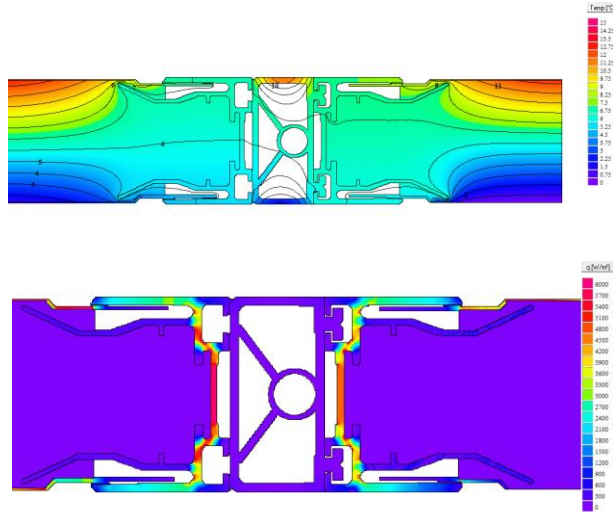
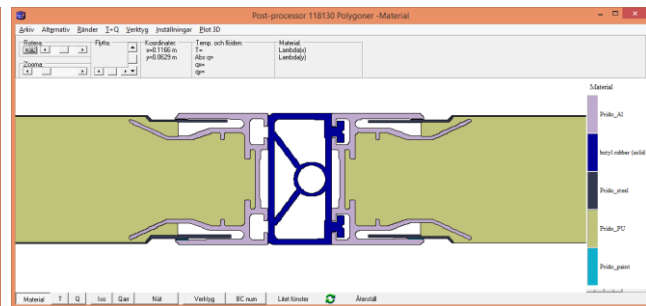
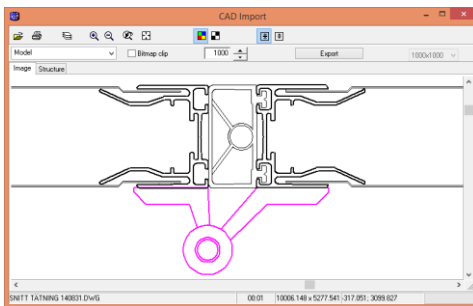
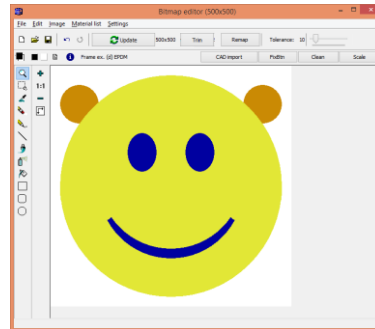
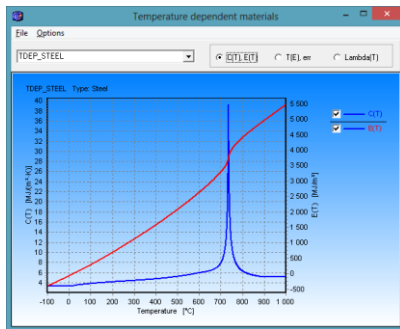


HEAT2 version 9

A PC-program for heat transfer in two dimensions - Update manual



December 19, 2014
Revised Oct 20, 2015

BLOCON

www.buildingphysics.com

Contents

1. WHAT'S NEW IN HEAT2 VERSION 9.....	5
1.1 TEMPERATURE DEPENDENT MATERIAL PROPERTIES	5
1.2 IMPORT OF CAD-FILES	5
1.3 INCREASED SIZES FOR DATA FIELDS	5
1.4 MISCELLANEOUS	6
1.5 LIGHT VERSION.....	6
1.6 INSTALLATION AND REQUIREMENTS	7
2. TEMPERATURE DEPENDENT MATERIAL PROPERTIES	9
2.1 INTRODUCTION	9
2.2 ENTHALPY CALCULATIONS.....	9
2.2.1 <i>Freezing of soil</i>	10
2.3 USING TEMPERATURE DEPENDENT MATERIALS.....	11
2.4 HOW TO DEFINE TEMPERATURE DEPENDENT MATERIALS	11
2.4.1 <i>Type 1, Freezing of soil</i>	11
2.4.2 <i>Type 2, piecewise linear volumetric heat capacity and thermal conductivity</i> ...	13
2.4.3 <i>The material list</i>	20
2.4.4 <i>Option for overriding and using constant properties</i>	20
2.5 UPDATE OF CONDUCTANCES, HEAT CAPACITIES AND STABLE TIME-STEP	21
2.6 MISCELLANEOUS ABOUT TEMPERATURE DEPENDENT MATERIAL PROPERTIES	21
2.7 SNOW LAYER AND COLD SPELL	22
2.8 EXAMPLE	23
3. ENHANCEMENTS IN THE BITMAP EDITOR	33
3.1 INTRODUCTION	33
3.2 SETTING THE SCALE OF THE GEOMETRY	34
3.3 BITMAP RECTANGLES AND BITMAP UNDERLAYS	40
3.4 DRAWING GEOMETRY USING A SCRIPT	44
3.5 TIPS.....	49
4. IMPORT OF CAD-FILES.....	55
4.1 INTRODUCTION	55
4.2 INTRODUCTION	55
4.3 EXAMPLE	59
5. MISCELLANEOUS	79
5.1 SEVERAL INSTANCES OF HEAT2 ARE NOW ALLOWED	79
5.2 IMPROVED HANDLING OF MODIFICATIONS	79
5.3 IMPROVED HANDLING OF INPUT VIA TEXT.....	79
5.4 MULTILINGUAL OPTION	79
5.4.1 <i>Introduction</i>	79
5.5 SAVING PROJECT FILES TO ZIP-FILE	81
5.6 ENHANCEMENTS IN THE POST-PROCESSOR.....	82
6. REFERENCES	83

1. What's new in HEAT2 version 9

Blocon is proud to present a new version of HEAT2. Many new important features have been added. Up-to-date information is given on www.buildingphysics.com.

This update manual covers new features that have been added since version 8. New users should also read the update manuals for versions 6, 7, 8 and the full manual for version 5 at

http://www.buildingphysics.com/manuals/HEAT2_8_update.pdf

http://www.buildingphysics.com/manuals/HEAT2_7_update_english.pdf

http://www.buildingphysics.com/manuals/Heat2_6_update.pdf

http://www.buildingphysics.com/manuals/HEAT2_5.pdf

Tips for reading for beginners: For a quick start read Chapter 4 (pages 23-27) in [Manual HEAT2 5.0](#). The example in chapter 8 (pages 117-121) would also give a short introduction. After this, look at the update manuals for versions 6, 7, and 8. Also see the examples for the test cases: [ISO 10211 & 10077-2 validation test cases](#)

1.1 Temperature dependent material properties

HEAT2 now handles calculations for temperature dependent material properties (thermal conductivity and heat capacity) and phase change (e.g. applications within fire safety design and soil freezing, etc.). It also handles a variable snow layer and cold spells more convenient. See Chapter 2.

1.2 Import of CAD-files

HEAT2 v9 can import DWG, DXF, SVG, PLT, CGM, WMF, JPG, BMP, PNG, and some other formats. See Chapter 4.

1.3 Increased sizes for data fields

Many data fields have been increased in sizes. The maximum number of numerical nodes has been increased from 4,000,000 (2,000·2,000) to 6,250,000 (2,500·2,500). Other changes are shown below.

Maximum number of	HEAT2 version 9	HEAT2 version 8
Nodes	6,250,000 (2,500·2,500)	4,000,000 (2,000·2,000)
Bitmap size in bitmap editor	2,000x2,000	1,000x1,000
Material boxes	20,000	10,000
Boundaries	10,000	5,000
Number of resistance lines	100	50

1.4 Miscellaneous

Hundreds of minor improvements are made. The most important are shown below:

- Several instances of HEAT2 can now be started and active at the same time. This makes it possible to work on a new project while another one is simulated.
- Check for update is now made automatically.
- The distance from a reference point to the mouse cursor is now shown in the pre-processor. This may be useful for e.g. checking the width of a sloping material.
- The chart for temperatures/heat (menu item Output/Temperature along line boundaries) has been updated. The chart editor is also updated.
- The Solve-window (F9 and F10) now shows the sum of all positive boundary flows (earlier the sum of all absolute boundary flows was shown).
- The pre-processor is now unlocked after every simulation (option “Always unlock pre-processor after simulation” in the settings menu).
- Better quality (metafile) when exporting pre-processor image.
- Bitmap editor: Possibility to set the drawing color to black or white in order to erase areas using different shapes.
- Improved report builder.
- Improved calculation of automatic expansion factor.
- Earlier versions were minimized when focus was set to another application in Windows. This version does not have this behavior.

Enhancements in post-processing:

- Much faster drawing/updating due to direct video buffering (can be turned off in menu item “T and Q” in the post-processor).
- Mouse cursor sometimes “froze” for a small moment in earlier versions. This has been fixed.
- The number of colors (materials) and legends shown in the post the post-processor was limited to 15 in version 8. This is now increased to 30.
- The U-values (1D) in x,y-direction (U_x , U_y) are shown for any cut.

1.5 Light version

The light (demo) version has the following restrictions:

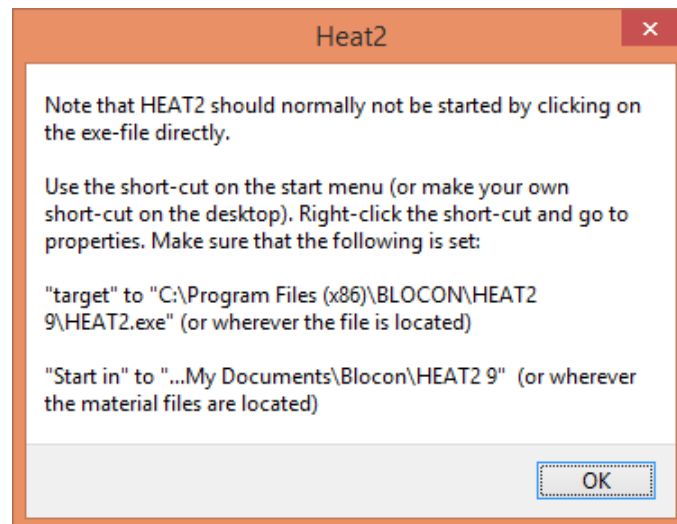
- Data cannot be saved.
- Materials cannot be added or changed.
- Max 1,000,000 (1,000·1,000) nodes.
- Results for heat flows are hidden for larger problems with more than 5,000 nodes and with more than 60 nodes in each direction.
- Thermal coupling coefficients (L2D) will not be calculated.
- Report preview is disabled.
- Some other restrictions apply.

1.6 Installation and requirements

HEAT2 v9 requires only about 0.2 GB RAM (a pc with 2 GB is recommended), and less than 10 MB hard disk space. HEAT2 runs on both 32-bit and 64-bit Windows 8.x, 7, Vista.

Install HEAT2 by clicking on the setup file. If there is any problem (such as HEAT2 cannot find the material files) please check the below.

Note that HEAT2/HEAT3 cannot be started directly by clicking on the exe-files. The programs should be started using the short-cuts on the start menu (or your own short-cuts on the desktop, see below how to create this).



By default, the setup program installs HEAT2_v9.exe and some other files to folder **C:\Program Files (x86)\BLOCON\HEAT2_v9**

The material files (*.mtl) and some other user files are placed in **C:\Documents and Settings*username*\My Documents\BLOCON\HEAT2 9**

This folder contains also user projects files. It needs local write/read rights.

Make sure that the short-cut for "HEAT2 v9" in the start-menu (right-click properties on the short-cut) has "**Target**" set to "**C:\Program Files (x86)\BLOCON\HEAT2_v9\HEAT2_v9.exe**" (or wherever the exe-file is located) and "**Start in**" set to "**...Documents\Blocon\HEAT2 9**" (or wherever the files are located).

The installation with the setup file will automatically create correct folders. Administrators rights are needed for the setup since the setup-program writes in the registry and program folders.

If needed, you can also make shortcuts on the desktop for each user with properties set to **Target:** "C:\Program Files\Blocon\HEAT2_v9\HEAT2_v9.exe"

and

Start in: "...User1\My Documents\Blocon\HEAT2 9" for user1

Start in: "...User2\My Documents\Blocon\HEAT2 9" for user2

etc.

Each folder should contain copies of the original material files (.mtl and Mtrl50.txt), ini-file, and projects files. (The same that was installed). The folders needs local write/read wrights. This way, each user has his/her own files.

2. Temperature dependent material properties

2.1 Introduction

The thermal conductivity and the heat capacity may change with temperature. This gives a non-linear thermal problem. For the two-dimensional case, the heat equation governing the thermal process is:

$$C(T) \cdot \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(\lambda(T) \cdot \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda(T) \cdot \frac{\partial T}{\partial y} \right)$$

Where

C	The heat capacity	$J/(m^3 \cdot K)$
λ	Thermal conductivity	$W/(m \cdot K)$
T	Temperature	K
x, y	Coordinates	m
t	Time (s)	

Fire safety design and soil freezing are typical applications where there are temperature dependent material properties resulting in non-linear thermal processes.

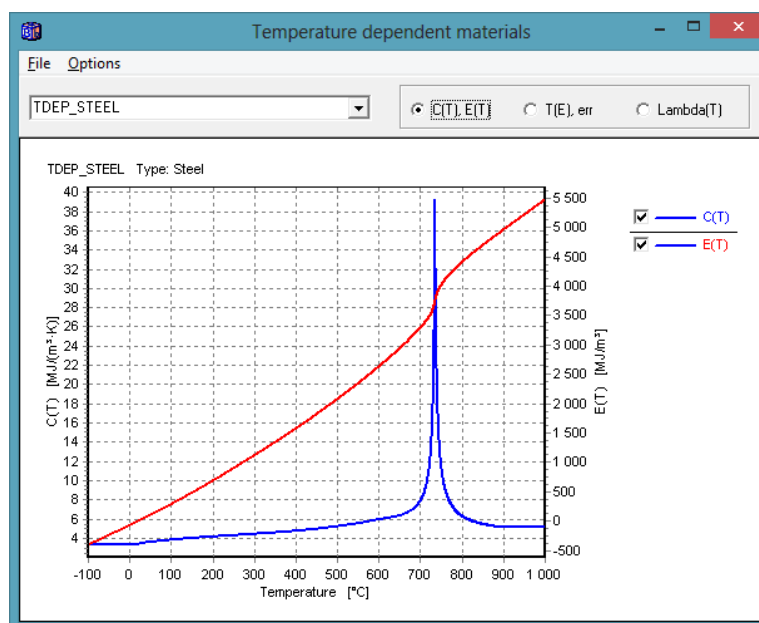
2.2 Enthalpy calculations

HEAT2 solves the heat equation using a numerical method as described in “*Theory of HEAT2 and HEAT3, (Doctorate thesis, Blomberg), (PDF), 2.5 MB, 202 pages, 1996*”, see http://www.buildingphysics.com/manuals/avh_TB.pdf

For all areas that have materials with a volumetric heat capacity that depends on temperature another approach is now used to solve the heat equation using the energy $E(T)$:

$$E(T) = \int_{T_{ref}}^T C(T) dT$$

The figure below shows an example of the volumetric heat capacity $C(T)$ and the integrated curve $E(T)$ for steel that has a large change in heat capacity with peak around 735 °C.



Volumetric heat capacity $C(T)$ and the integrated curve $E(T)$ for steel.

2.2.1 Freezing of soil

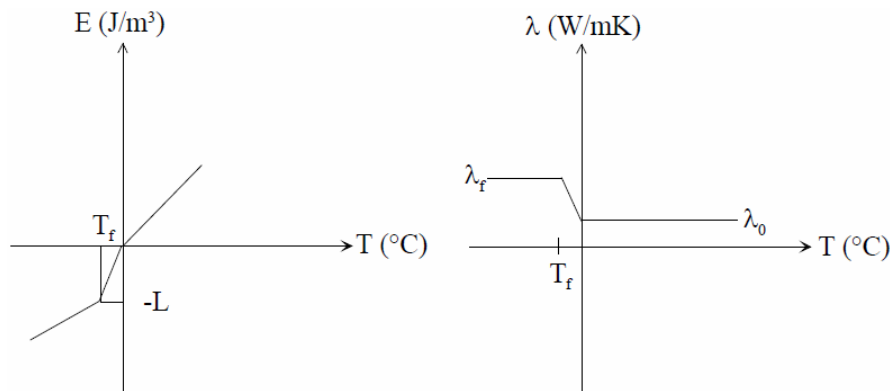
When soil is freezing, a moisture related phase change is occurring. Liquid water is transformed to ice. The thermal conductivity and the heat capacity will change with temperature. We have a non-linear thermal problem.

Let T_f denote the temperature in the soil when it is fully frozen. This temperature will be different for different soils. In clay, a considerable part of the water may be unfrozen at temperature below 0°C . The reason for this is that the water is located in small pores, where a lower temperature is required to freeze it. The material is unfrozen at 0°C . If we assume that the major part of the water in the soil is frozen at -1°C , then the freezing process will take place in the interval $0 - (-1^\circ\text{C})$. We will assume that the freezing process is linearly distributed within this interval. The energy per volume unit is called E and the change of energy, ΔE , associated with a change of temperature equal to ΔT . We have the following three cases in the freezing process, see [1] and [2]:

$$\Delta E = \begin{cases} C_0 \cdot \Delta T \\ \frac{L}{-T_f} \cdot \Delta T \\ C_f \cdot \Delta T \end{cases}$$

Where

- ΔE Change of internal energy in the material (J/m^3)
- L Latent heat for phase change of material (J/m^3)
- C_f Volumetric heat capacity for frozen soil, ($\text{J}/(\text{m}^3 \cdot \text{K})$)
- C_0 Volumetric heat capacity for unfrozen soil, ($\text{J}/(\text{m}^3 \cdot \text{K})$)



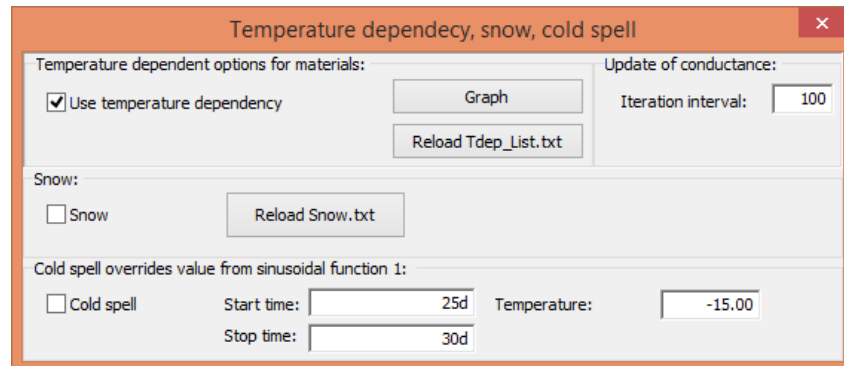
Consider the above figure. The relationship between the internal energy, E and the temperature is shown on the left hand side. In the right hand figure the relationship between thermal conductivity and temperature is given. The thermal conductivity for unfrozen soil is λ_0 and for fully frozen soil λ_f with a linear interpolation in the freezing interval. The parameter L is the latent heat. The latent heat for freezing of water is $330 \text{ kJ}/\text{kg}$. The relation between the moisture content per volume in a material, w (kg/m^3), and the latent heat L (J/m^3) reads:

$$L = w \cdot 330 \cdot 10^3 \quad (\text{J}/\text{m}^3)$$

The simulated frost penetration is assumed to follow the -1°C isotherm. By plotting the -1°C isotherm, we can analyze the risk for frost heave.

2.3 Using temperature dependent materials

It is quite easy to use temperature dependent materials in a calculation. Just draw the geometry as usual in the pre-processor (or bitmap editor) choosing pre-defined materials from the material list. Go to menu item “Temperature dependency, snow, cold spell” in the “Input” menu and check “Use temperature dependency”, see below.



2.4 How to define temperature dependent materials

There is one predefined material in HEAT2 for steel named “TDEP_STEEL”. It is quite easy to add new materials that have heat capacities and/or thermal conductivities dependent on temperature. These should be given in a text file named “TDep_List.txt” (located in the same folder as the material files). For each material, a name is given which has to have the prefix “TDEP_”. The row after gives the type (“1” or “2”), and the following rows contain data for that type:

```
TDEP_”name1”
Type
Data for type

TDEP_”name2”
Type
Data for type

etc.
```

There are currently two types available, see below.

2.4.1 Type 1, Freezing of soil

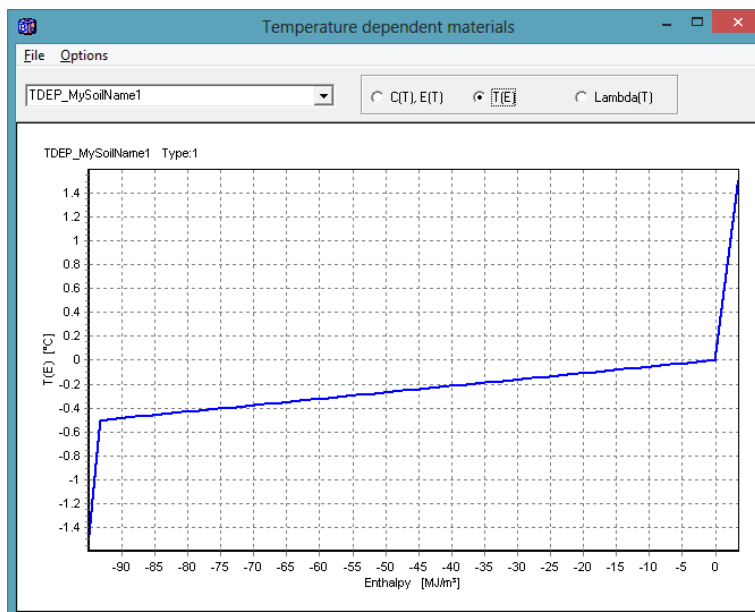
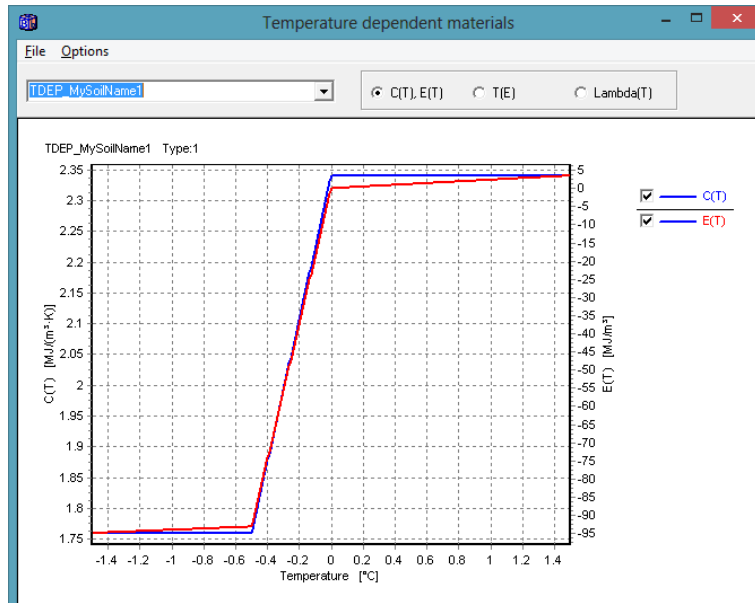
Type “1” means data used for freezing of soil with data as follows:

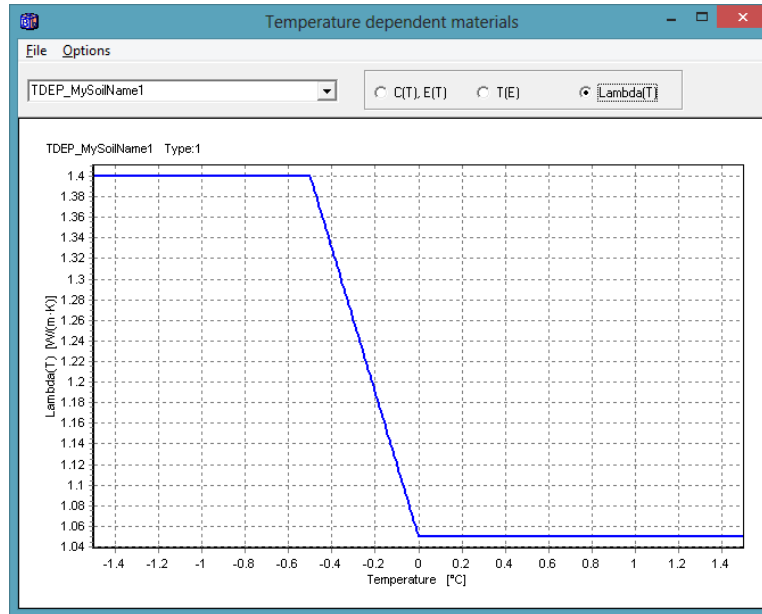
λ_f	Thermal conductivity for frozen soil	W/(m·K)
λ_0	Thermal conductivity for unfrozen soil	W/(m·K)
C_f	Volumetric heat capacity for frozen soil	J/(m ³ ·K)
C_0	Volumetric heat capacity for unfrozen soil	J/(m ³ ·K)
L	Latent heat in for phase change of material	J/m ³
T_f	Temperature in the soil when it is fully frozen	°C

An example is

```
TDEP_MySoilName1
1      type
1.40  Lf
1.05  L0
1.76e6 Cf
2.34e6 C0
9.32e7 L
-0.5  Tf
```

Note that it is OK to write comments after each value (as done above). This example gives the following curves for volumetric heat capacity $C(T)$, enthalpy $E(T)$, temperature as a function of energy $T(E)$, and thermal conductivity $\lambda(T)$:





2.4.2 Type 2, piecewise linear volumetric heat capacity and thermal conductivity

The second type “2” means that the heat capacity and thermal conductivity are assumed to vary piecewise linearly with temperature. Input data for each material are given as a number of temperature property-value pairs. The data are as follows:

- N_C Number of pairs for temperature and volumetric heat capacity, (max=100)
- T_1 Temperatures
- ...
- T_{N_C}
- C_1 Volumetric heat capacities
- ...
- C_{N_C}
- N_λ Number of pairs for temperature and thermal conductivity
- T_1 Temperatures
- ...
- T_{N_C}
- λ_1 Thermal conductivities
- ...
- λ_{N_λ}

An example is

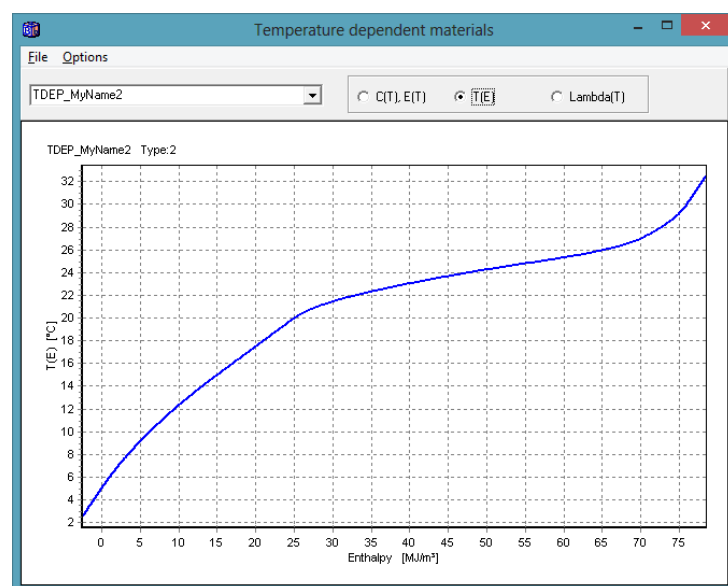
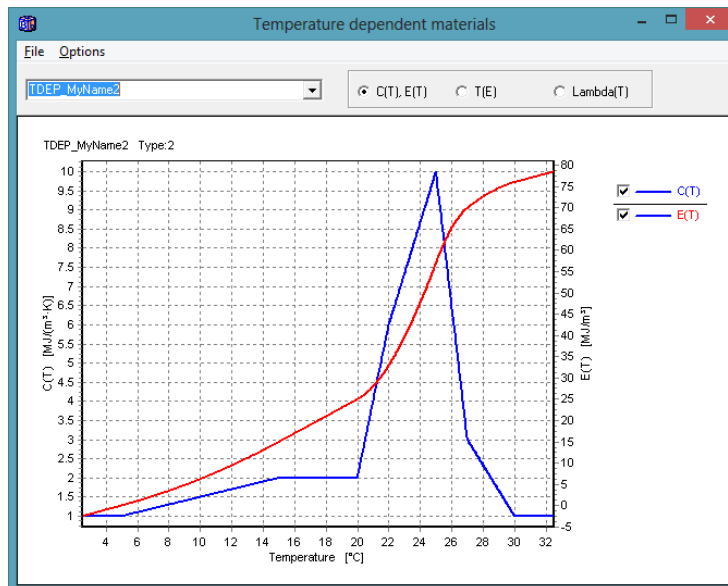
```

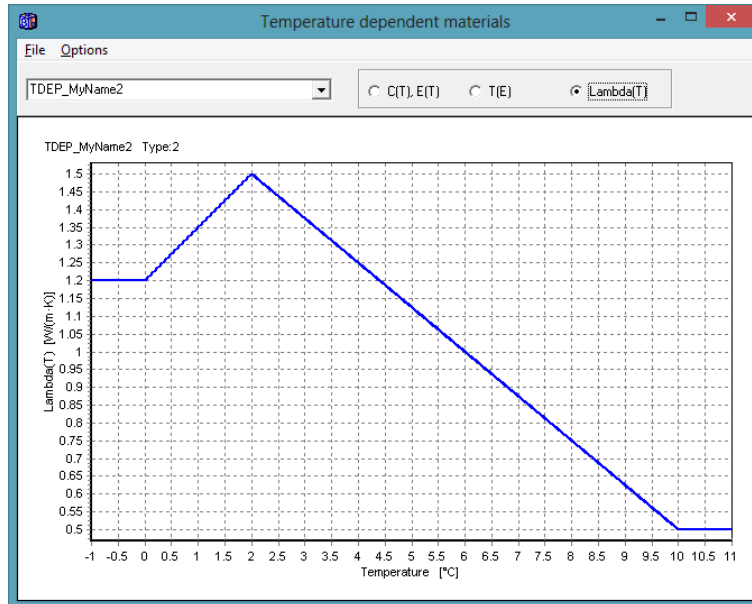
TDEP_MyName2
2      type
7      7 pairs for T,C
5      T1
15     T2
20     T3
22     T4
25     T5
27     T6
30     T7
1e6    C1

```

2e6	C2
2e6	C3
6e6	C4
10e6	C5
3e6	C6
1e6	C7
3	3 pairs for T, λ
0	T1
2	T2
10	T3
1.2	λ_1
1.5	λ_2
0.5	λ_3

The above example gives the following curves for volumetric heat capacity $C(T)$, enthalpy $E(T)$, temperature as a function of energy $T(E)$, and thermal conductivity $\lambda(T)$:





Note that the values are assumed to be constant before and after the end points for both the volumetric heat capacity and the thermal conductivity.

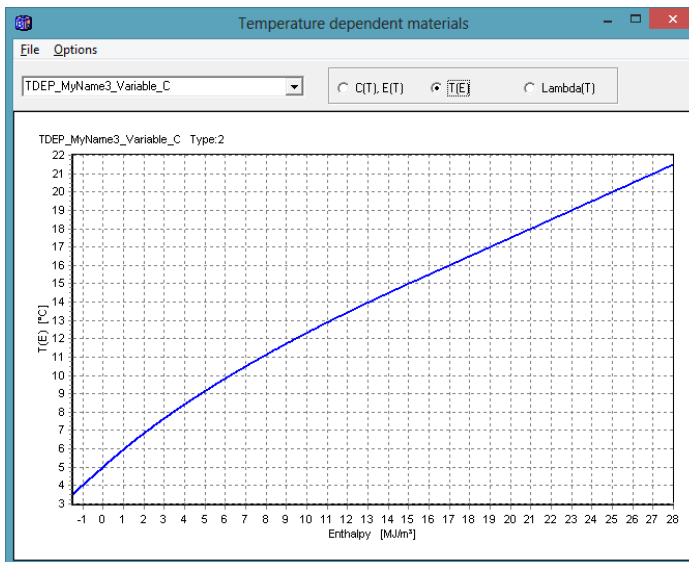
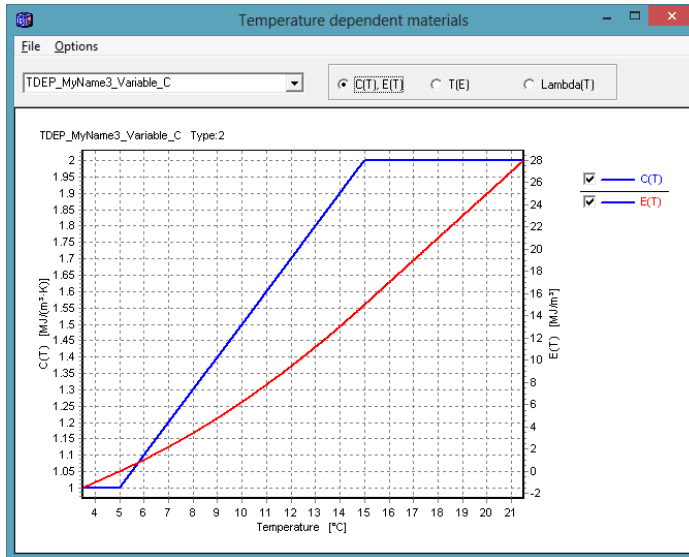
The following example shows a variable volumetric heat capacity but a constant thermal conductivity:

```

TDEP_MyName3_Variable_C
2      type
3      3 pairs for T,C
5      T1
15     T2
20     T3
1e6    C1
2e6    C2
2e6    C3
1      1 pair for T,  $\lambda$ 
5      T1 (any value can actually be given)
1.2     $\lambda$ 

```

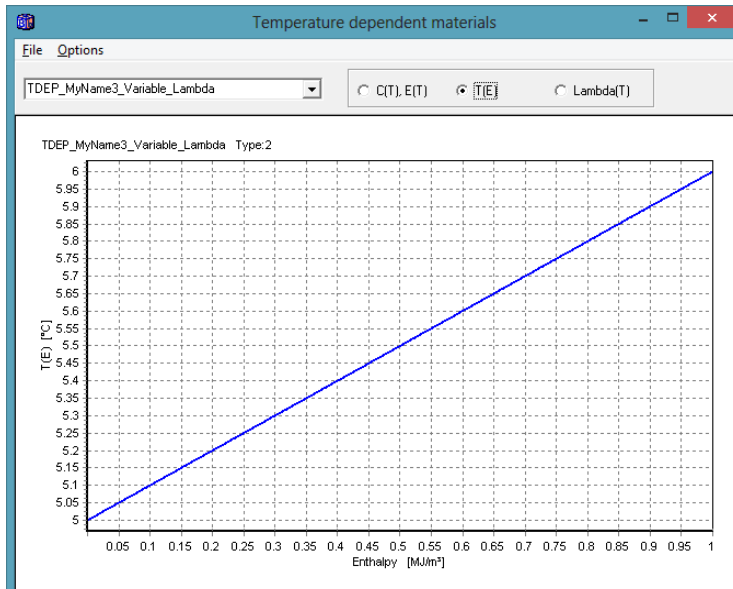
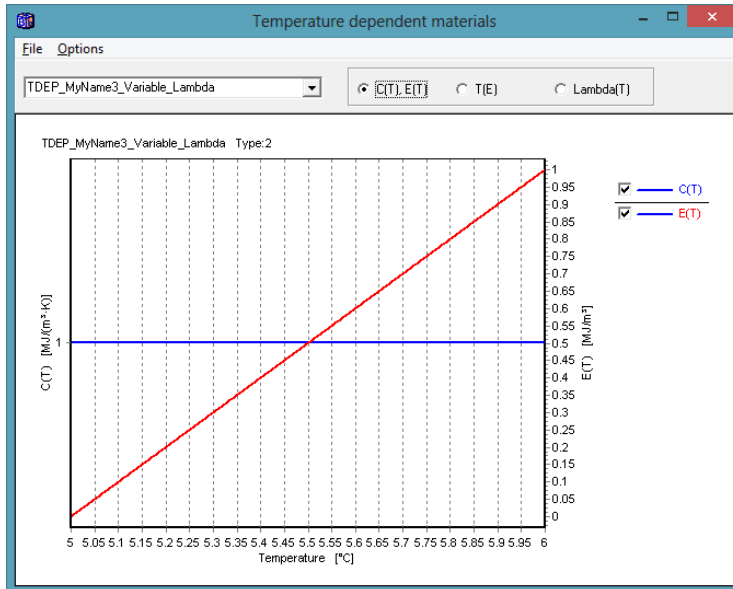
This will give a constant thermal conductivity of 1.2, and other curves as shown below.

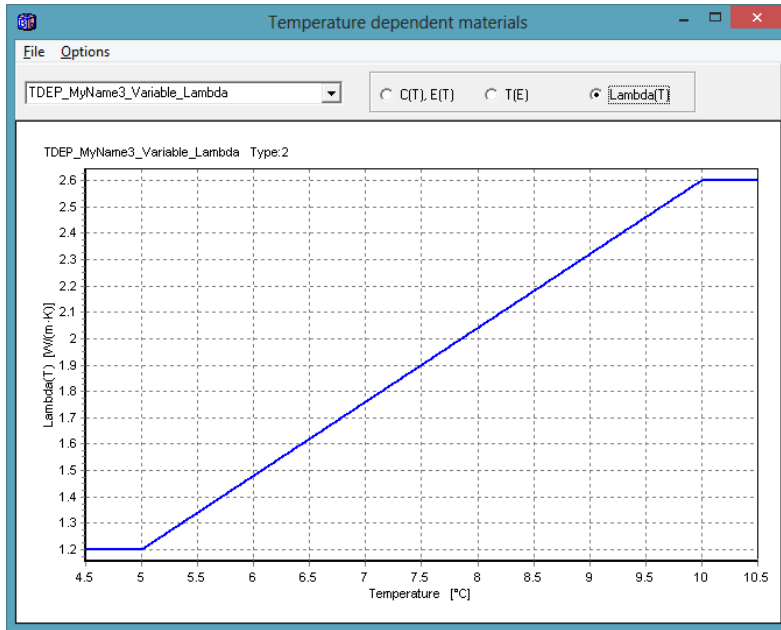


The following example shows a variable thermal conductivity but a constant volumetric heat capacity:

```
TDEP_MyName3_Variable_Lambda
2      type
1      1 pair for T,C
5      T1
1e6    C1
2      2 pairs for T, λ
5      T1
10     T2
1.2    λ1
2.6    λ2
```

This will give a constant volumetric heat capacity of 1e6, and other curves as shown below.



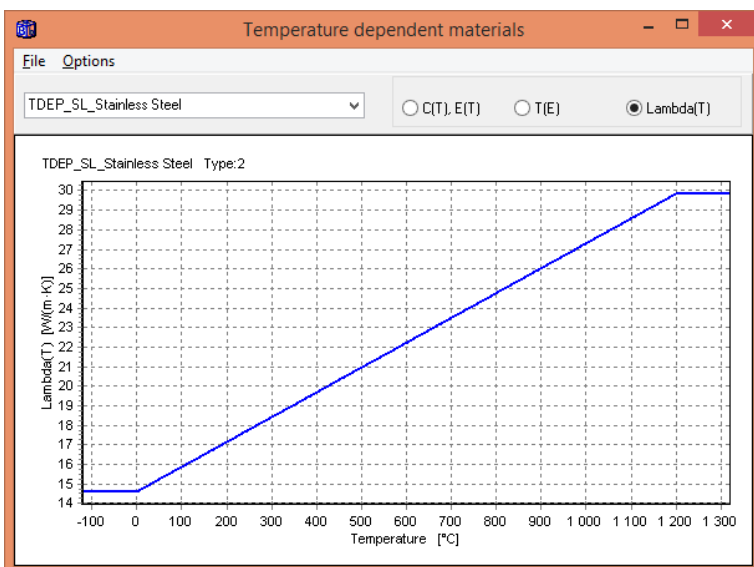
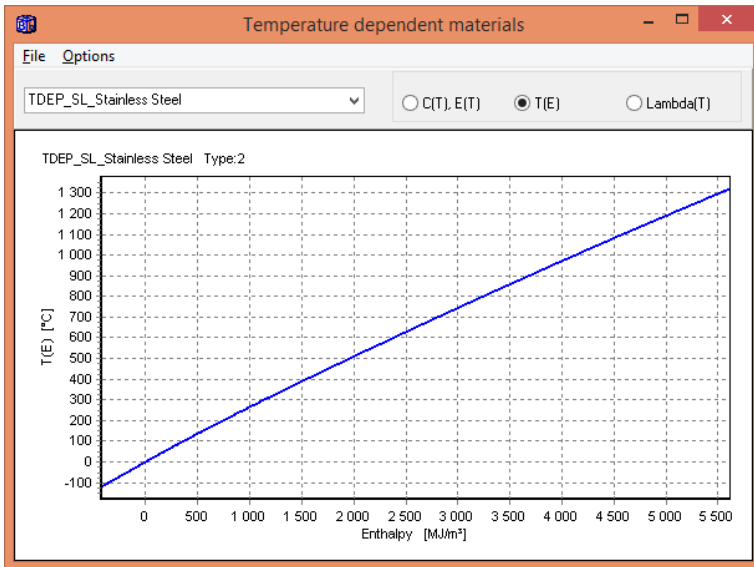
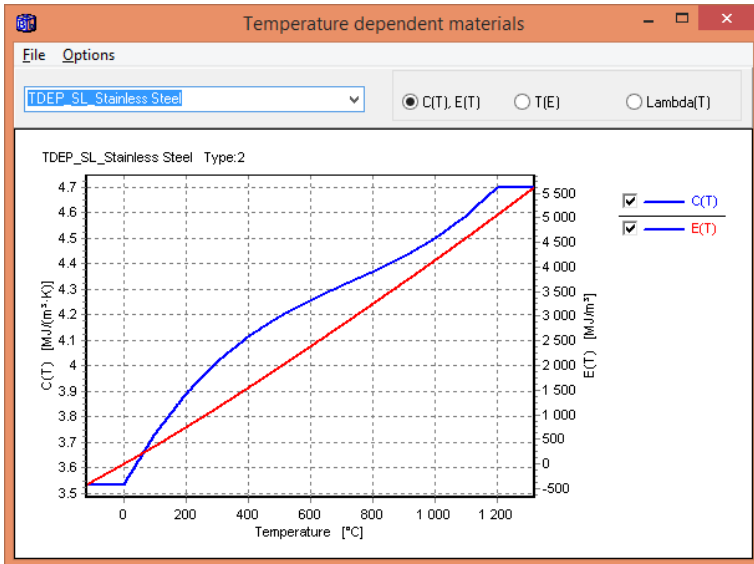


The following example shows data for stainless steel:

```

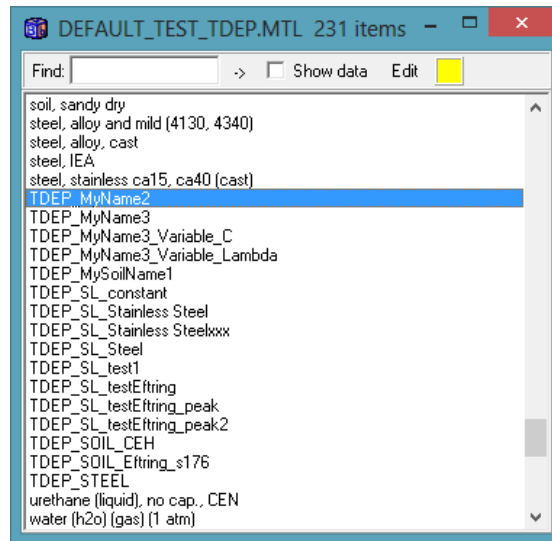
TDEP_SL_Stainless Steel
2      type
13     points for T,C
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
3.533e6
3.731e6
3.889e6
4.015e6
4.114e6
4.192e6
4.256e6
4.313e6
4.367e6
4.427e6
4.498e6
4.586e6
4.698e6
2      points for T,lambda
0
1200
14.6
29.84

```

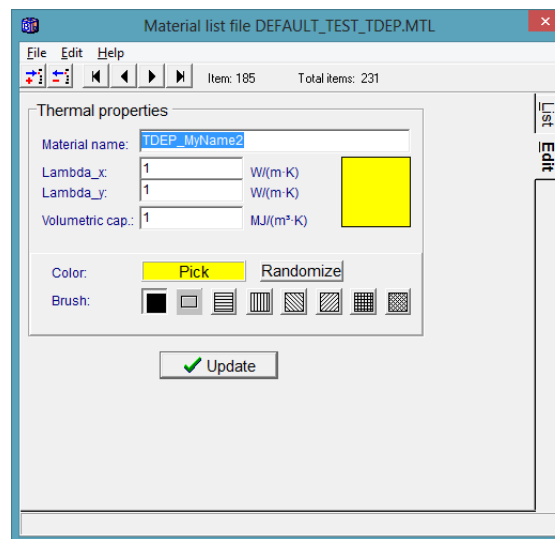


2.4.3 The material list

All the temperature dependent materials in file “TDep_List.txt” will be imported to the material list when HEAT2 is started. It is also possible to edit the file and reload the contents while HEAT2 is running using the “Reload” button, see Section 2.3.

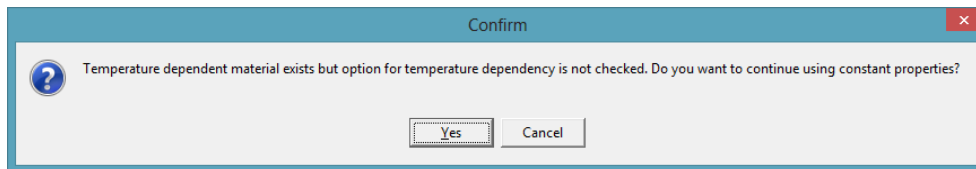


All temperature dependent materials will have the color yellow as default. It is however possible to change this color to another if needed by double clicking on the material which will bring forward the material editor, see below. You need to save the material file to keep any change.



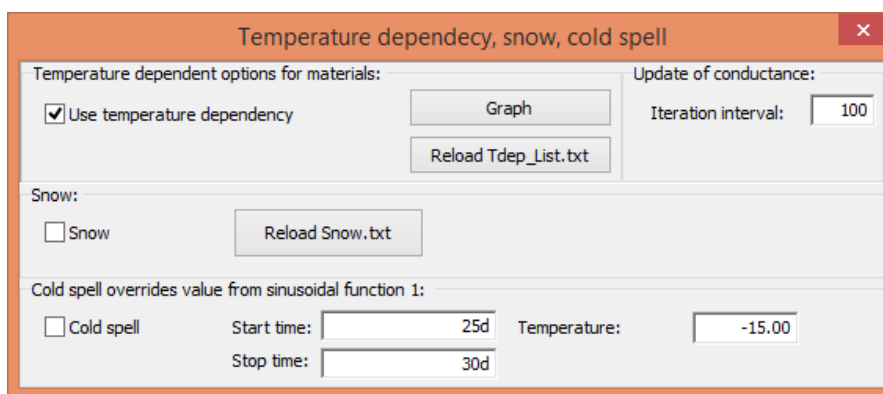
2.4.4 Option for overriding and using constant properties

In order to account for temperature dependent material properties you need to check the box “Use temperature dependency”. If there are materials with its name starting with “TDEP_” and this box is not checked, there will be a warning. There is an option to continue the calculation using constant properties. In this case the values from the material editor will be used (these can be changed from the default values 1).



2.5 Update of conductances, heat capacities and stable time-step

In the numerical iteration, there is no a need to update the conductances and heat capacities at every time-step since this is quite time consuming. The user can choose the update interval for this, see below. A default value of 100 is used meaning that an update will occur every hundredth iteration. This is normally sufficient (e.g. the time-step used for the slab on ground calculation in the example below is 52 seconds which means that update is made every second hour).



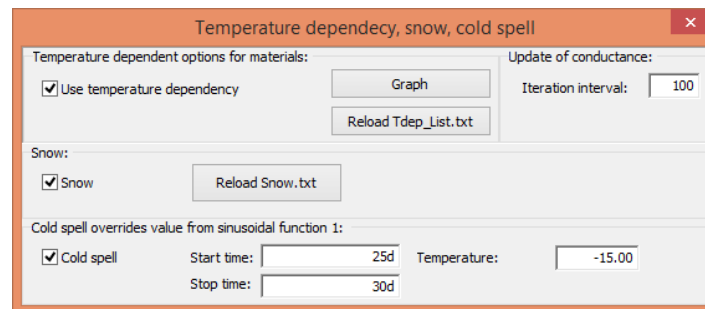
2.6 Miscellaneous about temperature dependent material properties

It should be relatively easy to define and use materials with temperature dependent material properties. The materials are picked from the material list and drawn in the pre-processor in the same way as constant materials. There are however some noteworthy comments:

- Note that you always need to have the file “TDep_List.txt” available when solving a problem that has temperature dependent material properties. If you e.g. need to email input data files to another user running HEAT2 on another PC, this file needs to be present (together with e.g. input files *.dat and *.h2p and material file *.mtl).
- The current data in file “TDep_List.txt” that describes the material properties are always used in the simulations. No data is actually saved in the input data files (*.dat or *.h2p) other than the material names. When HEAT2 encounters a temperature dependent material (i.e. when the name starts with “TDEP_”), the data are collected from the current file “TDep_List.txt” matching the name.
- You can make both transient and steady-state calculations. In the steady state case, the heat capacities do not matter and only the variation for the thermal conductivity is considered.
- Avoid using the same names for the materials. If the same name exists more than once in file “TDep_List.txt”, the last read data will be used.
- Different temperature dependent materials (and constant materials) can be used and mixed in a problem just like constant materials.
- There is one predefined material “TDEP_STEEL” that can be used without having to read the file “TDEP_List.txt”.

2.7 Snow layer and cold spell

This section is put in this chapter because snow layers and cold spells often are used for cases involving freezing of soil. Thus, the options are on the same window:

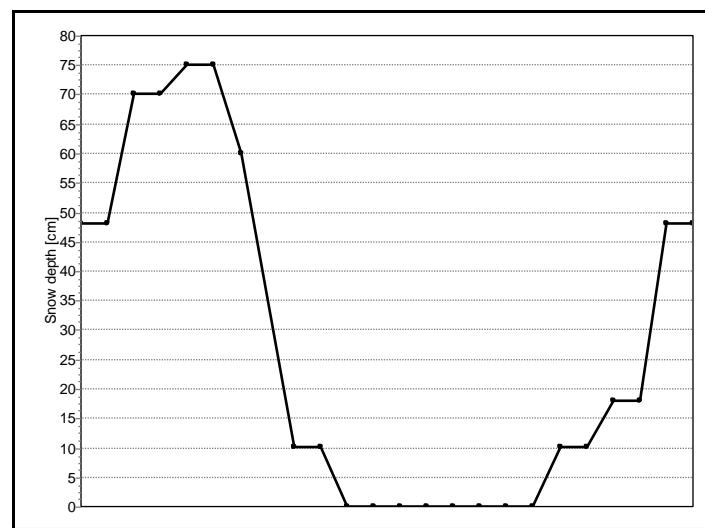


The snow is considered by adding a resistance to the surface resistance:

$$R_{\text{snow}} = d_{\text{snow}} / \lambda_{\text{snow}} \quad [\text{m}^2 \cdot \text{K} / \text{W}]$$

The values should be given in the file “snow.txt”. The first row should contain the thermal conductivity for the snow λ_{snow} , and the second row 24 values for the snow thickness in cm d_{snow} for a year (i.e. a value for every second week, starting at Jan 1). Here is an example:

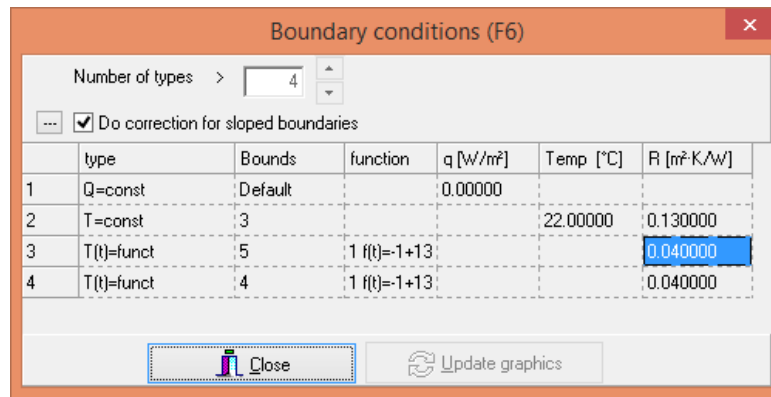
0.3 Kiruna North Sweden (Latitude: 67.85 Longitude: 20.22)
48 48 70 70 75 75 60 60 10 10 0 0 0 0 0 0 0 0 10 10 18 18 48 48



Snow depth (cm) for a year in Kiruna.

The resistance due to the snow layer is calculated during the simulation using linear interpolation between the snow depth values.

The extra resistance due to the snow layer will always be applied to **boundary condition type 3** and added to the normal surface resistance. In the example below, the ground surface (boundary 5) is coupled to BC type 3 that has a surface resistance of 0.04. The external side of a wall could e.g. have the same outdoor temperature function but does not have to be affected by the snow layer (boundary 4 connected to BC type 4 as below).



2.8 Example

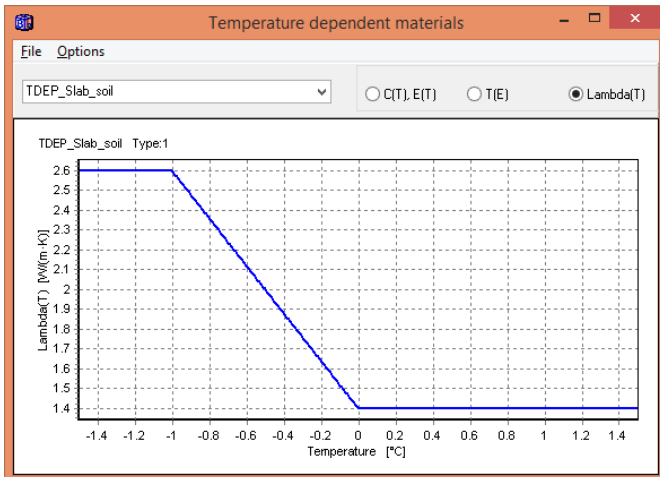
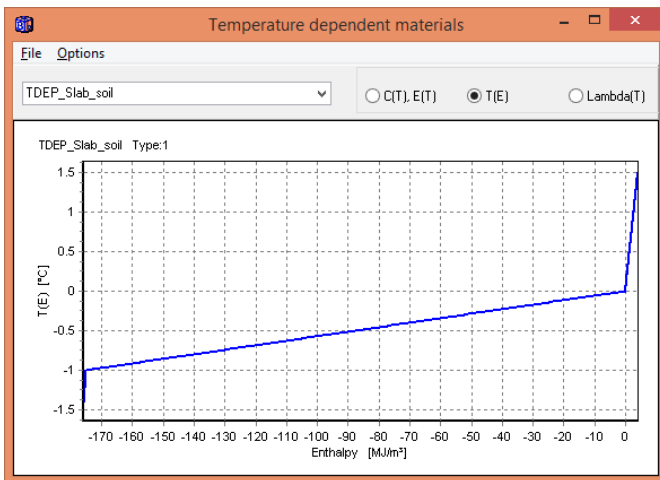
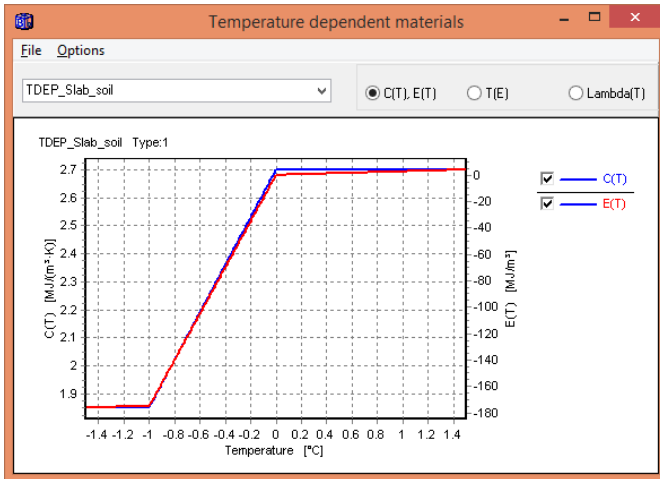
The following example (Slab_snow1.dat) shows a case with a temperature dependent material (soil), a periodic outdoor temperature with a sudden cold-spell, and a snow layer.

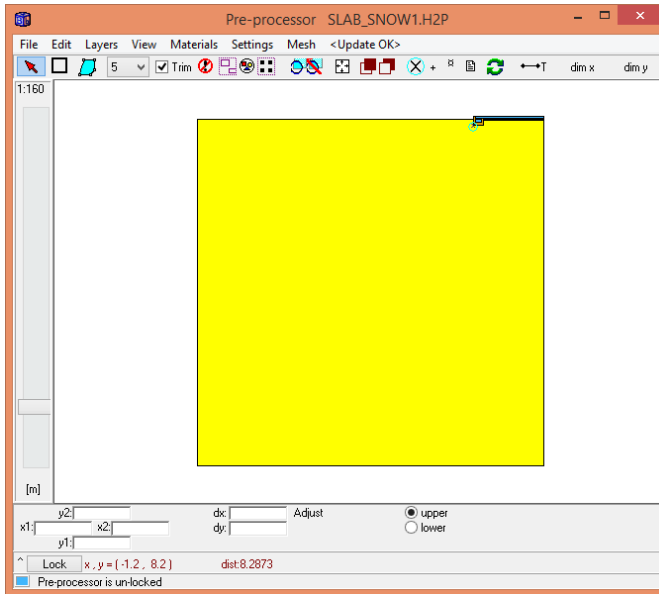
The location is Kiruna (North Sweden Latitude: 67.85 Longitude: 20.22). For the soil we use "TDEP_Slab_soil" with the following specification:

```

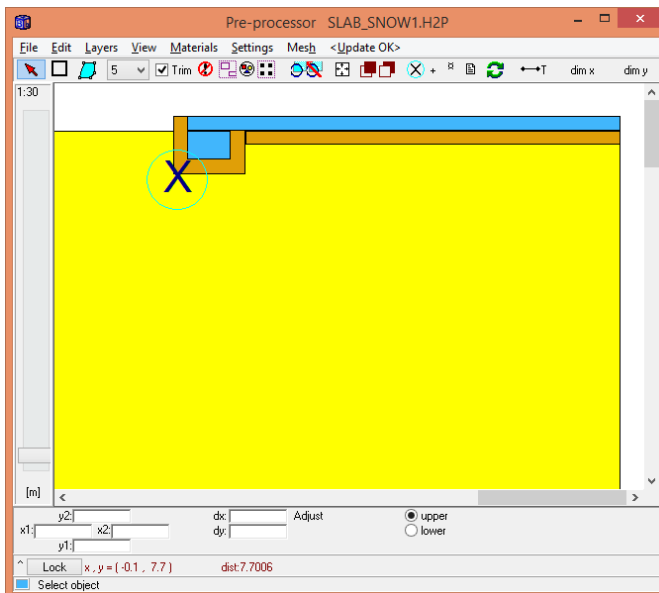
1      type
2.6    Lf
1.4    L0
1.85e6 Cf
2.7e6  C0
17.5e7 Latent heat in for phase change
-1     Tf

```





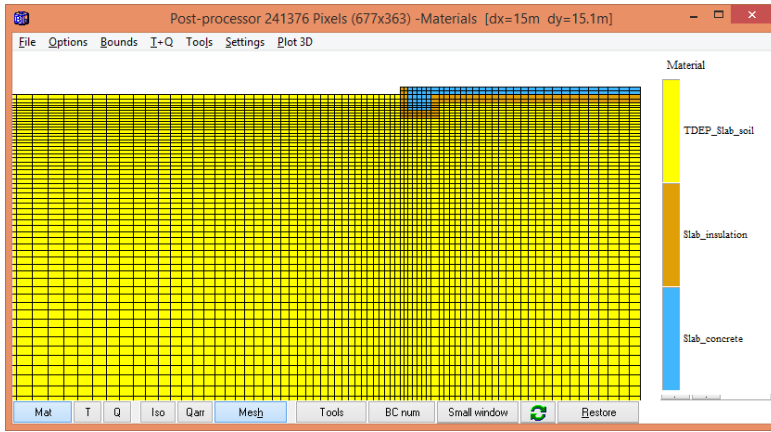
Pre-processor input



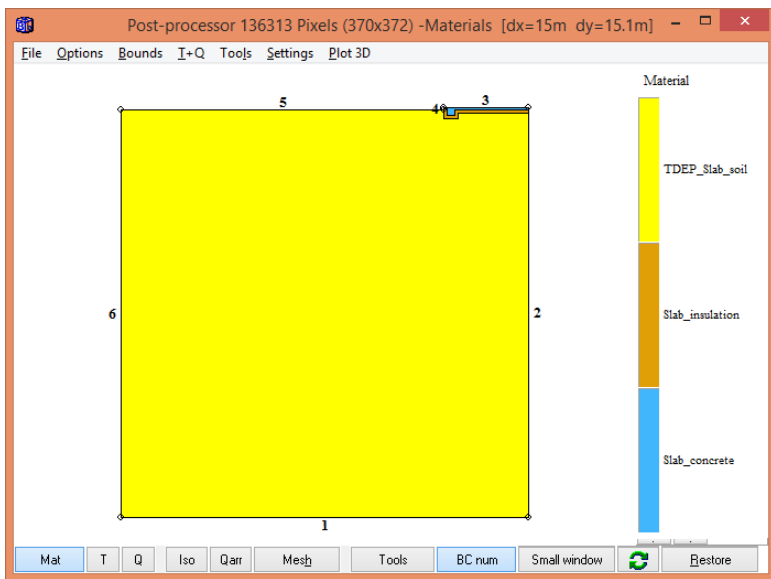
Zoomed detail with an expansion point

The thickness of the insulation, and of the concrete slab, is 1 dm.

We choose a mesh with about 10 000 nodes. The CPU-time to simulate 20 years is about 50 min on a i7 3770K 3.50GHz.

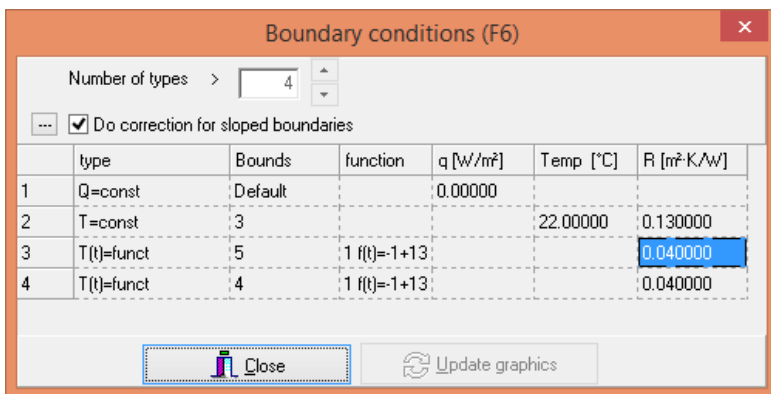


Mesh

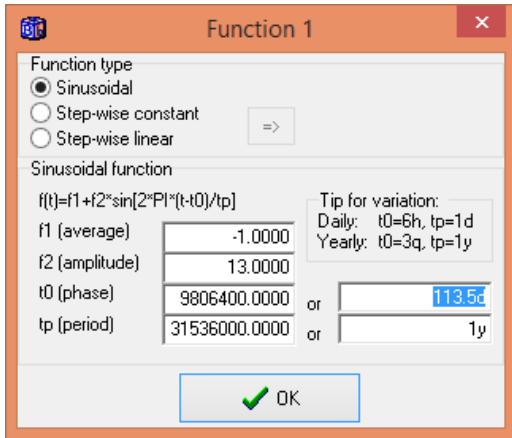


Boundaries 1-6

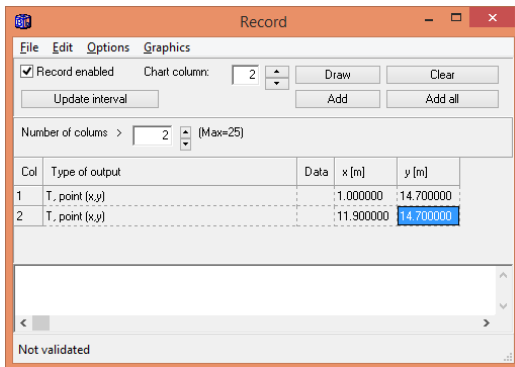
Note that BC type 3 is reserved for the optional snow layer (the resistance 0.04 will be added to the resistance of the snow layer), so we add type 4 for the external vertical side of the slab:



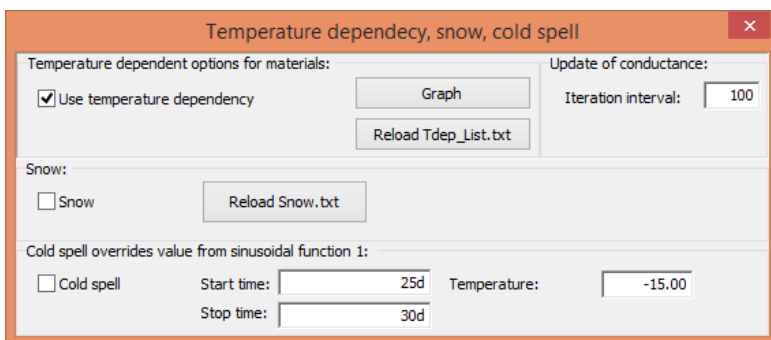
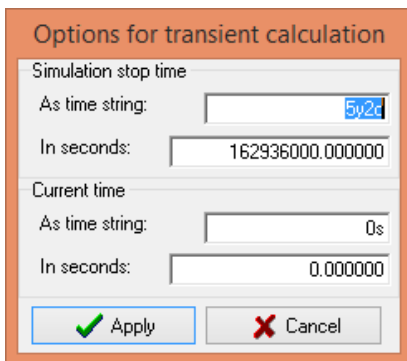
We use an outdoor variation as follows (the minimum will be at $t=24d$, i.e. end of January):



We are interested to follow the temperature at the lower left corner of the insulation (3dm beneath ground) and the undisturbed temperature (3dm beneath ground) 20 meters from the house. The points are added in the recorder:



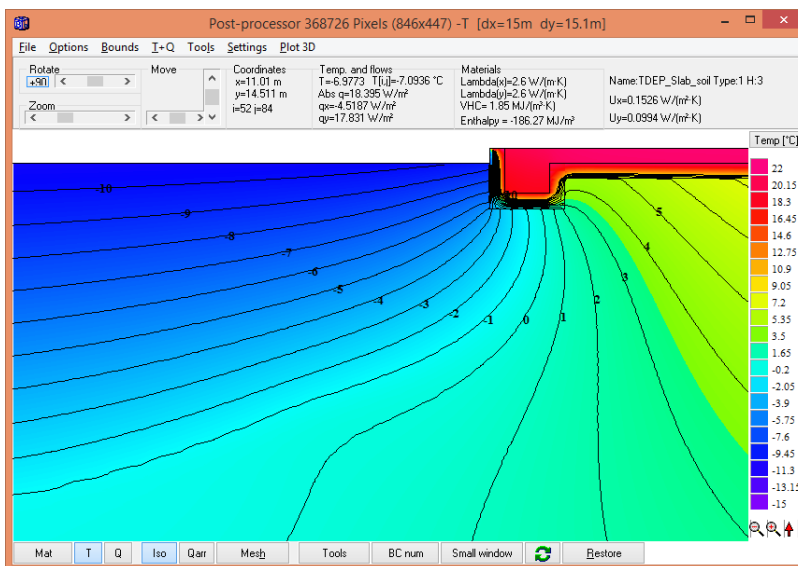
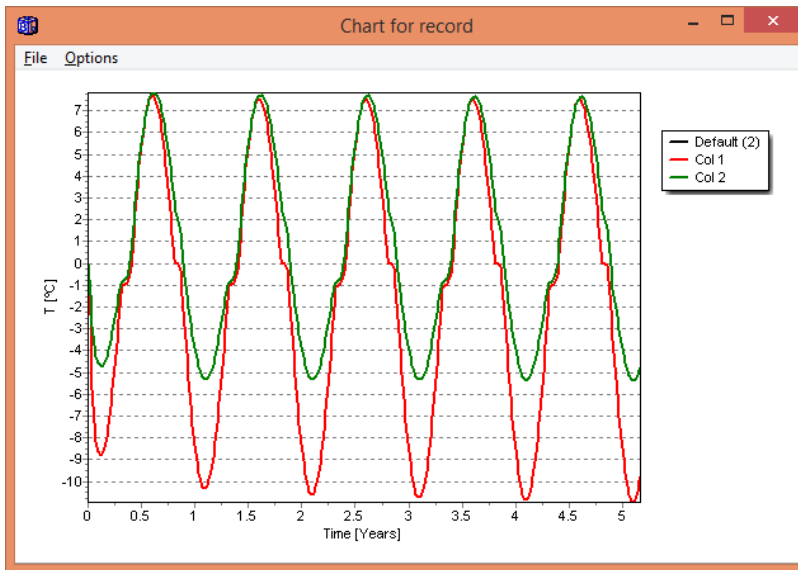
First we simulate 5 years and 2 months without any cold-spell and snow layer:



We get the following results:

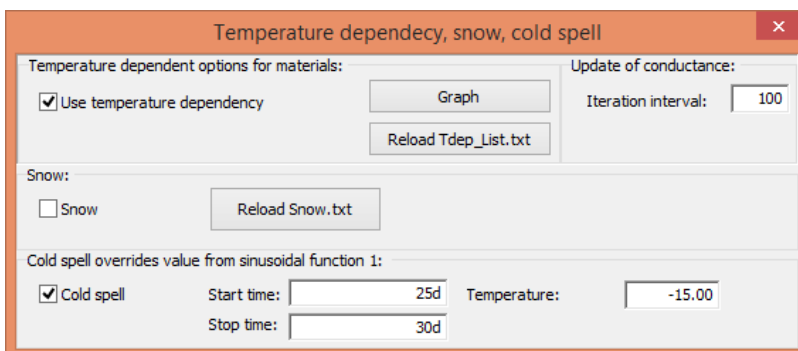
Green curve: Temperature at the lower left corner of the insulation (3dm beneath ground)

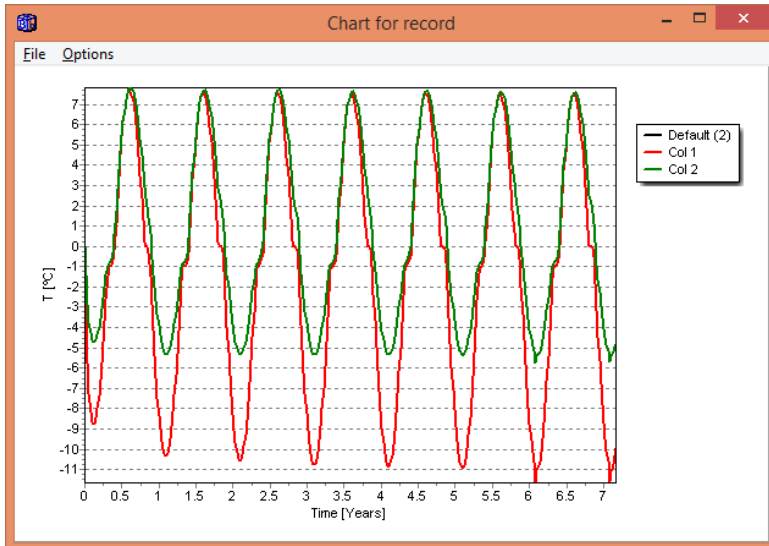
Red curve: Undisturbed temperature (3dm beneath ground) 20 meters from the house



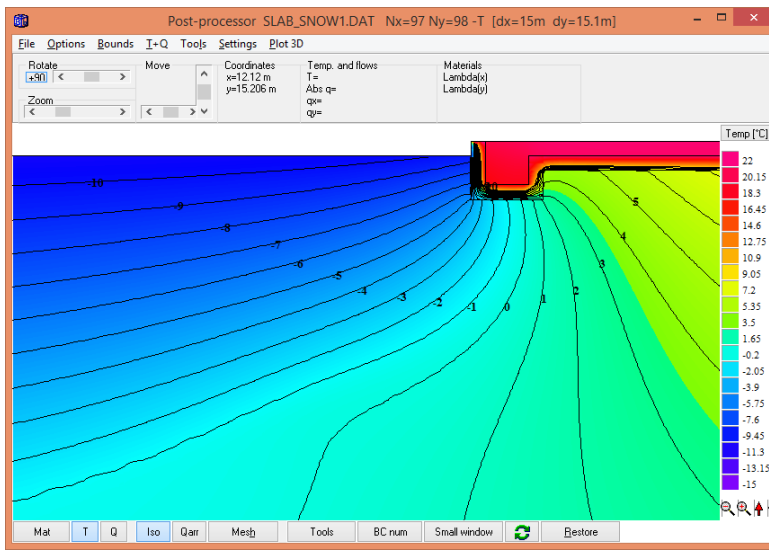
$t=5y$ 2months

Now we add a cold spell during 5 days in the end of January and continue the simulation for 2 years:





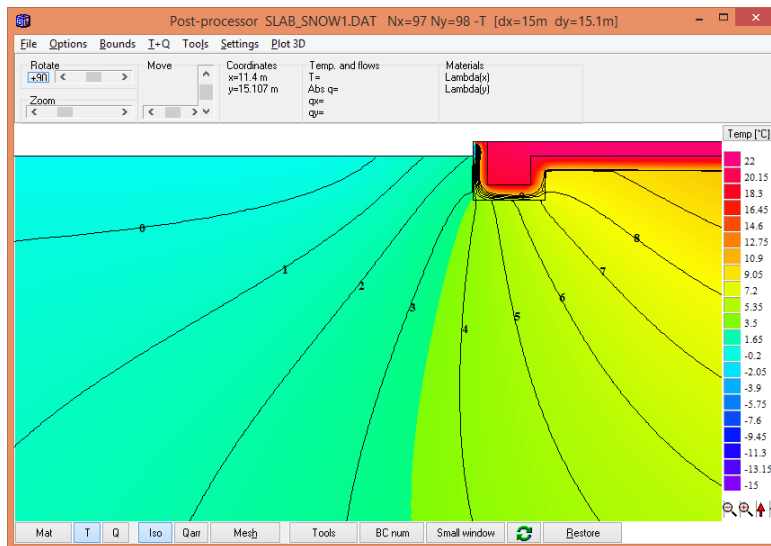
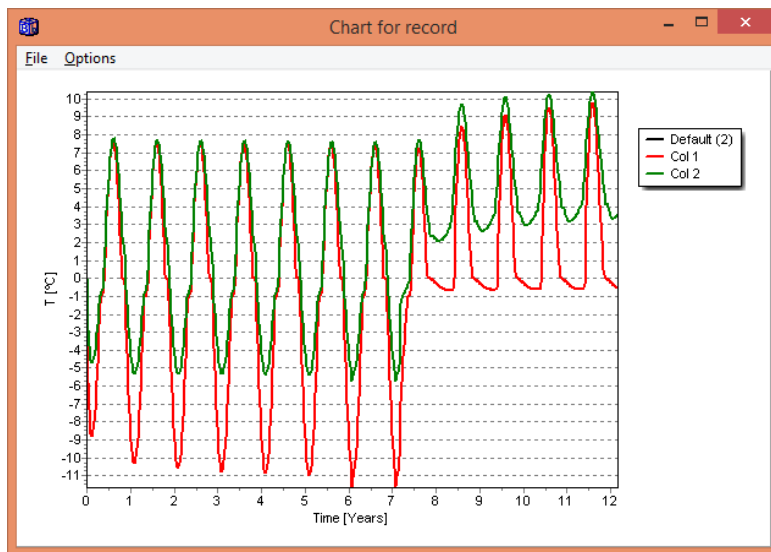
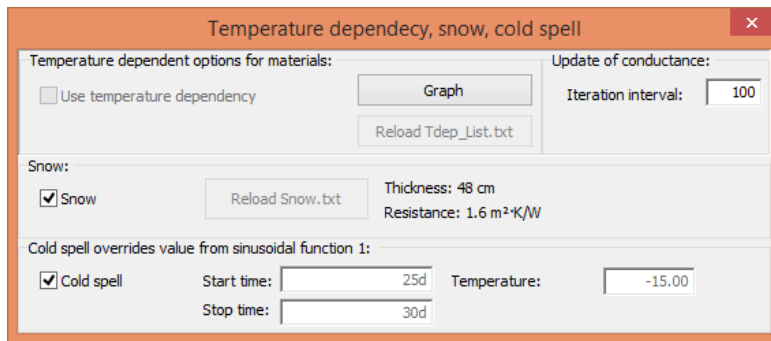
There is a small effect of the cold-spell (which is only 5 days long).



t=7y 2months

Now we also add the snow layer and continue the simulation for 5 more years:

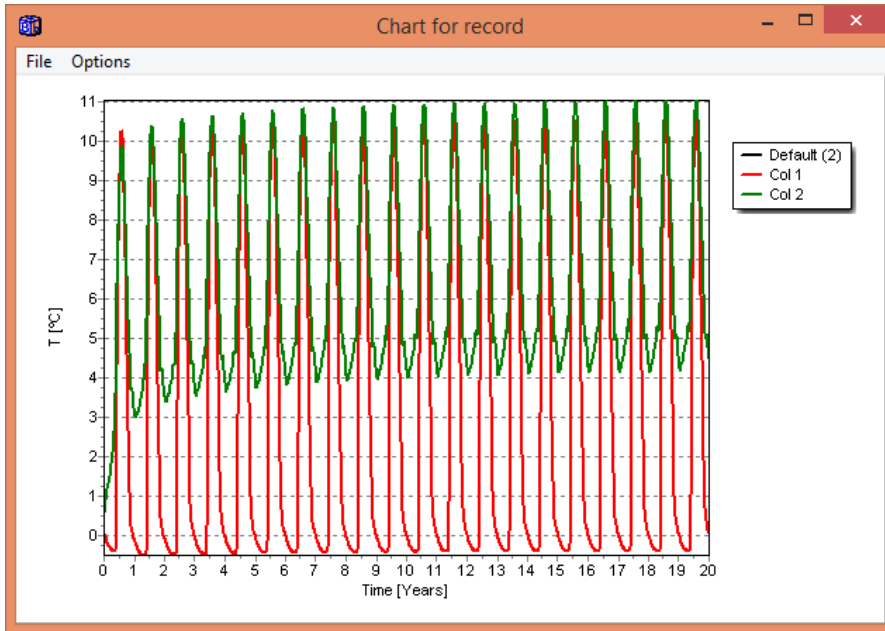
Note that during simulation, the current snow layer thickness and calculated resistance will be shown:



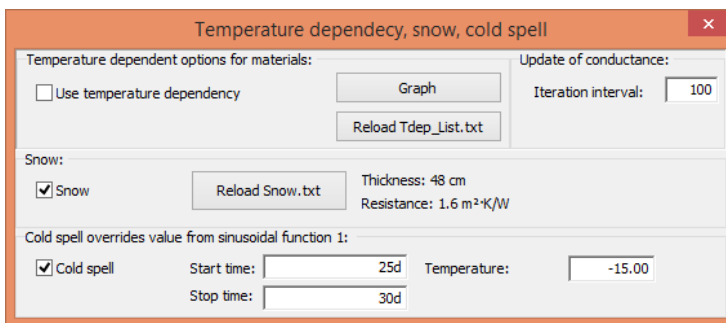
t=12y 2months

We see that the snow greatly increases the temperature in the ground. The lowest temperature at the lower left corner of the insulation (3dm beneath ground) is about 9°C higher (about -6°C for the case without snow and about +3°C for the case with snow).

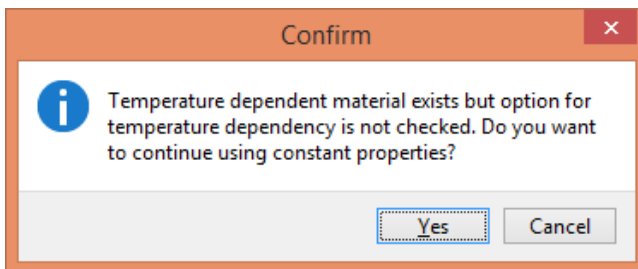
A simulation 20 years from the start (with snow and cold-spell) gives:



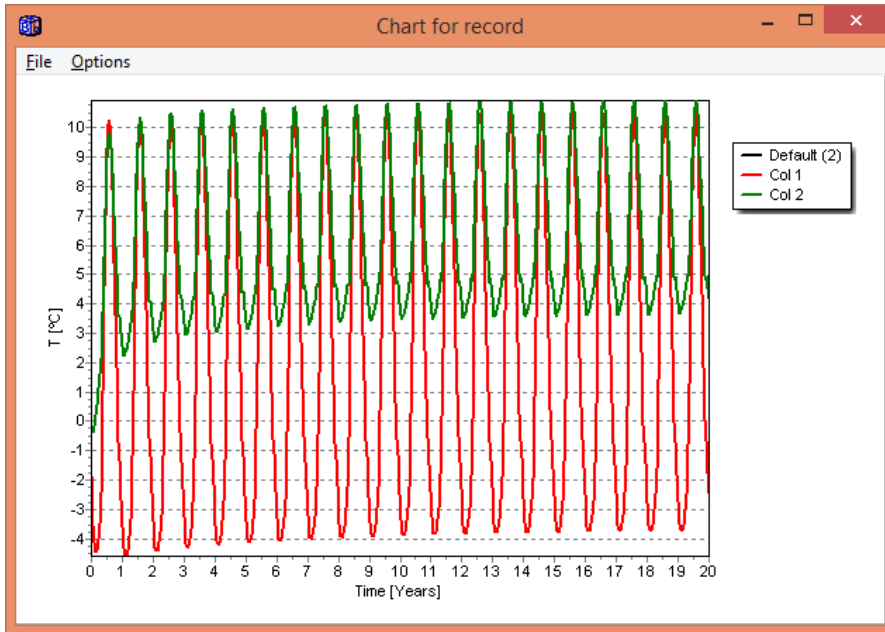
Now we run the case without temperature dependent materials, assuming the unfrozen values for the soil ($L=1.4$ and $C=2.7e6$):



Upon simulation the following message appears that we answer yes to:



The temperatures are somewhat lower when constant properties are used:

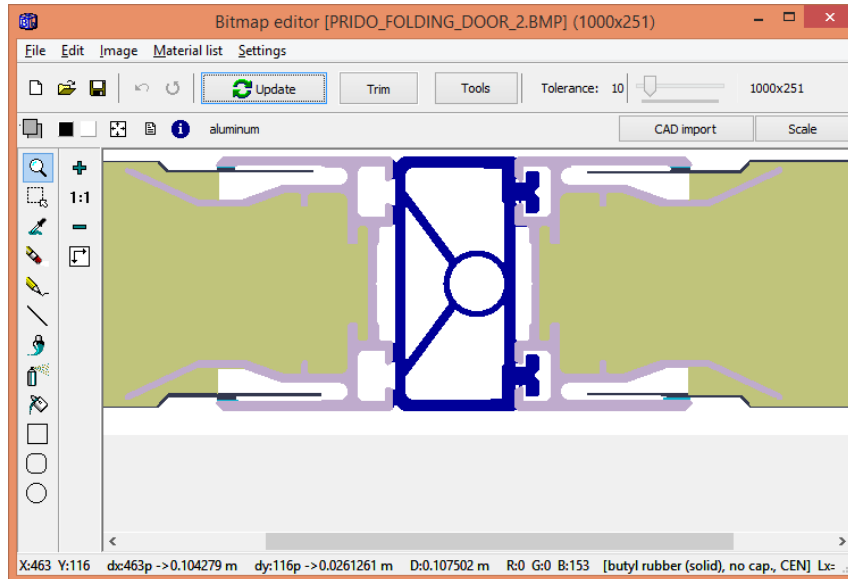


For the chosen mesh with about 10 000 nodes the CPU-time on a i7 3770K 3.50GHz to simulate 20 years is about 50 min for the case with temperature dependency, and about 14 min with constant properties. So, enthalpy calculation takes longer time.

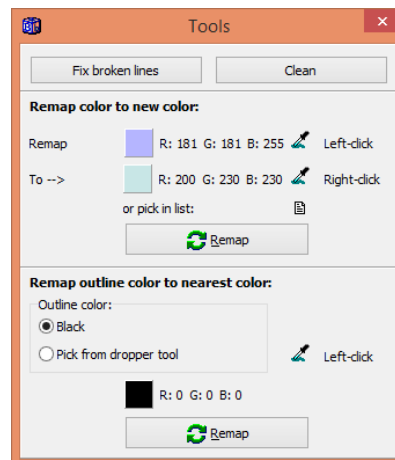
3. Enhancements in the Bitmap Editor

3.1 Introduction

There are some new features for the Bitmap Editor:



The **Tools** button opens this window:



Fix broken lines: Fixes lines that are broken so that the fill tool will work better, see below.

Clean: Removes unwanted single pixels from a bad cad-file, bitmap, etc.

Remap color to new color: Use the dropper tool and left-click on a material to get the source color. Right-click on a material, or pick a material in the material pick list to set the target color. Note that the tolerance (using the track-bar) can be set for the source color (a higher tolerance will remap “nearby” colors of the original color to the destination color), see update manual for version 8.

Remap outline color to the nearest color: Normally, the outline color is black. Use the dropper tool and left-click on the outline to get the source color if this is not black. The pixels at the outline will be replaced by the most common (non-white) neighboring pixel.

Press the undo-arrow to undo any changes.

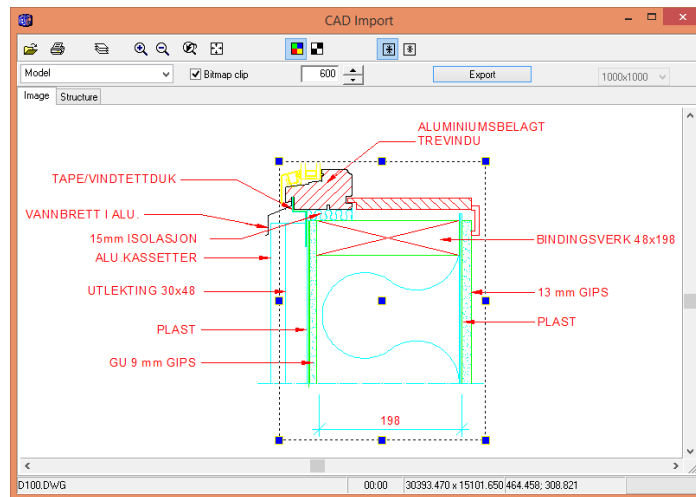
Note: An imported CAD image will automatically be “fixed” and “cleaned” unless menu item “Settings/Fix and clean image when importing from CAD” is unchecked. For more info about CAD import, see chapter 4.

There is a “Trim” checkbox in the upper tool-bar. If this is checked, the material pixels in the bitmap will “inflate” to better assure that materials are connected when the geometry is updated. See also the update manual for version 8.

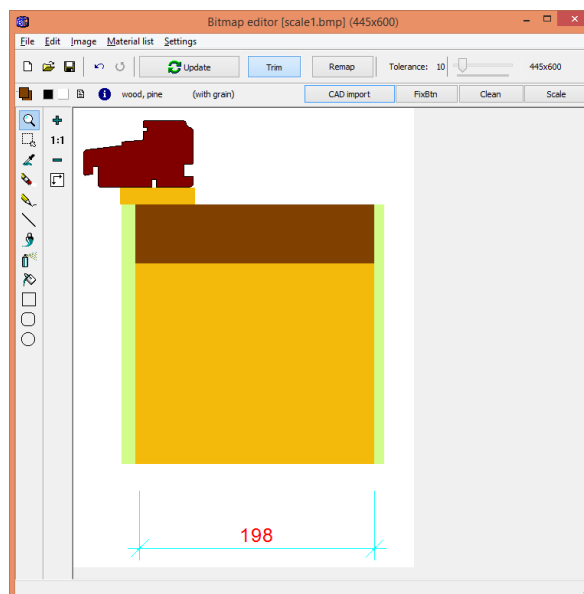
A white and black rectangle is added in the upper left side. You can e.g. choose the rectangle tool on the left hand side tool bar and pick the white color (by clicking on the white rectangle) and erase areas. It is now possible to use the mouse scroll to zoom in/out.

3.2 Setting the scale of the geometry

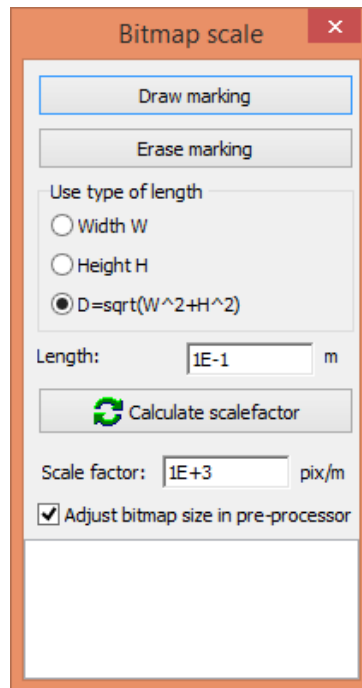
There is a new feature to specify the scale for the geometry (button **Scale**). This is illustrated by an example. Assume we have the following dwg-file:



The image is imported and edited to the following simplified bitmap:

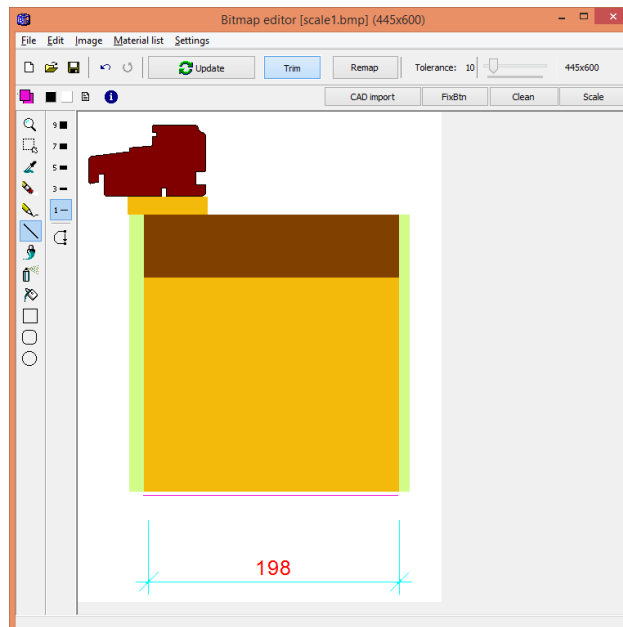


Now we want to specify the measurements. We know that the insulation is 198 mm wide. Open the scale window by pressing the scale button.

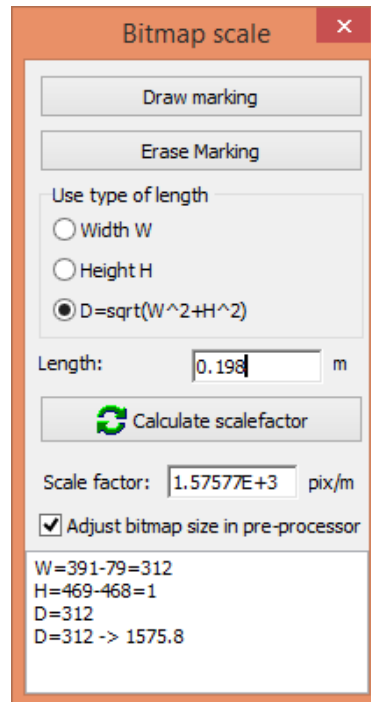


Do as follows:

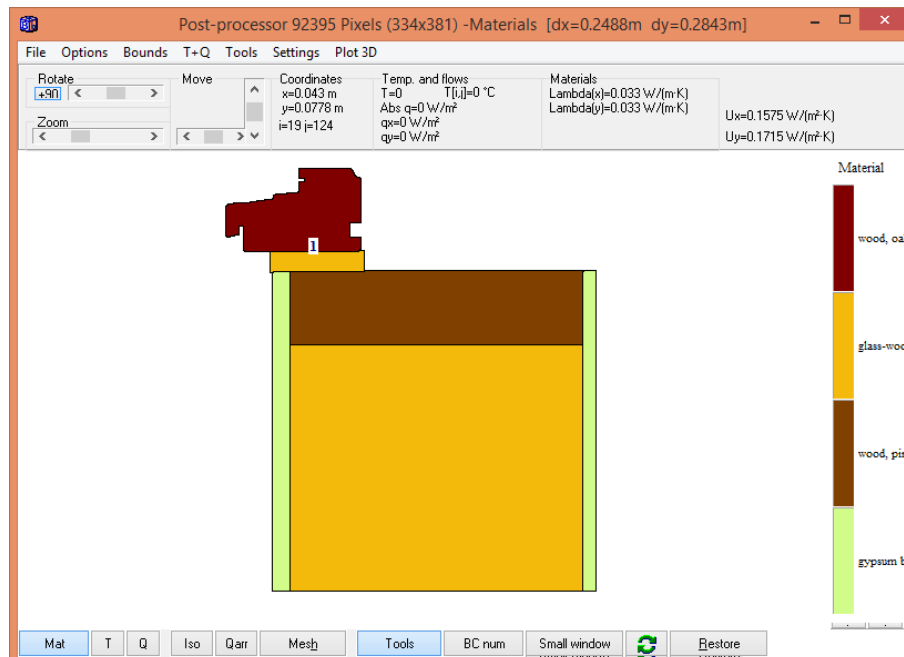
1. Press “Draw marking”. This will make it possible to draw a red line in the editor. The line is here drawn just beneath the insulation.



- Give the length for the line (0.198 m) and press “Calculate scale factor”. The scale factor is calculated to be 1.57577E3 pix/m. Here we use the default option “D” for the type of length since this is the same as the width.



- Press the ”Update” button in the Bitmap Editor to generate the final geometry as shown in the post-processor. It may be a good idea to check the geometry using the mouse (in this case we look at the x-coordinates where the wall insulation starts and ends). Also note the resulting total width and height as shown in the window title (dx=0.2488 m and dy=0.2843 m).

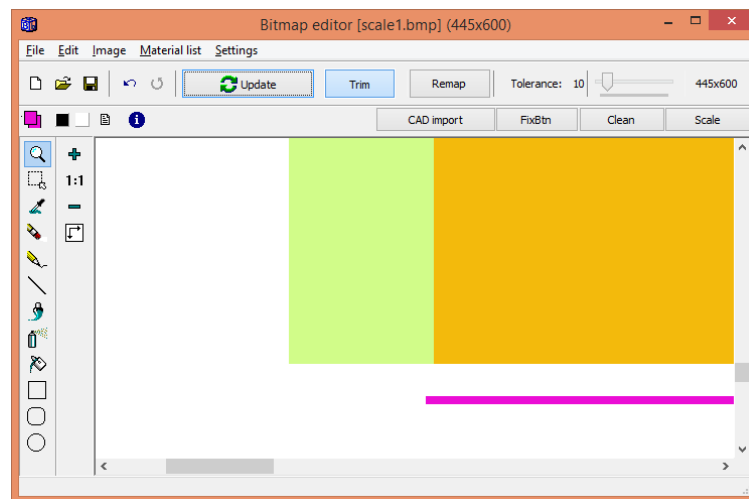


A few comments:

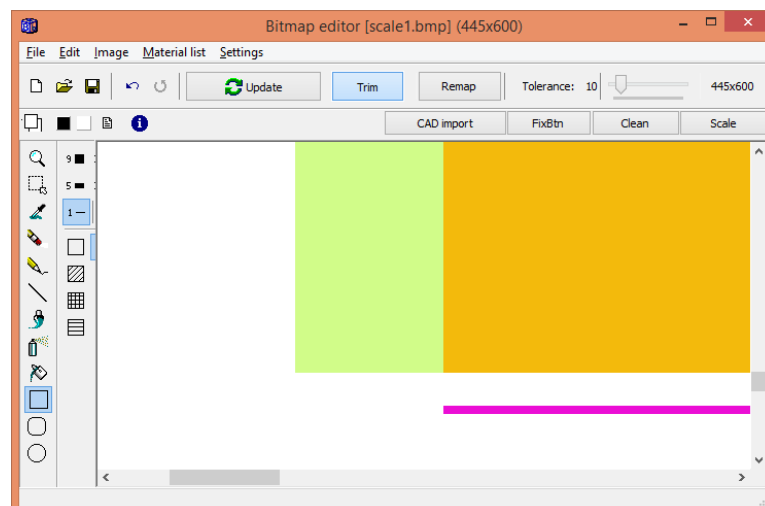
The bitmap rectangle in the pre-processor will always be adjusted to match the scale, unless “Adjust bitmap size in pre-processor” is unchecked. There may be times that you need to adjust the bitmap rectangle size yourself (e.g. if the scale is not the same in x- and y-directions).

You can set a scale factor directly (if you know this) without drawing a red marker line.

The size of the red line can easily be checked using the zoom tool. Here we adjust the line end using the eraser tool (or rectangle tool and white color):

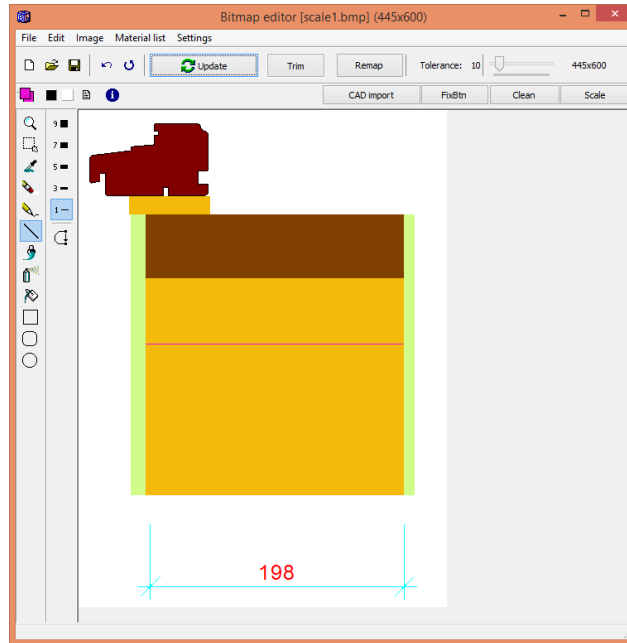


Line a bit too long.



Adjusted line (end of line erased).

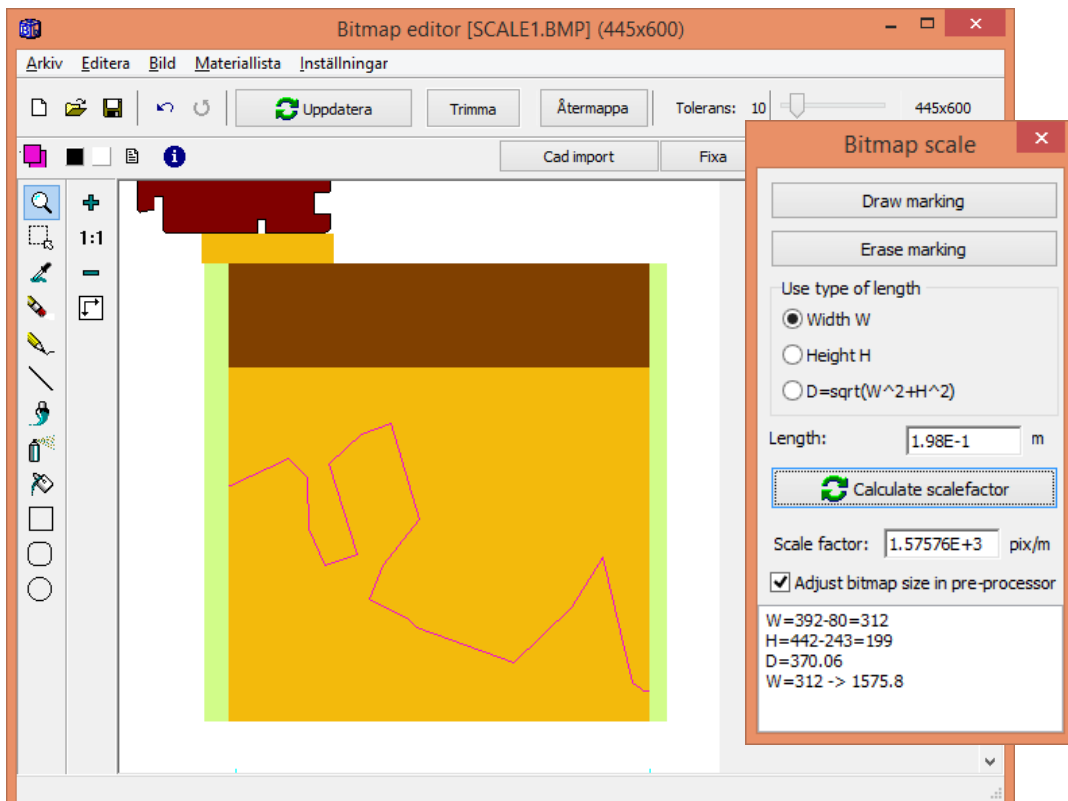
It would have been possible to draw the red scale mark line on the geometry as well, see below. However after the scale factor has been calculated (as above), the red line needs to be erased since it will not be recognized as a material upon updating. Erase the line using the undo-arrow in the editor or press button “Erase marking”.



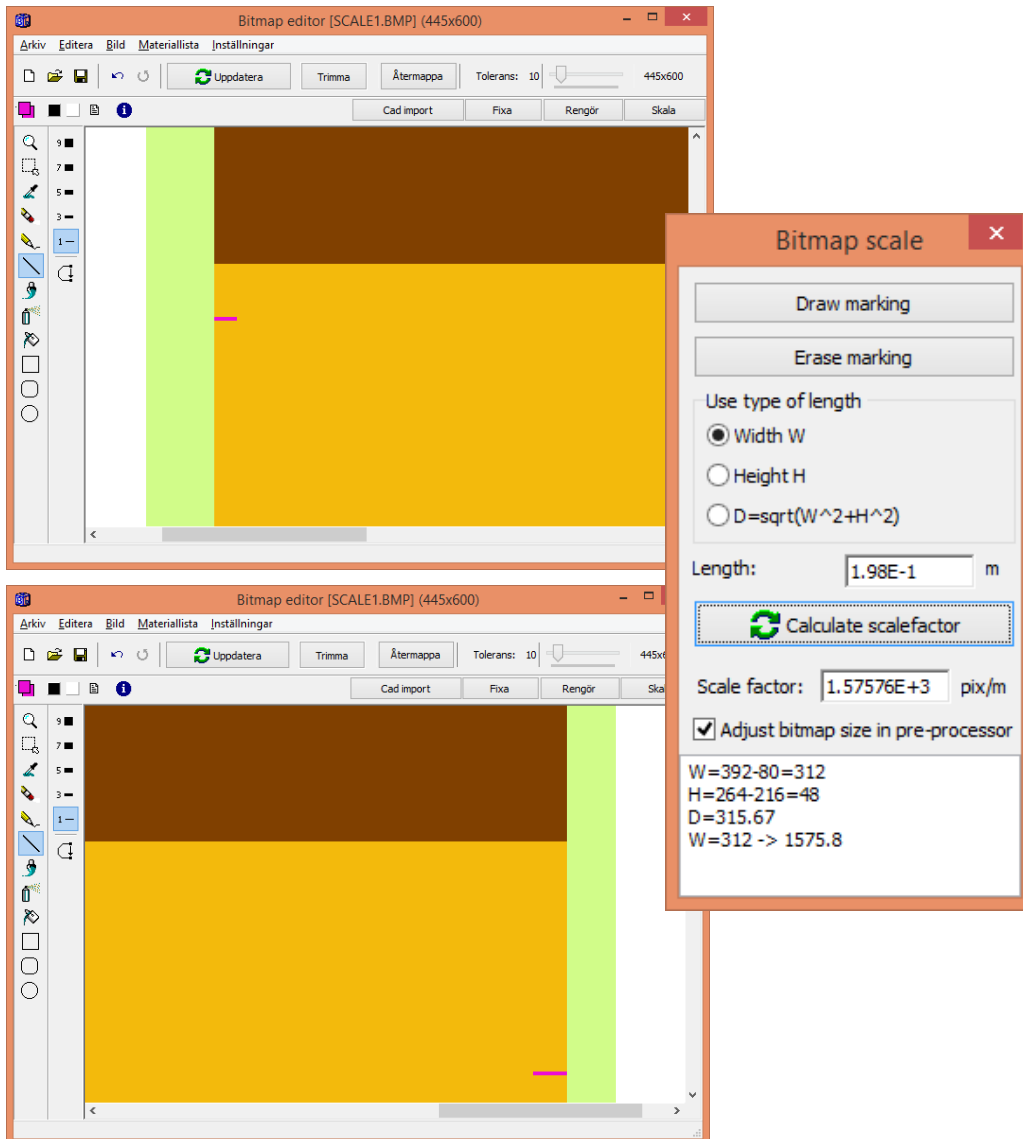
Red marker line needs to be erased after scale factor is calculated.

Type of length

The width W is just the extension for the red color in the x -direction ($\max_x - \min_x$), and the height H the extension in y -dir ($\max_y - \min_y$). The distance D is calculated as $D = \sqrt{W^2 + H^2}$. See example below where $W=312$, $H=199$, $D=370.06$. In this case we need to use the width W to calculate the scale factor correctly. The distance D is typically used to calculate the scale factor for a sloped measurement (not vertical or horizontal), but can also be used for horizontal lines ($D=W$) or vertical lines ($D=H$).

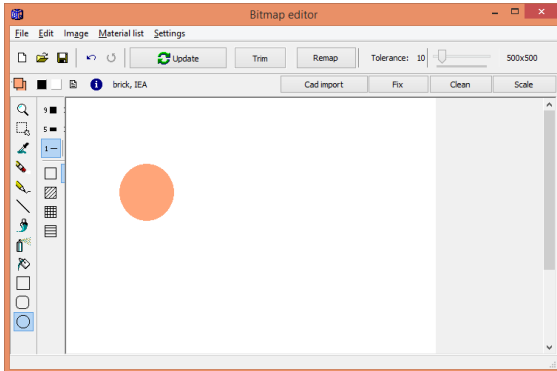


The scale marking does not have to be made using one line. You can put two lines with the red marker color (or even two pixels). A tip is to zoom in the area for better precision. Below we have two areas giving $W=312$ which is used to calculate the scale factor.

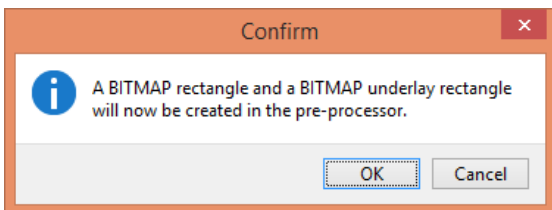


3.3 Bitmap rectangles and bitmap underlays

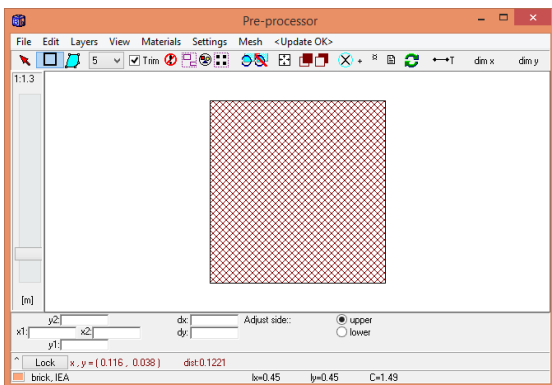
The bitmap rectangle in the pre-processor that maps the geometry from the Bitmap Editor is now automatically created (if it is not already is drawn). See also the update manual for HEAT2 v8.



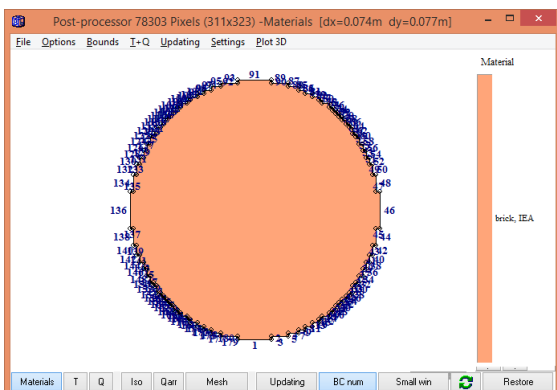
Circle drawn in the Bitmap Editor



Message when pressing "Update"

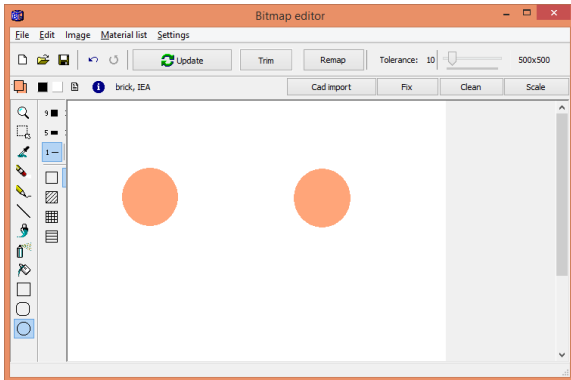


Created bitmap rectangle in the pre-processor

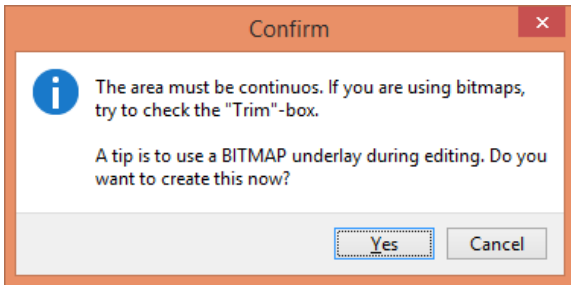


Final geometry as shown in the post-processor

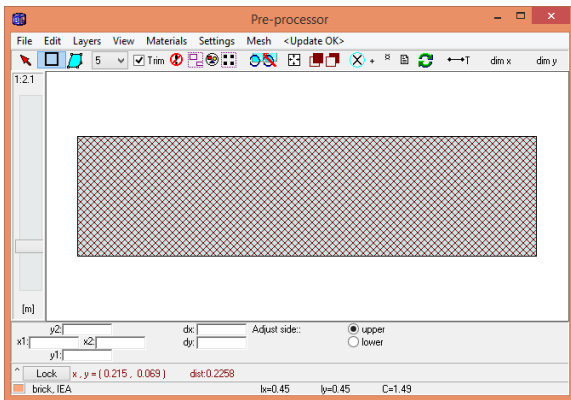
A **bitmap underlay** will also be created if the area is not continuous. The reason for this is that the final geometry (created by pressing “Update”) will not be valid if the area is not continuous and we cannot check the result in the post-processor. Using a bitmap underlay will allow us to edit and see the result in the post-processor during all steps in the input process.



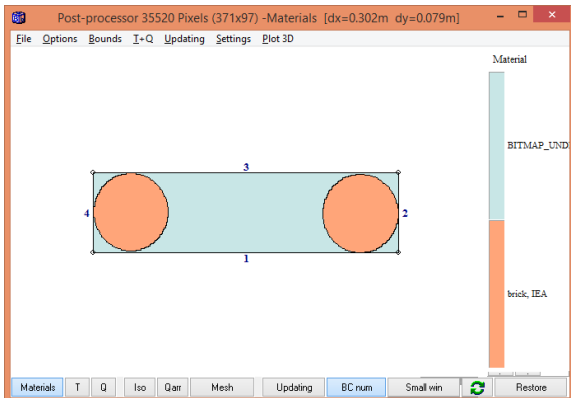
Circle added in the Bitmap Editor



Message when pressing "Update"

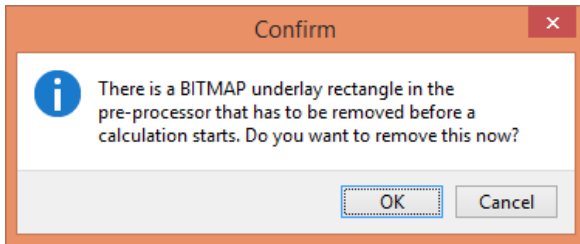


An underlay is created beneath the bitmap



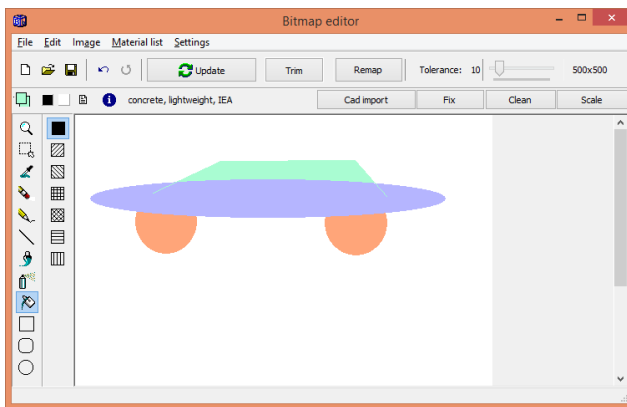
Result as shown in the post-processor

The underlay is just a dummy material in light-blue color. You can of course use any other material but the name BITMAP_UNDERLAY is reserved and not allowed when a calculation starts. The following message appears if we try to start a calculation for the geometry:

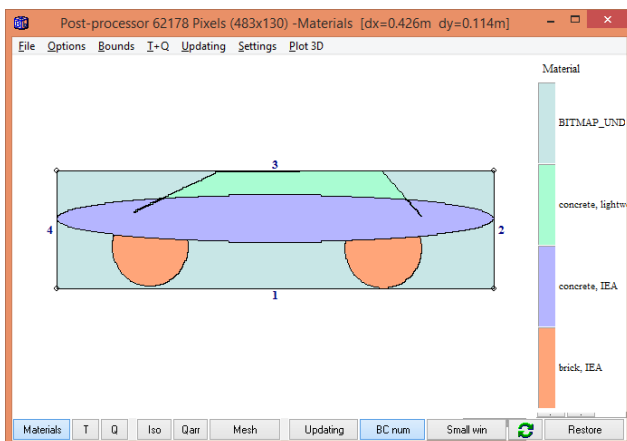


If we choose to remove it, we still cannot start a simulation (or even update the geometry) because the area is not continuous. When pressing "Update" we will get the earlier message about creating a new underlay, see above.

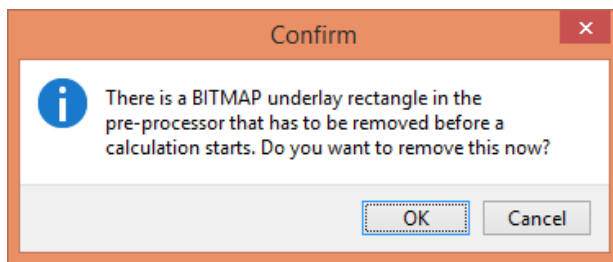
We continue with our project in the bitmap editor. Now the area will be continuous:



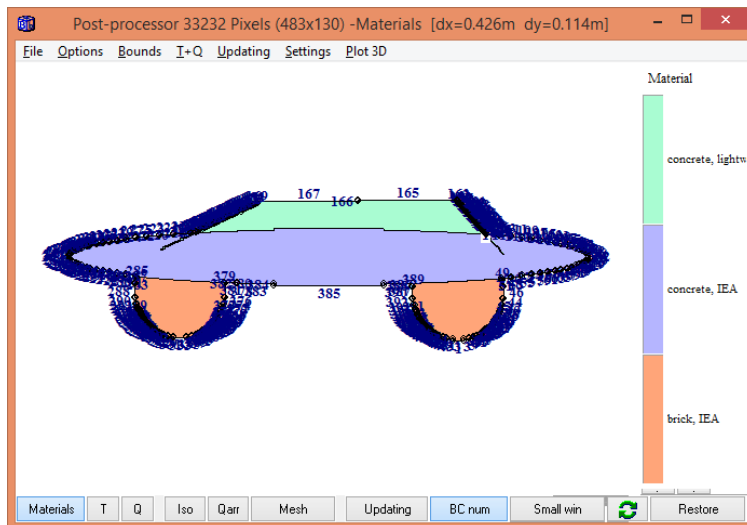
Updating the geometry gives this result (still with the bitmap underlay):



Now we should remove the bitmap underlay, by hand or by trying to start simulation:

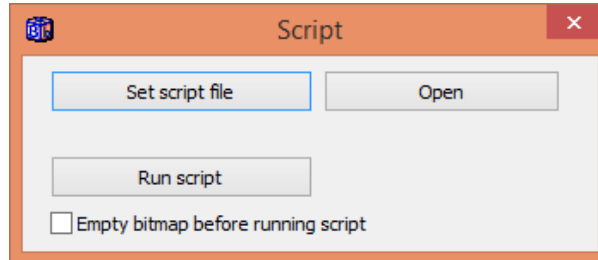


Press OK. After updating again, we get the final geometry as shown in the post-processor below. We can now see the final boundary numbers allowing boundary conditions to be set.



3.4 Drawing geometry using a script

The geometry in a bitmap can now be specified using a script file, see menu item “Edit/Script” in the bitmap editor. A script file can be opened and edited in e.g. Notepad. Running the script will draw all entities in the bitmap editor. If “Empty bitmap before running the script” is checked, the existing bitmap will be erased before the geometry of the script is added.



A script file is an ordinary text file (*.txt) containing commands for drawing:

L	x1 y1 x2 y2...xn yn	name	! draws line segments
E	x1 y1 x2 y2	name	! draws an ellipse
R	x1 y1 x2 y2	name	! draws an rectangle
M	x1 y1 x2 y2...xn yn	name	! draws a closed polygon
N	x1 y1 x2 y2...xn yn	name	! draws Bezier fitted lines
P	x1 y1 x2 y2 x3 y3 x4 y4	name	! draws a pie
A	x1 y1 x2 y2 x3 y3 x4 y4	name	! draws an arc
X	x1 y1	name	! draws a pixel
Y	x1 y1	name	! fills region bounded by black
Z	x1 y1	name	! fills region bounded by material

Comments begins with “!”. The optional “name” refers to the material name. The color of the material is used for both pen and brush. Make sure that the correct material file (*.mtl) is active in HEAT2. If the name does not exist, or no name is given, the drawing color will be black (and the ellipse, rectangle, and polygon will not be filled).

A few comments:

M: draws a set of connected beziers with the end point of the first curve being the start point of the next and so on. So the number of points in the array should be $4 + (n*3)$ where the number of curves required is $n-1$.

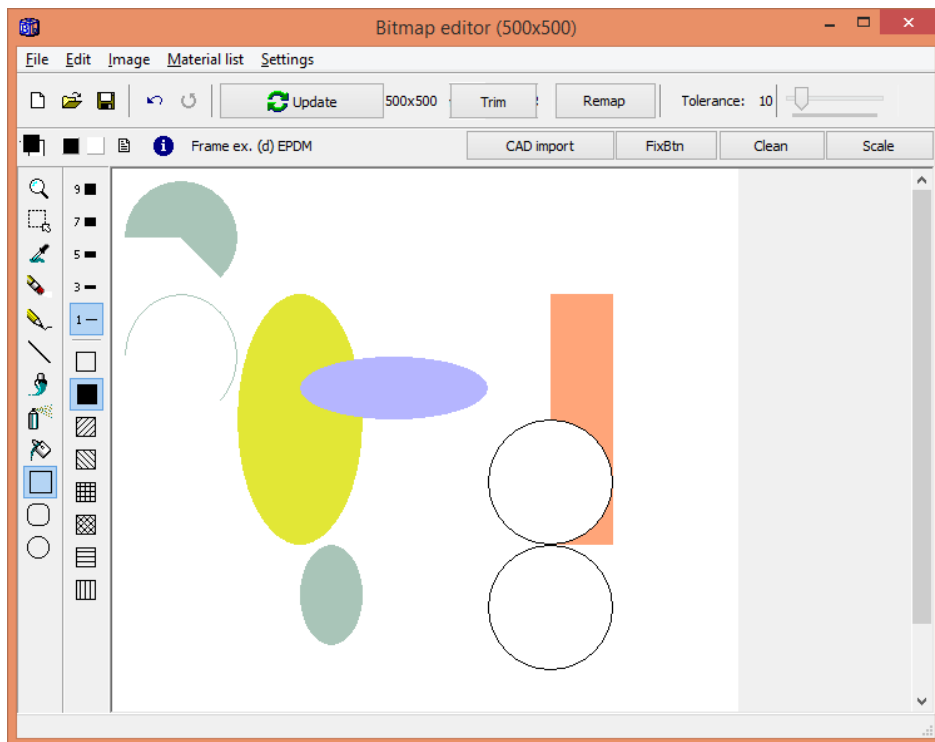
P and **A**: The wedge/arc is defined by the ellipse bounded by the rectangle determined by the points (X1, Y1) and (X2, Y2). The section drawn is determined by two lines radiating from the center of the ellipse through the points (X3, Y3) and (X4, Y4).

Y: The boundaries of the region to be filled are determined by moving outward from the point (x1,y1) until a black boundary is encountered.

Z: As Y except that the boundary color should be the same as the filling color (the material color).

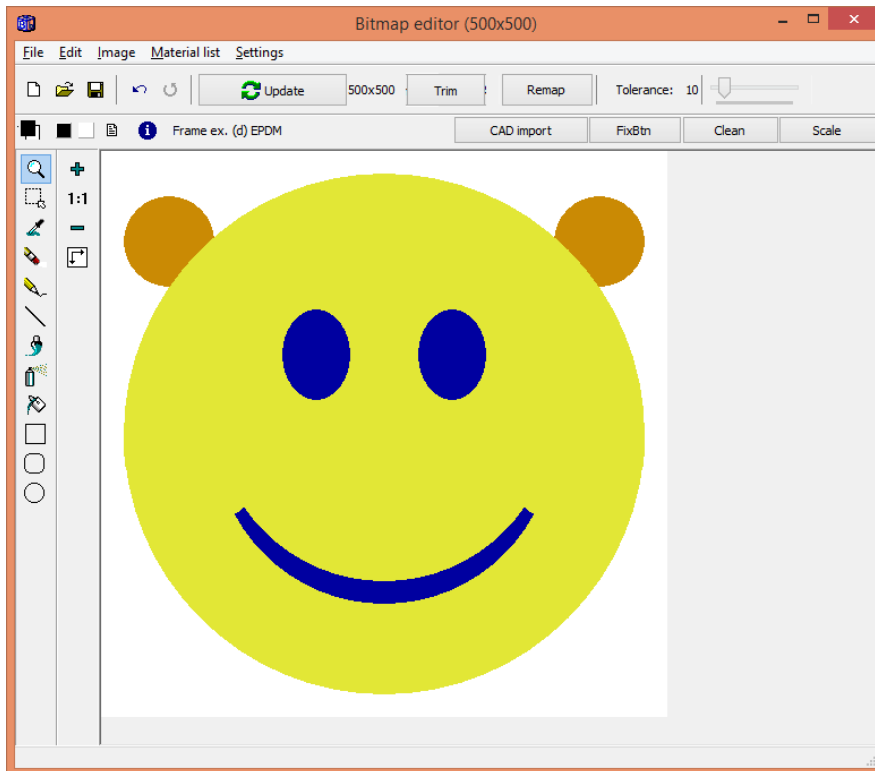
Example "Script_1.txt"

r 350 100 400 300		brick, IEA
p 10 10 100 100	100 100 10 55	Frame ex. (d) EPDM
a 10 100 100 200	100 200 10 150	Frame ex. (d) EPDM
e 100 100 200 300		cork, IEA
e 150 150 300 200		concrete, IEA
e 300 300 400 400	xxx	! material name does not exist => black color
e 300 200 400 300		! no material name gives black color
e 200 300 150 380		Frame ex. (d) EPDM



Example "Script_smiley.txt"

e 20 40 100 120 gold	! left ear
e 400 40 480 120 gold	! right ear
e 20 20 480 480 cork, IEA	! big yellow circle
e 100 100 400 400 Example 1, concrete	! blue circle mouth
e 100 80 400 380 cork, IEA	! yellow circle mouth
e 20 200 150 330 cork, IEA	! left end mouth
e 350 200 480 330 cork, IEA	! right end mouth
e 160 140 220 220 Example 1, concrete	! left eye
e 280 140 340 220 Example 1, concrete	! right eye



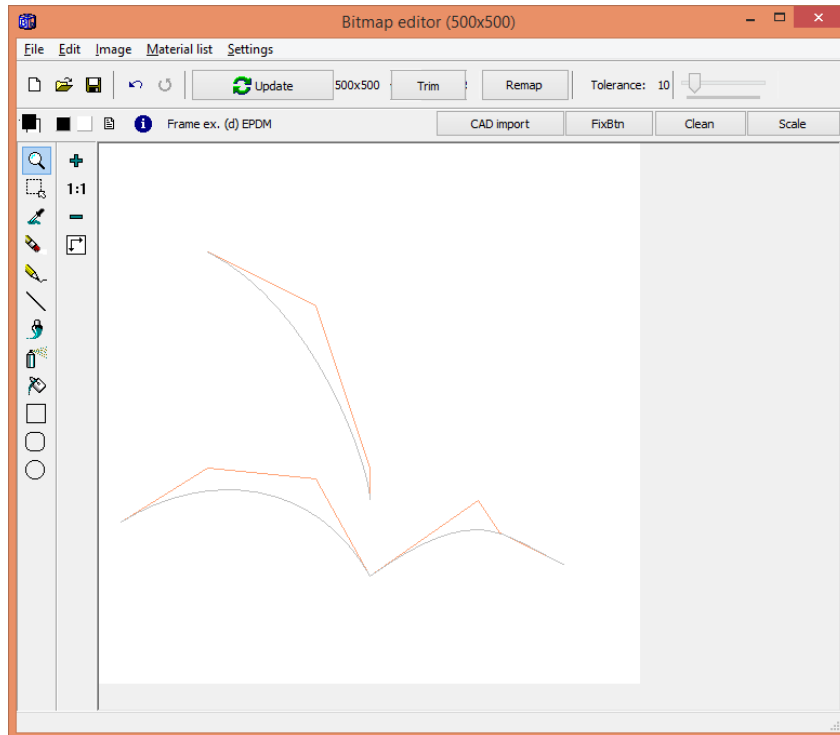
Example "Script_Bezier.txt"

L 100 100 200 150 250 300 250 330 brick, IEA

N 100 100 200 150 250 300 250 330 aluminum

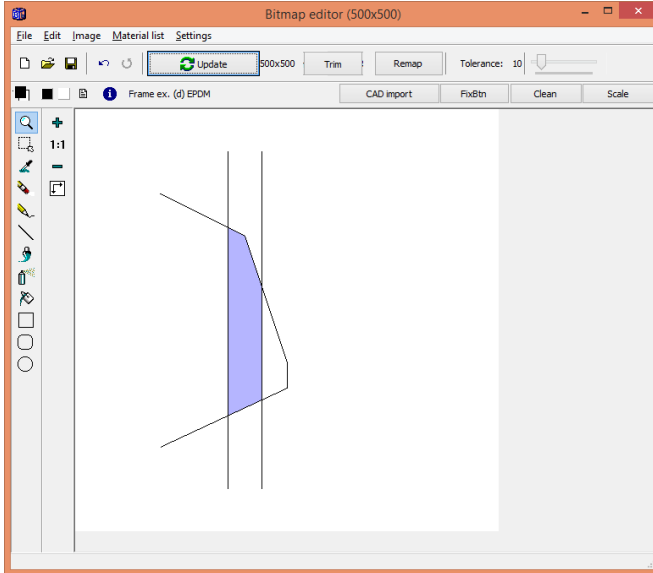
L 20 350 100 300 200 310 250 400 350 330 370 360 430 390 brick, IEA

N 20 350 100 300 200 310 250 400 350 330 370 360 430 390 aluminum



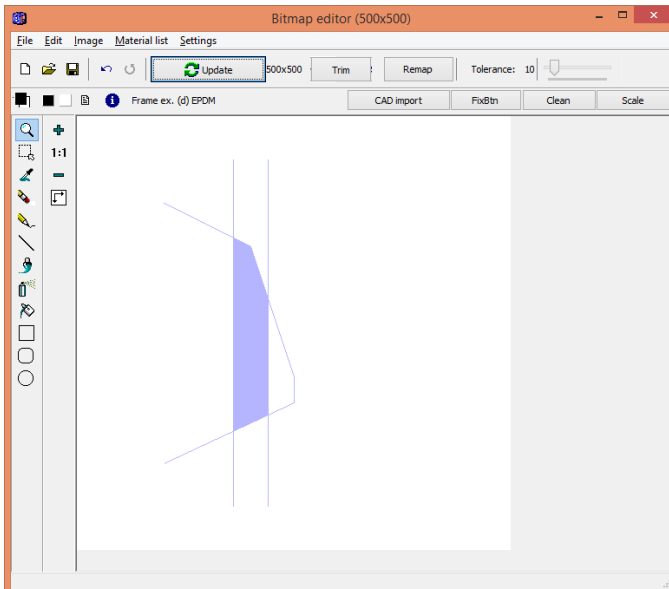
Example "Script_fill_1.txt"

L 100 100 200 150 250 300 250 330 100 400
L 180 50 180 450
L 220 50 220 450
Y 200 230 concrete, IEA



Example "Script_fill_2.txt"

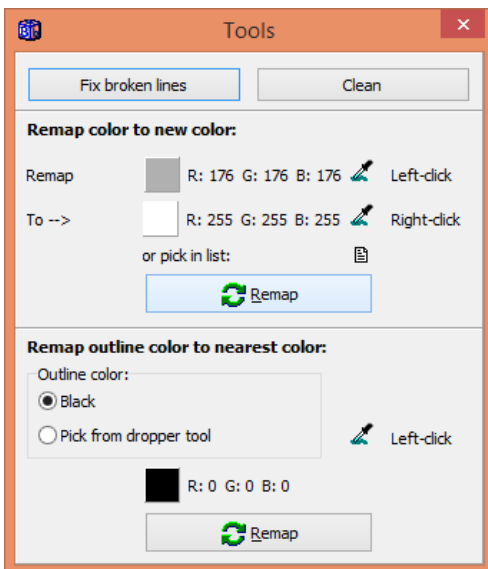
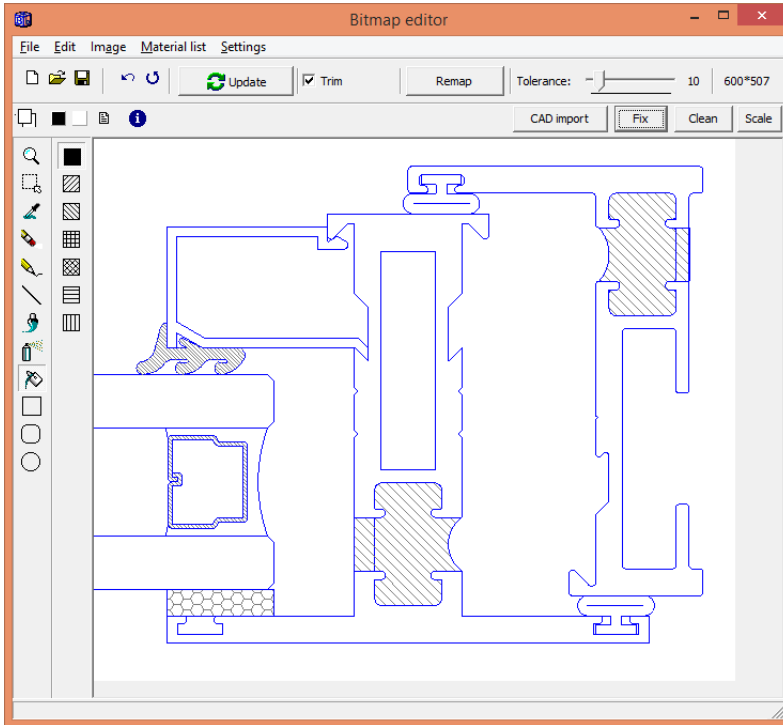
L 100 100 200 150 250 300 250 330 100 400 concrete, IEA
L 180 50 180 450 concrete, IEA
L 220 50 220 450 concrete, IEA
Z 200 230 concrete, IEA



3.5 Tips

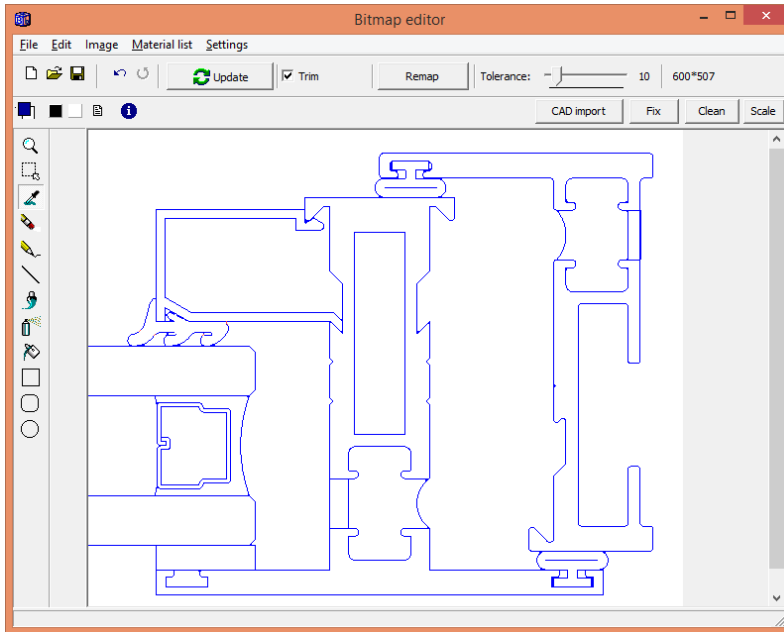
Areas with shading/hatches/etc.

Always try to use a “clean” picture/cad file without shading, etc. If a certain layer cannot be turned off in the CAD editor, you can try to remap the color to e.g. white using the remap tool:

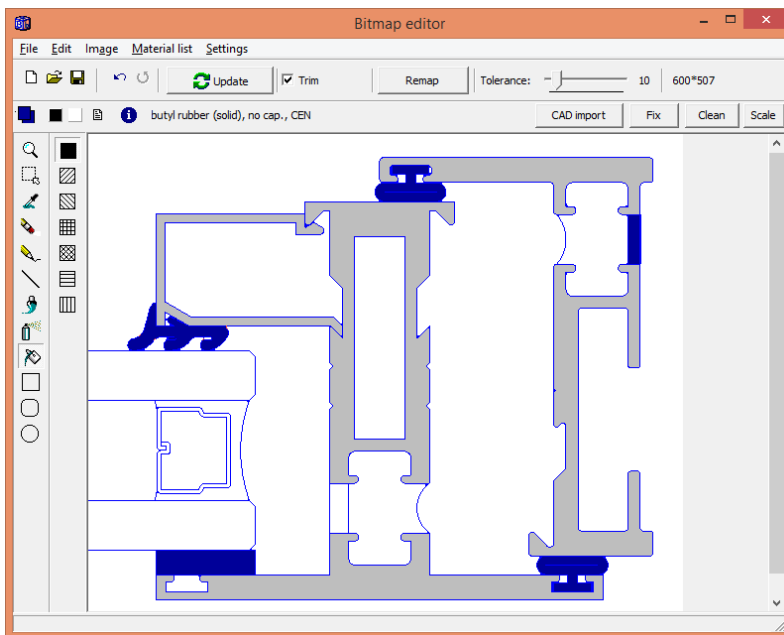


Note that the tolerance (using the track-bar) can be set for the source color (a higher tolerance will remap “nearby” colors of the original color to the destination color).

We now have a cleaner image:

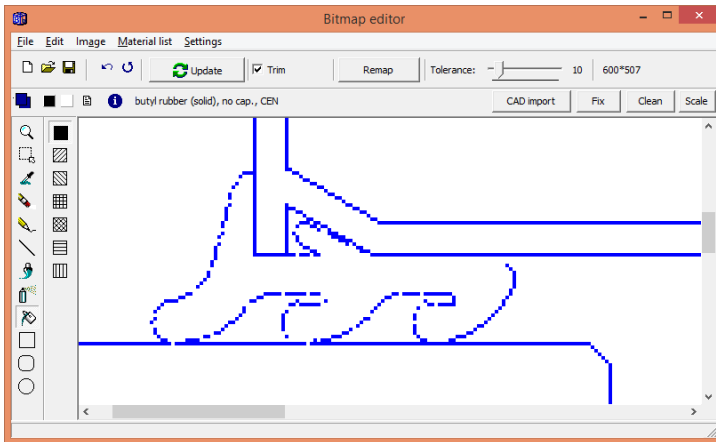


It's now easy to use the Fill tool:

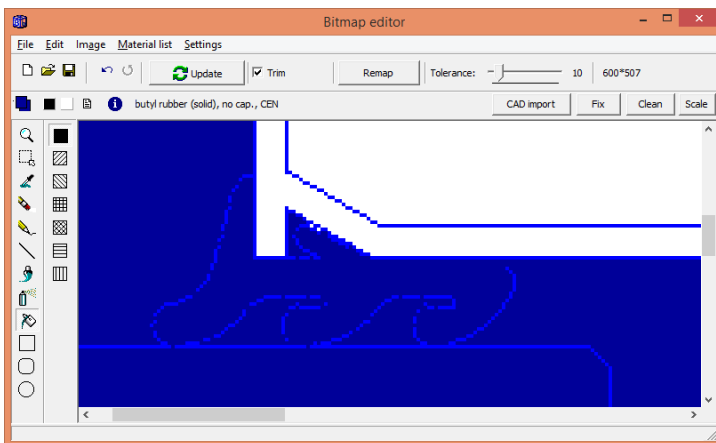


Broken lines

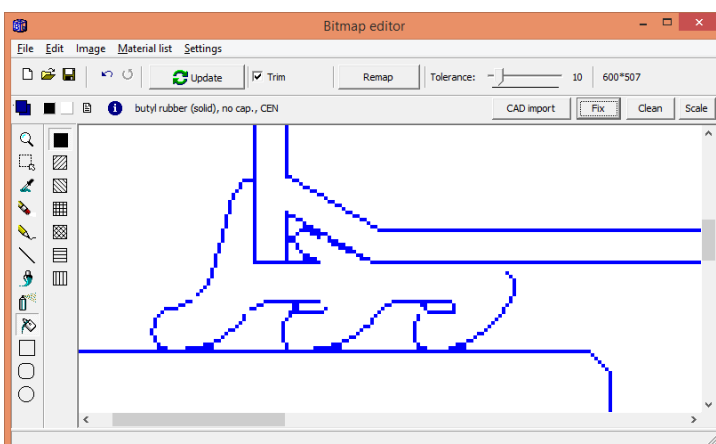
If the picture/CAD file has thin lines there may be a need to fix broken lines in order to use the fill tool:



This sealing has about 25 holes.

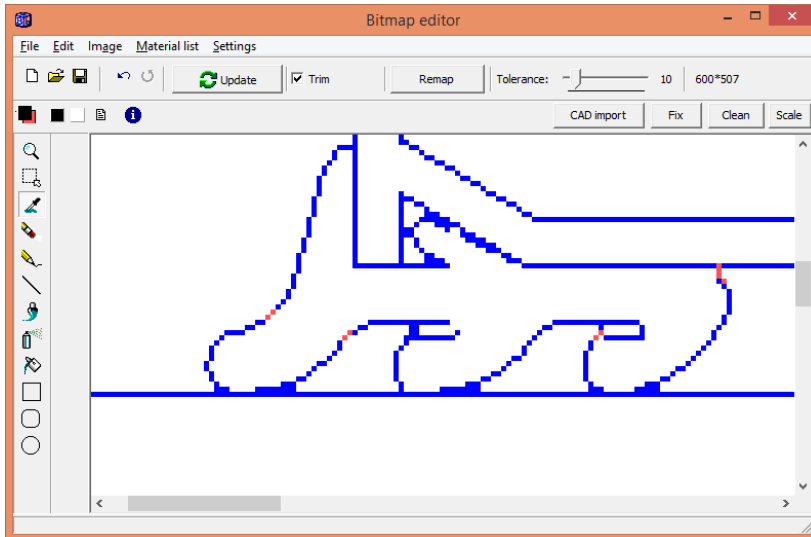


Result with leakage using the fill tool (not good).

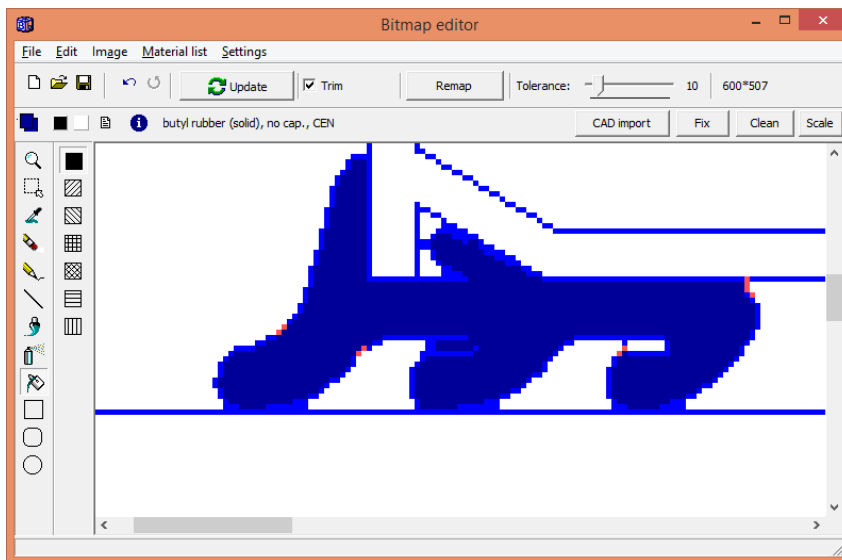


“Fix broken lines” applied.

The “Fix broken lines” tool binds together most part of the boundary line. There are still four holes that need to be fixed manually.



Use any color (usually black or the same material) to fix the line (here in red to illustrate the four holes).

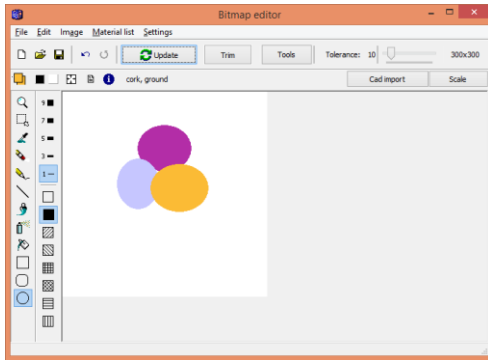


The filling tool can now fill properly with a material.

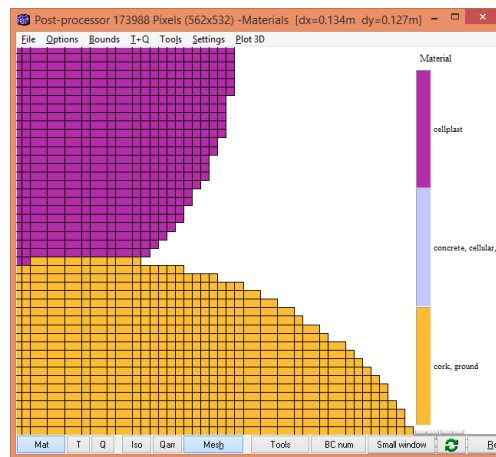
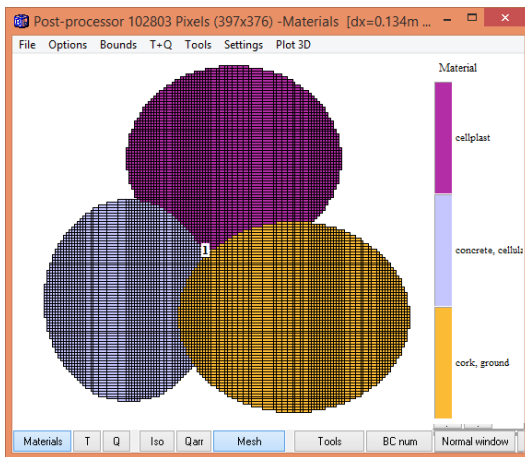
Bitmap size affects the numerical mesh and number of boundaries

A larger bitmap size will lead to that more numerical nodes will be used. Try to avoid too large bitmap size since this will increase the calculation time, especially for transient cases. You should on the other hand of course check that you have enough nodes to get a sufficiently small numerical error for your problem (see e.g. EN ISO 10211). Consider the two cases below with similar geometry where we illustrate the number of nodes that are required.

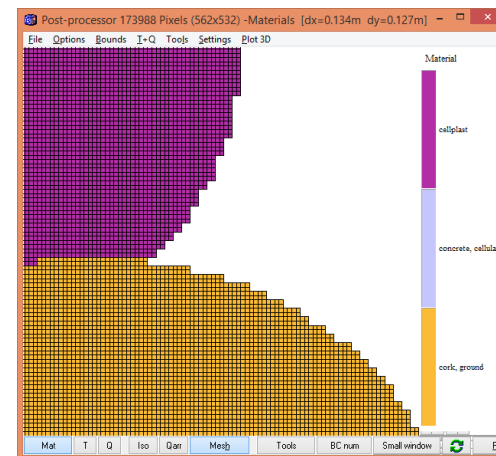
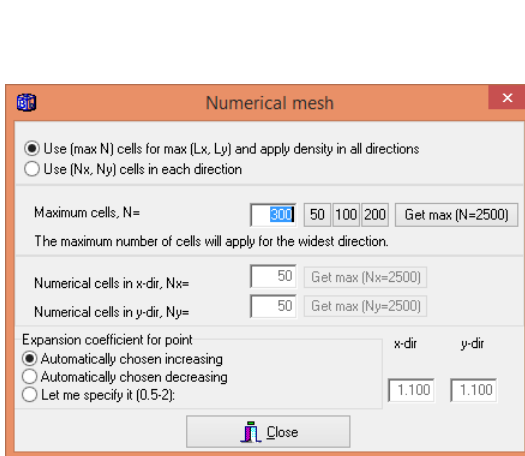
The first case is drawn on a 300x300 large bitmap canvas:



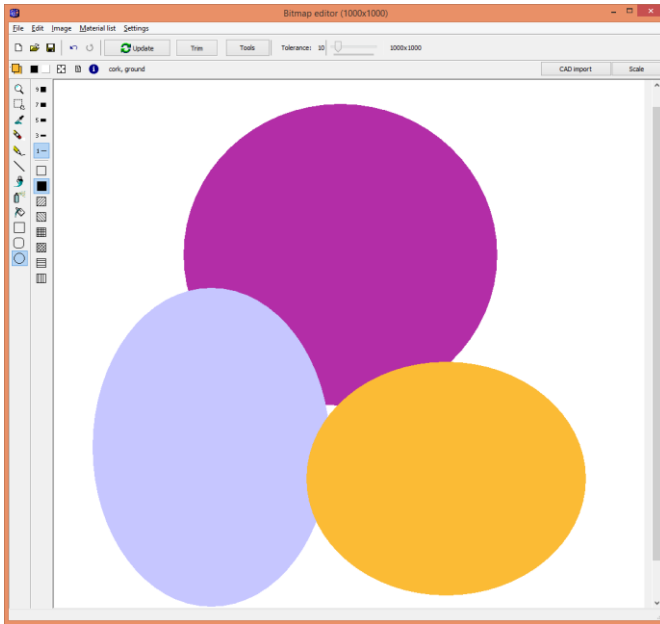
The resulting smallest mesh will have about 8000 nodes and 346 boundaries:



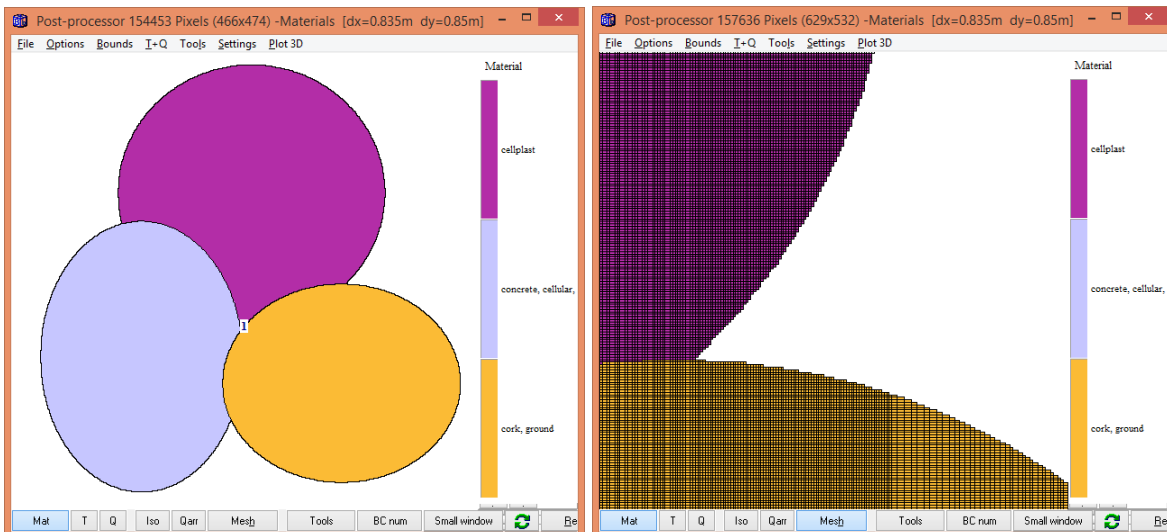
Note that we can change the mesh in the mesh dialog (menu item “Pre-processor/Mesh”). We will still have 346 boundaries:



The second case is drawn on a 1,000x1,000 large bitmap canvas:



The resulting smallest mesh will have about 400,000 nodes and 2,300 boundaries:



4. Import of CAD-files

4.1 Introduction

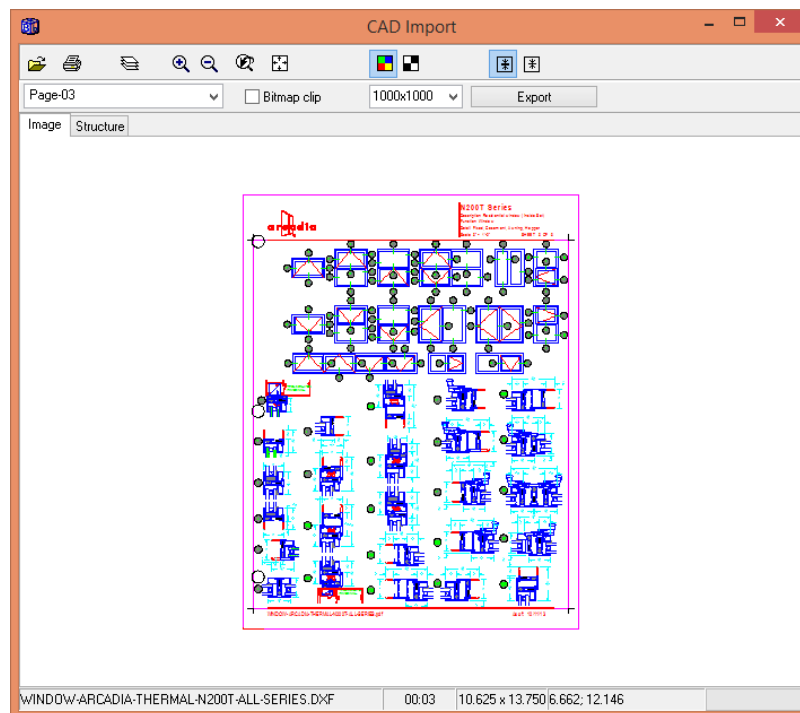
HEAT2 v9 can import DWG, DXF, SVG, PLT, CGM, WMF, JPG, BMP, PNG, and some other formats. Here is a short description:

- **DXF** (Drawing Interchange Format, or Drawing Exchange Format) is a CAD data file format developed by Autodesk[1] for enabling data interoperability between AutoCAD and other programs.
- **DWG** (from drawing) is a binary file format used for storing two- and three- dimensional design data and metadata. It is the native format for several CAD packages including DraftSight, AutoCAD, IntelliCAD (and its variants) and Caddie.
- **Computer Graphics Metafile (CGM)** is a free and open international standard file format for 2D vector graphics, raster graphics, and text, and is defined by ISO/IEC 8632
- **Scalable Vector Graphics (SVG)**: is an XML-based vector image format for two-dimensional graphics with support for interactivity and animation. The SVG specification is an open standard developed by the World Wide Web Consortium (W3C) since 1999.

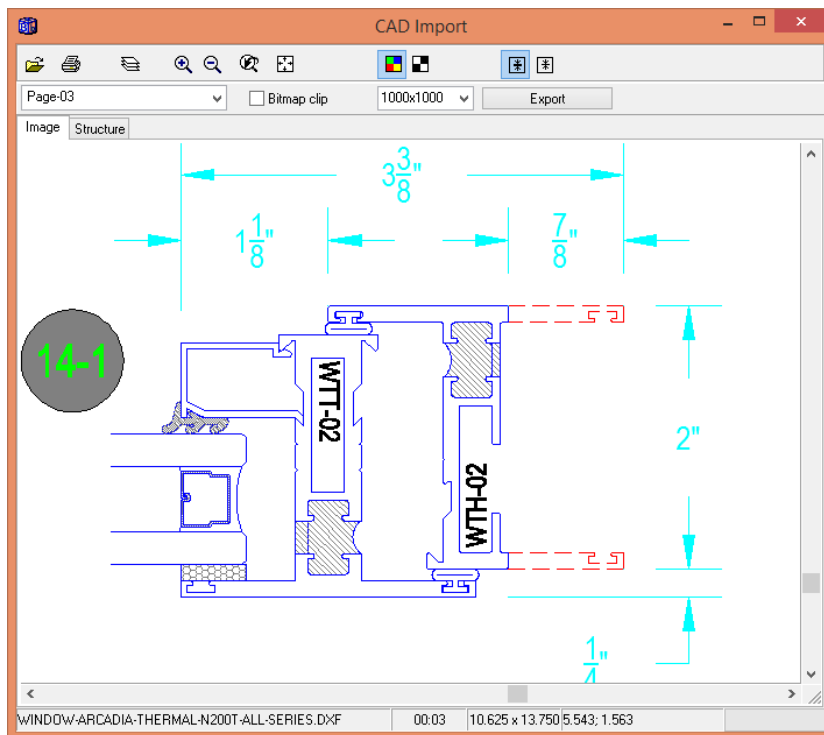
Note that Google Sketchup Pro can export dxf-files.

4.2 Introduction

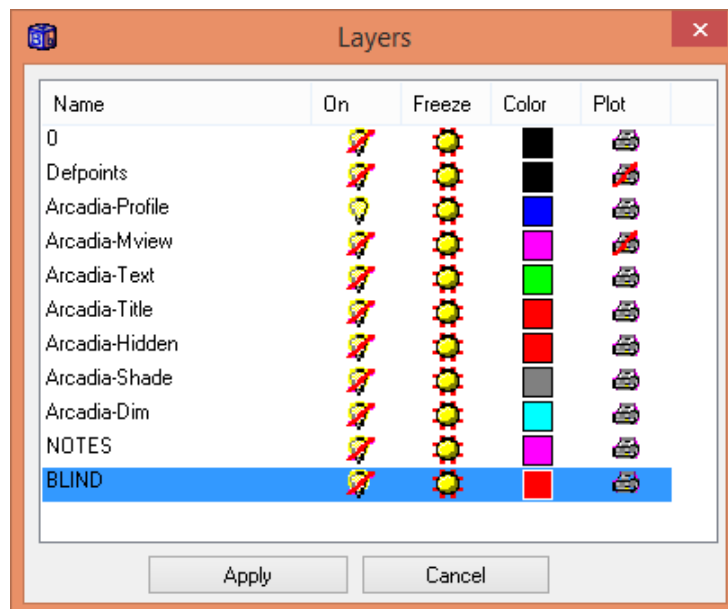
The CAD import window is reached from the bitmap editor (button “Cad Import”). The figure below shows a page with window frames from an imported dxf-file.



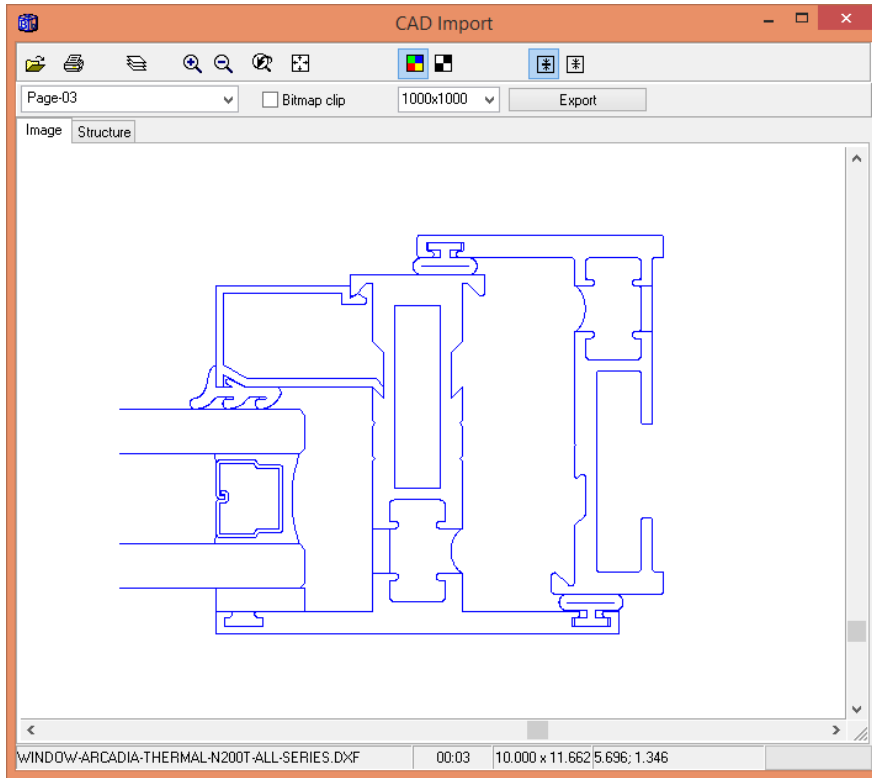
There are basic functionalities regarding zooming and panning. The figure below shows a zoomed detail.



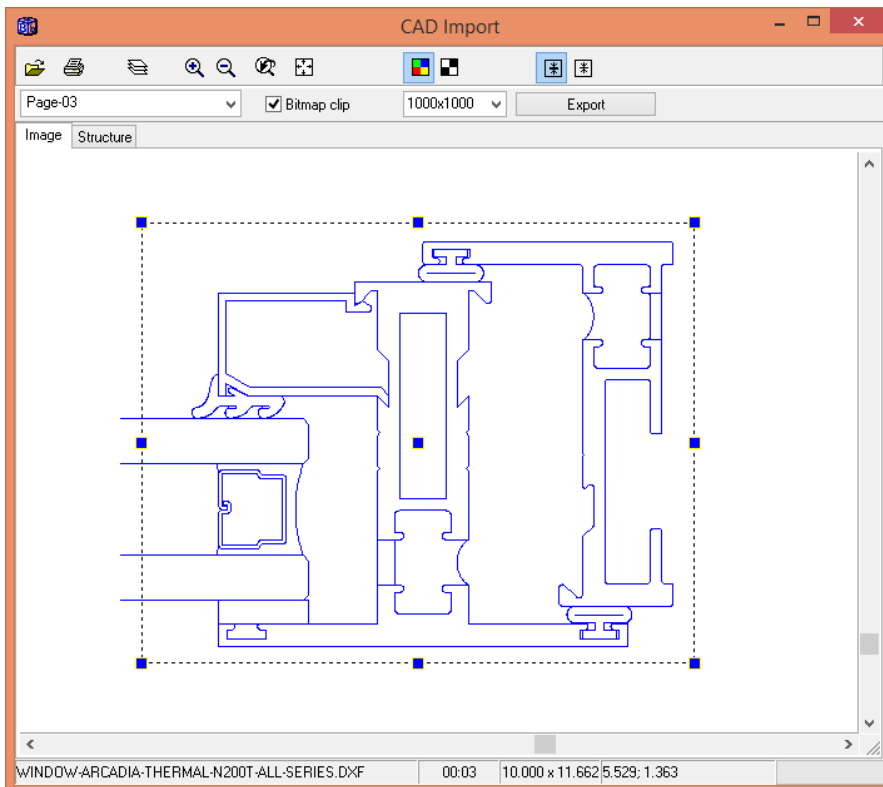
There is a layer manager that allows different layers to be visible. The color can be changed for any layer.



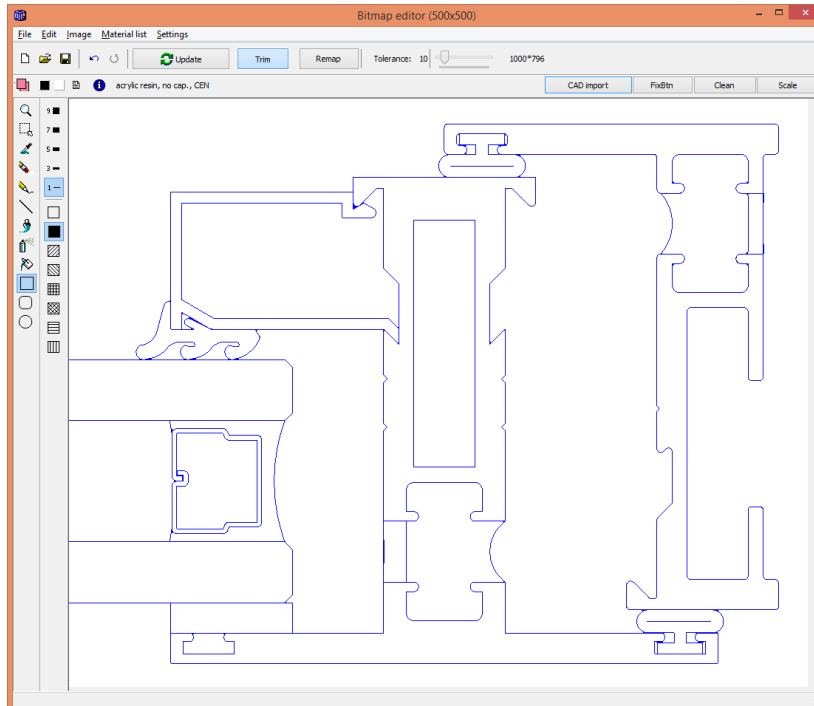
The following picture is now suited to be imported in the bitmap editor.



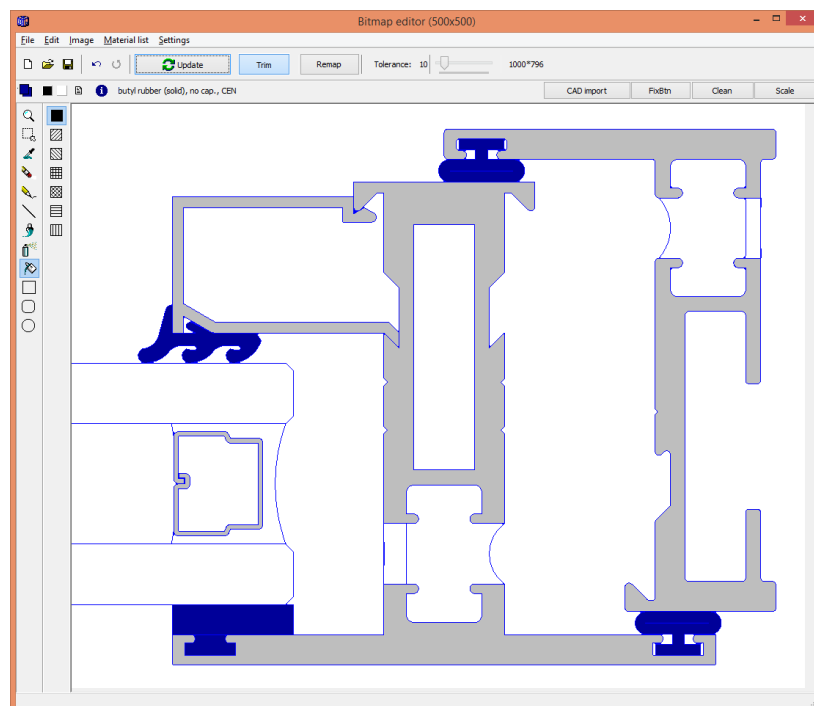
The final window to be exported to the bitmap editor is chosen using a bitmap clip (if no clip is used the whole window area will be exported). The size of the bitmap is chosen to be max 1000x1000 pixels.



Imported bitmap in the bitmap editor (final size is 1000x796 pixels):

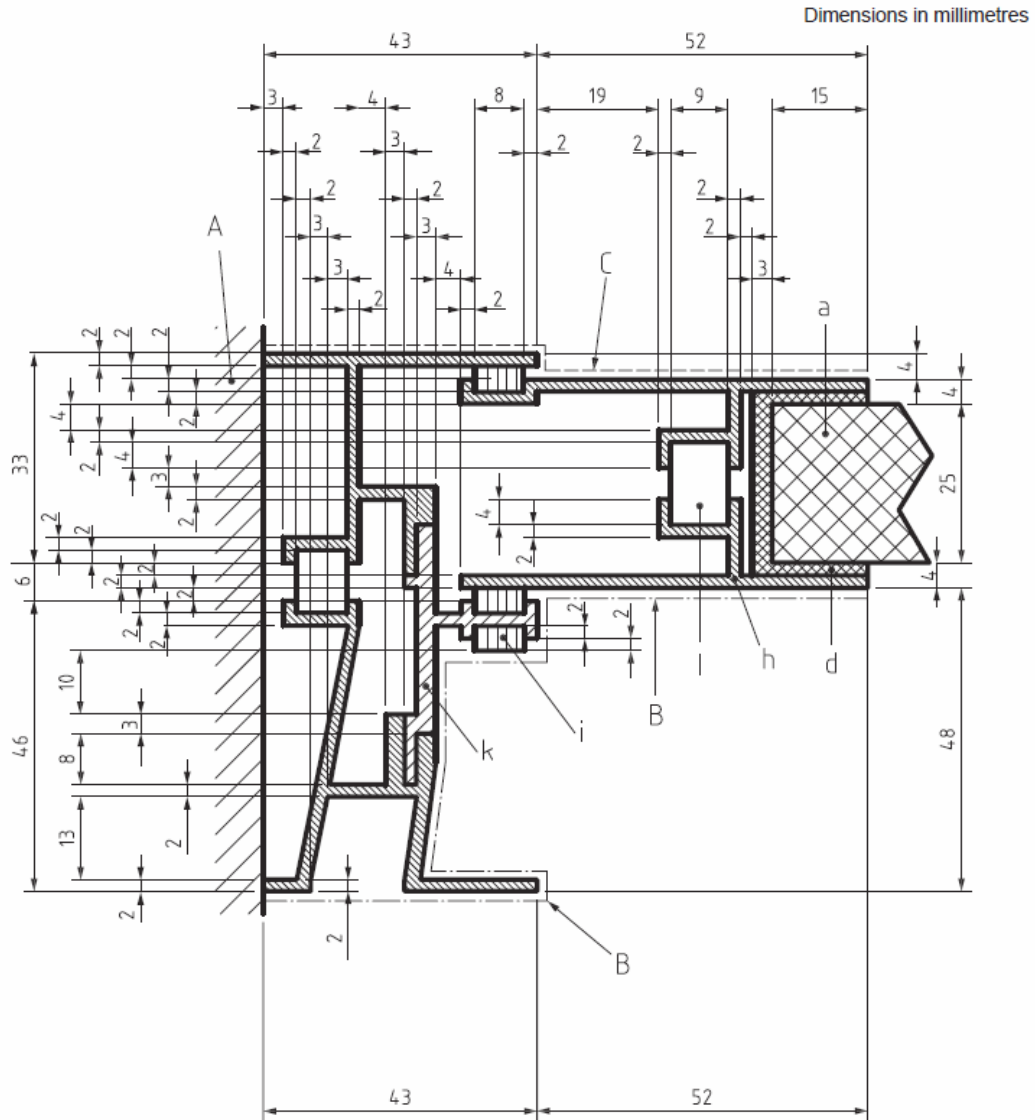


With a few clicks we have now generated the sealings and aluminium profile using the fill tool.



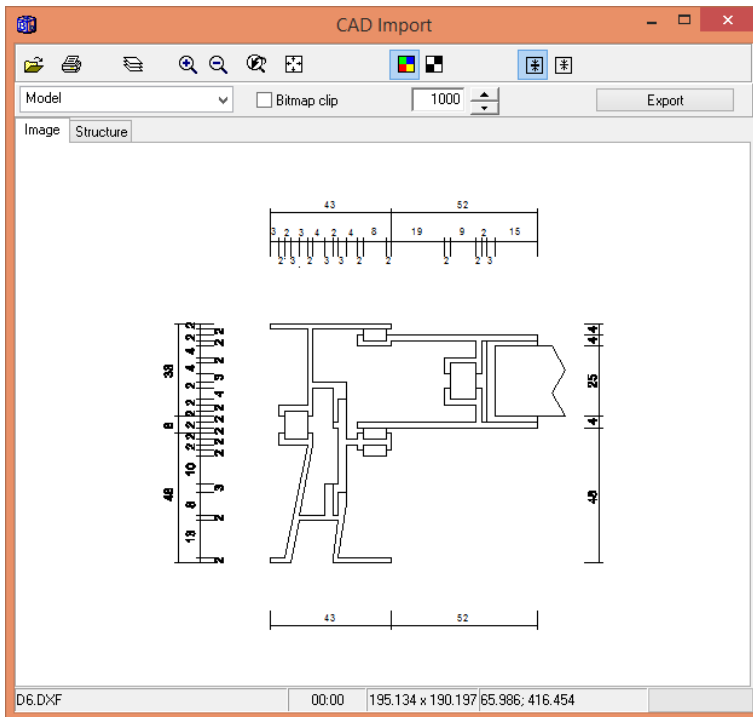
4.3 Example

Consider the test case D6 from EN ISO 10077-2 concerning a calculation for a window frame. The time to input and solve this case should not take more than a few minutes for an experienced user. All steps are shown below. See example **d6_example.dat**.

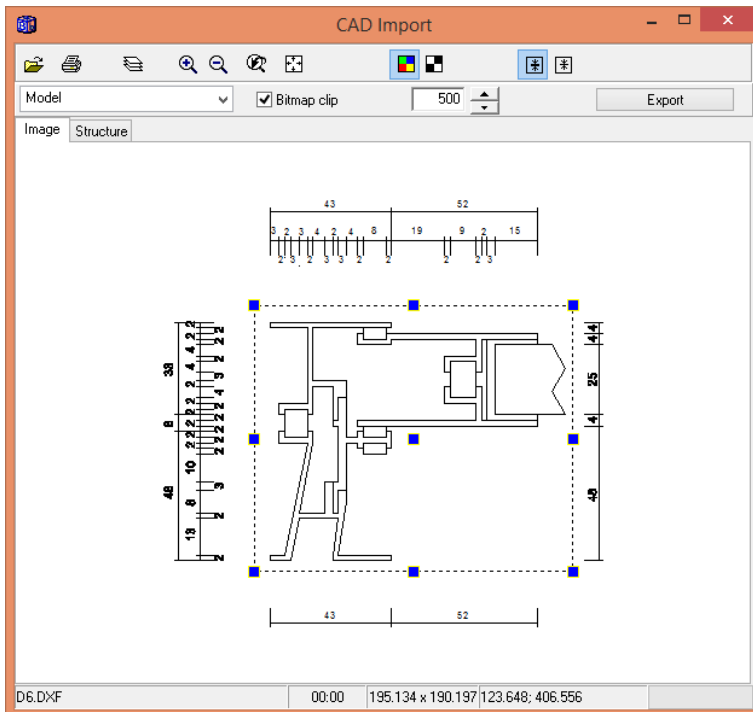


Sliding window frame section and insulation panel (bf = 95 mm)

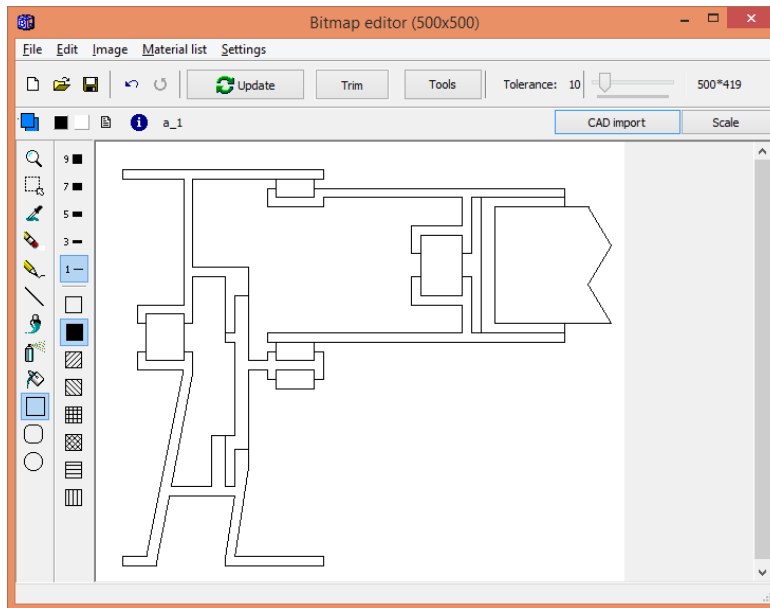
First, open the file d6.dxf:



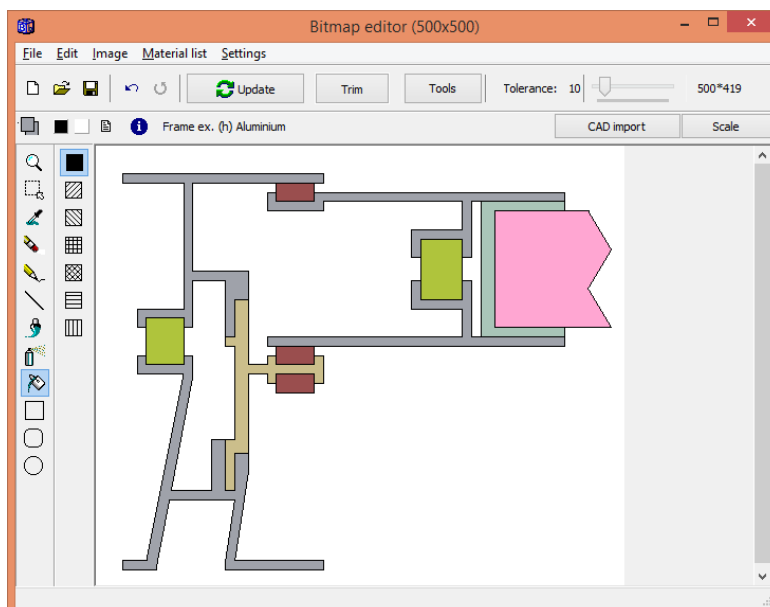
Check "Bitmap clip" and draw a clip rectangle around the area we want to export. Choose size 500 and press Export:



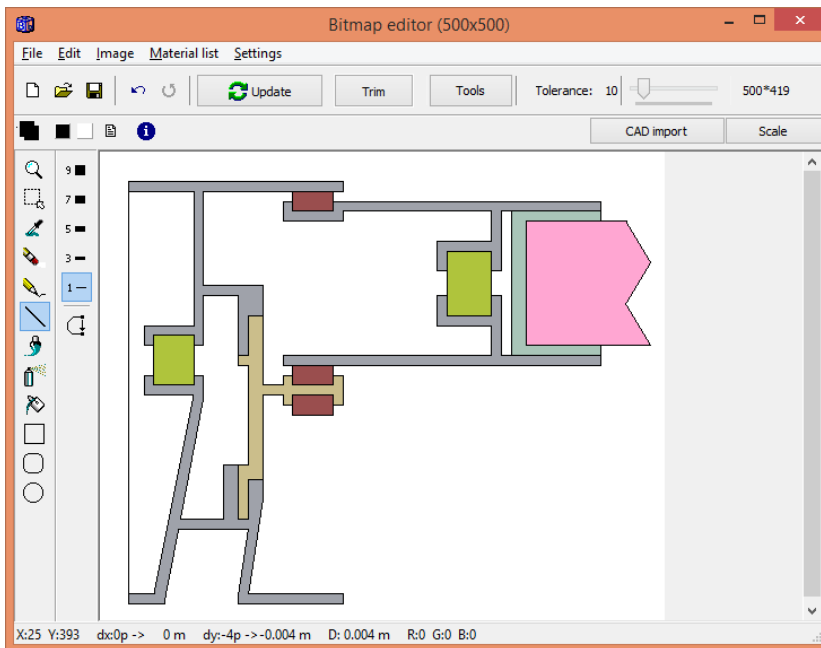
Exported bitmap to the Bitmap Editor:



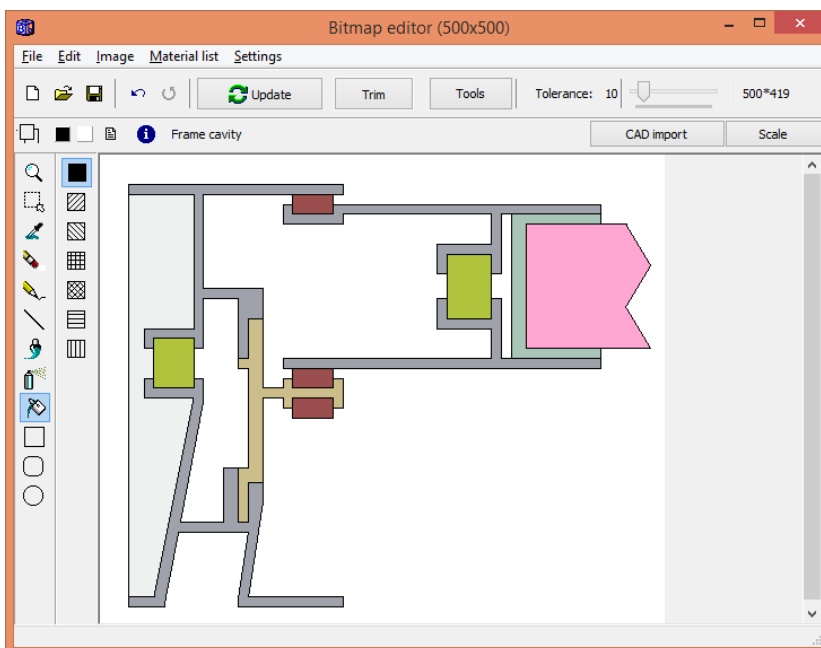
Use the Fill tool and fill the areas with materials from the pick list (12 clicks are needed):



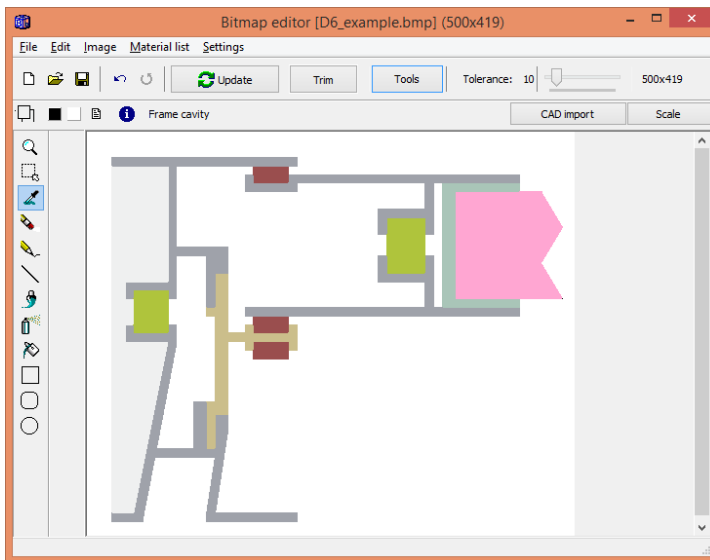
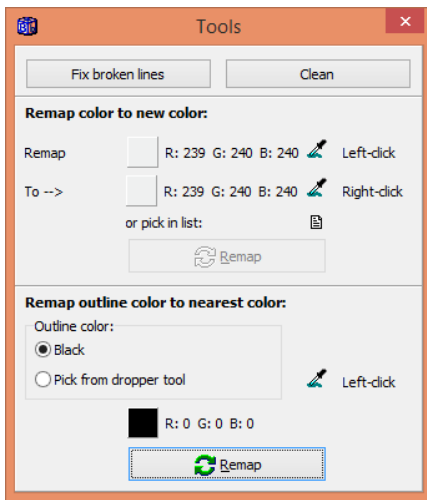
We need to draw a vertical line to the left in order to enclose a frame cavity. Pick the line tool, choose black color (from the upper tool bar) and draw a line (the start point should be where the black outline starts (RGB colors should be 0 at the lower toolbar)).



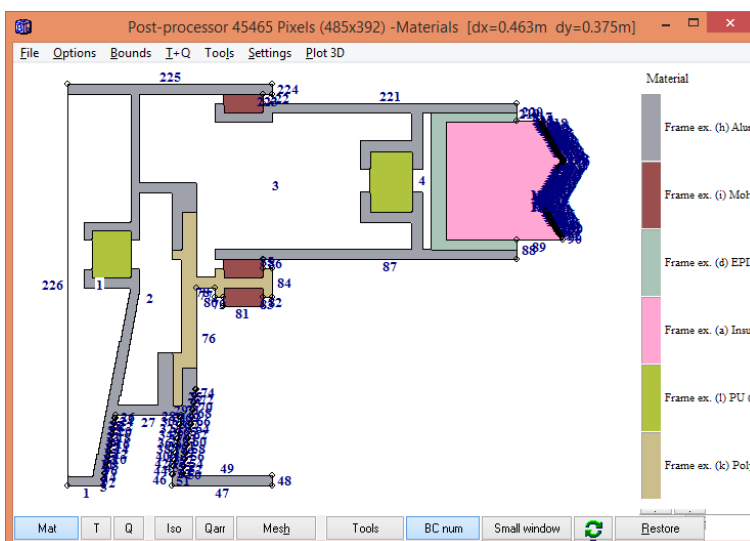
Use the Fill tool with material “Frame cavity” and fill the left area (all other enclosures will be automatically recognised as frame cavities):



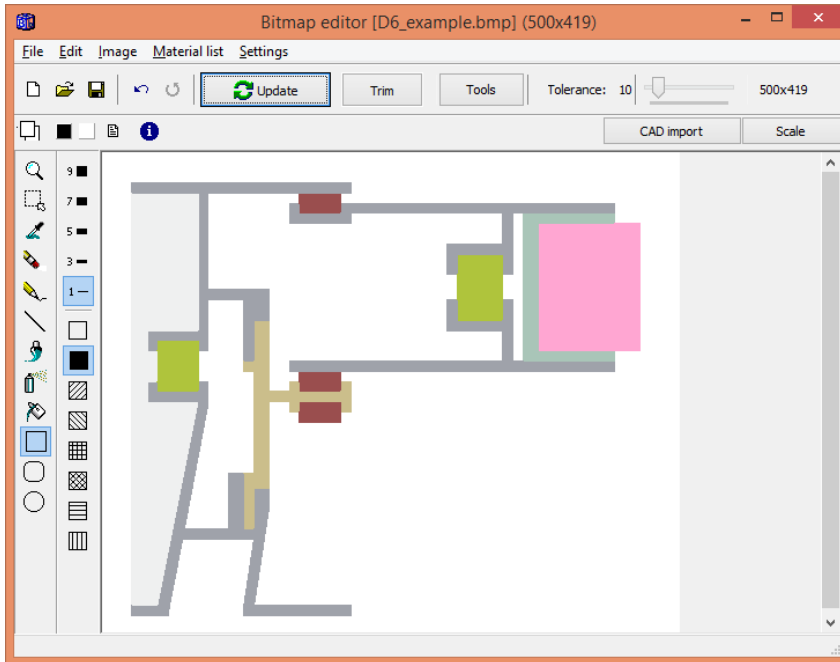
Open the Tools dialog and remap the black outline to nearest color:



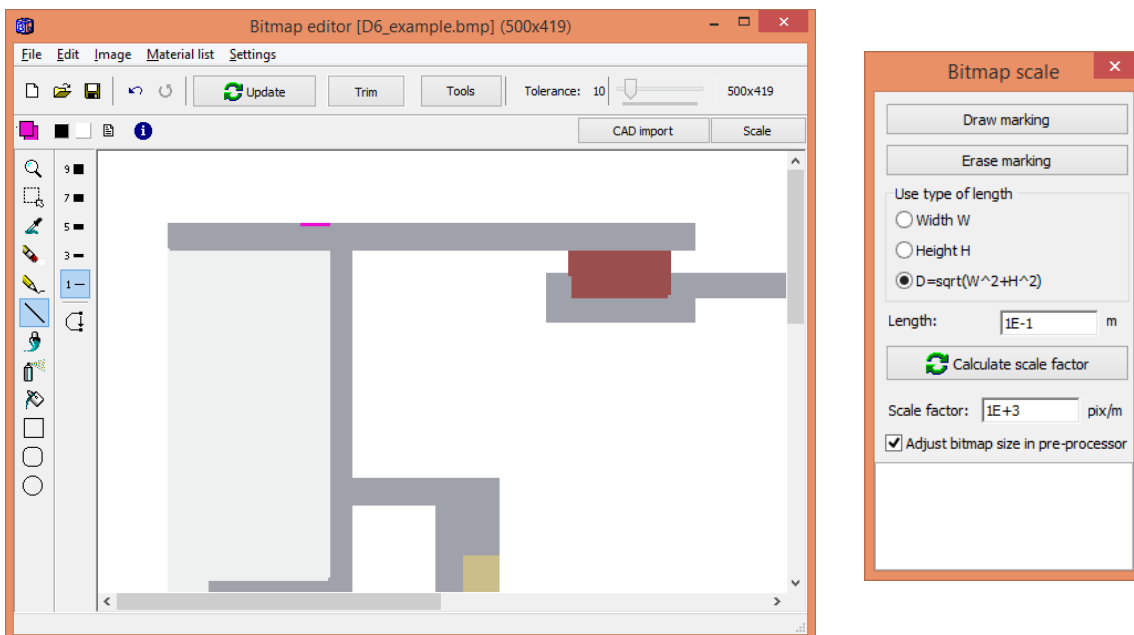
Press “Update” to check geometry in the post-processor:



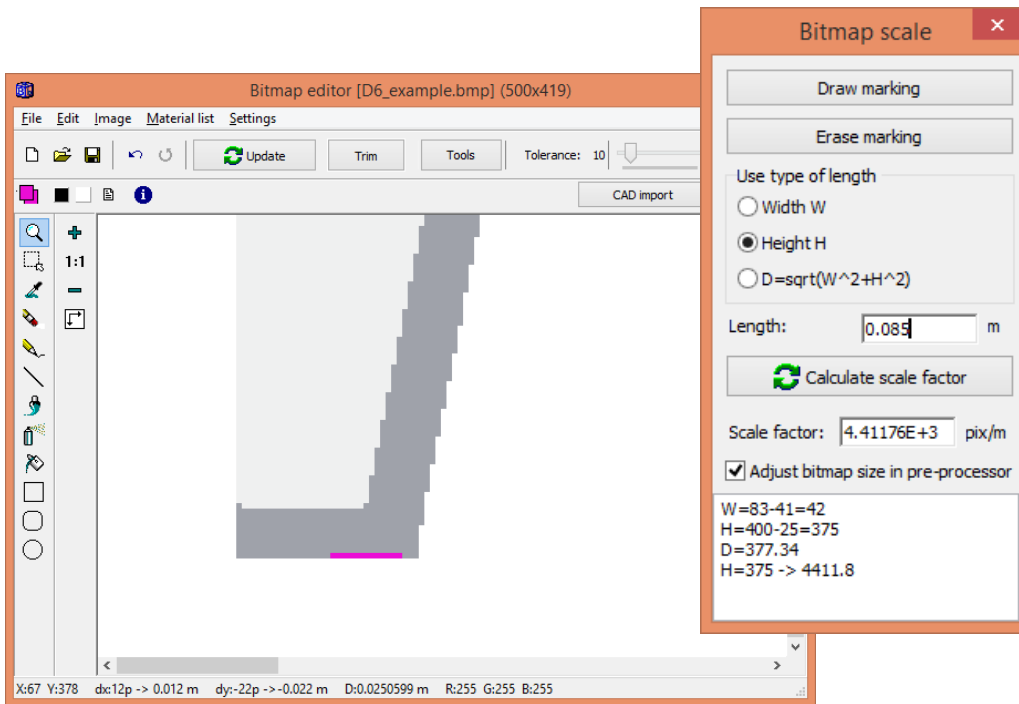
Draw a white rectangle to cut the right end of the insulation panel:



Open the “Scale” dialog and press “Draw marking”. Zoom in the upper part and draw a marking (see red line):

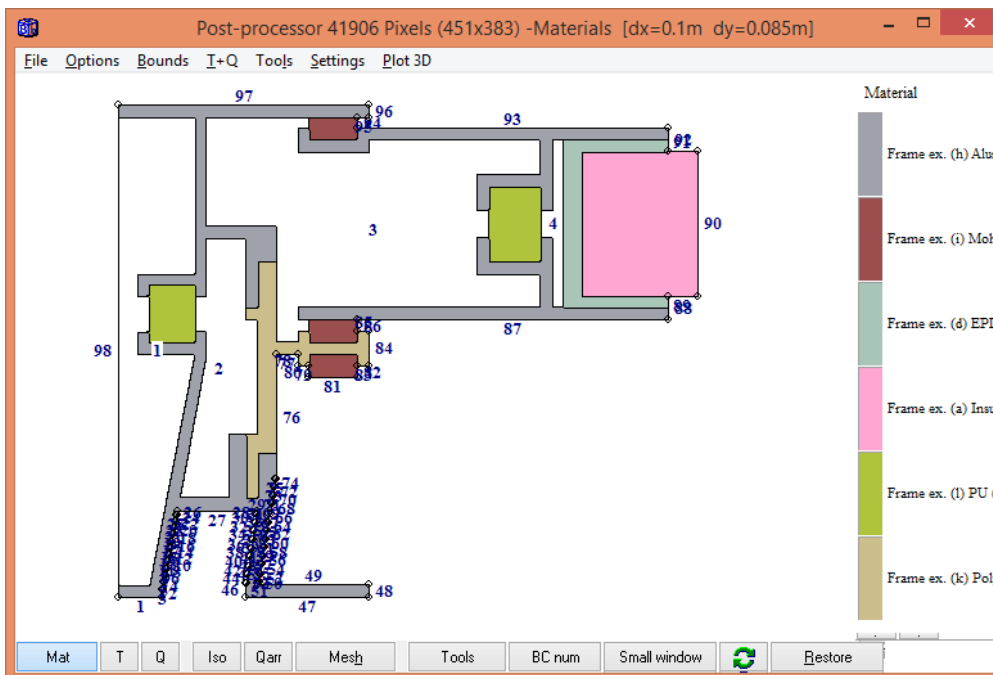


Draw a marking at the lower part. Choose the “Height” option and set the length (85mm) in the scale dialog. Press “Calculate scale factor”. The scale factor will be 4411.76 pixels per meter.



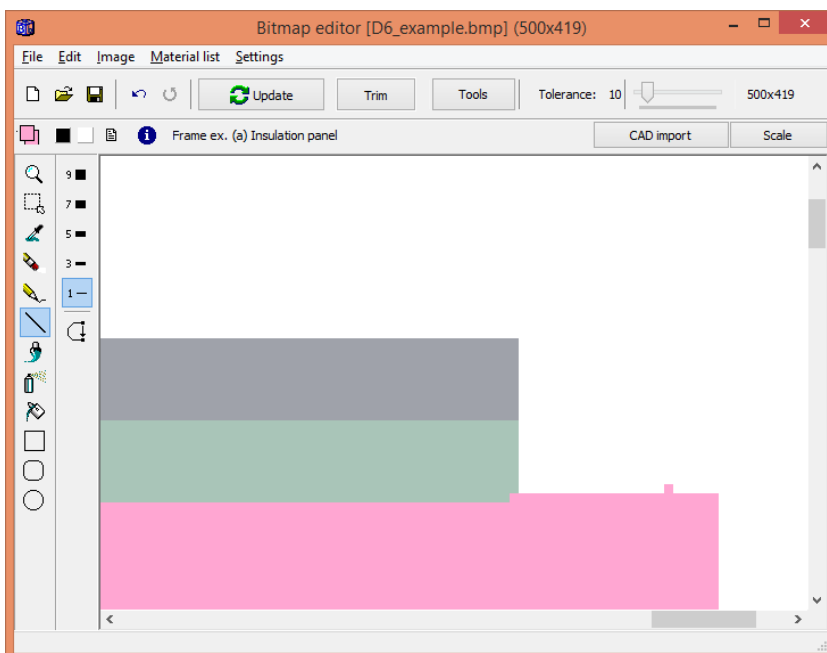
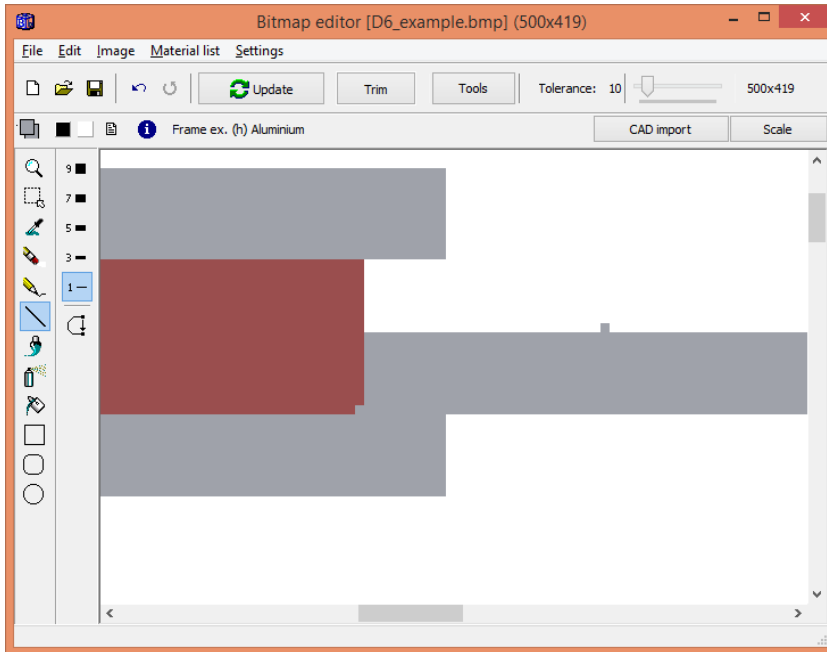
Important: since we have drawn the marking on a material we need to delete it. Press the undo-arrow twice (since we drew two markings) in the upper tool bar (or press “Erase marking”).

Press Update and check the geometry in the post-processor (the total width and height are shown in the caption and $dy=0.085$ is ok).

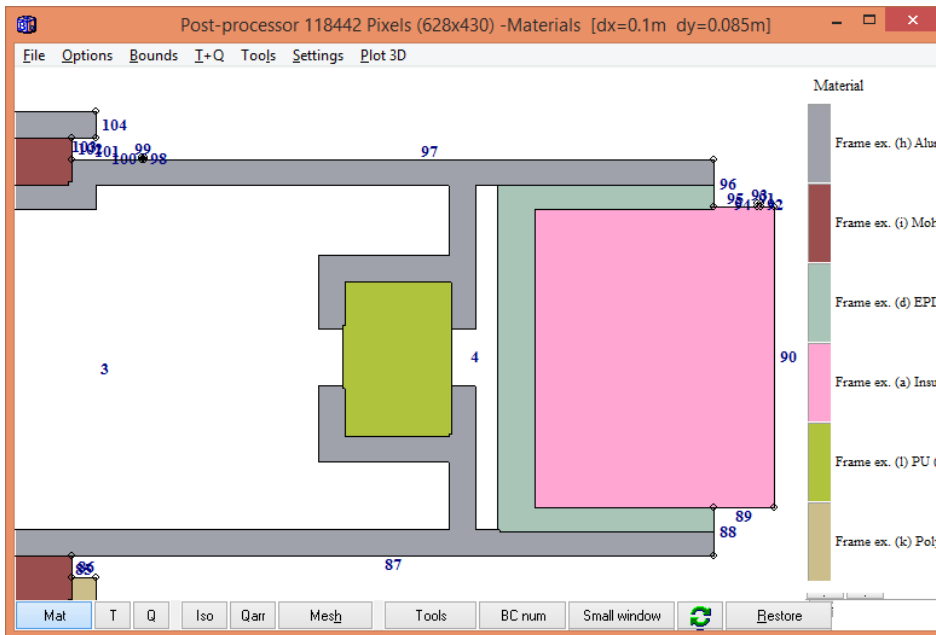


We have a couple of problems to deal with: the insulation panel should extend 190 mm to the right, and we need to specify different surface resistances according to EN ISO 10077-2.

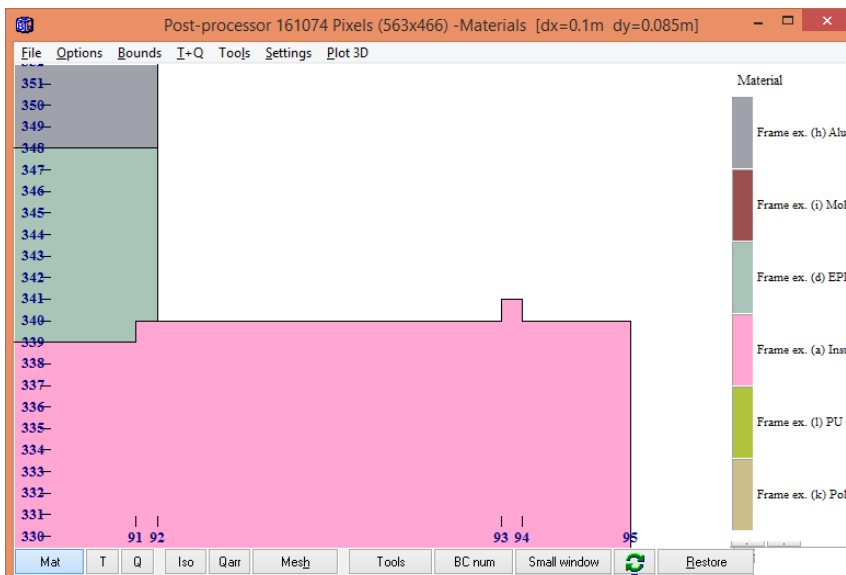
To be able to specify different boundary conditions we need to create some more boundary segments. This can be done by the following “dirty” trick. Just draw two small boxes (using e.g. the line tool or the pencil tool) located according to EN ISO 10077-2:



Now we have some more boundary segments where we can put the resistance $R=0.20$ (boundaries 95, 96, 101-104). The resistance $R=0.13$ is assigned to the other upper boundaries.



We can solve the other problem with 190 mm extension using another dirty trick and just change the length for those input mesh segments involved (or we could have imported a wider bitmap). First, check menu item “Options”/”Input mesh coordinates” to see the mesh coordinates:



Press F3 (or go to menu item “Input/Text input (old style)/Mesh in x-dir”) to see the lengths:

Mesh in x-direction (F3)

Between	Length [m]	Cells	Expansion
80 and 81 >	0.00023	1	1.000
81 and 82 >	0.01451	7	1.000
82 and 83 >	0.00204	1	1.000
83 and 84 >	0.00023	1	1.000
84 and 85 >	0.00861	4	1.000
85 and 86 >	0.00023	1	1.000
86 and 87 >	0.00204	1	1.000
87 and 88 >	0.00181	1	1.000
88 and 89 >	0.00023	1	1.000
89 and 90 >	0.00295	1	1.000
90 and 91 >	0.01473	7	1.000
91 and 92 >	0.00023	1	1.000
92 and 93 >	0.00363	2	1.000
93 and 94 >	0.00023	1	1.000
94 and 95 >	0.00113	1	1.000

Lx=0.1 Nx=118 Spread

Close Update graphics

The last three lengths (between 92 and 95) should sum up to 0.190 m so we change the last length to $0.190 - 0.00023 - 0.00363 = 0.18614$. Now the sum Lx is 0.285 which is fine (95mm+190mm=285mm).

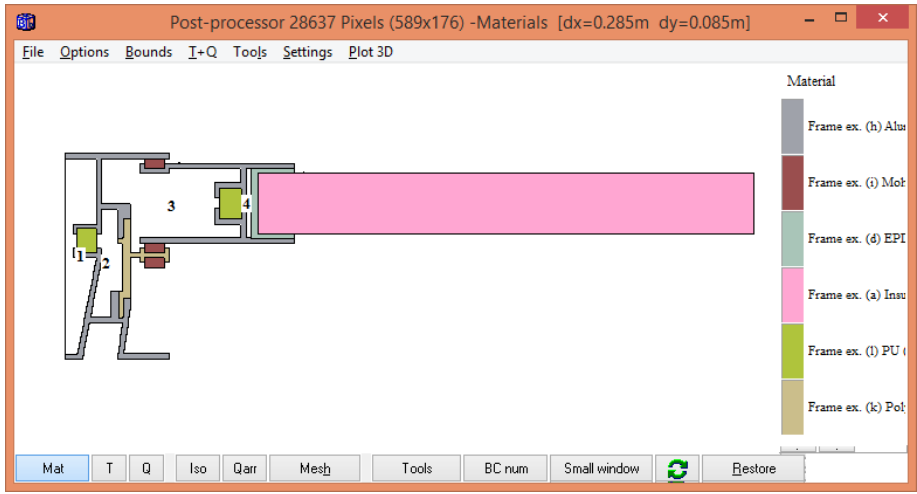
Mesh in x-direction (F3)

Between	Length [m]	Cells	Expansion
80 and 81 >	0.00023	1	1.000
81 and 82 >	0.01451	7	1.000
82 and 83 >	0.00204	1	1.000
83 and 84 >	0.00023	1	1.000
84 and 85 >	0.00861	4	1.000
85 and 86 >	0.00023	1	1.000
86 and 87 >	0.00204	1	1.000
87 and 88 >	0.00181	1	1.000
88 and 89 >	0.00023	1	1.000
89 and 90 >	0.00295	1	1.000
90 and 91 >	0.01473	7	1.000
91 and 92 >	0.00023	1	1.000
92 and 93 >	0.00363	2	1.000
93 and 94 >	0.00023	1	1.000
94 and 95 >	0.18614	1	1.000

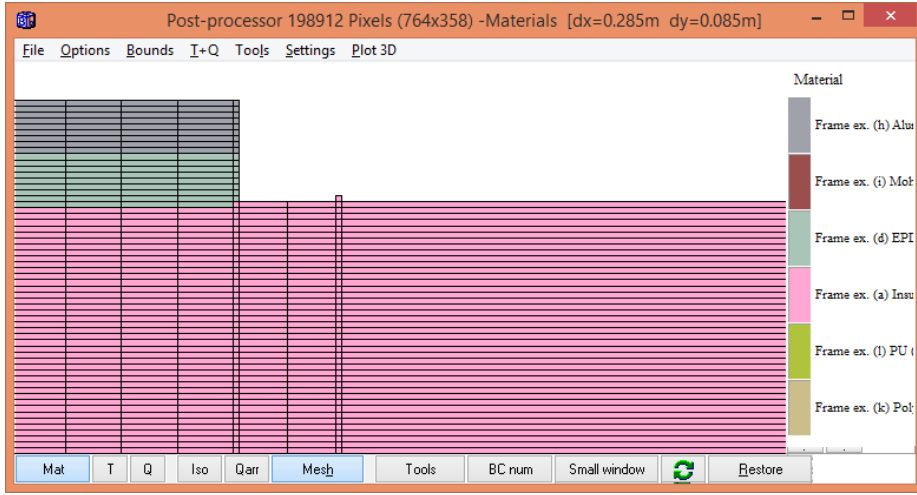
Lx=0.285 Nx=118 Spread

Close Update graphics

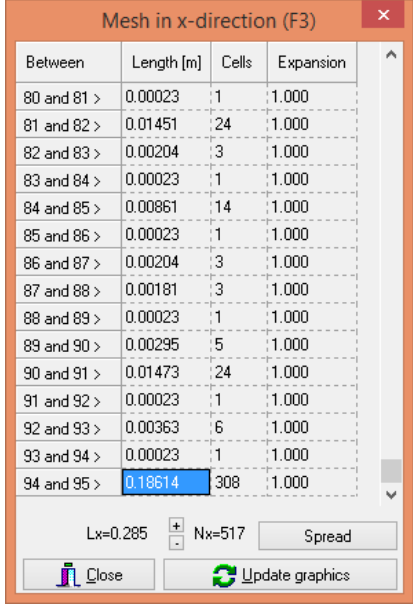
Press “Update graphics” and look at the post-processor window. We can also see in the caption that the total width is $dx=0.285$.

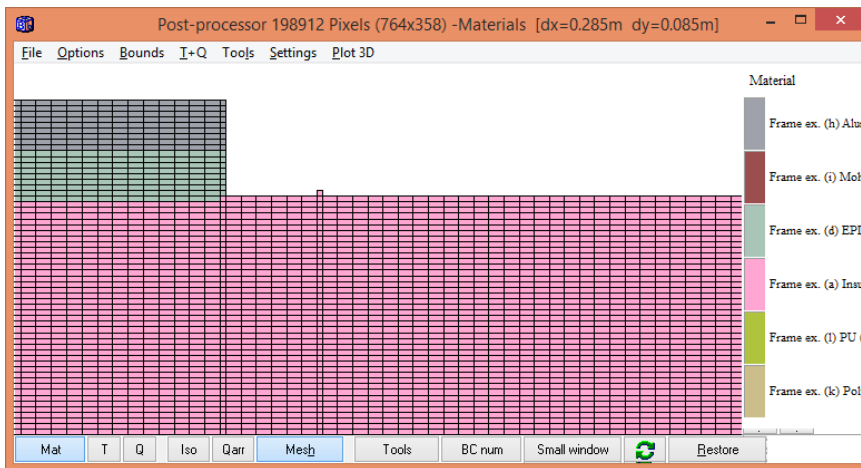


Press the mesh button and zoom up. We need to fix a numerical mesh better than this in the x-dir:



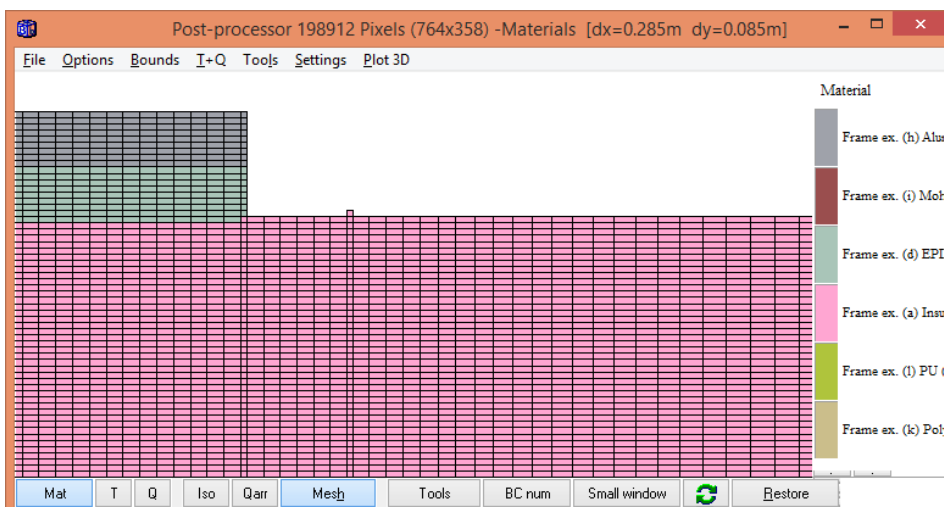
Goto F3 and press e.g. the “+” sign twice to increase the number of cells in x-dir, press “spread” and “update graphics”:





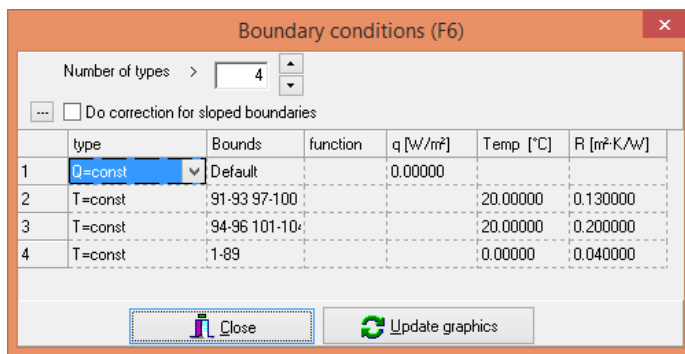
We don't need so many cells in the x-direction at the right end of the insulation panel since the heat flow is mainly in the y-direction towards the end. Choose e.g. 100 cells with an expansion factor of e.g. 1.02 for the last segment:

Between	Length [m]	Cells	Expansion
80 and 81 >	0.00023	1	1.000
81 and 82 >	0.01451	24	1.000
82 and 83 >	0.00204	3	1.000
83 and 84 >	0.00023	1	1.000
84 and 85 >	0.00861	14	1.000
85 and 86 >	0.00023	1	1.000
86 and 87 >	0.00204	3	1.000
87 and 88 >	0.00181	3	1.000
88 and 89 >	0.00023	1	1.000
89 and 90 >	0.00295	5	1.000
90 and 91 >	0.01473	24	1.000
91 and 92 >	0.00023	1	1.000
92 and 93 >	0.00363	6	1.000
93 and 94 >	0.00023	1	1.000
94 and 95 >	0.18614	100	1.020

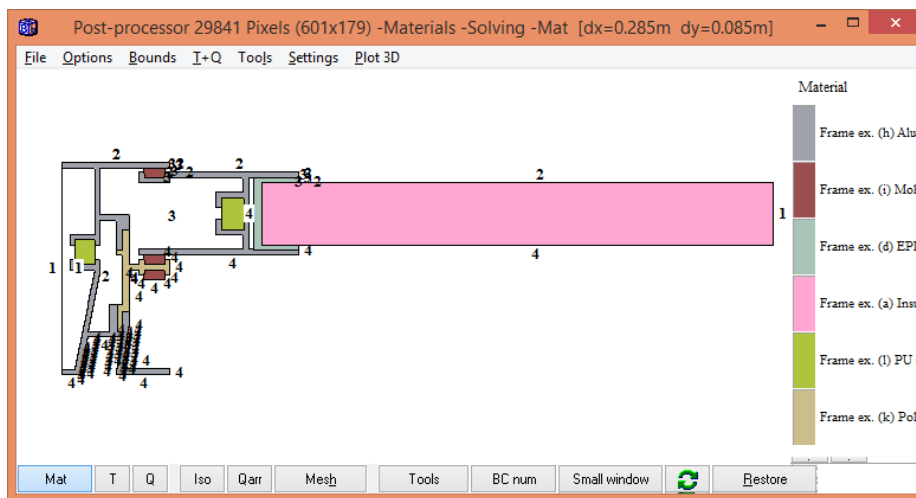


Resulting mesh expanding to the right.

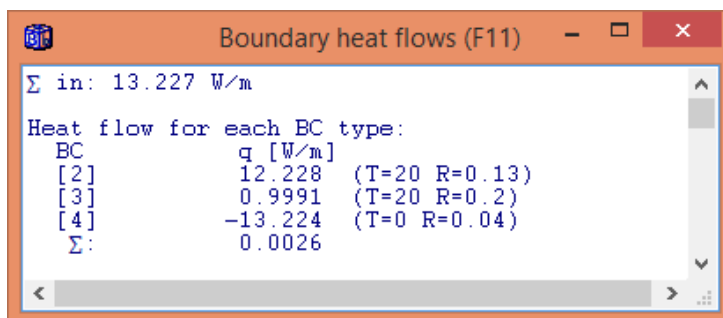
Finally, we specify the boundary conditions. For the first calculation we choose not to do correction for sloped boundaries:

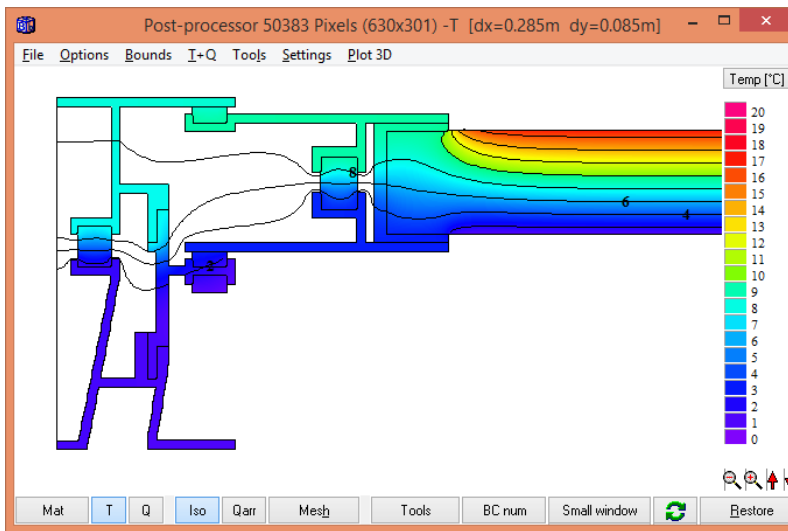


A good tip is to check the boundary condition types (menu item “Bounds/BC types”):

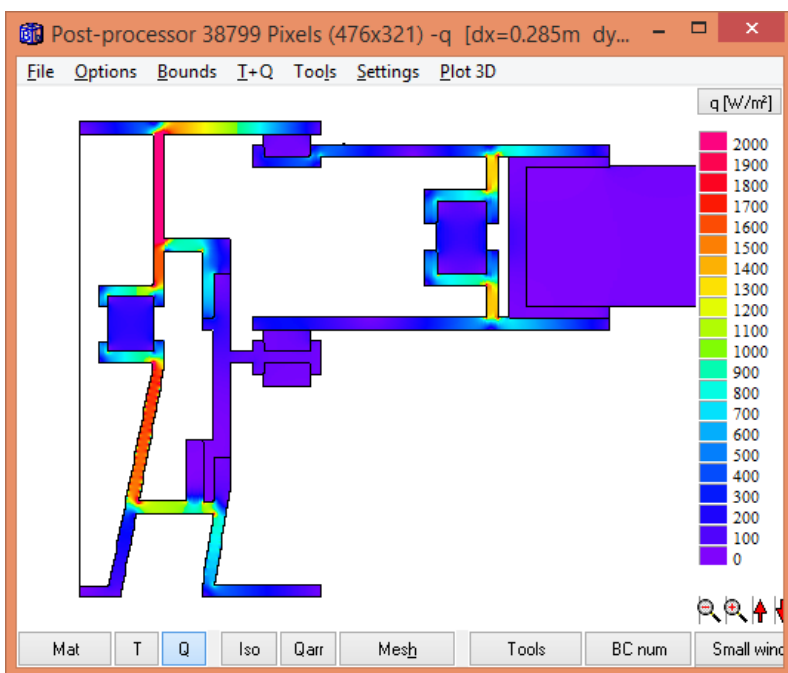


Now, start the calculation for steady-state. The calculated heat flow is 13.182 W/m which gives the thermal conductance $L^{2D}=13.227/20=0.661$ W/(m·K) (this value should be 0,659 +/- 0.008 according to EN ISO 10077-2). The CPU-time for this case with 60,000 nodes on a i7 (3770K 3.50GHz) is about 15 seconds.





Temperatures and isotherms

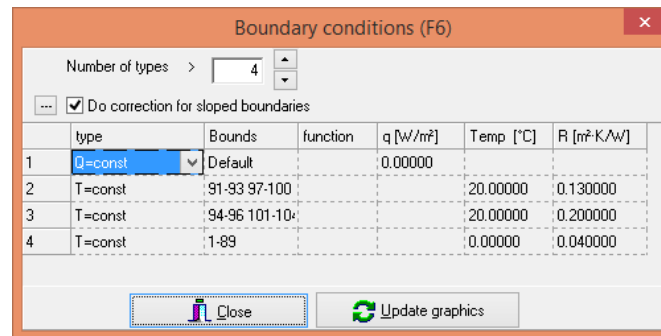


Heat flow intensities

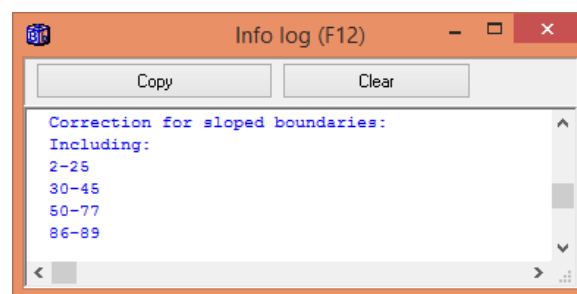
The above heat flows shows critical areas that could be improved regarding the thermal bridge effect.

Correction of surface resistances for sloping external boundaries

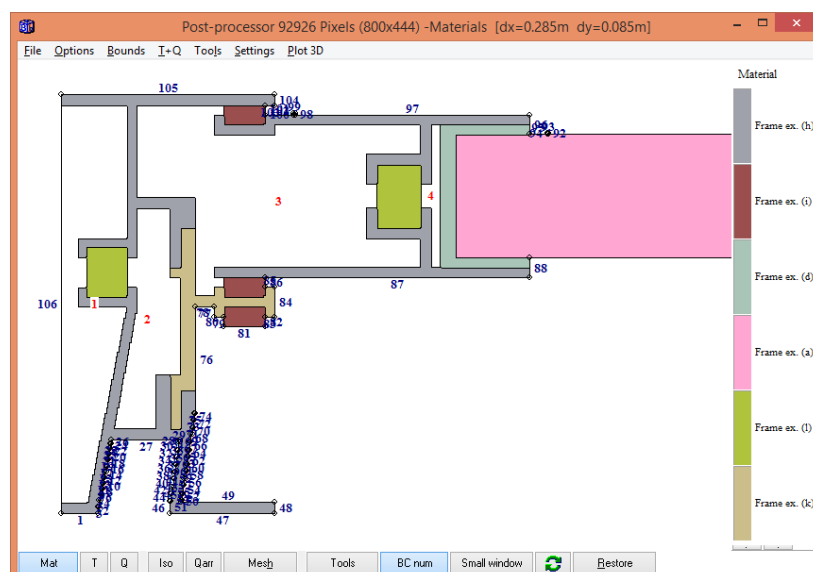
There is a smaller need (maybe only academic) to adjust the thermal resistance of a numerical cell lying at a boundary when a sloping boundary is approximated by steps in order to simulate a boundary condition with a temperature and a surface resistance, or a given heat flux, correctly. See Section 4.4 in the update manual of version 8. HEAT2 makes a guess of what boundaries are affected and corrects the surface resistances each time a simulation is started if the option “**Do correction for sloped boundaries**” is checked:



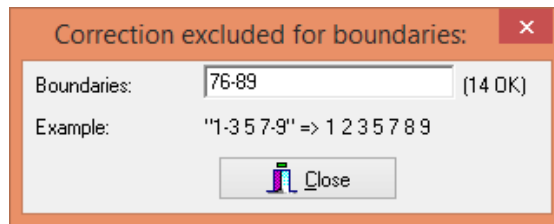
Turn on “Do correction for sloped boundaries”, start a steady-state simulation and open the info log (F12) to see what boundaries are guessed by HEAT2 to be affected:



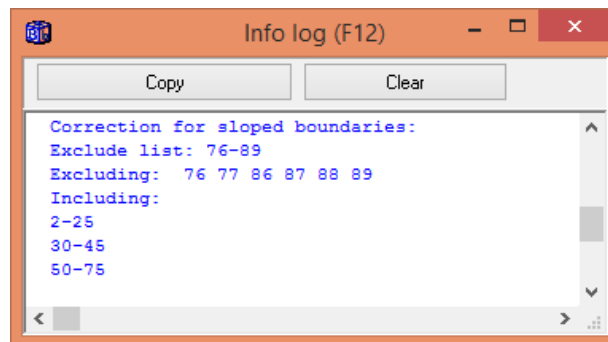
According to the figure below, we should however **not** do the correction for boundaries 76-77 and 86-89 (the other boundaries should be corrected because these are approximated by steps).



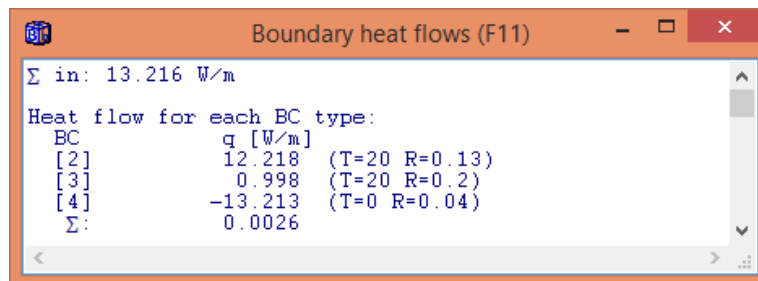
Press the button to the left of option “Do correction for sloped boundaries”, and give the **exclusion** for boundaries 76-77 and 86-89 (it is ok to enter 76-89 here):



Start the steady-state simulation and open the log window (F12). Now we see that the correction is made for the sloped boundaries only:



And the final calculated heat flow will be 13.216 W/m which gives $L^{2D}=13.216/20=0.661$ W/(m·K):



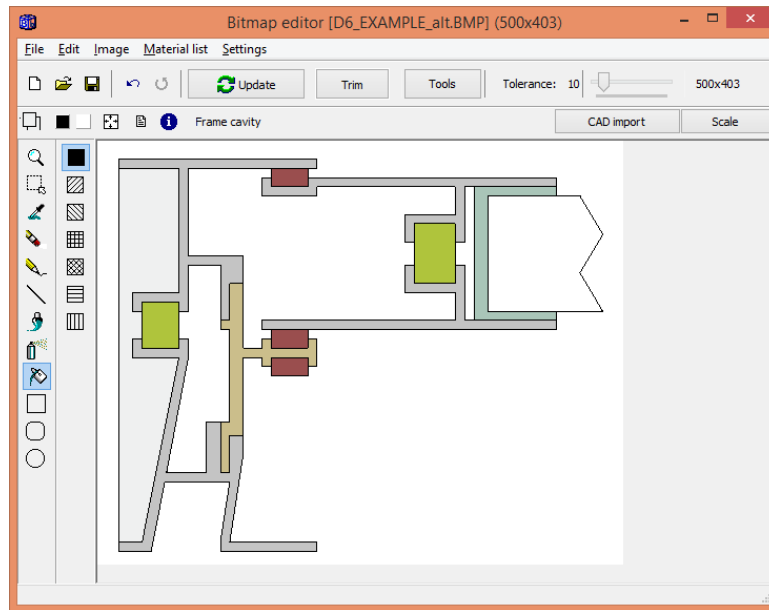
So, without the correction, the boundary resistances will be somewhat too low (leading to a somewhat too high total heat flow). On the other hand, the surface temperatures will be too low on the cold side, and too high on the warm side. This means we are on the safe side when calculating e.g. psi-values but not for condensation risk estimation.

The difference for the total heat flow is less than one per-mille, so the correction here is a bit academic.

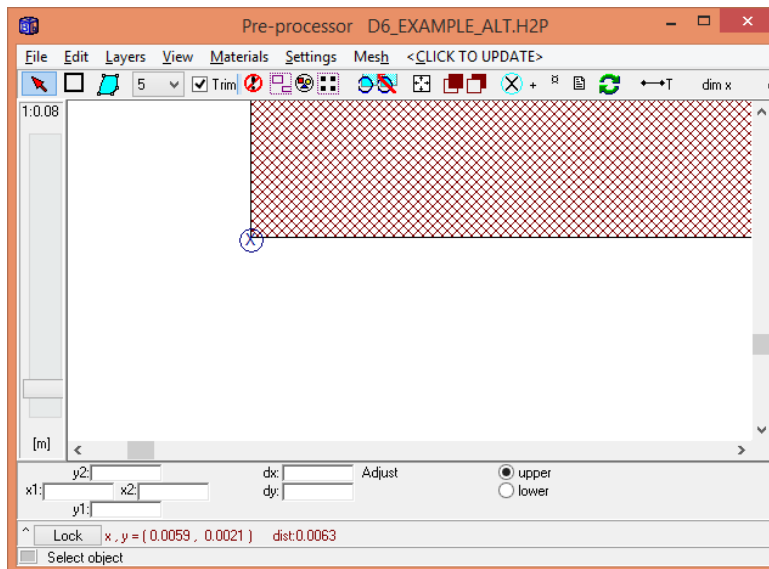
Alternative method for input

This method shows how to add materials in the pre-processor to the materials given in the bitmap editor. We will illustrate how to add the insulation panel in the pre-processor. See example **d6_example_alt.dat**.

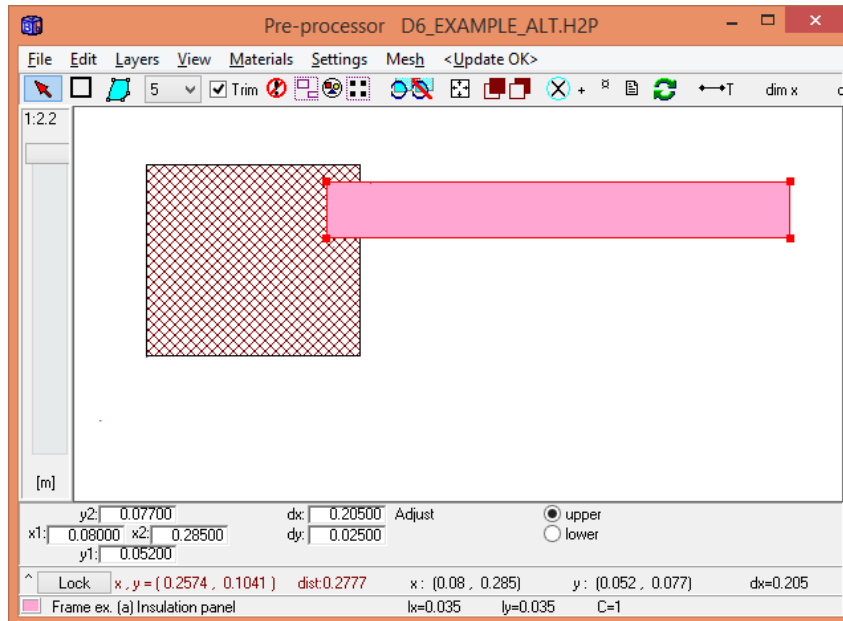
First, import the dxf, fill the materials (except the insulation panel), set the scale, and remap the black outline as above. We end up with this figure:



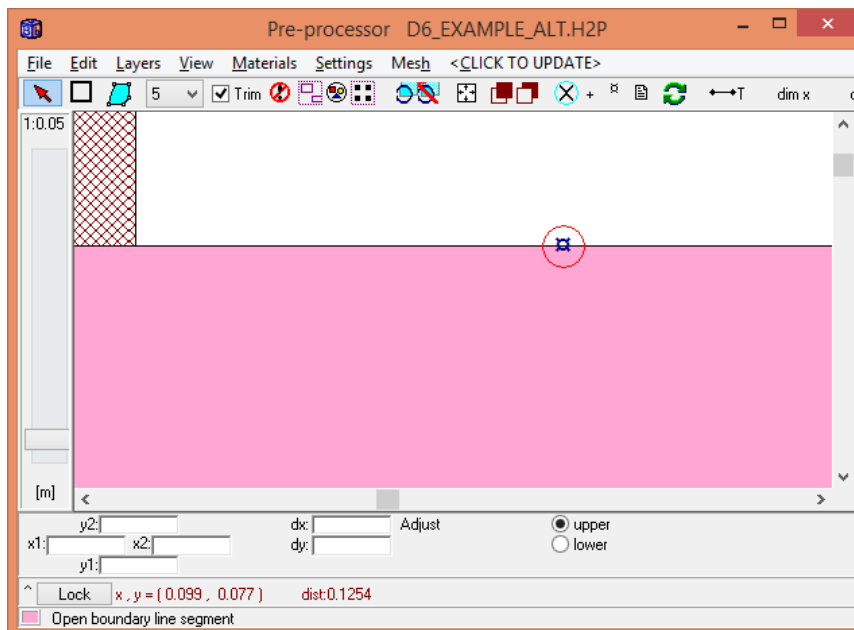
Press "Update". Open the pre-processor, zoom in the bitmap rectangle and put a reference point at the lower left in order to set origo:



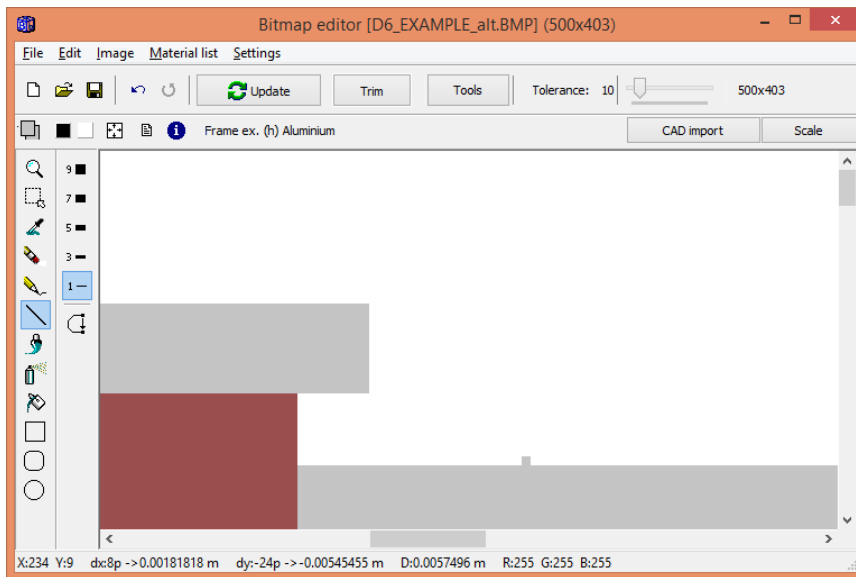
Now, draw a rectangle using the insulation material (“Frame ex. (a) Insulation panel”). Change the width to 205mm (this gives 190mm extension) and height 25 mm. Move the rectangle so that the lower left point at (0.080,0.052). Use the zoom to make it easier.



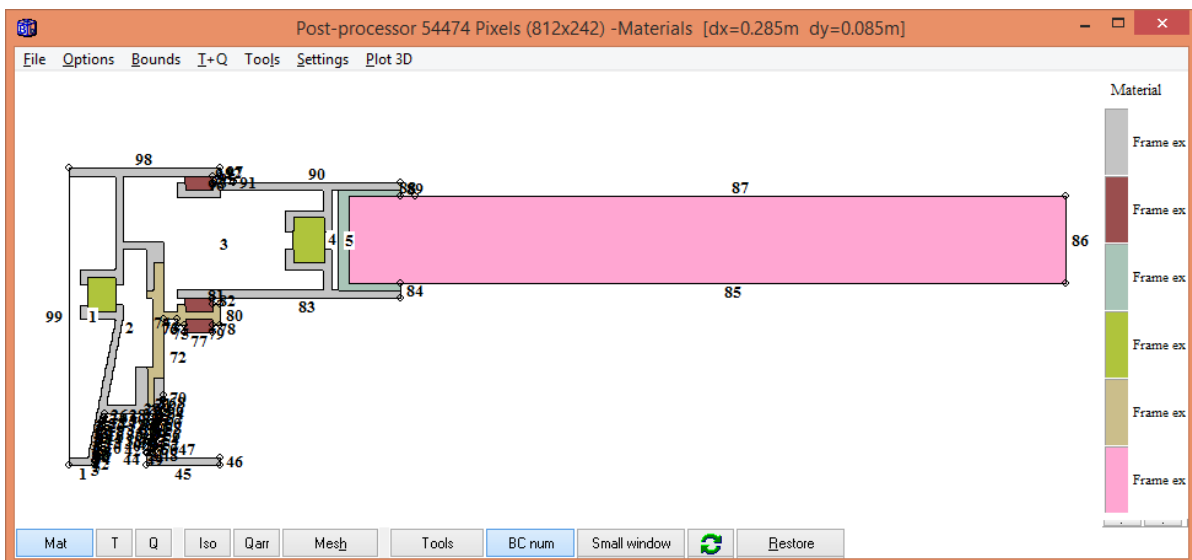
We can now put an "Open boundary line segment" point at (0.099, 0.077):



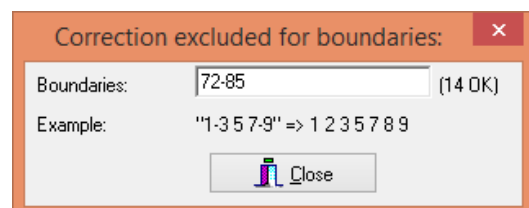
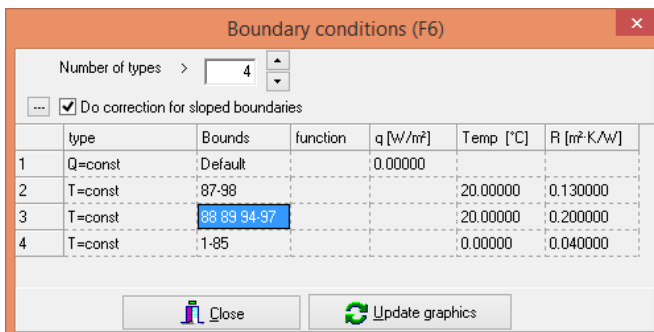
We create the other boundaries using the dirty trick as before:



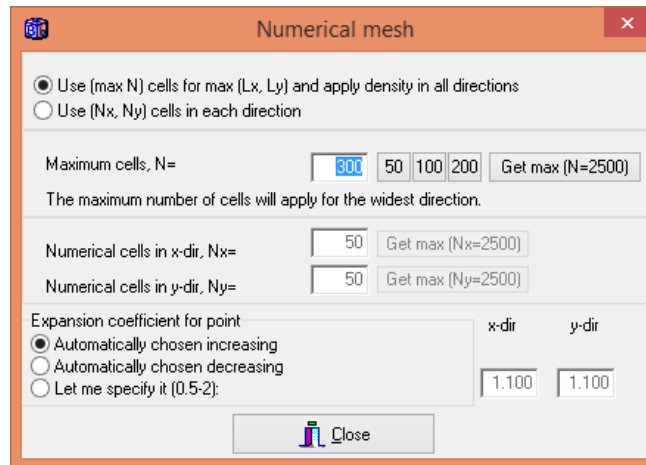
Boundaries shown in the post-processor:



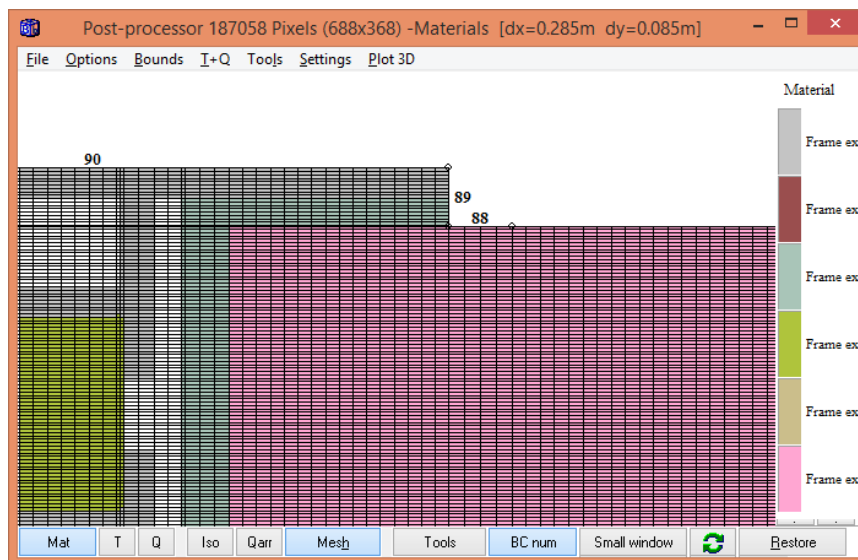
Now set the boundary conditions, (and optionally excluded boundaries if correction is checked for the surface resistance):



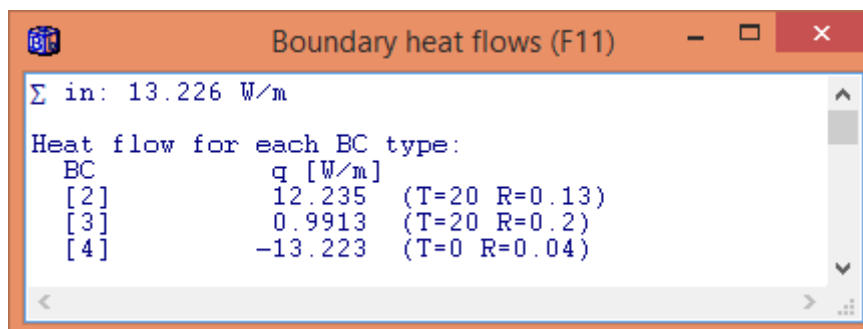
We need to improve the mesh:



This resulting mesh with about 60,000 nodes should be sufficient:



The calculated heat flow is 13.226 W/m which gives the thermal conductance $L^{2D}=13.226/20=0.661$ W/(m·K):



5. Miscellaneous

5.1 Several instances of HEAT2 are now allowed

Several instances of HEAT2 can now be started at the same time. This will allow you to work (input and solve) multiple cases at the same time. All instances of HEAT2 will share the available cores in the CPU (this is automatically handled by the OS system). The following table shows how efficient the OS system runs 1-5 number of instances of HEAT2 on an Intel Core i7-3770K (which has four CPU cores but 8 threads which means that 8 logical cores that can be used due to multi-threading). Assume that a problem to be solved in this case takes 52 seconds by one instance of HEAT2. If you e.g. run three instances of HEAT2 simultaneously, it will take 58 seconds to solve the three cases. This may be good to know if you need to solve several large transient cases that each can take several hours. Solving 3-4 cases parallel on a CPU with four cores, or 5-6 cases on a CPU with 6 cores should be very efficient.

1 instance	2 instances	3 instances	4 instances	5 instances
52s	53s	58s	71s	88s

5.2 Improved handling of modifications

In the old version 8, types and areas for internal modifications had to be specified/checked each time after the pre-processor content was updated via “<Click to update>”. Version 9 handles this better and there is only a need to check the modifications shown in the post-processor with the given data for the input for internal modifications types and areas.

5.3 Improved handling of input via text

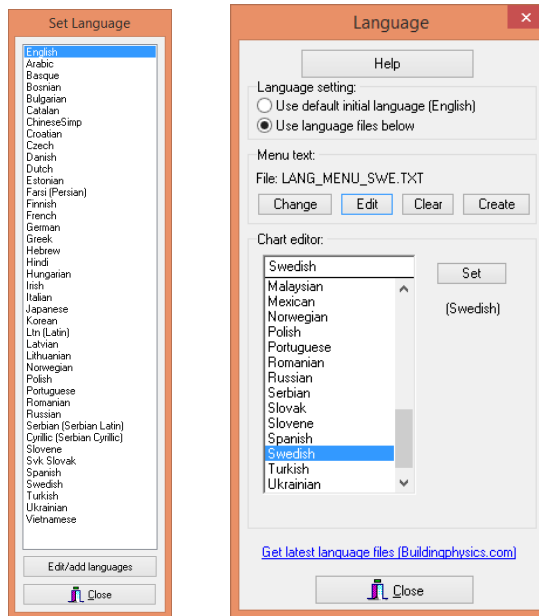
The old-style input using text has been improved for the thermal properties list (F5), and for mesh in x,y directions (F3 and F4) where lengths, number of cells and expansion coefficients are given. The connection to the post-processor is better.

5.4 Multilingual option

5.4.1 Introduction

The figure below shows the language option (menu item “Options/Language”). A click in the list will change the language. The most current language files are listed at <http://www.buildingphysics.com/index-filer/Page1724.htm>

The list of languages is given in a text file “Lang.txt” in the “Languages” folder.



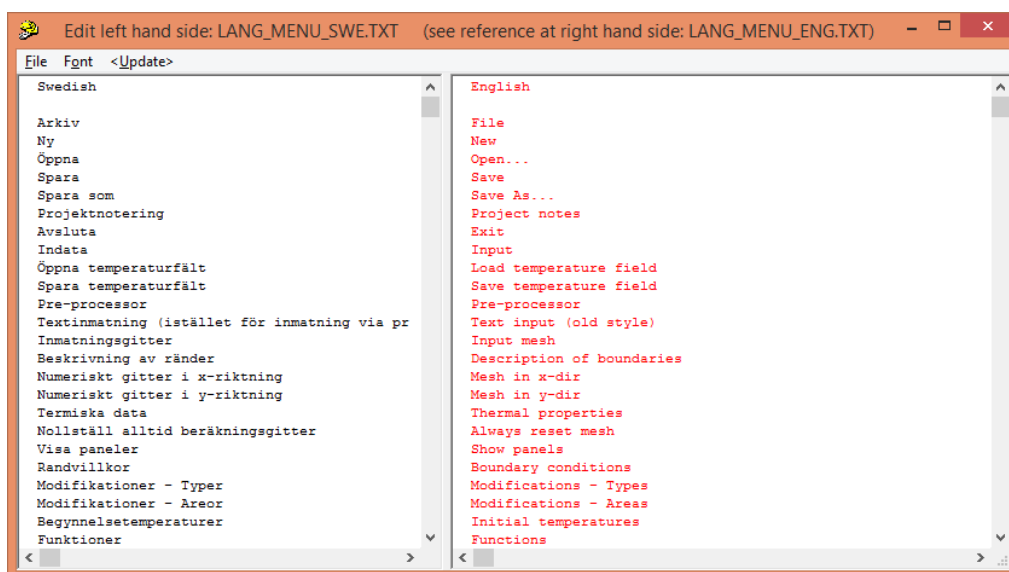
Dialog for changing and editing languages (list is defined in file “Lang.txt”).

New language files can easily be created and edited. For each language there is a file “Lang_menu_***.txt” where the string “***” should be replaced by the ISO 639-2 Code, see e.g.: http://www.loc.gov/standards/iso639-2/php/code_list.php

E.g. the Swedish files should be named “lang_menu_swe.txt” and the German “lang_menu_ger”.

If you create new language files please consider to email these to info@blocon.se and we will make these available to others. All strings are also in a excel file in the folder “...Documents\Blocon\HEAT2 9\Languages”.

To edit a file just open it in a text editor (e.g. Notepad). You can also use the button “Edit/add languages” which will open a new “Language” window, see above. As an example we choose Swedish and then click “Edit”. This will bring out the following window where you can edit each string. Click <Update> to see the updated text in HEAT2.



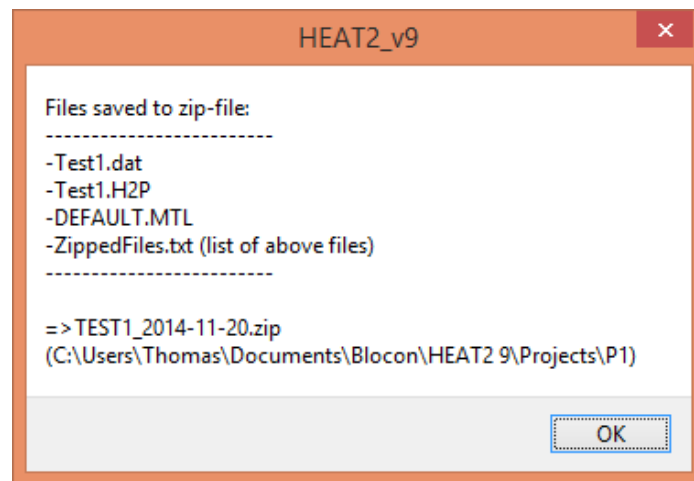
5.5 Saving project files to zip-file

All projects files can be saved to a zip-file for backup purposes. This makes it easier to share all needed files to another user by e.g. email. Use menu item “**Save to zip-file (for backup and email)**” in the File menu.

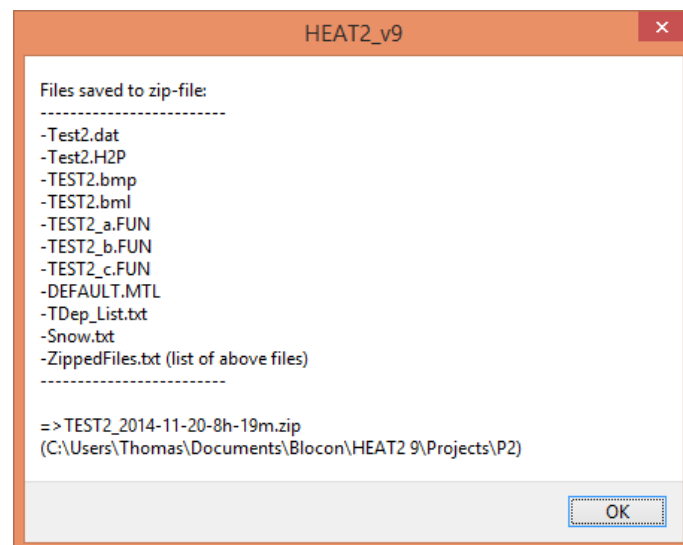
The name of the zip-file is automatically generated and will always start with the name of the dat-file followed by a date (yyyy-mm-dd), e.g. “TEST1_2014-11-20.zip”. If this file already exists, a time string (h-m) is added, e.g. “TEST1_2014-11-20-8h-9m”.

For the simplest project, the following example shows what files typically will be zipped:

Test1.dat	Input dat file
Test1.h2p	Pre-processor file
DEFAULT.MTL	Used material file
ZippedFiles.txt	Shows a list with the above files including paths



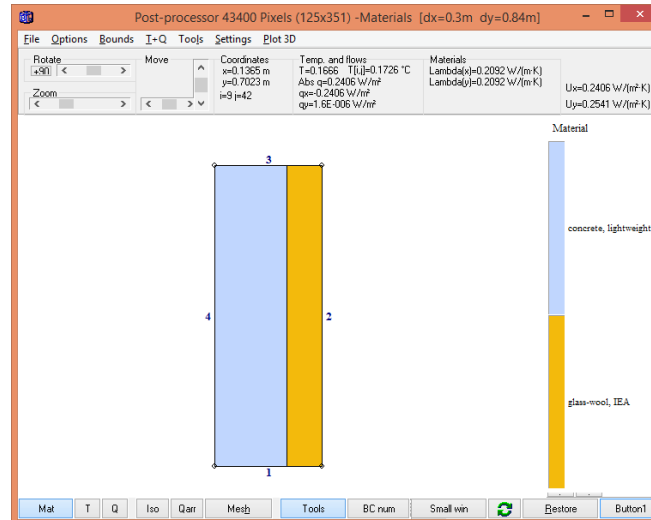
Here is an example for a case containing many files:



The zip-files are very small (around 10 kB in size). Even with the largest material file “General.mtl” (about 100 kb uncompressed) containing over 1200 materials and a 1000x1000 bitmap (around 1 MB uncompressed), the resulting size of the zip-file will only be around 30-40 kB.

5.6 Enhancements in the post-processor

The U-values (1D) in x,y-direction (U_x , U_y) are shown for any cut when the mouse cursor is moved on to a material. In the figure below, the U-value of the wall is shown as “ $U_x=0.2406 \text{ W}/(\text{m}^2 \cdot \text{K})$ ” which is calculated from $1/R=$ where R is the total resistance for the layers and the specified surface resistances, here $1/(0.13+0.04+0.2/0.2092+0.1/0.033)$. The value “ U_y ” can be ignored in this case.



6. References

[1] *Eftring B, 1991. Numerical calculation of thermal processes. A physical approach. (In Swedish) Dept.of Building Science. LUTADL/TABK-1007/1-221/1990. Lund 1990.*

[2] New design model for frost protection of a slab on grade. Peter Roots ; Carl-Eric Hagentoft (Institutionen för bygg- och miljöteknik, Byggnadsteknologi). 3rd International Conference on Research in Building Physics and Building Engineering, Montreal, August 2006.