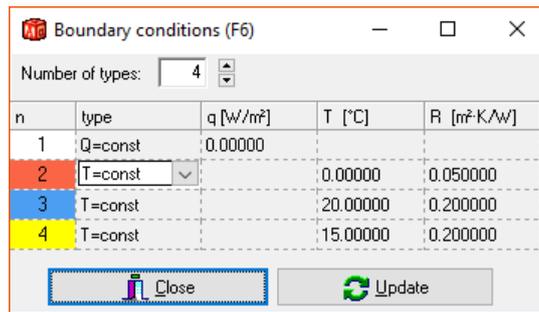
Example: Assume we have the following script command:

```
a 2 0 0.05 ! BC type 2 has T=0 and R=0.05
```

```
a 3 20 0.2 ! BC type 3 has T=20 and R=0.2
```

```
a 4 15 0.2 ! BC type 4 has T=15 and R=0.2
```

This will set the following boundary condition types (press F6):



It is not possible using a script command to define a boundary condition type that is a given heat flow, or where T is a function of time. This can however be made as usual in the HEAT3 user interface and a BC box can be linked to this type using the command “b”, see below.

Add **boundary conditions boxes** using "b" linked to boundary condition type number:

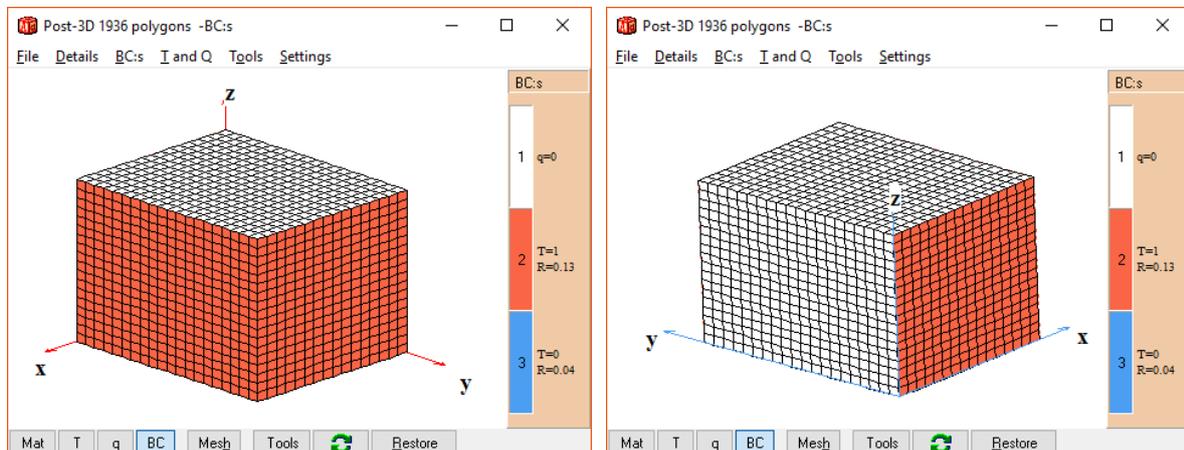
`b x1 y1 z1 x2 y2 z2 type`

=> BC-box using two points $P(x,y,z)$ and BC type number (as shown in F6-window above).

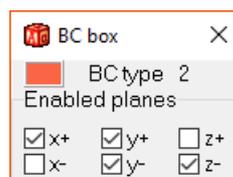
An optional string can be added to enable/disable each side of a BC box using "**%enable=nnnnnn**" where n is "0" or "1" for sides for x- x+ y- y+ z- z+. Here “x-“ means the bottom side in x-direction and “x+” the upper side in x-direction. Example with one material box and one BC box:

`p 0 0 0 0.47 0.57 0.51 acrylic resin, no cap., CEN`
`b 0 0 0 0.47 0.57 0.51 2 %enable=011110`

This will disable lower side (y,z-plane) in x-direction, and upper side (x,y-plane) in z-direction:



We can also check enabled sides in the pre-processor by double-clicking on the created BC-box:



If the enable command is not given, all 6 sides of a BC-box will be enabled by default.

Note that the %enable command rarely need to be used. The same BC:s as above could easily be generated just by moving the end points x1 and z2, so that those BC box sides do not touch the sides of the material box:

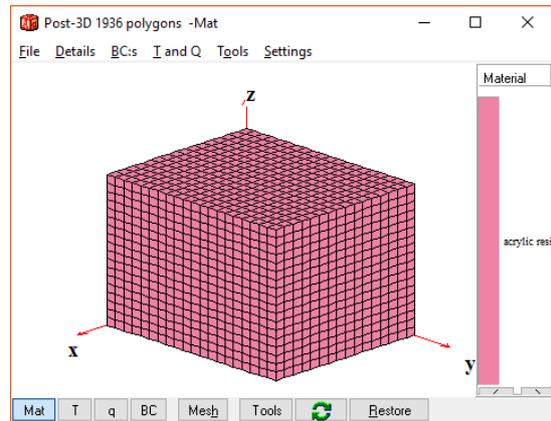
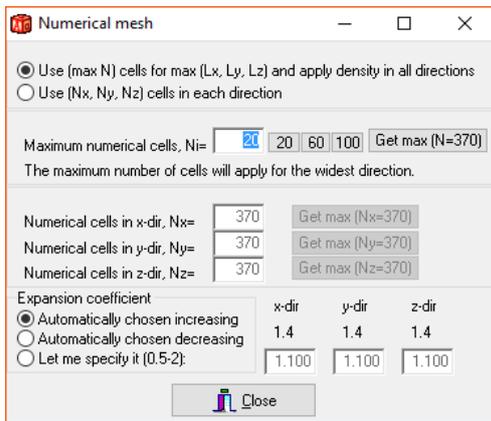
p 0 0 0 0.47 0.57 0.51 *acrylic resin, no cap., CEN*
 b -0.01 0 0 0.47 0.57 0.52 2

The **number of numerical cells** can be set as follows using “y”:

y N $[N_y \quad N_z]$ => *Max number of cells in each direction with optional Ny Nz*

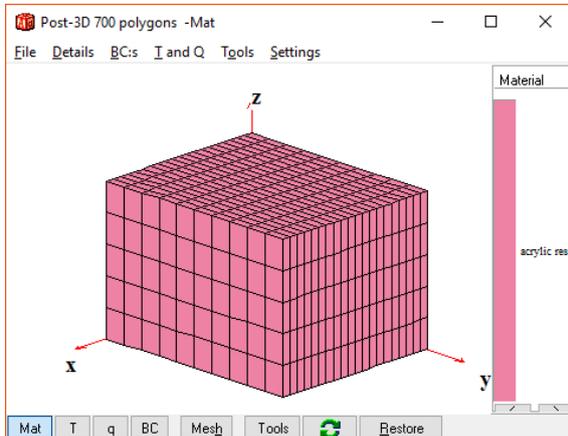
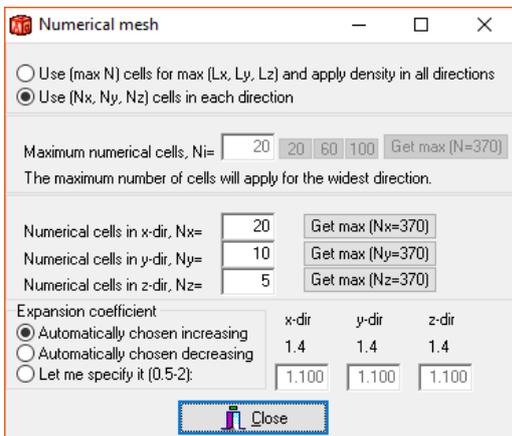
Examples: If only “N” is given, the value will apply to all directions, see mesh dialog below:

y 20 *! N=20*



y 20 10 5

! number of cells Nx Ny Nz



An **expansion point** can be set as follows using “x”:

x $x1$ $y1$ $z1$ $[e_x \quad e_y \quad e_z]$

=> expansion point at (x1,y1,z1) with optional expansion coefficients in each direction.

A coefficient larger than 1 will create an expansive mesh where cell size increases away from the point. A coefficient less than 1 will create on expansive mesh where cell size decreases towards to the point. The coefficient must be between 0.5 and 2. If no coefficients are given, HEAT3 will automatically choose a coefficient generating an increasing mesh.

Origo can be moved by
 $o \quad x1 \quad y1 \quad z1$

=> set new origo reference at $(x1,y1,z1)$

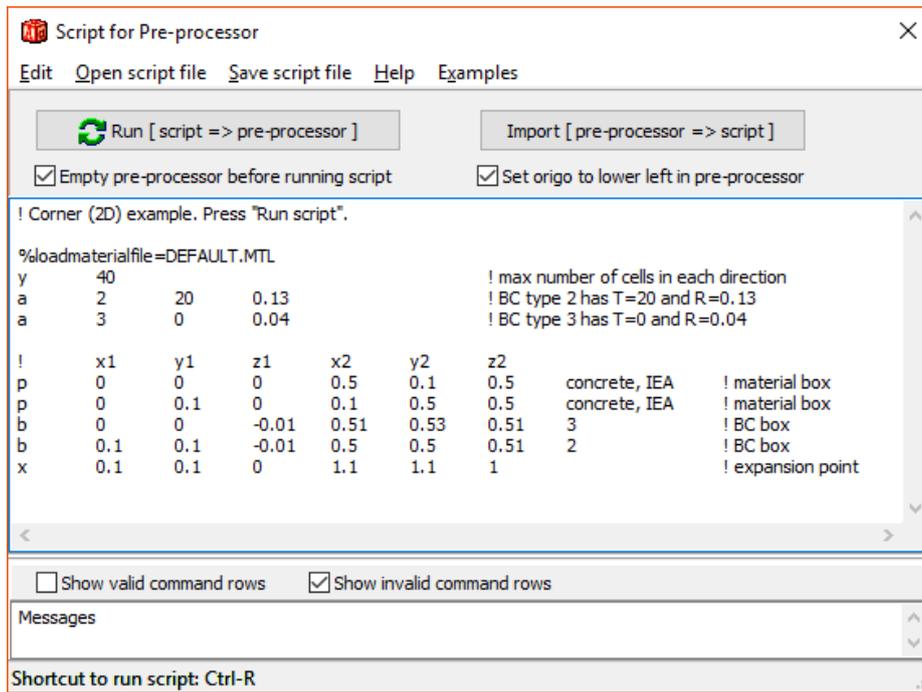
Objects drawn after this command will use this as a new reference point. See example in next section.

Add **temperature lines** (see menu item “Output/Temperature along line”) using “d”:

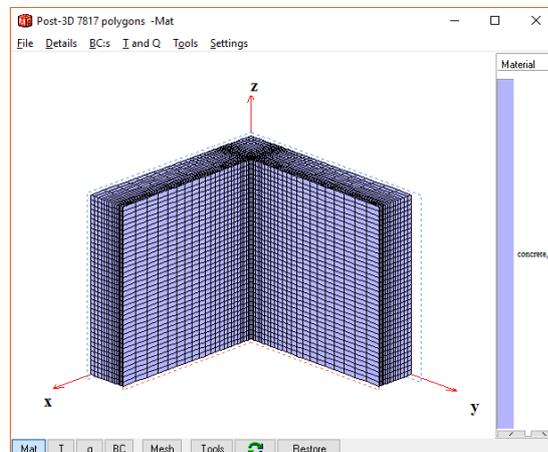
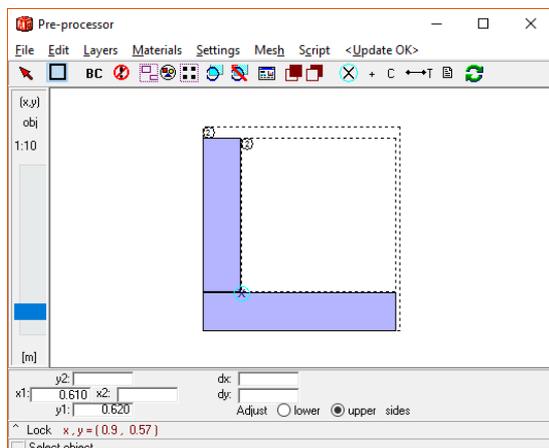
$! d \quad x1 \quad y1 \quad z1 \quad x2 \quad y2 \quad z2$ => temperature line using two points $P(x,y,z)$

2.3 Executing the script – an example

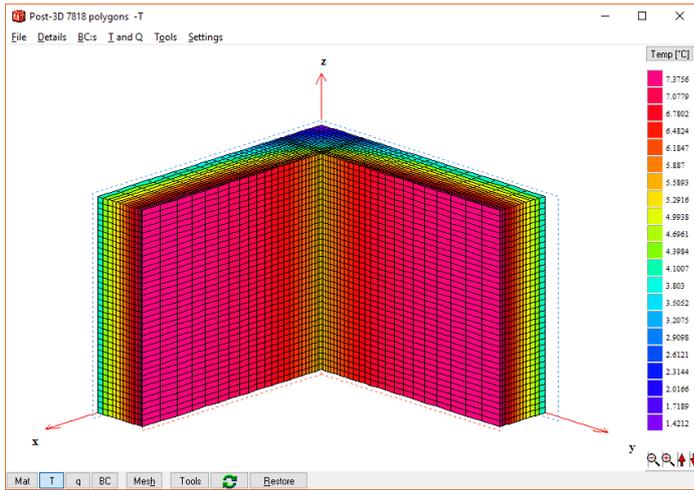
The script window is available from the top menu bar in the pre-processor:



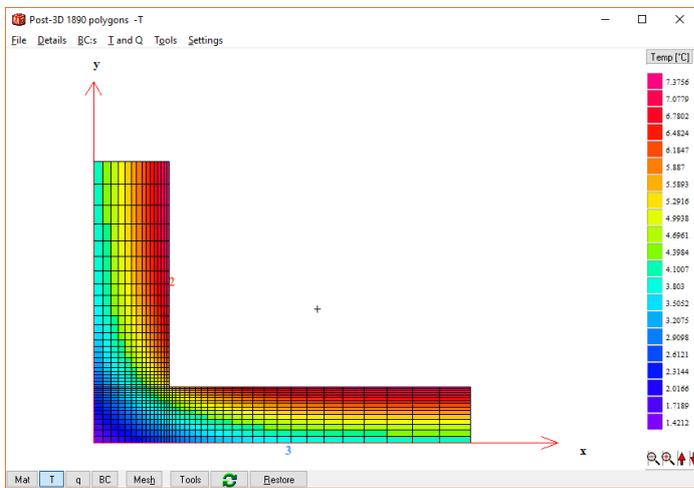
The button ”**Run (script => pre-processor)**” (or short cut $Ctrl+R$) will execute the script. The example above (click “Examples/Corner (2D) example”) will generate the following geometry:



Running the case for steady-state (press F9) gives the following calculated temperatures:

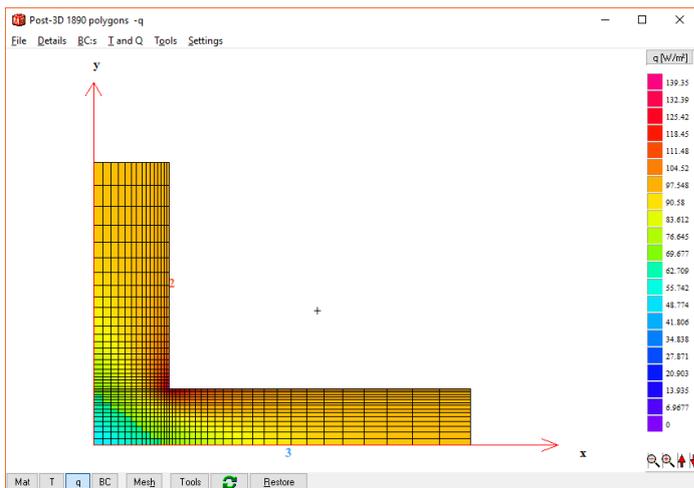


Temperatures

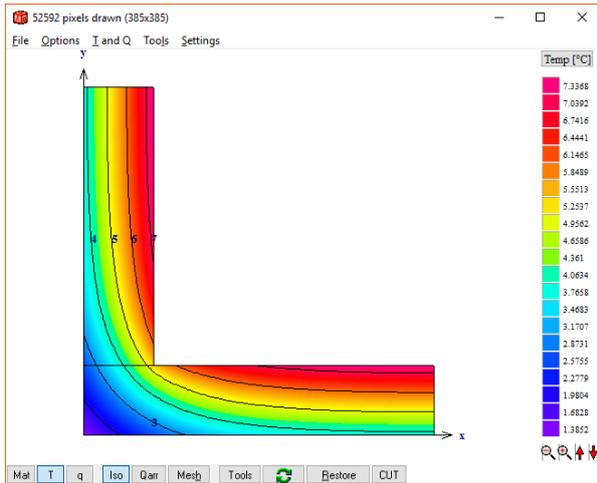


Temperatures

(Press "3" in the post-processor to get the x,y-plane view, press "0" to get back to 3D-view, see Tools/Help shortcuts)



Heat fluxes

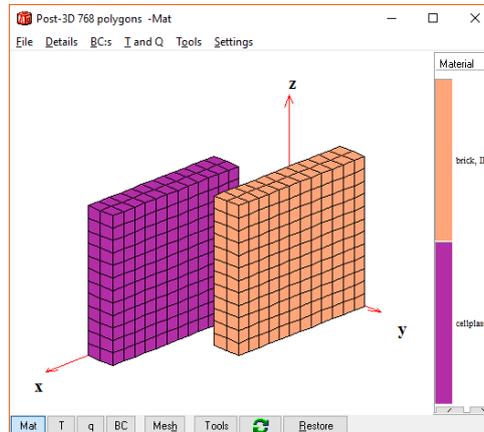


Isotherms can be viewed in the “Post-2D” window.

In the example below we have inserted “o 0.3 -0.2 0” which means that we move the origo with offset (0.2, -0.3, 0) and then all objects are added again referenced to the new origo.

y	20							<i>! max number of cells in each direction</i>
p	0	0	0	0.5	0.1	0.5	brick, IEA	<i>! material box</i>
o	0.3	-0.2	0					
p	0	0	0	0.5	0.1	0.5	cellplast	

This will give the following geometry:

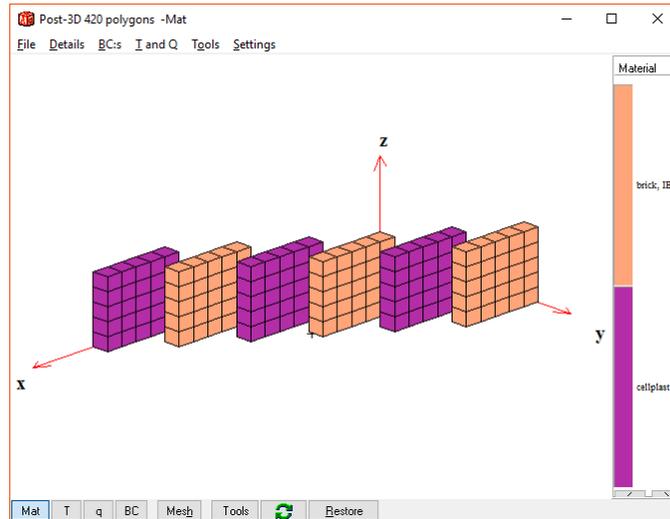


The geometry will be added to the existing pre-processor drawing If “**Empty pre-processor before running script**” is not checked.

Note that all objects for the last imported script will be marked (in red) in the pre-processor thus making it easier to move, delete, etc.

Run the script below three times with “**Empty pre-processor before running script**” un-checked the second and third time (you can just copy the rows below and paste it in the script editor):

y	20							<i>! max number of cells in each direction</i>
p	0	0	0	0.5	0.1	0.5	brick, IEA	<i>! material box</i>
o	0.3	-0.2	0					
p	0	0	0	0.5	0.1	0.5	cellplast	
o	0.3	-0.2	0					

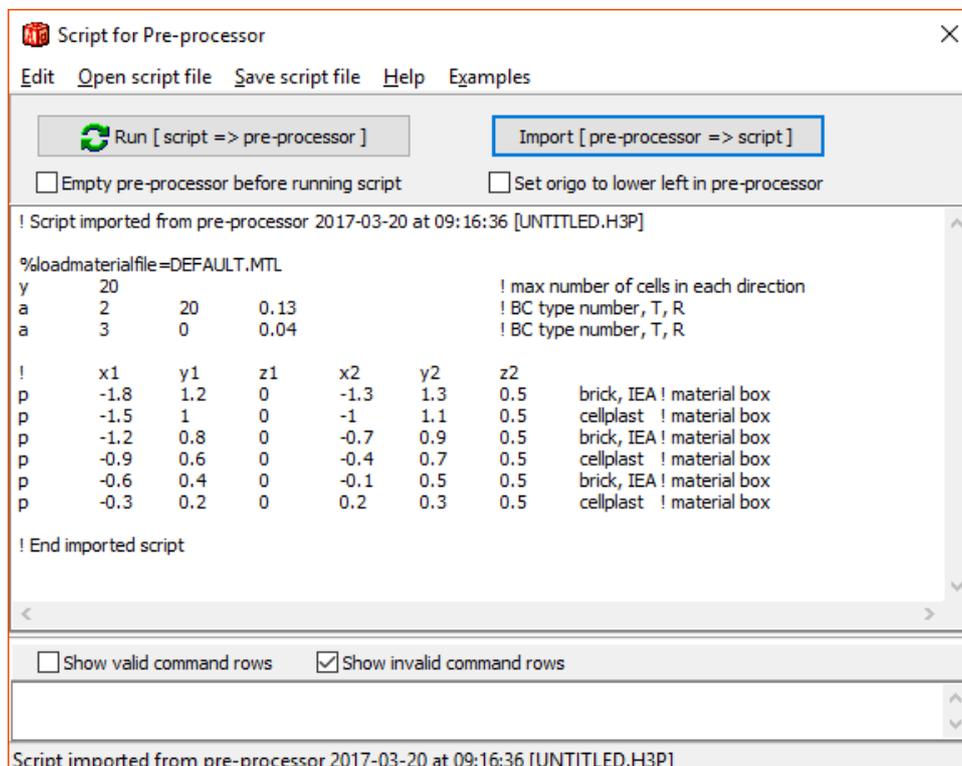


Setting a new origo can be useful e.g. in the following scenario: Assume you have a script that describes a detail (e.g. a window frame). You can use the same script over and over again and place the detail anywhere by first setting a new origo as a reference.

2.4 Importing geometry from pre-processor to a script

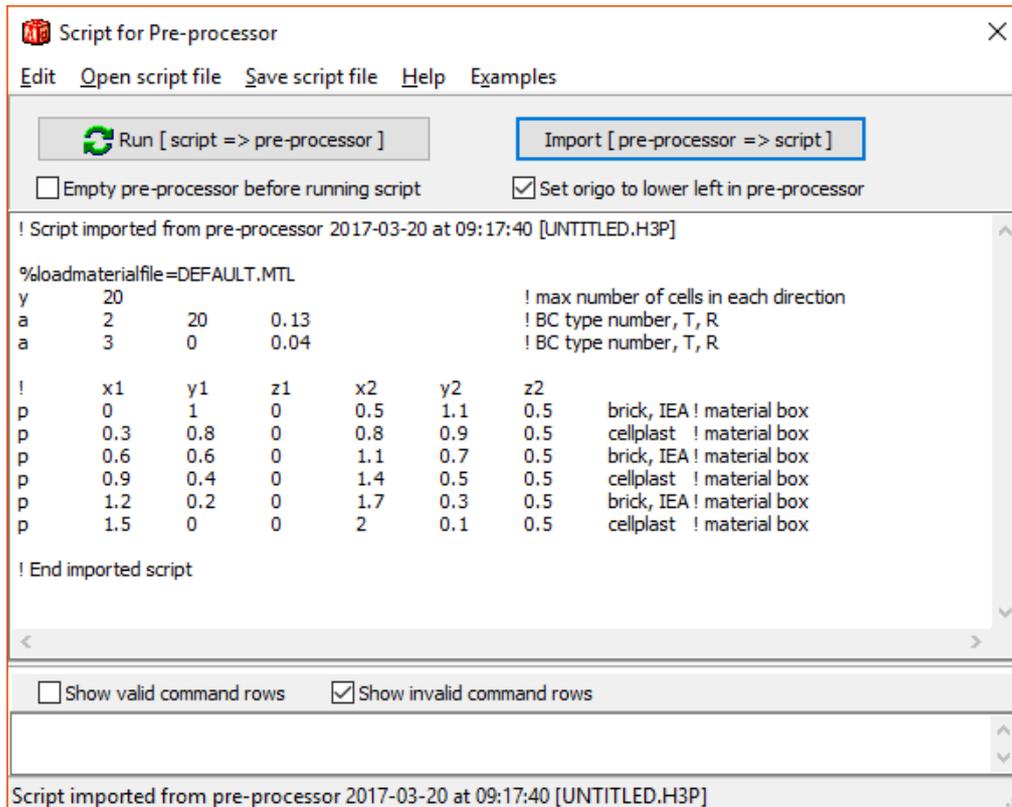
The button “**Import [pre-processor => script]**” will generate a script for the drawn objects. Assume we import the geometry as shown in the last picture.

First, uncheck “Set origo to lower left in pre-processor” and press the button “**Import [pre-processor => script]**”. The generated script will be as follows:



A comment is shown on the first row (and at the window bottom) when the script was generated. Note that info for material file, mesh and boundary condition types always will be included.

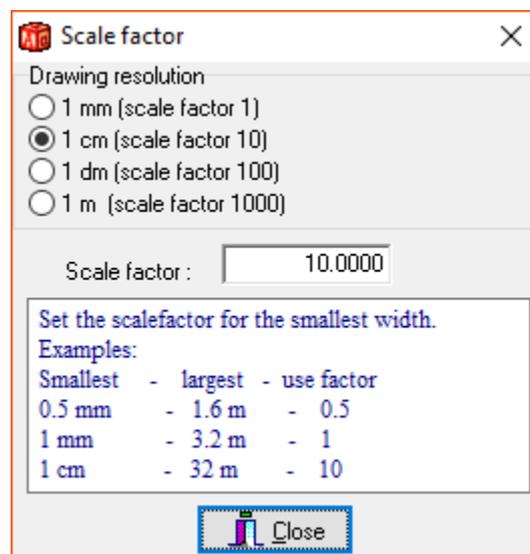
If “Set origo to lower left in pre-processor” is checked, the origo will always be at the lower left/down. The generated script will now be as follows:



2.5 Adjustment of the scale-factor

The scale factor should initially be set according to the smallest width that is to be drawn, see p. 49 at http://www.buildingphysics.com/manuals/HEAT3_4.pdf

By default, the scale factor is 10 which would give 10 mm for the smallest size to be drawn in the pre-processor when snap to grid is on:



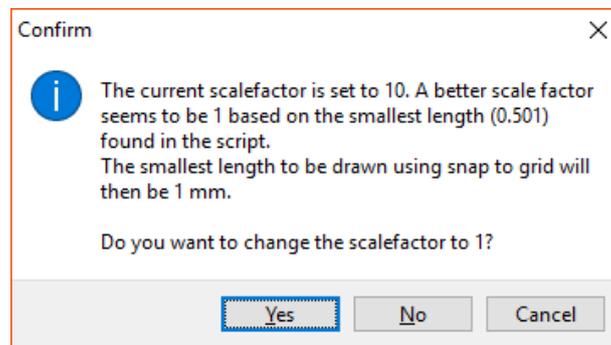
When a script is exported a suitable scale factor will be proposed if the current scale factor is too high. The reason is that it will be easier for the user to draw new objects aligned with the objects created by the script using a proper scale factor (by default there is a grid of 10 pixels in the pre-processor that the drawn objects will snap to).

It should be noted that it is not necessary to change the scale factor when executing the script. The correct geometry will be created regardless of the chosen scale factor. It will just be easier to edit/add objects using the existing alignment with a proper scale factor.

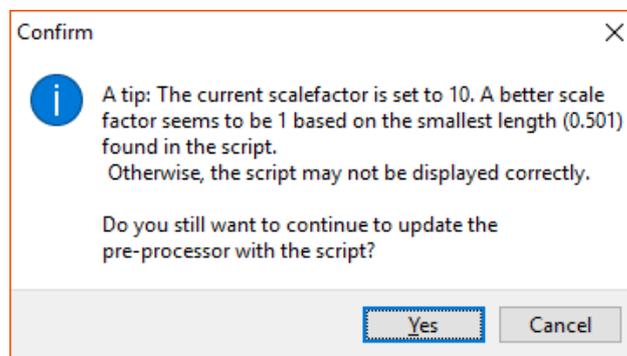
As an example, assume the default scale factor is “10” (meaning a resolution of 10 mm) and that we run a script with a rectangle that is 30x1 mm (that would require a resolution of 1 mm):

```
p    0    0    0    0.501 0.1    0.5    brick, IEA
```

There will be a suggestion to change the scale factor when running the script:



If “**Empty pre-processor before running script**” is not checked, and if the pre-processor is not empty, there will be a warning instead (if the current scale factor is too high):

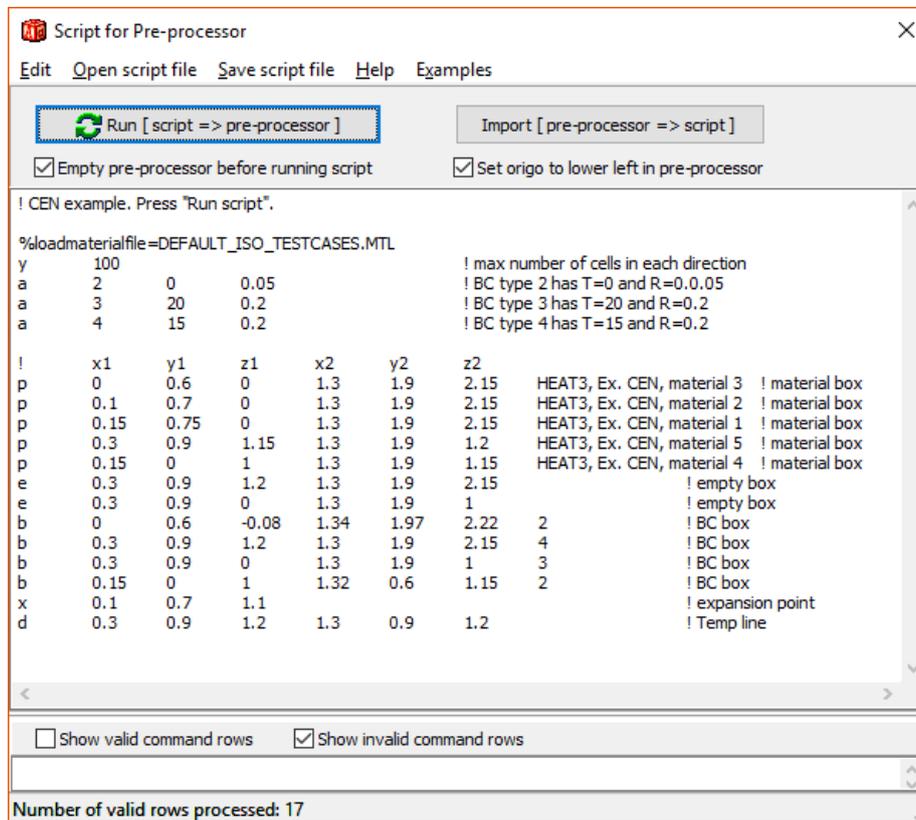


So, if the pre-processor is not empty, and we want to add objects from a script, the scale factor will not be changed if it is too high since this would affect the geometry already drawn. Only a warning will be shown.

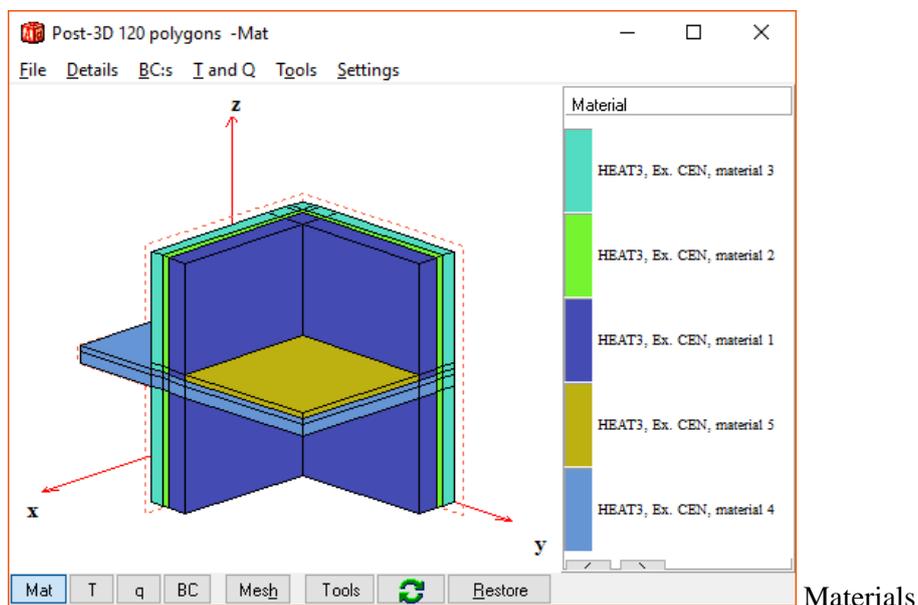
2.6 Example CEN

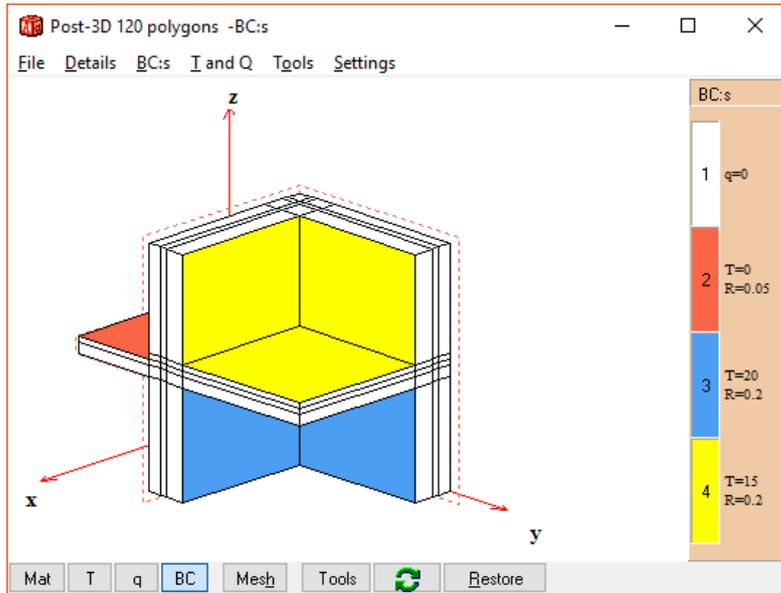
The following script describes the input for a test case in ISO 10211:2007, see <http://www.buildingphysics.com/index-filer/ISOValidation.htm>

Press menu item “Examples/Cen example” to view the script:



Run the script and press F9 for a steady-state solution. Below we will discuss the script rows. First, change the row for the mesh from “y 100” to “y 1”. This will make it easier to see the geometry in the post-processor. Press “Run”. The result is shown in the post-processor:



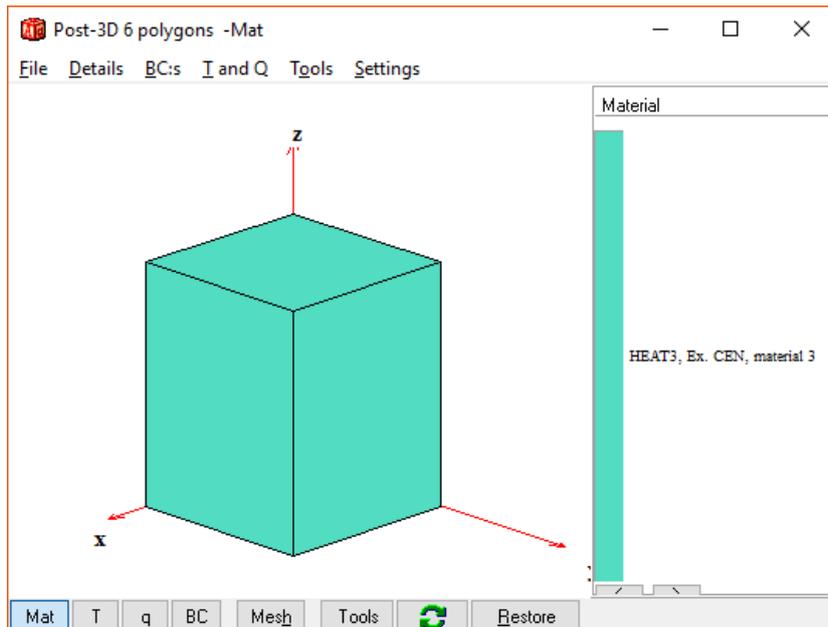


Boundary conditions

Note how materials are overlapped (last given box will replace earlier given boxes if they share the same space). To illustrate this comment out all rows with 'p' and 'e' except the first one using "!":

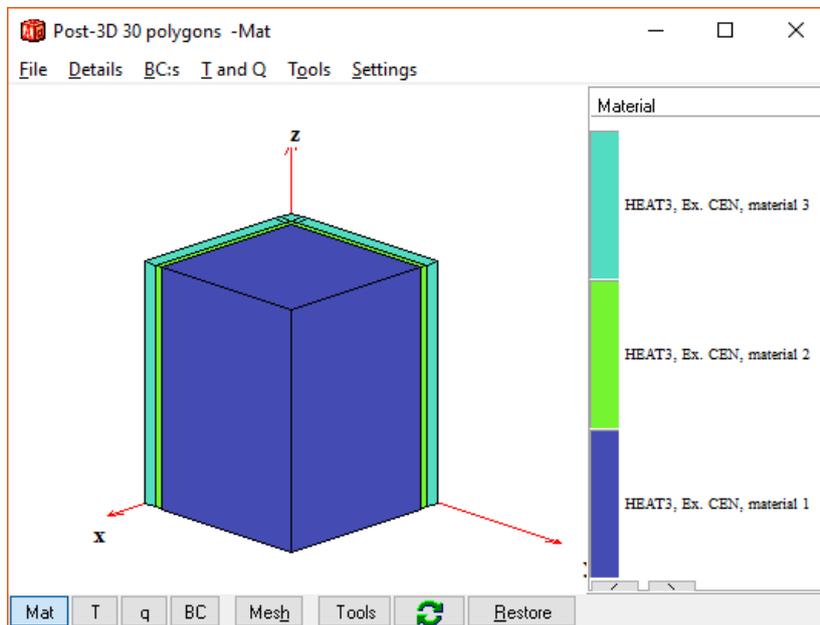
<i>p</i>	0	0.6	0	1.3	1.9	2.15	HEAT3, Ex. CEN, material 3 ! material box
<i>!p</i>	0.1	0.7	0	1.3	1.9	2.15	HEAT3, Ex. CEN, material 2 ! material box
<i>!p</i>	0.15	0.75	0	1.3	1.9	2.15	HEAT3, Ex. CEN, material 1 ! material box
<i>!p</i>	0.3	0.9	1.15	1.3	1.9	1.2	HEAT3, Ex. CEN, material 5 ! material box
<i>!p</i>	0.15	0	1	1.3	1.9	1.15	HEAT3, Ex. CEN, material 4 ! material box
<i>!e</i>	0.3	0.9	1.2	1.3	1.9	2.15	! empty box
<i>!e</i>	0.3	0.9	0	1.3	1.9	1	! empty box
<i>!b</i>	0	0.6	-0.08	1.34	1.97	2.22	2 ! BC box
<i>!b</i>	0.3	0.9	1.2	1.3	1.9	2.15	4 ! BC box
<i>!b</i>	0.3	0.9	0	1.3	1.9	1	3 ! BC box
<i>!b</i>	0.15	0	1	1.32	0.6	1.15	2 ! BC box

This will give:



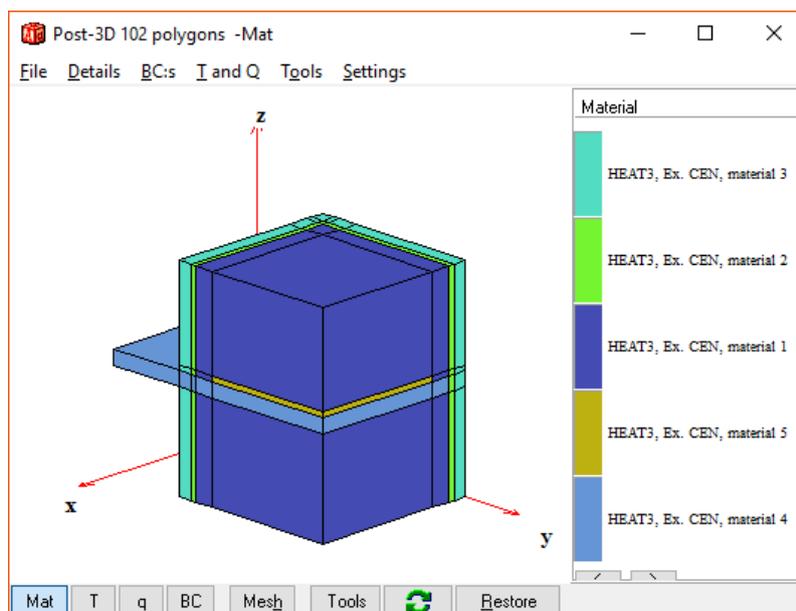
The second row below will replace material 3 with material 2, and so on:

p	0	0.6	0	1.3	1.9	2.15	HEAT3, Ex. CEN, material 3 ! material box
p	0.1	0.7	0	1.3	1.9	2.15	HEAT3, Ex. CEN, material 2 ! material box
p	0.15	0.75	0	1.3	1.9	2.15	HEAT3, Ex. CEN, material 1 ! material box



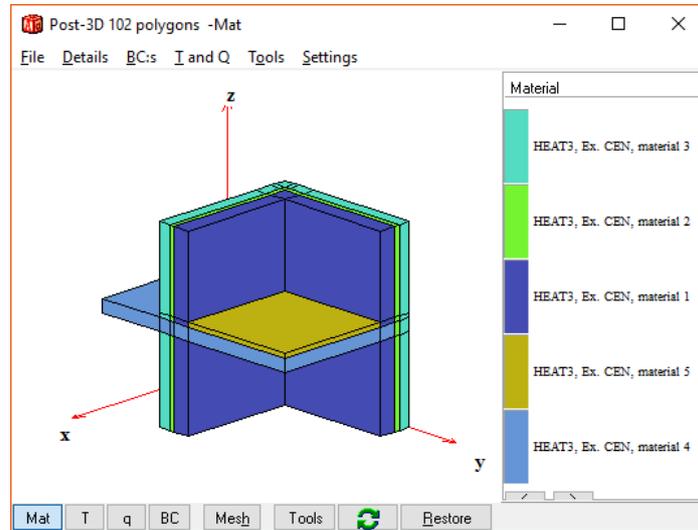
Now, we use the first 5 rows:

p	0	0.6	0	1.3	1.9	2.15	HEAT3, Ex. CEN, material 3 !material box
p	0.1	0.7	0	1.3	1.9	2.15	HEAT3, Ex. CEN, material 2 ! material box
p	0.15	0.75	0	1.3	1.9	2.15	HEAT3, Ex. CEN, material 1 ! material box
p	0.3	0.9	1.15	1.3	1.9	1.2	HEAT3, Ex. CEN, material 5 ! material box
p	0.15	0	1	1.3	1.9	1.15	HEAT3, Ex. CEN, material 4 ! material box

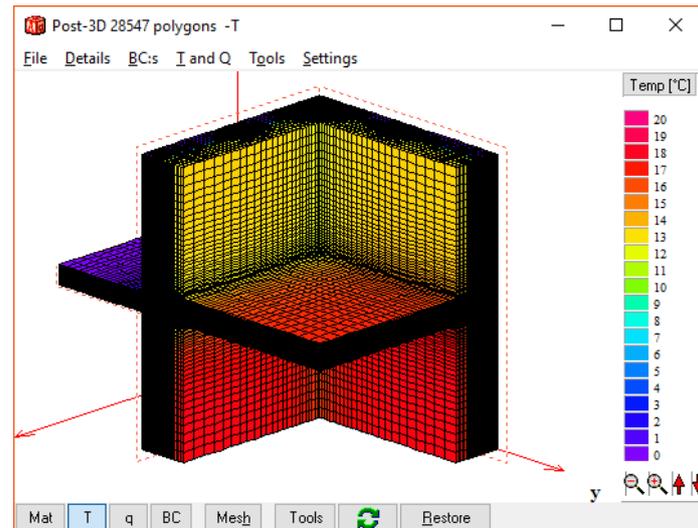


Now, we add the empty boxes that will cut out material:

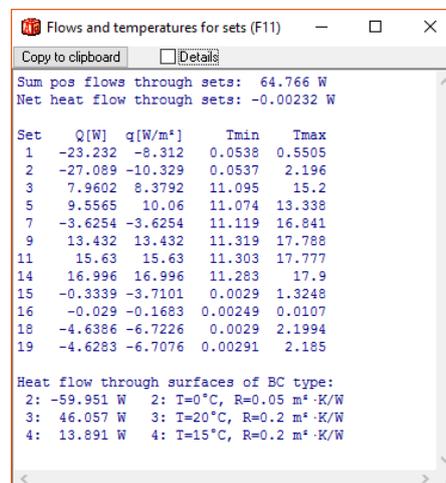
<i>e</i>	0.3	0.9	1.2	1.3	1.9	2.15	<i>! empty box</i>
<i>e</i>	0.3	0.9	0	1.3	1.9	1	<i>! empty box</i>



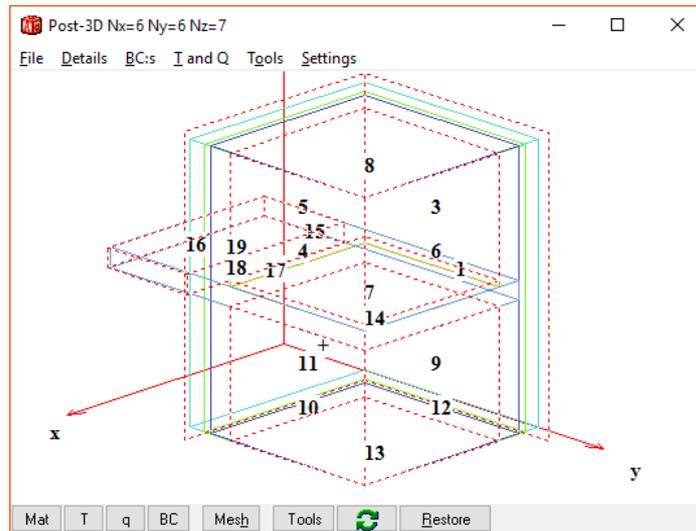
Change the row for the mesh back to “y 100” and run the complete script with defined boundary conditions. Press F9 to solve for steady-state:



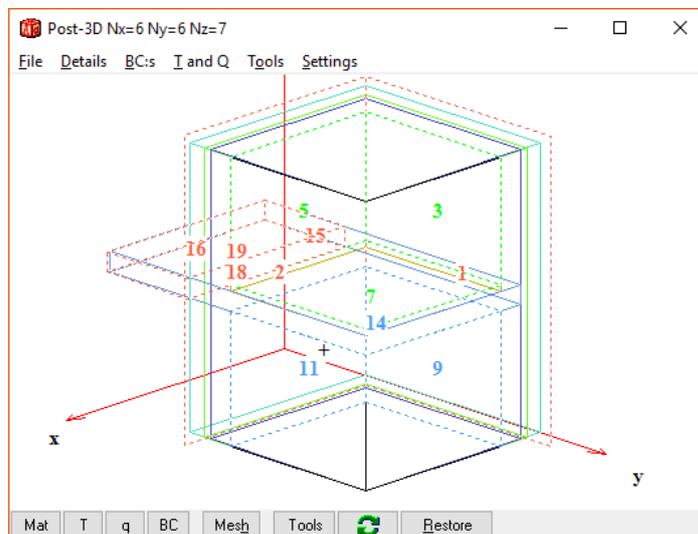
Press F11 to view calculated heat flows:



The “set”-number in the above window can be viewed by clicking menu item BC:s/Sets in the post-processor, see below.



The “set” numbers are the same as “surface” numbers. The visible difference is that set-numbers shows all sides of the BC-boxes, whereas surface-numbers only shows the sides used (connected to a material surface). The surface numbers will also be shown using the chosen color in boundary condition types (F6). See menu items BC:s/Sets and BC:s/Surfaces:



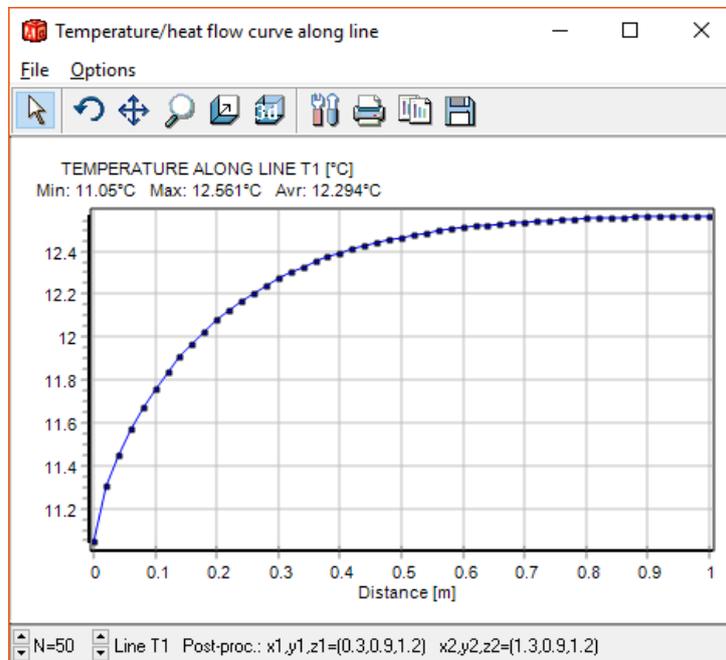
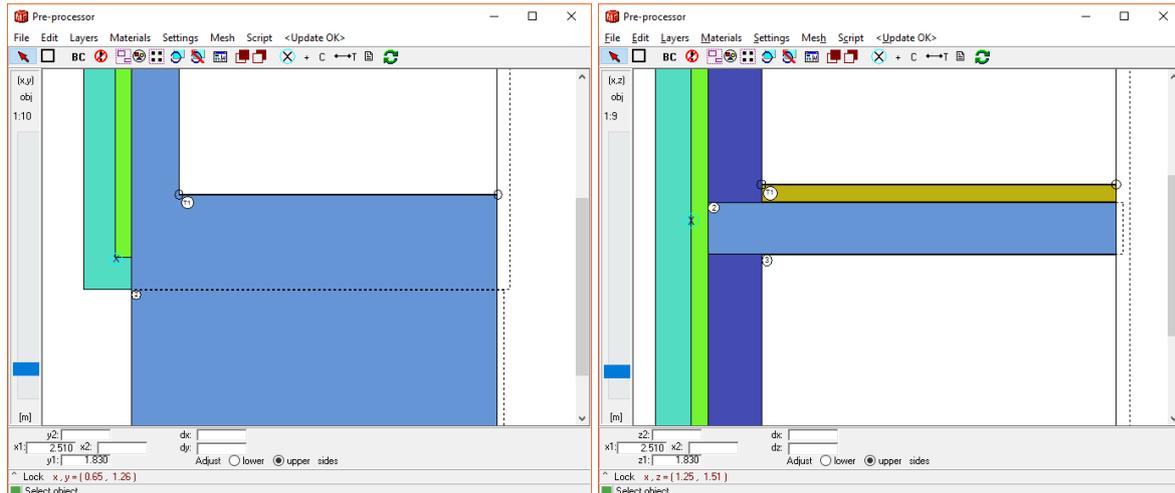
Boundary conditions (F6)

Number of types: 4

n	type	q [W/m²]	T [°C]	R [m²·K/W]
1	Q=const	0.00000		
2	T=const		0.00000	0.050000
3	T=const		20.00000	0.200000
4	T=const		15.00000	0.200000

Close Update

There is one temperature line defined (command "d" in script file) that lies on one edge wall/floor:



The lowest floor temperature is 11 °C at the corner.

2.7 Example Slab

The following example concerns a slab on the ground, see Figure 2.1. The simplification as shown in Figure 2.2 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 shows the material properties. The internal and external surface resistances are 0.13 m²·K/W and 0.04 m²·K/W, respectively.

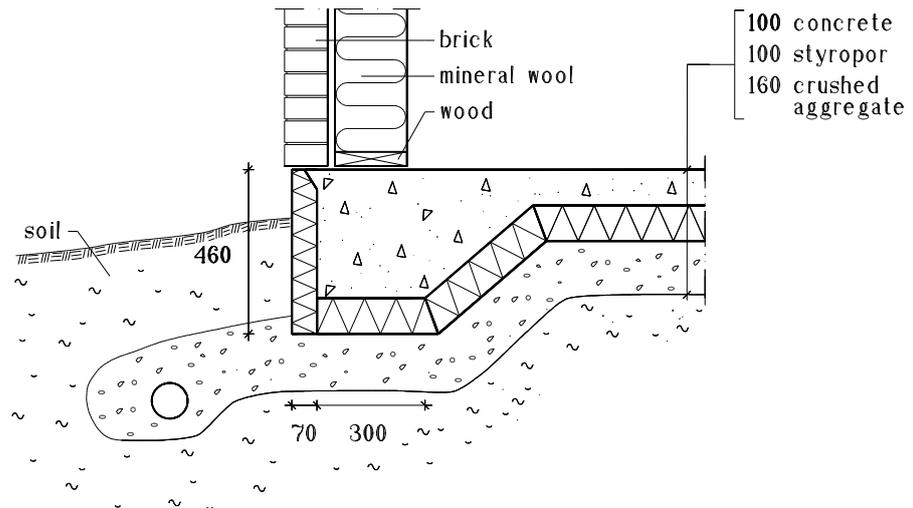


Figure 2.1: Slab on the ground.

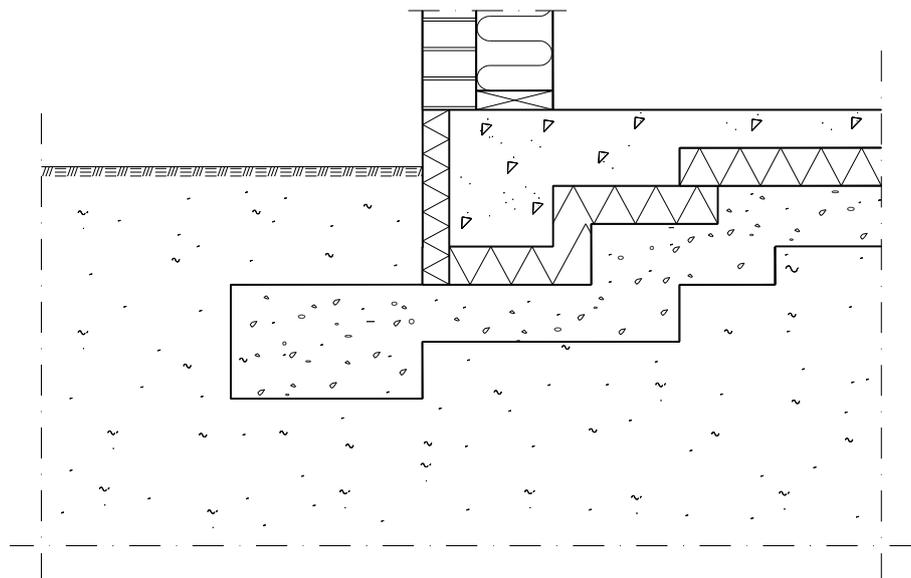


Figure 2.2: Simplified geometry for the calculations.

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 1: 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 2.3 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).

Each input parallelepiped in Figure 2.3 has a number in a circle referring to a parallelepiped in the script text, see the next section. Two numbers together means that there are two boxes extending in the x- and y-directions, respectively. Note that for overlapping boxes, the last mentioned prevails. Consider as an example box 1, 2, and 3. Box 1 (soil) has the coordinates (0,0,0,10,10,7). The overlapping box 2 (crushed aggregate) has the coordinates (1,1,1,10,10,3). Box 3 (soil) (7,7,2,10,10,3) will overlap box 2, etc. Box 6 and 7 have the coordinates (6,6,3,9,10,5) in the y-direction, and (6,6,3,10,9,5) in the x-direction (note the symmetry). This provides a rather compact way to describe the geometry.

The input in a complicated case as this one is quite tricky, but it is fairly simple to graphically check that the input is correct.

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 would be used instead of 20 m, the results would differ with less than 0.1% for this particular case.

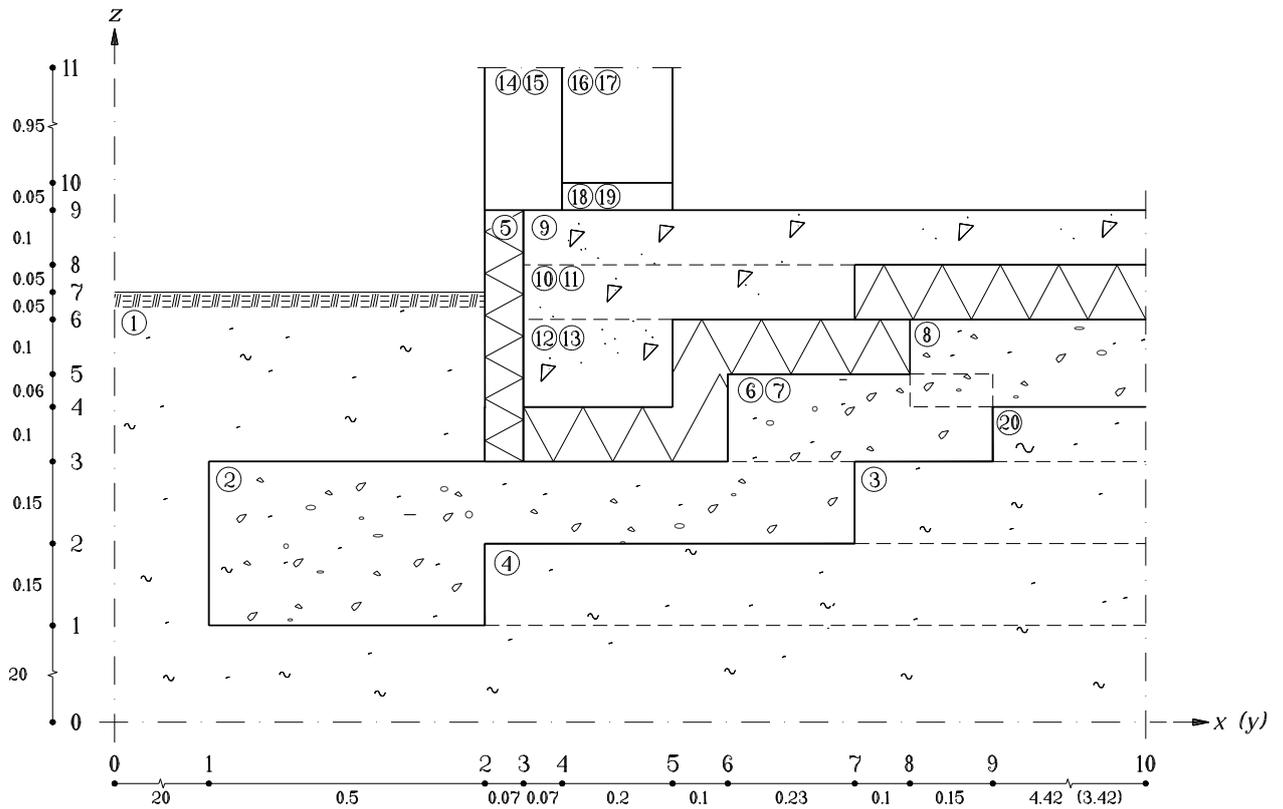


Figure 2.3: Input mesh coordinates and length of input mesh segments are shown. Each parallelepiped has a number in a circle referring to a parallelepiped in the script text.

The script input is made using real lengths. The table below shows lengths to the input mesh coordinate in each direction. As an example, box 1 (soil) with the coordinates (0,0,0,10,10,7) would be given as:

p 0 0 0 25.84 24.84 20.61 soil !circle 1

Coordinate	x	y	z
0	0	0	0
1	20	20	20
2	20.5	20.5	20.15
3	20.57	20.57	20.3
4	20.64	20.64	20.4
5	20.84	20.84	20.46
6	20.94	20.94	20.56
7	21.17	21.17	20.61
8	21.27	21.27	20.66
9	21.42	21.42	20.76
10	25.84	24.84	20.81
11			21.76

Press menu item Examples/Cen example to get the complete script. Press “Run” and then F9 to solve for steady-state.

```

Script for Pre-processor
Edit  Open script file  Save script file  Help  Examples

Run [ script => pre-processor ]  Import [ pre-processor => script ]

 Empty pre-processor before running script   Set origo to lower left in pre-processor

! Slab example. Press "Run script".

%loadmaterialfile=Slab_example.MTL
y      120
a      2      20      0.13
a      3      9.4      0.04
x      20.84  20.84  20.76

! max number of cells in each direction
! BC type 2 has T=20 and R=0.13
! BC type 3 has T=0 and R=0.04
! expansion point set at P(x,y,z)

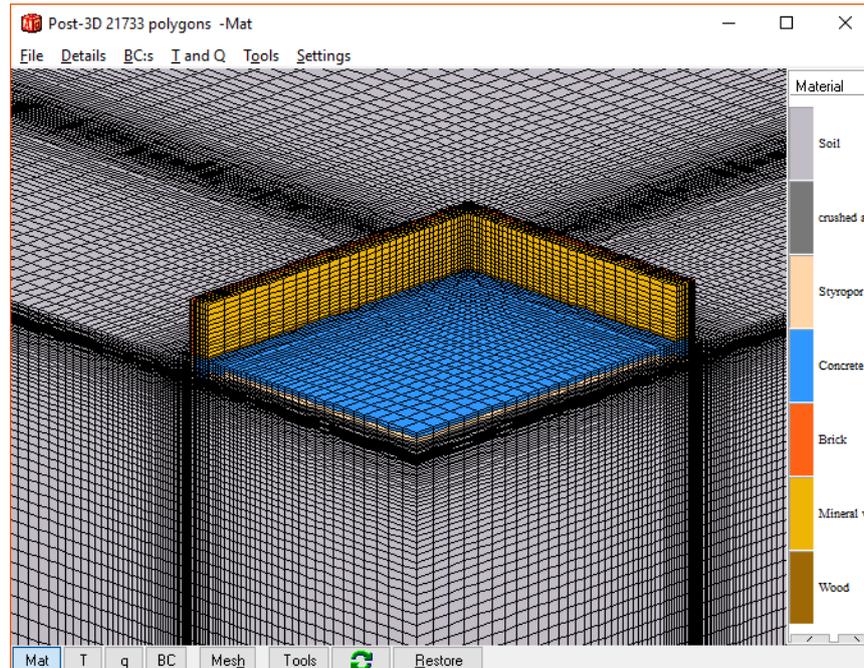
! "circle" below refers to figure in manual

!
! x1      y1      z1      x2      y2      z2
p 0 0 0 25.84 24.84 20.61 soil !circle 1
p 20 20 20 25.84 24.84 20.3 crushed aggregate !circle 2
p 21.17 21.17 20.15 25.84 24.84 20.3 soil !circle 3
p 20.5 20.5 20 25.84 24.84 20.15 soil !circle 4
p 20.5 20.5 20.3 25.84 24.84 20.76 styropor !circle 5
p 20.94 20.94 20.3 21.42 24.84 20.46 crushed aggregate !circle 6
p 20.94 20.94 20.3 25.84 21.42 20.46 crushed aggregate !circle 7
p 21.27 21.27 20.4 25.84 24.84 20.56 crushed aggregate !circle 8
p 20.57 20.57 20.66 25.84 24.84 20.76 concrete !circle 9
p 20.57 20.57 20.56 21.17 24.84 20.66 concrete !circle 10
p 20.57 20.57 20.56 25.84 21.17 20.66 concrete !circle 11
p 20.57 20.57 20.4 20.84 24.84 20.56 concrete !circle 12
p 20.57 20.57 20.4 25.84 20.84 20.56 concrete !circle 13
p 20.5 20.5 20.76 20.64 24.84 21.76 brick !circle 14
p 20.5 20.5 20.76 25.84 20.64 21.76 brick !circle 15
p 20.64 20.64 20.81 20.84 24.84 21.76 mineral wool !circle 16
p 20.64 20.64 20.81 25.84 20.84 21.76 mineral wool !circle 17
p 20.64 20.64 20.76 20.84 24.84 20.81 wood !circle 18
p 20.64 20.64 20.76 25.84 20.84 20.81 wood !circle 19
p 21.42 21.42 20.3 25.84 24.84 20.4 soil !circle 20
b 20.84 20.84 20.76 25.84 24.84 21.76 2 IBC type number 2 (T=20 R=0.13)
b 20.5 20.5 20.61 26 25 22 3 IBC type number 3 (T=9.4 R=0.04)
b 0 0 20.61 26 25 21 3 IBC type number 3 (T=9.4 R=0.04)

 Show valid command rows   Show invalid command rows

Number of valid rows processed: 27

```



2.8 Results

Figure 2.4 shows the heat flow through the floor (set or surface number 5 as shown below), through the walls (sets 1, 3, 7, and 8), to the ground (set 12), and the minimum and maximum surface temperatures.

The heat loss to the ground is the same as the heat that flows up from the ground (set number 12). The heat flow to the ground from the house during one year then becomes $61.91 \cdot 4 \cdot 24 \cdot 365 = 2.2$ MWh. (Note that if the heating period is part of the year, and we need to know the heating demand for the ground loss, we can make a transient calculation by using e.g. a sinusoidal external temperature variation or step-wise temperature function and then record the heat losses for the heating period taking the heat capacities of the ground and other materials into account. Note that correct values for the volumetric heat needs to be given in that case since they are set to “1” because they do not matter for the steady-state calculation).

Note that the calculated heat flows represent one quarter of the house, so we need to multiply all heat flows by 4.

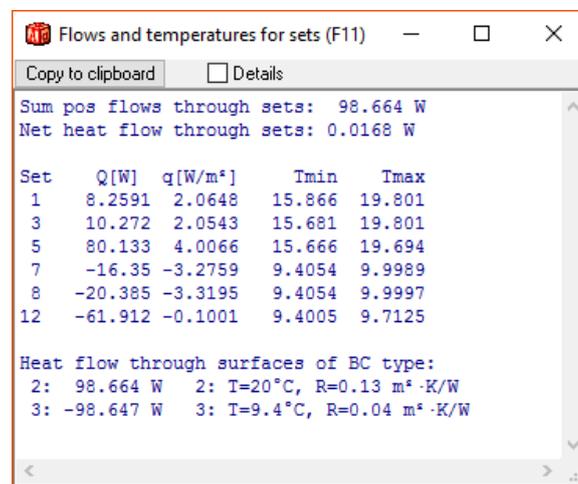
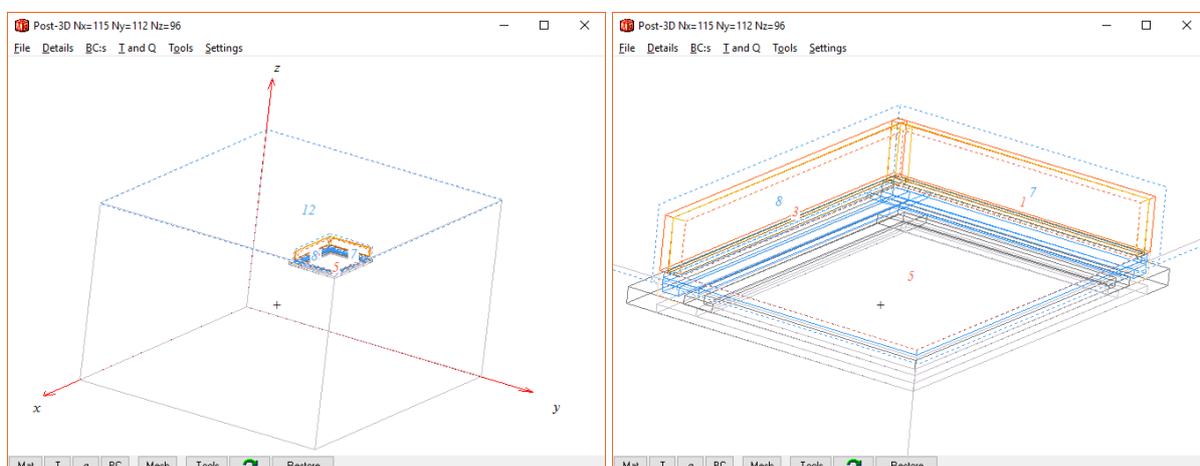


Figure 2.4: Calculated heat flows and surface temperatures.

The set numbers are shown below (1, 3 are internal side of walls, 7, 8 are external side of walls, 5 is floor, 12 is the ground surface). The “set” numbers are the same as “surface” numbers. The visible difference is that set-numbers shows all sides of the BC-boxes, whereas surface-numbers only shows the sides used (connected to a material surface). The surface numbers will also be shown using the chosen color in boundary condition types (F6). See menu items BC:s/Sets and BC:s/Surfaces.



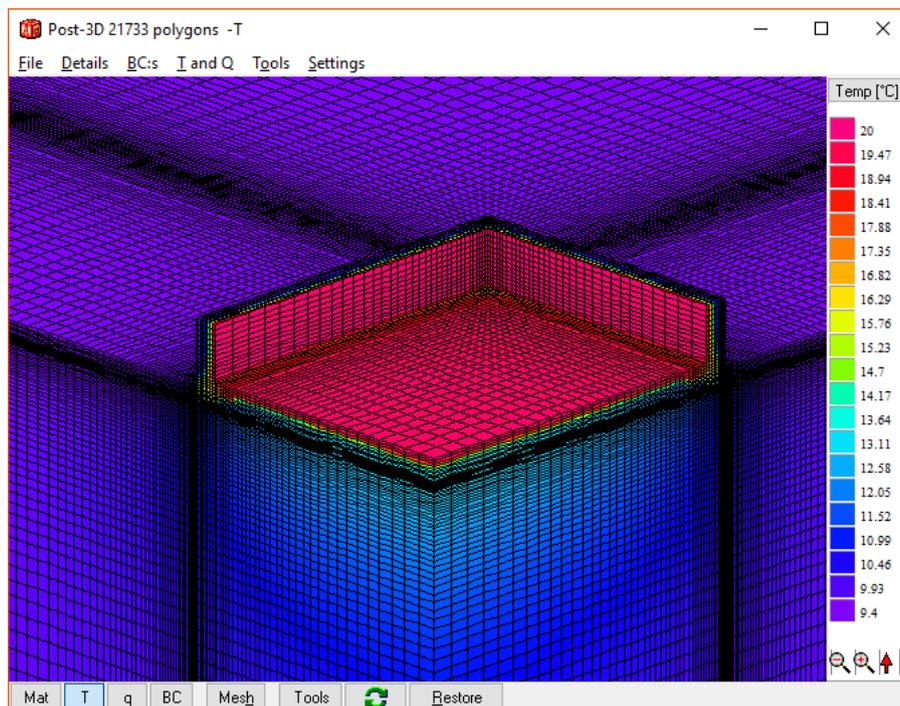
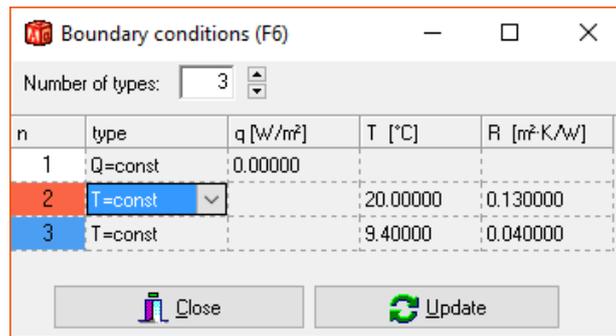


Figure 2.5: Surface temperatures and the mesh around the house.

We can make the soil and the crushed aggregate invisible by adding “%hide” in the rows (4 rows for the soil and 4 rows for the crushed aggregate) as below

```
p    0    0    0    25.84  24.84  20.61  soil    %hide          !circle 1
p    20   20   20   25.84  24.84  20.3   crushed aggregate  %hide !circle 2
etc
```

Running the script again would require a new calculation since the temperature field will be initialized unless “Always reset temperatures when input changes” is unchecked in the Solve menu item.

2.9 Estimation of numerical error

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, 2007). 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 1%. Further mesh division is required if the difference is larger.

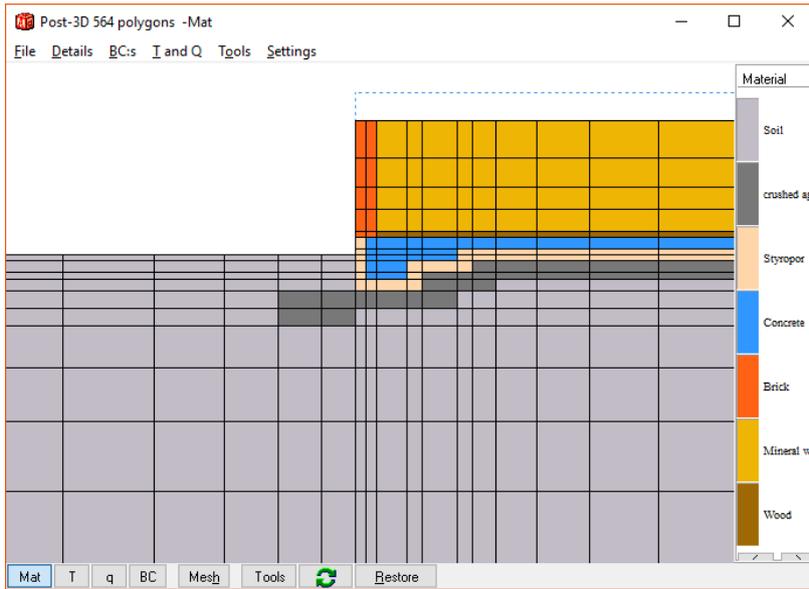
The table below shows calculated heat flows and how long the simulation took (cpu-time, on an Intel i7-3770K CPU @ 3.50GHz, 8 logical cores) for some different meshes.

Max cells in each direction (script command “y”)	Total number of cells, Ntot	Heat flow to ground (W)	Total heat loss (W)	cpu-time
20	6000	60.27	93.87	<1 sec.
28	12000	60.70	94.34	<1 sec.
40	29000	61.10	96.14	<1 sec.
60	88000	61.43	97.42	2 sec.
80	227000	61.72	97.76	5 sec.
100	453000	61.71	97.78	18 sec.
120	789000	61.91	98.66	35 sec.
140	1.25 million	62.08	99.21	53 sec.
370	32 million	62.21	99.32	35 min.

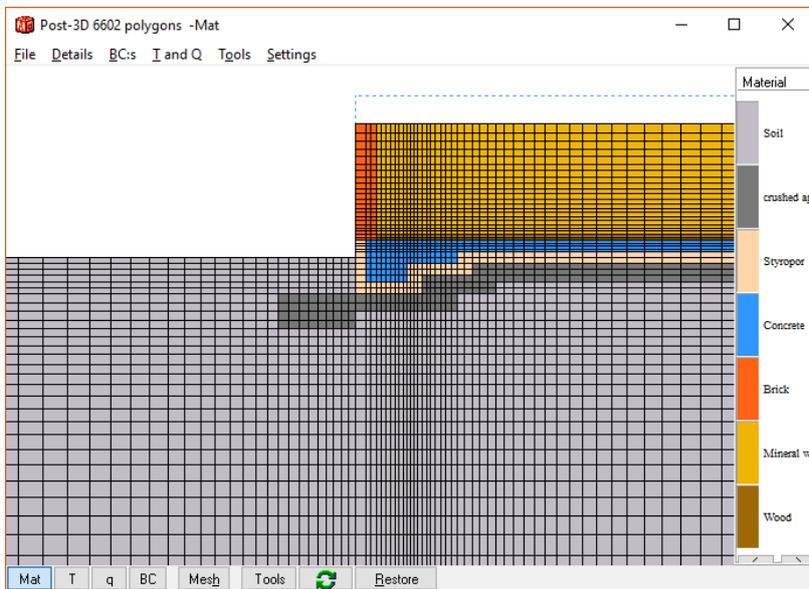
The case with 789000 nodes (“y 120”) takes 16 seconds on a Intel I7-6900K @3.20 GHz (16 logical cores).

Using the above results for 6000 and 12000 cells the ratio for the CEN criteria is $1-93.87/94.34=0.5\%$ (<1%) which would mean that about 12000 cells would be enough. However, we see that this solution for 12000 cells will be at least 5% ($1-94.34/99.32=0.05$) off the correct value. It is clear that this criterion should be used with caution. It is recommended to use about 800,000 nodes for this case. The results should then be about 1% off from the “true” values ($1-98.66/99.32=0.007$ if we assume that 32 million nodes gives more or less a correct value). The numerical mesh around the edges are shown below for some cases (press ”1” in the post-processor to get the y,z-plane view, press “0” to get back to 3D-view, see Tools/Help shortcuts).

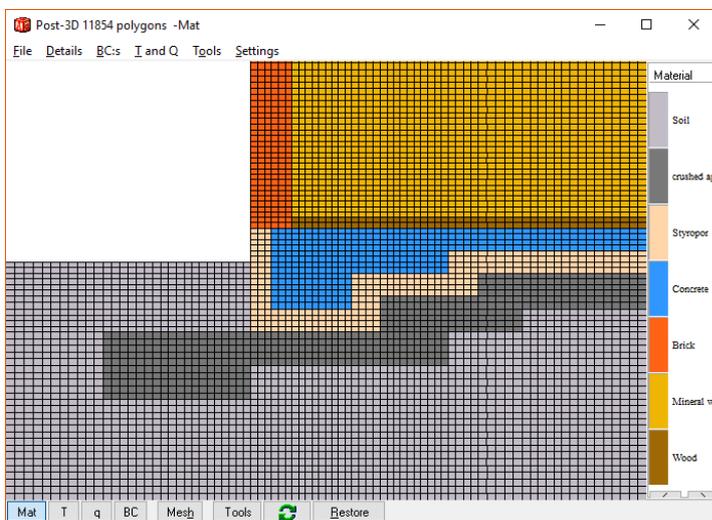
Another drawback with the CEN criterion is that the user can put a dense grid where it is not needed and still obtain conformance even though it is a badly chosen mesh. The smaller cells should be concentrated where the temperature gradients are high (see option “q” in the post-processor).



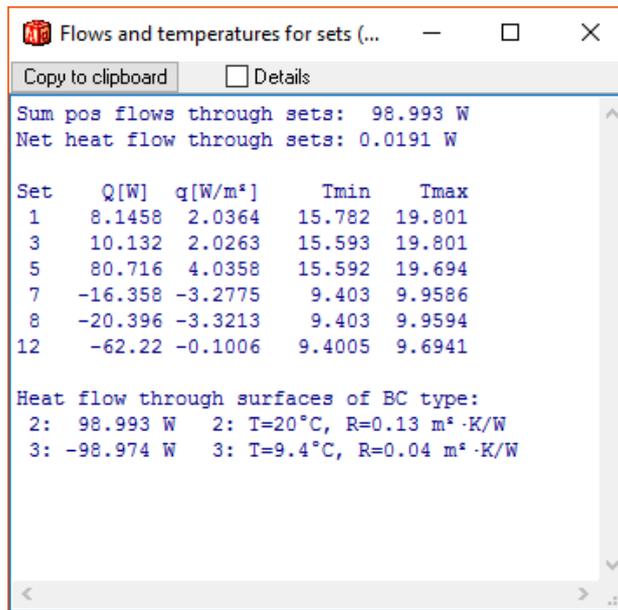
$N_x=N_y=N_z=28$, $N_{tot}=12000$



$N_x=N_y=N_z=120$, $N_{tot}=789000$



$N_x=N_y=N_z=370$, $N_{tot}=32$ million



N=32 million nodes (y=370), cpu-time 37 min.

2.10 Improving the mesh (“advanced users”)

The mesh can be “fine-tuned” using the text editor (menu item Editor/Show editor). For more info about the text editor, see chapter 5.1.2 in http://www.buildingphysics.com/manuals/HEAT3_4.pdf

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

Open the text editor (you might need to close the pre-processor and other windows in order to see the text). The complete text input for the case with 780000 cells is shown below. This is the actual data that describes the whole problem (geometry, BC:s, etc) that is used in the calculation and post-processor view. It is possible to change all data.

```
11
49 9 1 2 7 4 7 2 3 30 1  Nx=115
0.9434 0.9434 0.9434 0.9434 0.9434 1.06 1.06 1.06 1.06 1.06 1
20 0.5 0.07 0.07 0.2 0.1 0.23 0.1 0.15 4.42 0.16  Lx=25.84

11
49 9 1 2 7 4 7 2 3 26 1  Ny=111
0.9434 0.9434 0.9434 0.9434 0.9434 1.06 1.06 1.06 1.06 1.06 1
20 0.5 0.07 0.07 0.2 0.1 0.23 0.1 0.15 3.42 0.16  Ly=24.84

13
50 2 2 2 1 2 1 1 4 2 6 14 1  Nz=88
0.9434 0.9434 0.9434 0.9434 0.9434 0.9434 0.9434 0.9434 1.06 1.06 1.06 1
20 0.15 0.15 0.1 0.06 0.1 0.05 0.05 0.1 0.05 0.19 0.76 0.24  Lz=21.76

20 material boxes

0 0 0 10 10 7 2.2 1E006 0 1 Soil
1 1 1 10 10 3 2 1E006 0 2 crushed aggregate
7 7 2 10 10 3 2.2 1E006 0 3 Soil
```

2 2 1 10 10 2 2.2 1E006 0 4 Soil
2 2 3 10 10 9 0.039 1E006 0 5 Styropor
6 6 3 9 10 5 2 1E006 0 6 crushed aggregate
6 6 3 10 9 5 2 1E006 0 7 crushed aggregate
8 8 4 10 10 6 2 1E006 0 8 crushed aggregate
3 3 8 10 10 9 1.7 1E006 0 9 Concrete
3 3 6 7 10 8 1.7 1E006 0 10 Concrete
3 3 6 10 7 8 1.7 1E006 0 11 Concrete
3 3 4 5 10 6 1.7 1E006 0 12 Concrete
3 3 4 10 5 6 1.7 1E006 0 13 Concrete
2 2 9 4 10 12 0.44 1E006 0 14 Brick
2 2 9 10 4 12 0.44 1E006 0 15 Brick
4 4 10 5 10 12 0.036 1E006 0 16 Mineral wool
4 4 10 10 5 12 0.036 1E006 0 17 Mineral wool
4 4 9 5 10 10 0.14 1E006 0 18 Wood
4 4 9 10 5 10 0.14 1E006 0 19 Wood
9 9 3 10 10 4 2.2 1E006 0 20 Soil

0 empty boxes

3 BC types

0 0
1 20 0.13
1 9.4 0.04

13 BC rect

5 5 9 5 10 12 2
10 5 9 10 10 12 2
5 5 9 10 5 12 2
5 10 9 10 10 12 2
5 5 9 10 10 9 2
5 5 12 10 10 12 2
2 2 7 2 11 13 3
2 2 7 11 2 13 3
2 2 7 11 11 7 3
0 0 7 0 11 11 3
0 0 7 11 0 11 3
0 0 7 11 11 7 3
0 0 11 11 11 11 3

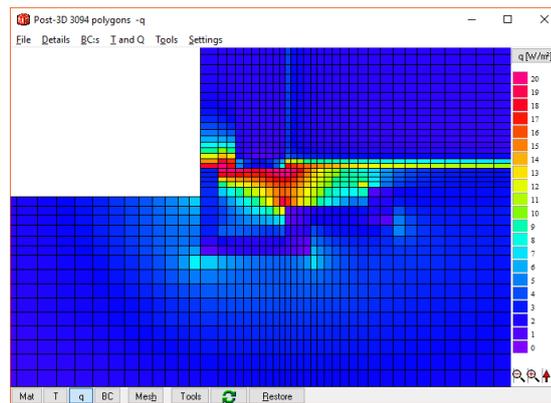
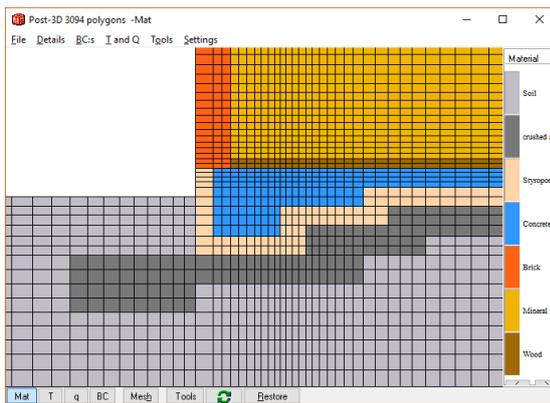
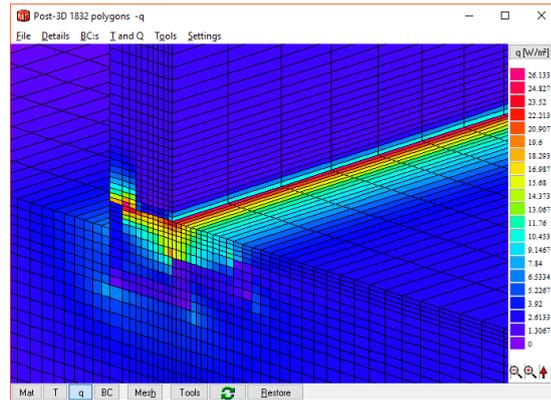
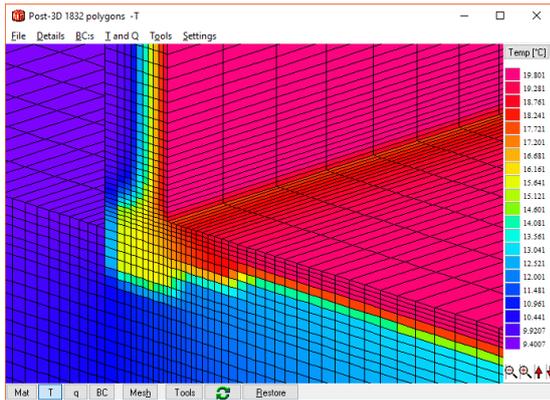
0 Res planes

13 Sets...

1 1
1 2
1 3
1 4
1 5
1 6
1 7
1 8
1 9
1 10
1 11
1 12
1 13

0 Heat sources

Let us now look at the automatically generated mesh using the expansion point. Calculated temperatures and heat flows are shown below.



(x,z-plane)

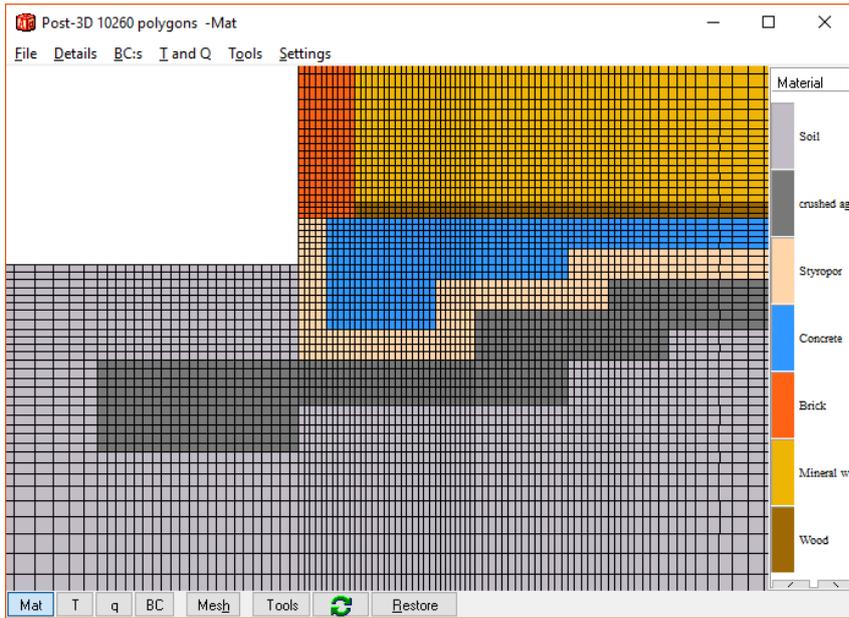
Change the rows above for mesh in x- , y- , and z- directions (number of cells and expansion coefficients for each input mesh segment as shown in figure 3 above) to these rows:

```
11
40 22 5 5 14 8 14 6 8 30 1
0.9 1 1 1 1 1 1 1 1.06 1.1 1
20 0.5 0.07 0.07 0.2 0.1 0.23 0.1 0.15 4.42 0.16
```

```
11
40 22 5 5 14 8 14 6 8 30 1
0.9 1 1 1 1 1 1 1 1.06 1.1 1
20 0.5 0.07 0.07 0.2 0.1 0.23 0.1 0.15 3.42 0.16
```

```
13
40 5 5 4 2 4 2 2 5 3 8 15 1
0.9 1 1 1 1 1 1 1 1 1 1 1 1
20 0.15 0.15 0.1 0.06 0.1 0.05 0.05 0.1 0.05 0.19 0.76 0.24
```

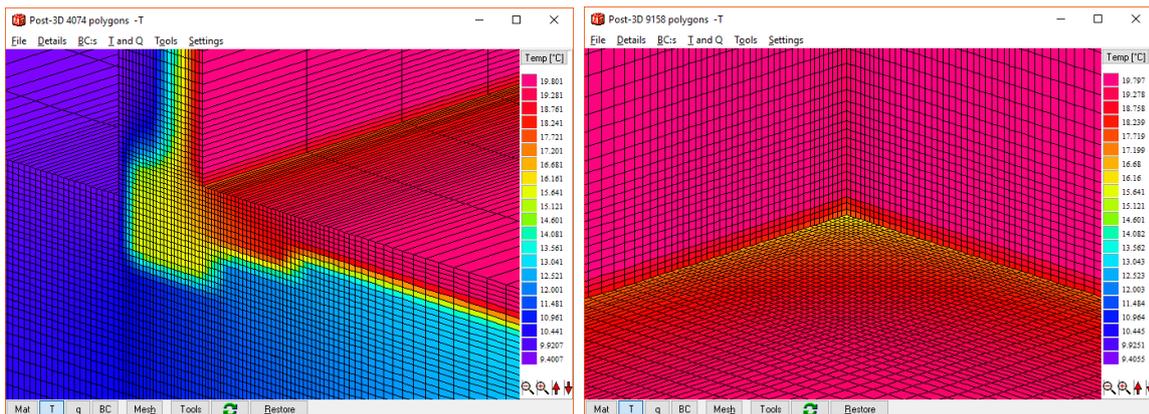
This will give the following mesh:

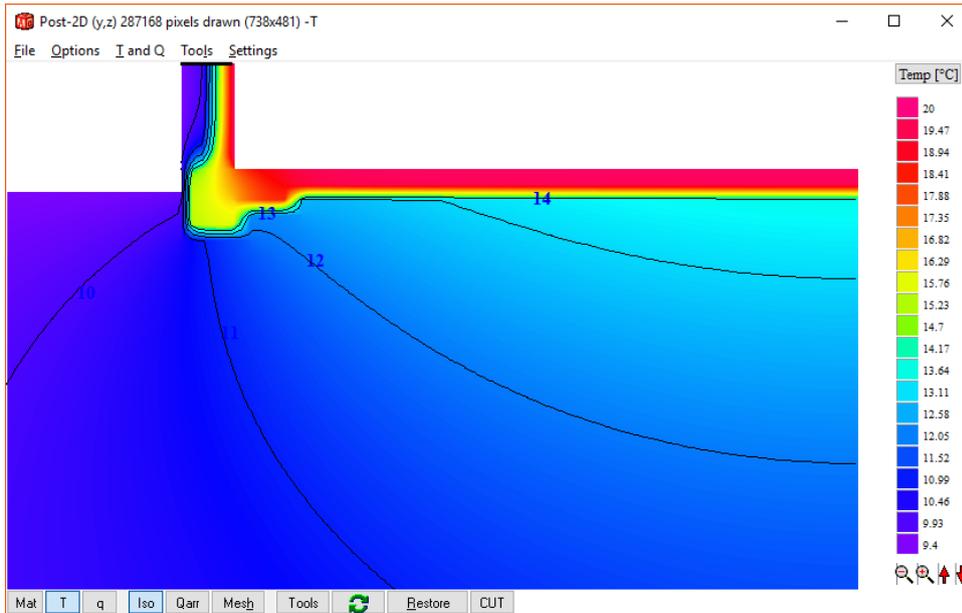


The calculation results are shown below. This mesh gives more accurate heat flows compared to the case above when 32 million nodes was used. The cpu-time needed is also much lower (2 minutes compared to 32 minutes).

	Total number of cells, Ntot	Heat flow to ground (W)	Total heat loss (W)	Cpu-time
	1.59 million	62.22	99.59	106 sec.

This case takes 61 seconds on a Intel I7-6900K @3.20 GHz (16 logical cores).





Temperatures and isotherms displayed by “Post-2D” window. Cut at center of slab.

2.11 Save/open script files

The scrips can be saved and opened to/from text files. A suggestion is to put the files in folder
“...Documents\Blocon\HEAT3 8\Scripts_preprocessor”

3. A few tips

3.1 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.

3.2 Numerical aspects

3.2.1 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, 2007). 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 1%. Further mesh division is required if the difference is larger.

3.2.2 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.

3.2.3 Steady-state calculations

The following criterion for when to stop a calculation for steady-state is recommended as a European standard (CEN, 2007). 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 0.0001.

3.2.4 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.

3.2.5 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.

3.2.6 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.

4. Miscellaneous

4.1 Adapted for Windows 7 and 8

HEAT3 is now fully adapted for Windows 7, 8, and 10. HEAT3_v8.exe is placed in folder %ProgramFiles(x86) during installation.

The startup folder is "My Documents\Blocon\HEAT3 8". This means that the path should be "C:\Users\[user name] \Blocon\HEAT3 8". The materials files (*.mtl) should be placed in this folder. This set up is similar for HEAT2 v10.

The program should normally be started using the short-cut on the start menu. To start HEAT3, press the windows key and write "HEAT3 v8". The icon should be displayed. Click this.

Do as follows if you want to create a short-cut on the desktop:

1. Right-click the desktop, choose New/Short-cut. Browse to "C:\Program Files (x86)\BLOCON\HEAT3_v8" and add "HEAT3_v8.exe".
2. Right-click on the icon ("HEAT3_v8.exe".) created on the desktopn and choose properties.

Make sure that "target" is set to
"C:\Program Files (x86)\BLOCON\HEAT3_v8\HEAT3_v8.exe"
(or wherever the file is located)

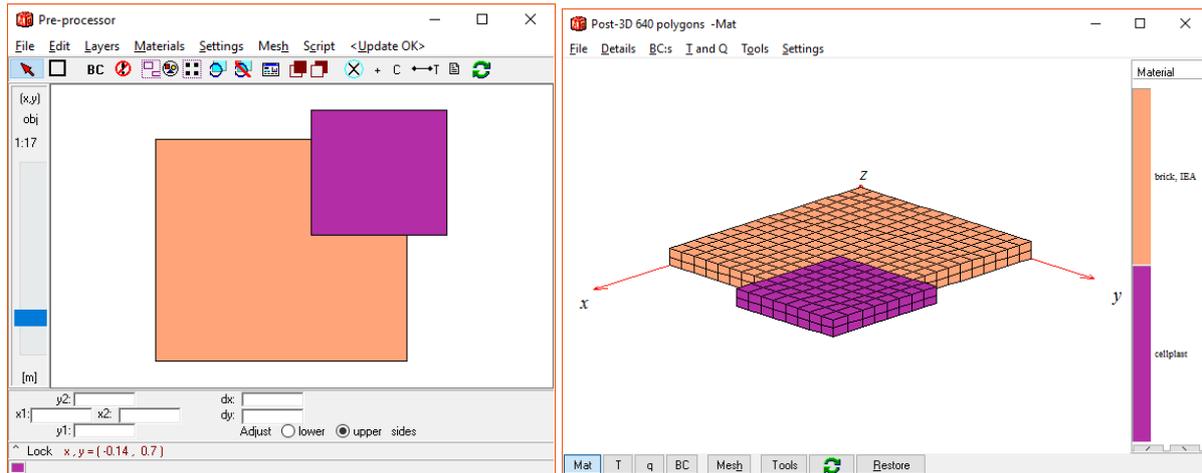
and "Start in" is set to
"...My Documents\Blocon\HEAT3 8" (or wherever the files are located)

4.2 Improvements for “Text editor”

The material name in the “Text editor” (not to be confused with the script editor) will now use the color defined in the material list if the name can be found there. If the name is not found, the color will be defined as usual by an integer, see chapter 7.7 in http://www.buildingphysics.com/manuals/HEAT3_4.pdf

Example

Assume we have drawn one “brick, IEA” (orange) and one “cellplast” (purple) box in the pre-processor:



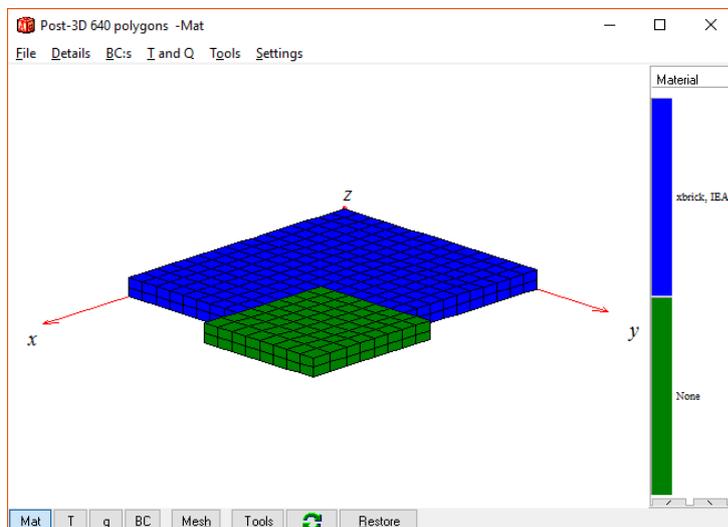
Now, go to menu item “Editor/Show editor”. The following lines can be found

```
0 0 0 2 2 1 0.45 1.49E006 0 1 brick, IEA
1 1 0 3 3 1 0.04 1E006 0 2 cellplast
```

If these are replaced with

```
0 0 0 2 2 1 0.45 1.49E006 0 1 xbrick, IEA
1 1 0 3 3 1 0.04 1E006 0 2
```

The result will be one blue box (“1”) and one green box (“2”):



Changing the color number to “0” will make the box invisible in the post-processor (however still accounted for in the calculations):

```
0 0 0 2 2 1 0.45 1.49E006 0 1 xbrick, IEA  
1 1 0 3 3 1 0.04 1E006 0 0
```

