SOLIDO

computer program to calculate
3D steady state heat transfer
in free-form objects

version 5.0

physibel

This manual is intended as a complete reference for the SOLIDO application.

Example projects to get started are available as **Tutorials**
.../Documents/Physibel/SOLIDO5/Tutorials/

Documented software application examples, tutorials & exercises, short practical guides (How To's) and validation cases are accessible via the Physibel Knowledge base (www.physibel.be/en/knowledge).

Table of contents

CHAPTER A INTRODUCTION

A.1. Program overview

SOLIDO is a thermal analysis program for 3D steady state heat transfer in objects of any shape. SOLIDO can be considered as an extension of the Physibel program TRISCO, which is restricted to rectangular shapes.

The geometry in SOLIDO is composed of a list of blocks. Each block is a simple solid, derived from a rectangular block oriented along the axes of the coordinate system (as in TRISCO). In SOLIDO the vertices of the block can get free positions, which allows to introduce a non-rectangular form, as a prism or a pyramid. Also the block surface can be rounded to form e.g. a cylinder, a cone or a sphere. As in TRISCO there is a superposition principle for the blocks in the list: when blocks intersect one another, then blocks further in the list have priority over previous blocks, and overwrite the common parts.

The surfaces of the blocks are triangulated, based on a given mesh size (defined in the **Blocks window**). For a rounded block surface the triangulation delivers an approximation of the real surface. The lower the given mesh size, the better the approximation. The intersections of the triangulated blocks are resolved conform to the superposition principle in the list of blocks.

Besides the blocks defined directly in SOLIDO, also blocks referring to externally defined triangulations of objects can be processed. Files created in external programs in STL format ("Stereolithography Tessellation Language") can be imported in SOLIDO. Each STL file contains the triangulation of the surface of a solid, and can be created by a solid modelling program (also e.g. AutoCAD). The STL file is the standard format in "rapid prototyping", but is also used in "finite elements" programs. The externally created STL file can be linked to a single SOLIDO block, and can then be handled similarly as blocks defined directly in SOLIDO, without any difference.

The link between blocks and thermal properties of materials and boundary conditions is made through the use of colour indices, as in TRISCO. For cavities modelled as materials, the equivalent thermal conductivities can be derived from the geometric sizes of the cavities, based on the methods described in standards, but generalised to non-rectangular shapes.

The calculation grid of nodes is based on a rectangular grid as in TRISCO. This rectangular grid can be created automatically, and adjusted using different grid functions (if necessary). The geometry of triangulated surfaces in SOLIDO is rasterised based upon the grid, using a 3D "raster image processor", comparable to the creation of a 2D pixel bitmap starting from a vector contour description of an image. A thin material which (partially) disappears after rasterisation can be recovered through a swell function. The grid can be fixed to the object geometry (as is always the case in TRISCO, because block coordinates are directly expressed using grid layer numbers), so that changes in the grid are immediately reflected in changes in the object geometry.

The unique strength of SOLIDO is that the rectangular grid is adapted to the original (possibly non-rectangular) geometry, by fitting the nodes of the grid to the closest points on the block surface triangulations (see A.2). In this way a more correct simulation can be obtained for the three components of heat transfer:

- <u>Conduction</u>: the fitted geometry approximates better the original volumes and shape proportions.
- <u>Convection</u>: the fitted geometry approximates better the contact surfaces to the environment.

In the rasterised geometry before node fitting all contact surfaces are reduced to orthogonal surfaces, all parallel to the coordinate planes. The contact surface in the rectangular geometry is in general larger than the contact surface in the real non-rectangular geometry.

- Radiation: the fitted geometry approximates better the original surface orientations, which are important in calculating view factors.
  The rasterised geometry only recognises the three orientations of the coordinate axes, which is a strong simplification of the real distribution of orientations.

Even though the rectangular grid is fitted to the non-rectangular geometry of the model, the system of equations can be solved as fast and as stable in SOLIDO as the system based on the rectangular grid (as in TRISCO). This is because SOLIDO ensures that the fitted system has the same favourable characteristics as the rectangular system (i.e. a sparse positive definite linear system). The required amount of memory is linearly proportional to the number of nodes.

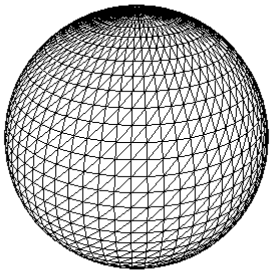
Also the RADCON module for separate simulation of convection and radiation is available in SOLIDO. Without the RADCON module all heat transfer to the environment is modelled as convection, using global heat transfer coefficients.

The graphic output allows to make detailed inside views of temperatures and heat fluxes, also with object clipping to grid planes, and/or clipping off certain materials (via colour indices). Text output shows temperatures, heat flows from or to boundary conditions, global view factors, temperature factor, linear thermal transmittance, equivalent thermal transmittance.

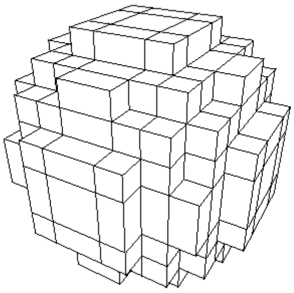
A.2. Principle drawings of object meshing

The following drawings of a spherical object show the different steps in the definition of the calculation mesh. These steps are essential in the use of SOLIDO and are further explained in the manual.

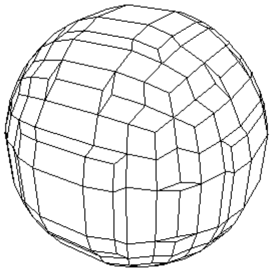
1. Create a surface triangulation of the block surfaces of the input geometry (solids defined in SOLIDO and/or STL files).
   In the program terminology this is called the triangulated object (obtained after the function *Image → Fill Blocks (Triangulate)*).

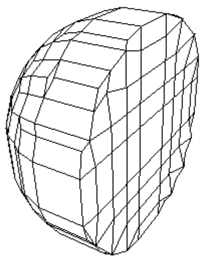


2. Then the triangulated object is rasterised with reference to a rectangular grid defined in the **Grid window**.
   In the program terminology this is called the rasterised object (obtained after the function *Image → Fill Grid (Rasterise)*).



3. Finally the surface grid nodes of the rasterised object are fitted by projection to the triangulated object, which results into the calculation mesh. In the program terminology this is called the fitted object (obtained after the function *Image → Fit Grid).*
   The right figure below is a cross section, showing internal grid nodes.





A.3. SOLIDO: brief history

The development of the first version of SOLIDO was partially funded by a research grant of the Flemish Institute for the Promotion of Industrial Scientific and Technological Research (project no. VLIET/970397).

- Version 1.0 (October 2001) was based on meshing with tetrahedra.
- Version 2.0 (October 2003) based on a rectilinear grid (as in TRISCO) with node fitting.

The detailed version history from SOLIDO v3.0 (2009) onwards can be found on the website: www.physibel.be/en/products/solido/versions.

A.4. Technical specifications

- SOLIDO is a 64-bit Windows program. SOLIDO is installed in C:\Program Files (x86)\Physibel\ SOLIDO5.
  In this folder the executable file is SOLIDO5.exe.
- The standard version of SOLIDO allows to solve systems with up to 5,000,000 nodes. An add-on feature to calculate an unlimited amount of nodes is available for purchase (feature Unlimited Nodes). The maximum number of nodes then depends on the RAM installed. For example, 8 Gb RAM allows to solve about 25,000,000 nodes, 16 Gb RAM allows to solve about 50,000,000 nodes.
- SOLIDO relies on OpenGL for use of hardware accelerated graphics. A dedicated GPU is recommended to ensure fluent 3D visualisation of complex models.
- SOLIDO is protected by either a hardware key, provided by Physibel, or a software licence (A.5).

The Physibel 64-bit programs need several shared libraries (.dll files) from Microsoft.
The Microsoft Visual C++ 2017 Redistributable Package (x64) is included in the installation files of SOLIDO, but can also be downloaded from https://support.microsoft.com/en-us/help/2977003/the-latest-supported-visual-c-downloads.
This package installs mfc140.dll and other files in the directory C:\Windows\SysWOW64, required by 64-bit applications developed with Microsoft Visual C++ 2017.

A.5. Licence

The Physibel applications are protected by a licence, to prevent illegal use of the software. This can be either a hardware licence (Sentinel USB key) or a software licence.

A.5.1. Hardware licence

Perpetual licences are protected by a hardware licence (Sentinel USB key). The licence is locked to this dongle. It is strongly recommended to take a financial assurance for loss of the key by theft or any other incidence.

Before SOLIDO can be used, the driver (delivered with purchase) for the hardware key needs to be installed. Administrator rights are needed for this installation. It might be necessary to restart your computer after installing the driver.

During use of SOLIDO, the dongle needs to be attached to the computer at all times.

A.5.2. Software licence

Subscription licences are protected by a software licence. After first installation of SOLIDO, the user needs to activate the licence using the activation code received upon purchase of the software. When first opening SOLIDO, the Software Licence Manager will open automatically and prompt the user to activate the licence, or to register in the network floating database, depending on the licence type. The computer needs to be connected to the internet for this licence activation. Cloud-based floating licences are activated every time when opening the application.

After activation, the Software Licence Manager is available from the menu for stand-alone licences:
Help → Software Licence Manager...

Users with a floating network licence can check out a licence from the database for a maximum period of 3 days.
Help → Check out/in floating licence...

The software licence key for SOLIDO is linked to the computer where it was activated. When you want to move the application with a stand-alone licence to another computer, you can deactivate the licence in the Software Licence Manager. The licence can then be reactivated on a new computer.

If the Software Licence Manager is unavailable from the application itself, you can also open it from the command prompt. You need to first change directory to the Program Files folder where the application is installed. Then, open the Software Licence Manager with the following command:
QlmLicenseWizard.exe /settings "SOLIDO 5.0.lw.xml"

A.6. User Account Control (UAC)

Administrator rights are necessary to install SOLIDO on a computer.

The installation of the application is machine-level, meaning all user accounts can access the application after installation.

Each user account has its own copy of the colour database (see F.2) to work with. These files can be found in the user's application data directory (C:/Users/*username*/Appdata/Roaming/Physibel/SOLIDO5/).

In case of a software licence, the licence for SOLIDO is linked to the computer, but is stored at the user level. Each user account on a computer has to activate the licence in order to have access to the licence.

A.7. Overview of delivered files and file structure

SOLIDO is installed (when the user accepts the default destination) in the folder C:\Program Files (x86)\Physibel.

The following files are installed:

Folder	*File*	*Description*
...\Physibel\SOLIDO5	SOLIDO5.exe	SOLIDO (64-bit) program
	*.dll, *.pat, *.rgb	Program resources
	QlmLicenseWizard	Software Licence Manager files
...\Physibel\Materials	*.phm	Material databases
...\Physibel\Manuals	SOLIDO5Manual.pdf	SOLIDO manual
	SOL***.pdf	SOLIDO How to's

SOLIDO tutorials are installed in Documents/Physibel/SOLIDO5/Tutorials (Personal Folder).

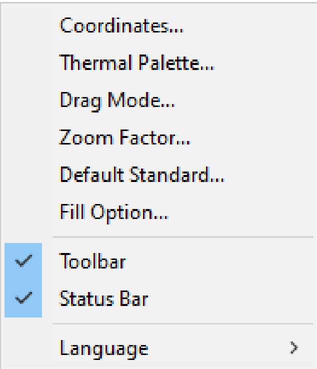
The colour database file ColourDatabase.sld and back-up ColourDatabase_Physibeldefault.sld (see F.2) together with the parameter file SOLIDO.prm (see A.11) are installed in C:/ProgramData/Physibel/SOLIDO5 (machine-level), and copied to each user account's application data folder when first opening SOLIDO (C:/Users/*username*/AppData/Roaming/Physibel/SOLIDO5).

A.8. Text convention

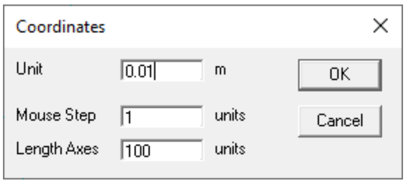
Menu paths are printed in *italic*, with format *Submenu → Command.*
E.g. *File → Open* (Open command in File submenu).

A.9. Program settings

Global program settings can be edited via the main menu (*Settings → ...*).

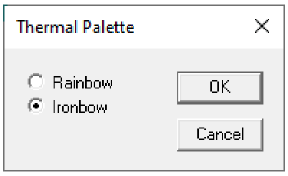


– *Settings → Coordinates...*

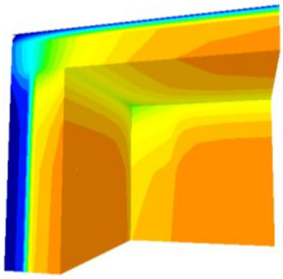


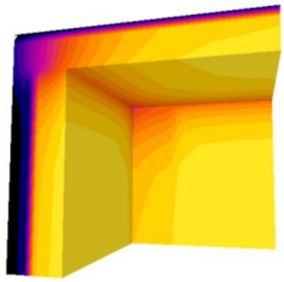
The parameters in this dialog all refer to the world coordinate system (see B.7.9), and are saved in the SOLIDO data file.
<u>Unit</u>
Specifies the length unit in metres of all coordinate measurements.
Defines the scale of the object.
<u>Mouse step</u>
Length of smallest mouse movement in the **Image window**, when using mouse dragging of block points (*Image → Drag Point*, see B.4.10) or blocks (*Image → Drag Block*, see B.4.11).
<u>Length axes</u>
Visible length of coordinate axes in the **Image window** (red line for X axis, green line for Y axis, blue line for Z axis).

– *Settings → Thermal Palette...*: to select the thermal palette used to visualize the temperature gradient in the solution.

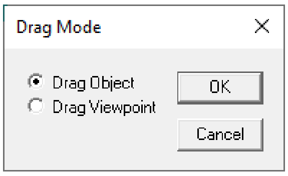


The user can choose between the traditional Physibel 'Rainbow' thermal colour palette, or the 'Ironbow' thermal colour palette, which is often used by IR camera's.



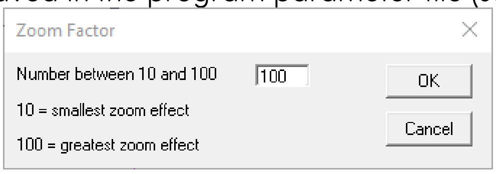


– *Settings → Drag Mode...:* to define how the view of an object (in the **Image window** or the **Graphic Output window**) changes as a result of a rotate or pan command (using the mouse or an arrow key).

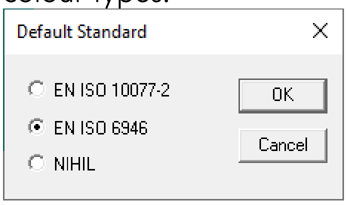


"Drag object" moves the viewpoint position with respect to the object. "Drag viewpoint" moves the object with respect to the viewpoint.
The selected drag mode is saved in the program parameter file (see A.11).

– *Settings → Zoom Factor...:* to customise the zooming factor of the mouse scroll wheel. The selected zoom factor is saved in the program parameter file (see A.11).



– *Settings → Default Standard...*: set EN ISO 10077-2, EN ISO 6946 or NIHIL as default standard for the definition of colour types.



– *Settings → Fill Option...*: see B.4.6 for a detailed explanation.
– *Settings → Toolbar*: presence of the toolbar giving access to menu commands through icons.
– *Settings → Status Bar*: presence of the status bar at the bottom of the SOLIDO application window, containing context depending information.
– *Settings → Language...*: to select the user interface language (English, Dutch, French, German). The selected language is saved in the parameter file (SOLIDO.prm). After exiting SOLIDO (*File → Exit*) and restarting SOLIDO, the new language is active.

A.10. General editing commands

<u>Copy – paste</u>

– *Edit → Copy Value*: copies the value from the active input cell into the clipboard.
– *Edit → Paste Value*: copies the value from the clipboard into the active input cell.

Undo – redo

- *Edit → Undo*
  Undoes the last edit action (e.g. to cancel a mistake).
  This command can be repeated to undo more previous edit actions.
  Only the last 100 edit actions are stored for undo.
  The undo command is impossible after saving a file, closing the current file or starting a calculation.
- *Edit → Redo*
  Restores the previous edit action that was made undone using *Edit → Undo*.
  In other words: "redo" undoes "undo".
  This command can be repeated to restore more edit actions that were made undone.

A.11. Program parameters

The global default values (calculation parameters, output parameters, interface language, etc.) are saved in the file SOLIDO.prm in the application data folder (C:/Users/*username*/Appdata/Roaming/Physibel/SOLIDO5). When the program SOLIDO is started, the program parameters are loaded from the parameter file.

The program parameter default values can be changed from several input dialog boxes. When the button "Set As Default" in an input dialog box is clicked, the corresponding program parameters are saved in the program parameter file SOLIDO.prm at exiting SOLIDO.

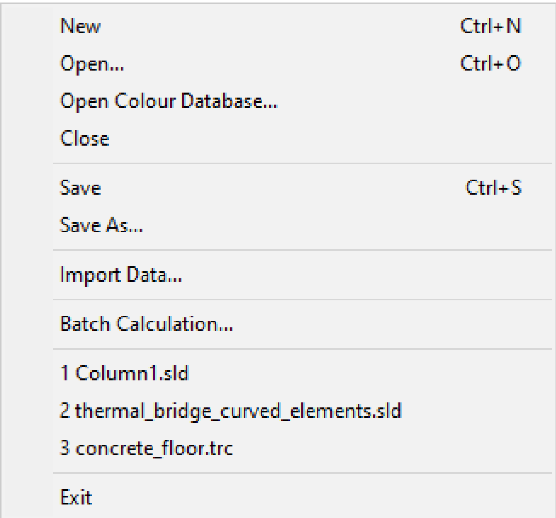
CHAPTER B EDITING INPUT DATA

B.1. SOLIDO start up

When SOLIDO is launched, the parameter file SOLIDO.prm is read (if present) to load global program settings (see A.11).
Then the default colour database ColourDatabase.sld (see F.2) is read (if present) to initialise material and surface boundary condition properties of the **Colours window**, which is useful to minimise the input for new data files.

B.2. File functions

File submenu:



B.2.1. New project

Menu command: *File → New*

This command opens a new project, after closing the current open project (if present).
The new project starts from a default cubic block.

All parameters are reset to their default values (as stored in the parameter file SOLIDO.prm, A.11), except for the colour definitions. This allows to reuse the colour definitions of the previous open file.

B.2.2. Open project

Menu command: *File → Open...*

After closing the current open project (if present), an Open dialog box appears.
Select a SOLIDO data file (with extension .sld) from the list in the current folder. If necessary, choose another directory.

When a file is opened, all parameters are reset to their default values (as stored in the parameter file SOLIDO.prm), except for the colour definitions. This allows to reuse the colour definitions of the previous open file. Then all data from the chosen SOLIDO data file are read and loaded into memory. Colour definitions that are not present in the data file, remain untouched.

How to load a solution file

When a solution file (with extension .sol, possibly also a file with extension .flx) is present, the following steps must be accomplished to load it:

- Open the data file (*File → Open...*). This shows the *wireframe object* in the **Image window**.
- Calculate the *triangulated object* (*Image → Fill Blocks (Triangulate)*).
- Calculate the *rasterised object* (*Image → Fill Grid (Rasterise)*).
- Calculate the *fitted object* (*Image → Fit Grid*).
  At the end of this step the solution file is automatically loaded.

To view the calculated temperatures, execute the following command:

- *Image → Show Temperatures* to show the calculated temperatures in the **Image window**.

B.2.3. Open the colour database

Menu command: *File → Open Colour Database...*

The colour database contains predefined material and boundary condition properties (stored in the file ColourDatabase.sld in the application data folder, see F.2), which is loaded at every SOLIDO start-up (see B.1). It is advisable to add frequently used materials and boundary conditions to the colour database.

These predefined colours of the colour database are then available for new colours in any document.

B.2.4. Import TRISCO data

Menu command: *File → Import Data...*

Files of type: TRISCO Data Files (*.trc)

TRISCO data are saved in .trc files. The corresponding solutions are saved in .sol and .flx files. The .flx files are only created when the RADCON module is used in TRISCO.

SOLIDO can read .trc and corresponding .sol files (not the .flx files).

The node boundary conditions are not read, because these are not defined in SOLIDO.

When TRISCO data containing up to 100 blocks are imported, automatically the following program options are set:

- Show grid number coordinates instead of world coordinates (cf. command *Blocks → Grid Number Coordinates*, see B.7.9).
- Link grid to block points (cf. command *Grid → Link Block Points to Grid*, see B.11.12).
- Automatic fill blocks (cf. command *Settings → Fill Option,* see B.4.6).

The results of the calculations in TRISCO and in SOLIDO for the same data are identical, except for minor differences in object surface temperatures (where SOLIDO applies an additional local smoothing of surface temperatures, cf. section D.8).

See B.20 for hints on using SOLIDO for users familiar with TRISCO.

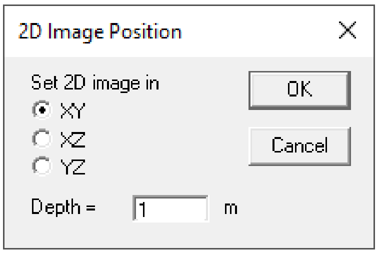
B.2.5. Import Trisco2D data

Menu command: *File → Import Data...*

Files of type: Trisco2D Data Files (*.tr2)

Trisco2D data are saved in .tr2 files which can be read by SOLIDO.

The 2D geometry can be set in XY, XZ or YZ. The size of the third (extrusion) dimension is a user input (Object depth).



When Trisco2D data containing up to 100 blocks are imported, automatically the following program options are set:

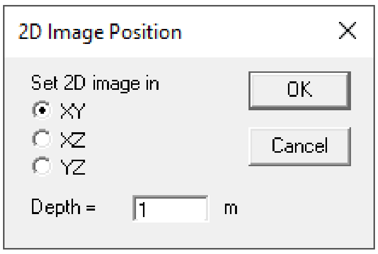
- Show <u>grid number coordinates</u> instead of world coordinates (cf. command *Blocks → Grid Number Coordinates*, see B.7.9).
- <u>Link grid to block points</u> (cf. command *Grid → Link Block Points to Grid*, see B.11.12).
- <u>Automatic fill blocks</u> (cf. command *Settings → Fill Option*, see B.4.6).

B.2.6. Import BISCO data

Menu command: *File → Import Data...*
Files of type: BISCO Data Files (*.bsc)

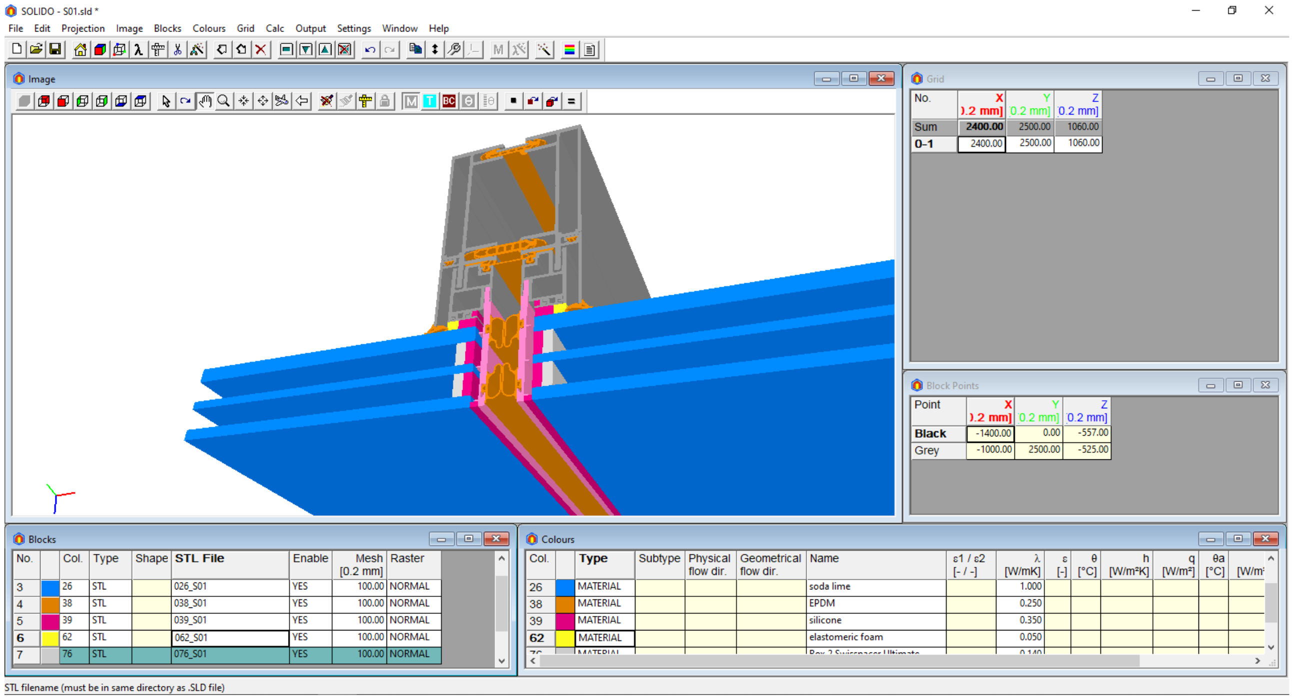
Required input files (all having the same file name):
- .bsc file (with thermal properties and coordinate unit)
- .bmp file (for colour palette definition)
- .tri file (for triangulation data of material colours)

BISCO is the Physibel program to calculate two-dimensional steady state heat transfer in free-form objects. BISCO is the 2D equivalent program of the 3D program SOLIDO. The input of BISCO is based on a 256 colours bitmap (in BMP format). The thermal properties are defined in an analogous way as in SOLIDO (through a **Colours window**), and saved in a .bsc file. When a calculation is started in BISCO, the material colours (type MATERIAL and EQUIMAT) of the bitmap are first triangulated and saved in a .tri file. This .tri file is imported in SOLIDO together with the .bsc file and .bmp file, and is interpreted as base of an extruded object. The bitmap can be set in XY, XZ or YZ. The size of the third (extrusion) dimension is a user input (Object depth).



The individual bitmap colours (only of material colours) are converted in SOLIDO to blocks of type STL. There is one STL block per bitmap (material) colour. BISCO boundary conditions are not triangulated in the .tri file, and therefore not imported in SOLIDO. If necessary first modify in BISCO the boundary conditions into materials, before calculating the .tri file. The conversion from the .tri file into STL blocks in SOLIDO is straightforward, because both .tri and STL blocks contain triangulation data. The STL blocks are automatically saved to physical .stl files (as separate files on disk) after importing the BISCO file, in the directory of the BISCO file.

Before starting a calculation in SOLIDO for imported BISCO data, boundary conditions have to be added, the calculation grid has to be defined (in the **Grid window**), and the calculation parameters must be set.



B.2.7. Export TRISCO data

Menu command: *File → Save As...*
Save as type: TRISCO Data Files (*.trc)

Saves the *rasterised object* (after *Image → Fill Grid (Rasterise)*) as rectangular blocks in a TRISCO data file. The TRISCO blocks do not use any superposition; the blocks are set side by side (in juxtaposition, without any overlap).
All colour definitions, border boundary conditions and calculation parameters are written into the .trc file.

<u>Remark on file names</u>

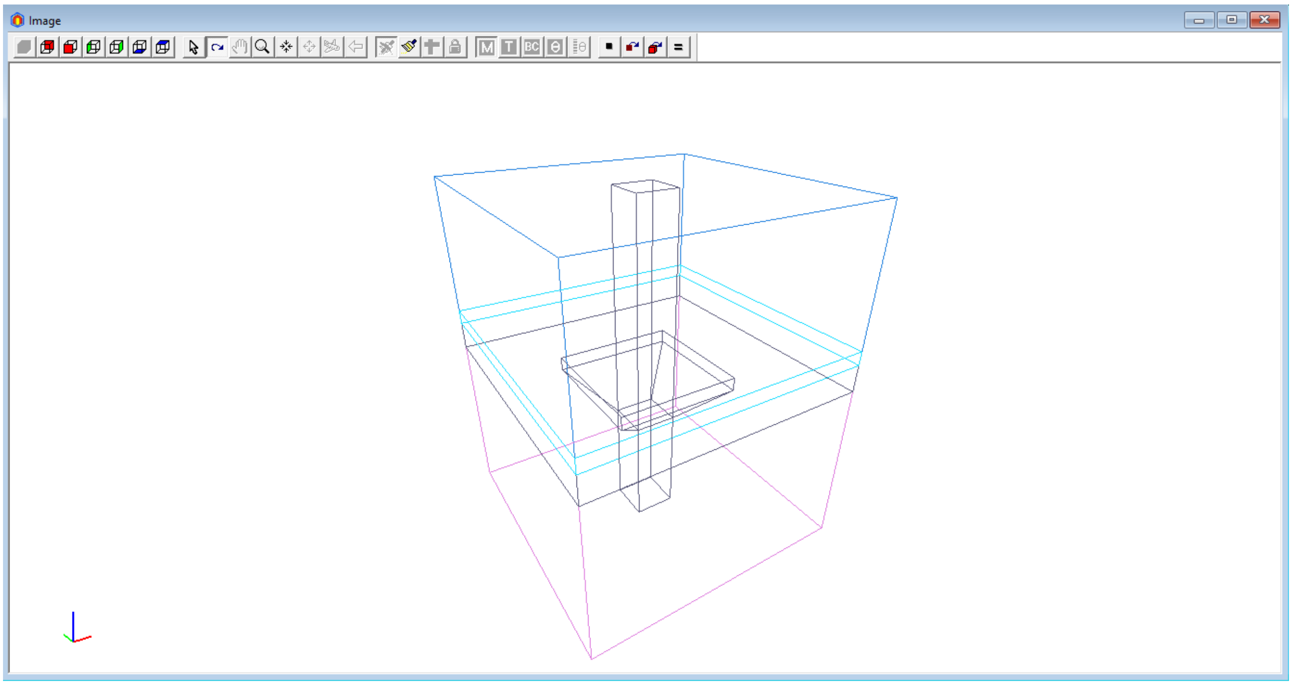
When saving as a TRISCO data file, it is highly recommended to change the file name from the default (i.e. the file name of the current SOLIDO project) when saving in the same location as the current SOLIDO project. Otherwise, a TRISCO simulation would overwrite the .sol results file from a SOLIDO simulation or vice versa.

B.3. Image window

Menu command: *Edit → Image*

The **Image window** shows a view of the object model (*wireframe object, triangulated object, rasterised object* or *fitted object*). The view of the object in the **Image window** can be altered using image functions. In the bottom left corner of the **Image window**, the 3 main coordinate axes are shown for reference (red for X, green for Y, and blue for Z).

After any geometric edit operation the model returns to a *wireframe object*, which is not filled (as in the picture above). The object must then be converted into the different filled models (*triangulated object, rasterised object, fitted object*) as required steps before the system calculation can be executed.



Possible OpenGL errors

In the **Image window** hardware accelerated graphics is used with OpenGL functions. When OpenGL produces an error in a very complex image (e.g. due to not enough memory), image painting is interrupted and a partial (or empty) object representation may be shown. Long image repainting can be interrupted by pressing the <Esc>key. The full image can always be shown in the **Graphic Output window**, which does not use OpenGL.

B.3.1. View coordinates

When the **Image window** is active (i.e. its title bar is blue), the coordinates X, Y and Z of the object point under the mouse cursor point is displayed in the status bar.

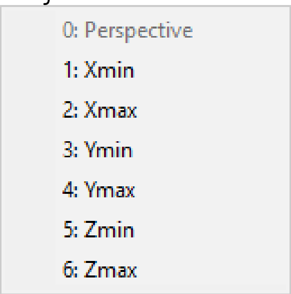
In the *wireframe object* only coordinates of block points can be shown.

After a fill operation (*triangulated object, rasterised object* or *fitted object*), the coordinates of any object surface point can be shown. When the temperature solution is available and the option *Image → Show Temperatures* (see B.4.7) is activated, also the temperature in that point is shown.

If the filled object is too complex (i.e. more than 100,000 triangles in the *triangulated object*, or for the *rasterised object* or *fitted object* more than 1,000,000 grid voxels), then the automatic coordinates view is disabled (because it would be too slow), and the command *Image →Pick Coordinates* (see B.4.8) should be used.

B.3.2. Projection

Projection submenu

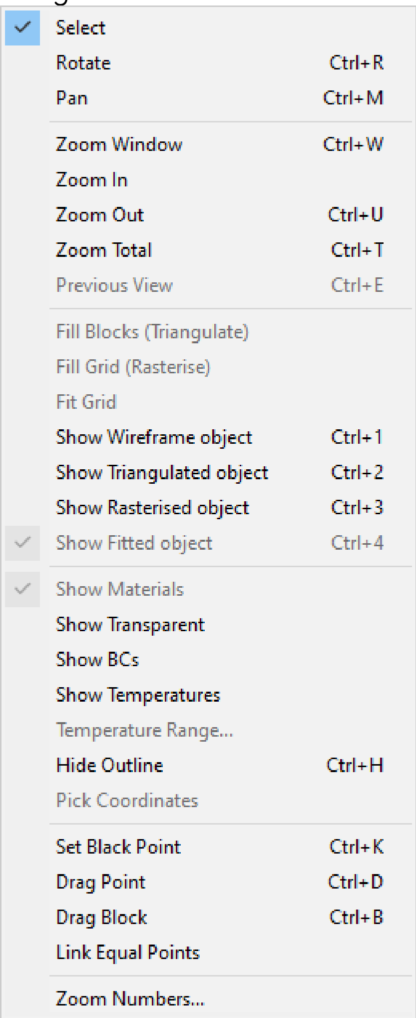


The projection of the model in the **Image window** can be toggled between

- *Projection → Perspective*: perspective view of the 3D model.
- *Projection → Xmin...:* 2D orthogonal view of the model onto the YZ plane with minimum X coordinates, looking in the positive X direction.
- *Projection → Xmax...:* 2D orthogonal view of the model onto the YZ plane with maximum X coordinates, looking in the negative X direction.
- *Projection → Ymin...:* 2D orthogonal view of the model onto the XZ plane with minimum Y coordinates, looking in the positive Y direction.
- *Projection → Ymax...:* 2D orthogonal view of the model onto the XZ plane with maximum Y coordinates, looking in the negative Y direction.
- *Projection → Zmin...:* 2D orthogonal view of the model onto the XY plane with minimum Z coordinates, looking in the positive Z direction.
- *Projection → Zmax...:* 2D orthogonal view of the model onto the XY plane with maximum Z coordinates, looking in the negative Z direction.

B.4. Image commands

Image submenu:



B.4.1. Viewpoint

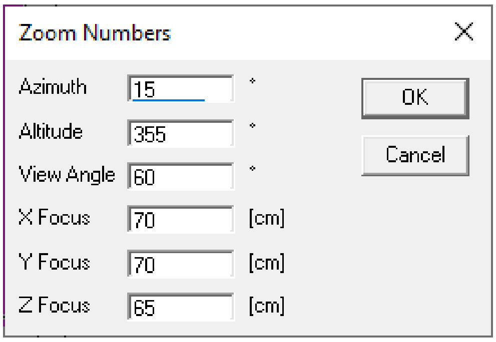
The viewpoint can be rotated or panned (i.e. moved laterally).

- *Image → Rotate*
  Enters rotate mode. Mouse dragging or pressing arrow keys (left, right, up, down, with **Image window** active) rotates the relative position of the viewpoint with respect to the object centre.
  The Z axis of the XYZ coordinate system should point upwards to the sky to get a good behaviour of the rotate function.
  Rotate mode is activated automatically by holding down the mouse wheel when the **Image window** perspective (3D) projection is active (see B.3.2).
- *Image → Pan*
  Enters pan mode (only possible when the current view is already zoomed in). Mouse dragging or pressing arrow keys (with **Image window** active) moves the relative position of viewpoint and object centre parallel to the projection plane.
  Pan mode is activated automatically by holding down the mouse wheel when the **Image window** 2D orthogonal projection is active (see B.3.2).

The command *Settings → Drag Mode* allows to define the relative movement of viewpoint and object (as either “drag viewpoint” or “drag object”, cf. section A.9).

B.4.2. Zoom commands

- *Image → Zoom Window*
  Enters "zoom window" mode. This allows to draw a zoom rectangle using the mouse in the **Image window**, which defines the borders of a zoomed view.
  Press Esc to cancel the current zoom rectangle (before releasing the left mouse button).
- *Image → Zoom In*
  Enters "zoom in" mode. A following click using the left mouse button in the Image window defines the centre point of the next zoomed view. The zoom factor is a program setting (see A.9).
  Zooming in is also possible using forward scrolling with the mouse wheel (when the **Image window** is active).
- *Image → Zoom Out*
  Zooms out using the current zoom factor (or restores the total zoom, when zoomed out enough).
  Zooming out is also possible using backward scrolling with the mouse wheel (when the **Image window** is active).
- *Image → Zoom Total*
  Shows total object.
- *Image → Previous View*
  Shows the previous zoom view. (More than one previous view cannot be recalled.)
- *Image → Zoom Numbers...*
  Opens a dialog box with the current zoom parameters.
  Enter new parameters to change the view. Use preferably rounded numbers to facilitate restoring the same view at a later moment.



B.4.3. Fill blocks (Triangulate)

Menu command: *Image → Fill Blocks (Triangulate)*

A triangulation of all block surfaces is calculated and painted in the **Image window**. The resulted object model is called the *triangulated object.*

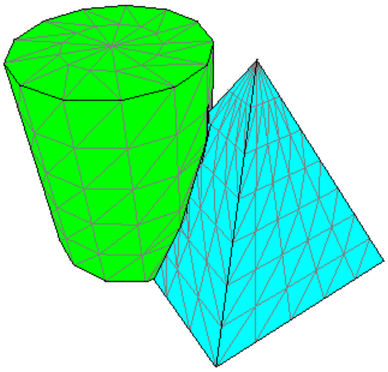
The triangulation of the boundary surfaces of each block (of type SOLID or STL; blocks of type POINT are ignored) are based on the mesh size of the block (as defined in the **Blocks window**). The intersections between the triangulated surfaces of the different blocks are determined, and the superfluous triangles are deleted following the superposition rule (blocks with higher block number overwrite blocks with lower block number).

Larger mesh sizes require less memory space, and allow a faster calculation of the *triangulated object*. Rounded solid surfaces (for solids having cylindrical or spherical shape) are made smoother using a smaller mesh size.

After the system calculation (see C.4), temperature images projected on the *triangulated object* are interpolated between the temperatures in the vertices of the *triangulated object*. The temperature field is smoother for smaller block mesh sizes.

Triangulation mesh visualisation
The triangulation mesh of the *triangulated object* can be viewed in the **Graphic Output window**. Use the following commands:

– *Image → Fill Blocks (Triangulate)* (to calculate the triangulation mesh)
– *Output → Graphic Output* (to switch to the **Graphic Output window**)
– *View → Grid Lines* (to show the triangulation mesh)



Disable fill blocks
The command *Fill Blocks* can be disabled using the command *Setting → Fill Option...*, in order to switch off the calculation of block intersections (cf. section B.4.6).

B.4.4. Fill grid (Rasterise)

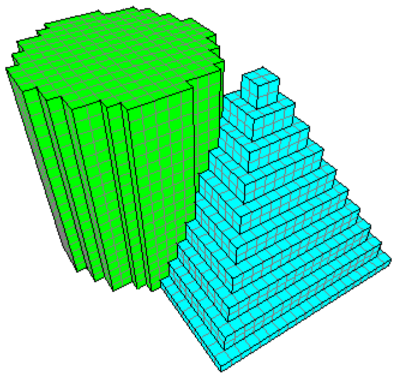
Menu command: *Image → Fill Grid (Rasterise)*

Input: *triangulated object* (i.e. after *Image → Fill Blocks (Triangulate)*) + grid definition (in **Grid window**) + rasterisation method per block (in **Blocks window**, column "Raster")
Output: *rasterised object*

Starting from the grid definition (in the **Grid window**) a 3D voxel space is set up, in which each *voxel* (as short name for "volume element", by analogy with *pixel* in 2D as short name for "picture element") corresponds to an elementary cell in the rectangular grid. The width, depth and height of each voxel correspond to three individual grid mesh sizes along X, Y and Z in the **Grid window**. The voxels are coloured based on the colours in the voxel centres with reference to the *triangulated object* (as obtained after *Image → Fill Blocks (Triangulate)*, starting from the blocks of type SOLID and STL).
Thereafter the swelling blocks (as defined in the **Blocks window**, having the rasterisation attribute SWELL instead of NORMAL) are considered. If a voxel contains any part of a triangulated surface (of the *triangulated object*) that borders a swelling block, then it is repainted with the colour of that block.
Finally the blocks of type POINT are considered. For each block of type POINT a seed fill is applied using the block colour starting from the seed point position.
The final result in the voxel space is called the *rasterised object*.

Voxel space
The **Grid window** defines the measurements of the individual voxels along X, Y and Z.
The total number of voxels equals the product of the numbers of grid meshes along X, Y and Z.
For best performance the total number of voxels should be not greater than 1,000,000.
The fill grid operation determines a colour value (from the present colours in the **Colours window**)

for each voxel. Voxels outside all blocks (defined in the **Blocks window**) remain uncoloured. These void voxels can be detected using the command *Blocks → Convert Void to Blocks...* (see B.7.6).

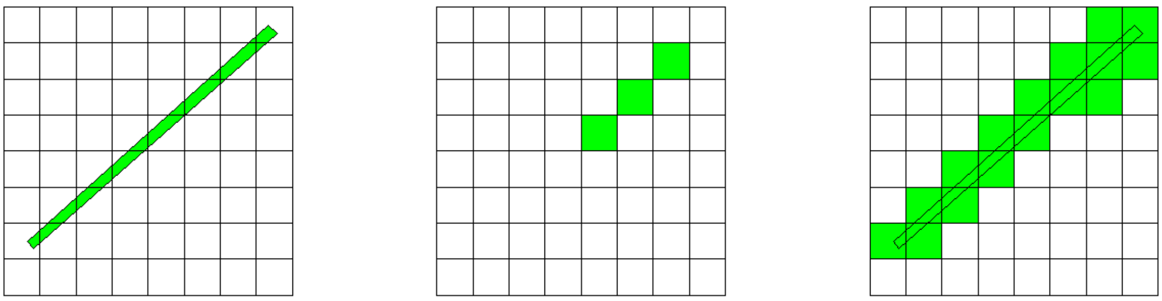


Rasterisation based on voxel centres

The rasterisation is based on the colour values in the voxel centres. Each voxel (as a volume) gets the colour value obtained in its centre (as a point). This method is called "aliasing". It is possible that a voxel compared to the *triangulated object* contains several subparts in different colours. The colour of the subpart that contains the voxel centre (even if this subpart is smaller than other subparts with different colours), is determinant for the whole voxel volume.

Rasterisation of swelling blocks

After the rasterisation of all blocks based on the voxel centres, the blocks with attribute SWELL in the **Blocks window** ("swelling blocks") are considered.
If a voxel contains any part of a triangulation surface bordering a swelling block, then this voxel is recoloured using the colour of the swelling block. The effect is that materials composed of thin plates or thin pins defined by swelling blocks, that would normally disappear in the voxel space due to aliasing, now are recovered.
If a material layer is thin compared to the grid mesh (cf. figure below left), normal rasterising (with respect to the centres of the voxels) can make it disappear partially or even completely in the raster image (cf. figure in the middle).



The "raster option" in the **Blocks window** allows to alter the rasterisation method for each individual block. The option NORMAL is the default. The option SWELL also adds all voxels passing through the triangulated block surface (regarding the entire voxel volume and not only the voxel centre), after finishing the normal rasterising for all colours (based on the colours in the voxel centres). The figure above right shows the result after swelling up thin material block.

A swelling block with higher priority in the blocks list (i.e. with higher block number in the **Blocks window**) overwrites the voxels of swelling blocks with lower priority.

Processing blocks of type POINT

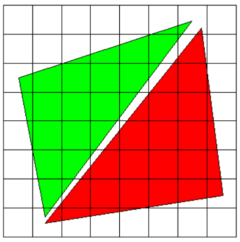
When blocks of type POINT are present in the **Blocks window**, these blocks are taken into account after the rasterisation based on voxel centres and the rasterisation of swelling blocks.
Traverse the list of blocks from the top to the bottom (i.e. with increasing block number).

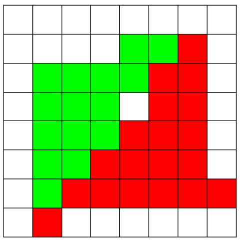
For each block of type POINT the position is searched with reference to the grid, where the corresponding (black) block point (defined in the **Block Points window**) is lying. If this point is lying within a grid voxel, the colour of this voxel is replaced with the colour attributed to the considered block (of type POINT). This recoloured voxel is then taken as start for a seed fill in the voxel space, so that all neighbouring voxels having the same colour as the original colour of the start voxel (the seed) are also recoloured using the new colour (of the considered block of type POINT). The seed fill never crosses neighbouring voxels having other colours. In this way the connected colour region is replaced by the new colour.

<u>Rasterised object</u>
The final result of the voxel colouring is called the *rasterised object*.
All vertices of material voxels (i.e. voxels having a colour of type MATERIAL or EQUIMAT, as defined in the **Colours window**) are system nodes in the thermal system.

<u>Possible rasterisation errors due to small gaps between object blocks</u>
Be careful with object blocks having common or touching surfaces.
The *triangulated object* may hold small gaps between originally coincident (flat or curved) surfaces that are triangulated using a different block mesh size.

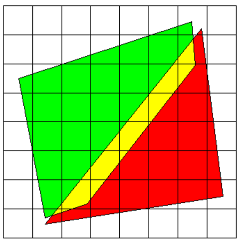
The following 2D example illustrates the problem.
The figure below left shows a *triangulated object* with a green triangle (top left) and a red triangle (bottom right) with quasi common edge.
A square grid is laid over the object.
Rasterising based on the colour in the centre of the grid cells yields the figure below right.

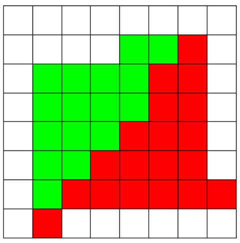




The *rasterised object* contains one white pixel between the green and red pixel regions (because the centre of this pixel lies accidentally in the small gap between both triangles), which causes a problem (through aliasing) in the further processing of the thermal calculation.
Space which is not occupied by any defined colour is supposed to be an adiabatic boundary condition (i.e. an environment without heat exchange to the object).

Often it is safer to make use of block superposition to avoid aliasing.
The figure below illustrates this solution.
Suppose the green triangle has lower priority than the red triangle (meaning the green block comes before the red block in the blocks list, as defined in the **Blocks window**).





Redefine the green block in the **Blocks window**, so that there is a common margin between the green and red block (which corresponds to the yellow colour in the figure above left), which

prevents the creation of small gaps in the triangulated object.
After rasterisation which takes account of the superposition rule, we obtain the figure above right without any gap between both colours.

The function *Blocks → Convert Void to Blocks...* allows to detect void voxel zones (i.e. not filled by a colour, hence adiabatic) from the *rasterised object* (see B.7.6).

<u>Object model validity</u>
Before continuing with the object nodes fitting (*Image → Fit Grid*, B.4.5) first check the validity of the created object model by showing alternately the material colours and the colours of the boundary conditions active upon the object surfaces (see B.4.7). In this way rasterisation errors (e.g. due to aliasing) may be revealed (cf. example above). It can be useful to apply transparent mode (see B.4.7) object clipping (see 0) or colour clipping (see B.16) to view the inside of the object model.

B.4.5. Fit grid

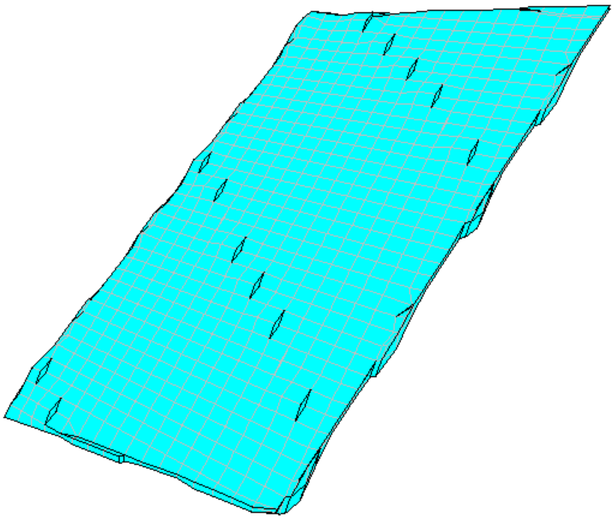
Menu command: *Image → Fit Grid*

Input: *triangulated object* + *rasterised object* (i.e. after *Image → Fill Grid (Rasterise)*, B.4.4)
Output: *fitted object*

The nodes on the surfaces between different colours of the *rasterised object* are fitted to the closest sharp vertices, sharp edges or triangle surfaces of the *triangulated object*. Preference is given to projection onto the closest sharp vertex, then onto the closest sharp edge and finally onto the closest triangle surface, in order to approximate the original object as close as possible.
A *sharp edge* is an edge of the *triangulated object* between two triangles bounding the same colour which form an angle greater than 30°.
A *sharp vertex* is a common endpoint between two sharp edges.

<u>Swelling blocks</u>
In general node fitting for swelling blocks (with attribute SWELL in the **Blocks window**) is less good than for normal colours (with attribute NORMAL), because the projection distances of the surface nodes to the triangulated object are larger, and thus more distortion is possible, compared to normal rasterisation (based on the voxel centres).

The figure below shows the *fitted object* of a 3D thin plate in a slanted position based on a coarse grid.



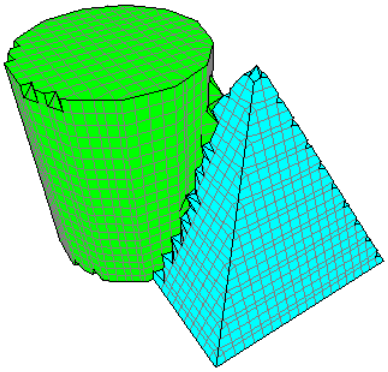
Volume test

After all node projections are finished, a volume integrity test is executed for each material voxel. If the volume of a material voxel using the new node coordinates is not positive (as required by the further system solution), the node fittings for the eight vertices of that voxel are moved gradually towards the original node positions in the rectangular grid until the volume becomes positive.

Fitted object

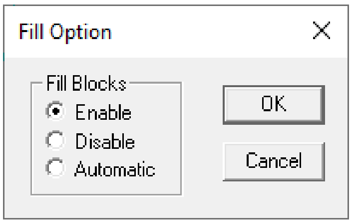
The output is called the *fitted object*.

The result of the intersection between different blocks may be crispy. This is illustrated in the figure below, which shows that the intersection between the cylinder and the pyramid of the *fitted object* is jagged in contrast to the smooth lateral surfaces of the cylinder and pyramid.



B.4.6. Fill blocks setting

Menu command: *Settings → Fill Option...*



One of three options can be selected:

- Enable Fill Blocks
  Option for normal use. With this option the best results for object fitting (*Image → Fit Grid*) are obtained, because intersection lines between different blocks are calculated (during *Image → Fill Blocks (Triangulate)*) and used as snapping lines in node fitting.
- Disable Fill Blocks
  Option for use for a complex object (for which the calculation of block intersections would take too much time and memory).
  The menu command *Image → Fill Blocks (Triangulate)* is disabled.
  The menu command *Image → Fill Grid (Rasterise)* first calculates a surface triangulation of the individual block surfaces, but does not calculate any block intersection lines (which is part of the disabled *Image → Fill Blocks (Triangulate*). After this step the normal object rasterising is executed.
  The menu command *Image → Fit Grid* fits the nodes of the *rasterised object*, lying in the interface between different colours, onto the *triangulated object*. No fitting is done to the closest sharp edges of triangle intersection lines, because these are not available.
  The system calculation (*Calc → Calc System*) delivers a more robust, but less refined calculation, because the quality of the node fitting in the *fitted object* is in general less good than with the option "Enable fill blocks" (near block intersections). This calculation option needs less memory and may be recommended for complex object models.
- Automatic Fill Blocks
  Option for use for a simple object (e.g. composed of only rectangular blocks).
  Imported TRISCO and Trisco2D data (see B.2.4 and B.2.5) automatically use this option up to 100 blocks.
  The object is always automatically filled after each edit action (as happens in the programs TRISCO and Trisco2D).

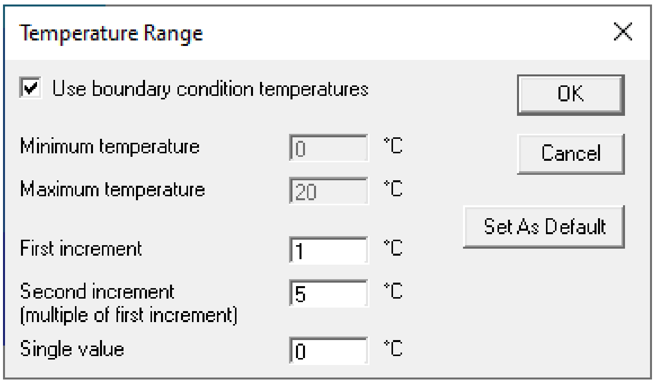
Disabling fill blocks can be useful when *Image → Fill Blocks (Triangulate)* has to solve a lot of block intersections between the different blocks in the blocks list. These intersections are dictated by the superposition rule for blocks.
Many intersections can cause the following problems:

- The contacts between the different colour regions are not always preserved, by insertion of narrow gaps between the colour regions, due to numerical round off errors in the calculations of intersection points (cf. section B.4.4).
- The calculation of intersection points demands additional memory and slows down the calculation speed.

B.4.7. Show commands

- Show Wireframe object
  Menu command: *Image → Show Wireframe object*
  Recalls the *wireframe object*. (No object clipping is possible.)
- Show Triangulated object
  Menu command: *Image → Show Triangulated object*
  Recalls the *triangulated object*. (No object clipping is possible.)
- Show Rasterised object
  Menu command: *Image → Show Rasterised object*
  Recalls the *rasterised object.*
- Show Fitted object
  Menu command: *Image → Show Fitted object*
  Recalls the *fitted object.*

- Show materials
  Menu command: *Image → Show Materials*
  Shows the colours of the material blocks, as defined in the **Colours window**.
- Show transparent
  Menu command: *Image → Show Transparent*
  Shows the object with transparent materials, so that the inside of the object can be viewed.
- Show boundary conditions
  Menu command: *Image → Show BCs*
  Shows the material blocks with the colours of the adjacent boundary conditions. If no boundary condition is active on a material surface (i.e. an implicit adiabatic boundary condition, without any heat exchange to the environment), a light yellow colour is used. This command is useful in verifying the action regions of the applied surface boundary conditions.
- Show temperatures
  Menu command: *Image → Show Temperatures*
  Shows the calculated temperature distribution on the material surfaces. The legend of the rainbow colours can be viewed in the **Graphic Output window** (D.6).
  The thermal palette making up the colours of the temperature scale are interpolated from values defined in a .rgb file (see A.9). This is a text file containing red, green and blue components (values from 0 up to 255) for increasing temperature values.
  The temperatures are calculated for the nodes in the *fitted object*. SOLIDO also allows to show the temperatures on the *triangulated object*. The temperatures are obtained by reverse fitting from the temperature field in the *fitted object*. The accuracy improves when the block mesh sizes, which define the precision of the object triangulation, get smaller. If required, recalculate the *triangulated object* (by executing *Image → Fill Blocks (Triangulate)*, B.4.3) after modifying the block mesh sizes in the **Blocks window**.
- Temperature range
  Menu command: *Image → Temperature Range...*
  Opens a dialog box to define the parameters of a temperature view.
  The option “Use boundary condition temperatures” initialises the minimum and maximum temperatures from the boundary condition temperatures of types BC_SIMPL present in the **Colours window**.
  The first increment is used as step value in *Image → Show Temperature*.
  (The second increment is only used with isothermal lines in the **Graphics Output window**)
  This command is only enabled when the menu option “Show Temperatures” is checked.



- Hide selection outline
  Menu command: *Image → Hide Selection Outline* (operates as an on/off switch).
  The current item in an input window is highlighted in the **Image window** when this toggle is switched off. There are no selection feedback lines in the **Image window** when the toggle is switched on.

B.4.8. Pick coordinates

Menu command: *Image → Pick Coordinates*

For simple objects the coordinates are dynamically shown when moving the mouse over the object in the **Image window** (cf. section B.3.1). For more complex objects this dynamic view of coordinates is automatically disabled, because it would be too slow. In that case use the function *Image → Pick Coordinates.*
Each following click with the left mouse button in the **Image window** shows the coordinates (and temperature) of the point on the object surface under the mouse cursor point. The coordinates are displayed in the status bar (at the bottom of the SOLIDO application window).

B.4.9. Set black block point

Menu command: *Image → Set Black Point*

Enters the input mode for defining new black point locations. The black point serves as the anchor point or local origin of the block and may be useful in modifying a block shape (in particular for non-symmetrical blocks of type SOLID, such as PBAL and PCYL, for which the geometry is defined based on the location of the black point).

The input only concerns blocks of type SOLID (as defined in the **Blocks window**) of the *wireframe object*.
From the *triangulated object*, *rasterised object* or *fitted object*, switch back to the *wireframe object* by choosing *Image → Fill Off*.

After choosing *Image → Set Black Point*, each following click with the left mouse button in the **Image window** with the mouse cursor point upon a vertex of a SOLID block defines this vertex as new black point of the corresponding block.
As a result the **Block Points window** and **Image window** are repainted, showing the adjusted block points.

Modifying the black point location may affect the shape of the block.

B.4.10. Drag block point

Menu command: *Image → Drag Point*

Enters the input mode for dragging block points in the **Image window** to new positions using the mouse (thereby modifying the shape of the block).

Only the coloured block points (which are defined in the **Block Points window**) of a block of type SOLID can be dragged (see B.7.7). From the *triangulated object*, *rasterised object* or *fitted object*, dragging a block point will result in automatically switching back to the *wireframe object* (unless Automatic Fill Blocks is enabled, see B.4.6).

First activate the block in the **Blocks window** by clicking on it to show the block outline and the coloured block points (the coordinates of which are shown in the **Block Points window**) in the **Image window**.

To drag a block point in the **Image window**:
- Pick the block point with the left mouse button. Keep the mouse button pressed down.
- Move the mouse cursor (while pressing down the left mouse button) to drag the block point in a plane parallel to XY.
- Alternatively, hold down the <Shift> key to drag the block point in a direction parallel to the X axis, or hold down the <Alt> key to drag the block point in a direction parallel to the Y axis, or hold down the <Ctrl> key to drag the block point in a direction parallel to the Z axis.
- The coordinates change in multiples of the defined "mouse step" (via *Settings → Coordinates...*, see A.9).
- When the dragged point is moved over another block point (possibly of another block), the dragged point adopts the coordinates of that other point (snapping, indicated by the cursor changing its shape from the standard arrow to a cross). Snapping to any other block point is always possible, overruling unidirectional dragging holding down the <Shift>, <Alt> or <Ctrl> key.
- When the option *link equal points* is enabled (using the command *Image → Link Equal Points*, see B.4.12), coincident block points (belonging to the same block or to other blocks) are fixed to the dragged point and get the same new coordinates.
- To cancel the current drag operation, press the <Esc> key before releasing the left mouse button.
- The current drag operation is finished by releasing the left mouse button.
- The new coordinates of the dragged block point are shown in the **Block Points window** upon releasing the left mouse button.

B.4.11. Drag block

Menu command: *Image → Drag Block*

Enters the input mode for dragging blocks in the **Image window** to new positions using the mouse.

A block of type SOLID can be picked up for dragging on any block point. A block of type STL can be picked up for dragging on any block point or on any vertex of the triangulated surface of the STL file. In the latter case, this point is indicated in blue while dragging the block. If there are one or more blocks selected in the **Blocks window** (see B.5), only the selected blocks are available for dragging and the selection as a whole is moved (thus allowing to move multiple blocks in a single dragging move).

From the *triangulated object*, *rasterised object* or *fitted object*, dragging one or more blocks will result in automatically switching back to the *wireframe object* (unless Automatic Fill Blocks is enabled, see B.4.6).

To drag one or more blocks in the **Image window**:
- Pick the block point or STL vertex with the left mouse button. Keep the mouse button pressed down.
- Move the mouse cursor (while pressing down the left mouse button) to drag the block(s) in a plane parallel to XY.
- Alternatively, hold down the <Shift> key to drag the block(s) in a direction parallel to the X axis, or hold down the <Alt> key to drag the block(s) in a direction parallel to the Y axis, or hold down the <Ctrl> key to drag the block(s) in a direction parallel to the Z axis.

– The coordinates change in multiples of the defined "mouse step" (via *Settings → Coordinates...*, see A.9).
– When the dragged point is moved over another block point or STL vertex, the dragged point adopts the coordinates of that other point (snapping, indicated by the cursor changing its shape from the standard arrow to a cross). Snapping to any other block point is always possible, overruling unidirectional dragging holding down the <Shift>, <Alt> or <Ctrl> key.
– The option *link equal points* (*Image → Link Equal Points*, see B.4.12) is never used when dragging blocks.
– To cancel the current drag operation, press the <Esc> key before releasing the left mouse button.
– The current drag operation is finished by releasing the left mouse button.
– The new coordinates of the block points of the dragged block are shown in the **Block Points window** upon releasing the left mouse button.
– Dragging a block of type STL transforms the coordinates of the vertices of the STL file. When the user saves the SOLIDO data file (.sld), the option is therefore given to either overwrite the existing STL file(s) or save the changed STL file(s) with a different name (see B.7.4).

B.4.12. Link equal block points

Menu command: *Image → Link Equal Points* (operates as an on/off switch).

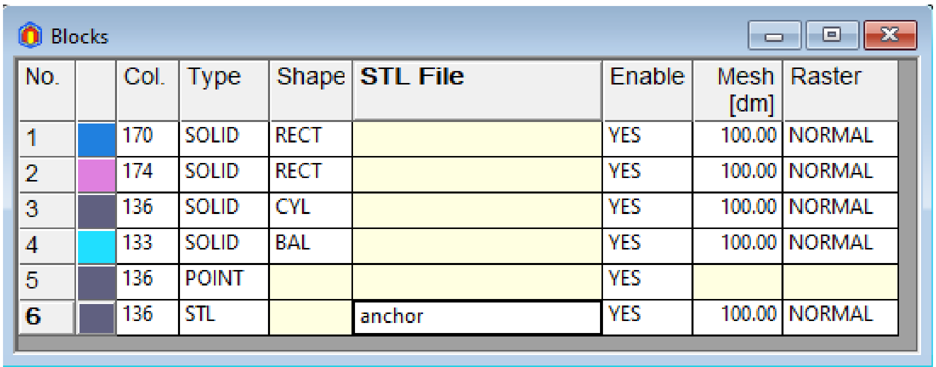
Enables or disables the option to link coincident block points together.

It does not matter whether the coincident block points belong to the same block or to different blocks.

If the coordinates of one of the coincident block points are modified (via input in the **Block Points window** or via point dragging after choosing *Image → Drag Point*), the coordinates of the other linked block points are modified also, so that the block points remain coincident.

B.5. Blocks window

Menu command: *Edit → Blocks*


Blocks

No.		Col.	Type	Shape	STL File	Enable	Mesh [dm]	Raster
1		170	SOLID	RECT		YES	100.00	NORMAL
2		174	SOLID	RECT		YES	100.00	NORMAL
3		136	SOLID	CYL		YES	100.00	NORMAL
4		133	SOLID	BAL		YES	100.00	NORMAL
5		136	POINT			YES		
6		136	STL		anchor	YES	100.00	NORMAL

The **Blocks window** is the main input window for the geometric description of the object model. The object is composed of blocks, which are ordered in a list. Each block refers to either a solid, a STL file or a point, and has a colour, which refers to the **Colours window** (in which thermal properties are defined, 0). The coordinates of block vertices are defined in the **Block Points window**.

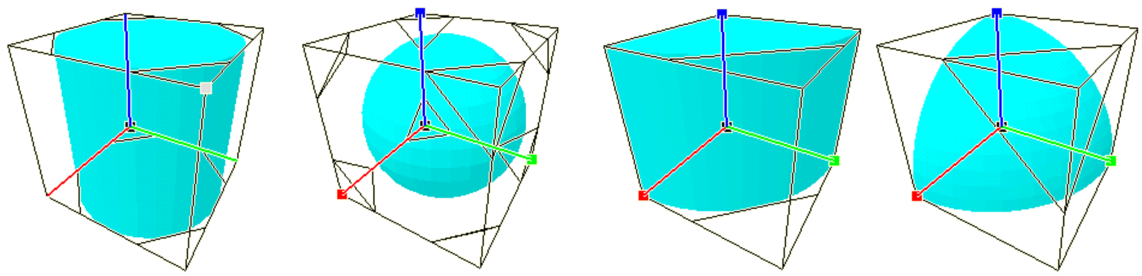
Blocks with higher block number have priority over (i.e. overwrite the common parts of) blocks with lower block number. This is called the *superposition rule*.

Fields per block

Column	Value	Description
Col.	0-255	Colour number (as reference to **Colours window**)
Type	SOLID	Solid block defined in SOLIDO
	STL	Block defined by external STL file
	POINT	Block defined by a seed point (cf. section B.4.4)
Shape	RECT	Rectangular shape
	CYL	Cylindrical shape
	BAL	Spherical shape (ball)
	PCYL	1/4 part of cylinder
	PBAL	1/8 part of sphere
STL File	File name	File name of imported STL file (must be stored in same directory as SLD data file)
Enable	YES	Correspondent block enabled in the model geometry
	NO	Correspondent block disabled in the model geometry
Mesh [unit]	Number	Mesh size measure for triangulation of the object block surfaces into the *triangulated object*. The mesh size affects the approximation precision of a curved or rounded surface and the quality of a temperature image in the *triangulated object* (B.4.3).
Raster	NORMAL	Rasterisation based on grid voxel centres
	SWELL	Swelling block (see B.4.4)

SOLID shapes

Apart from the default rectangle, the other possible shapes are a cylinder, a ball, 1/4 part of a cylinder and 1/8 part of a ball.



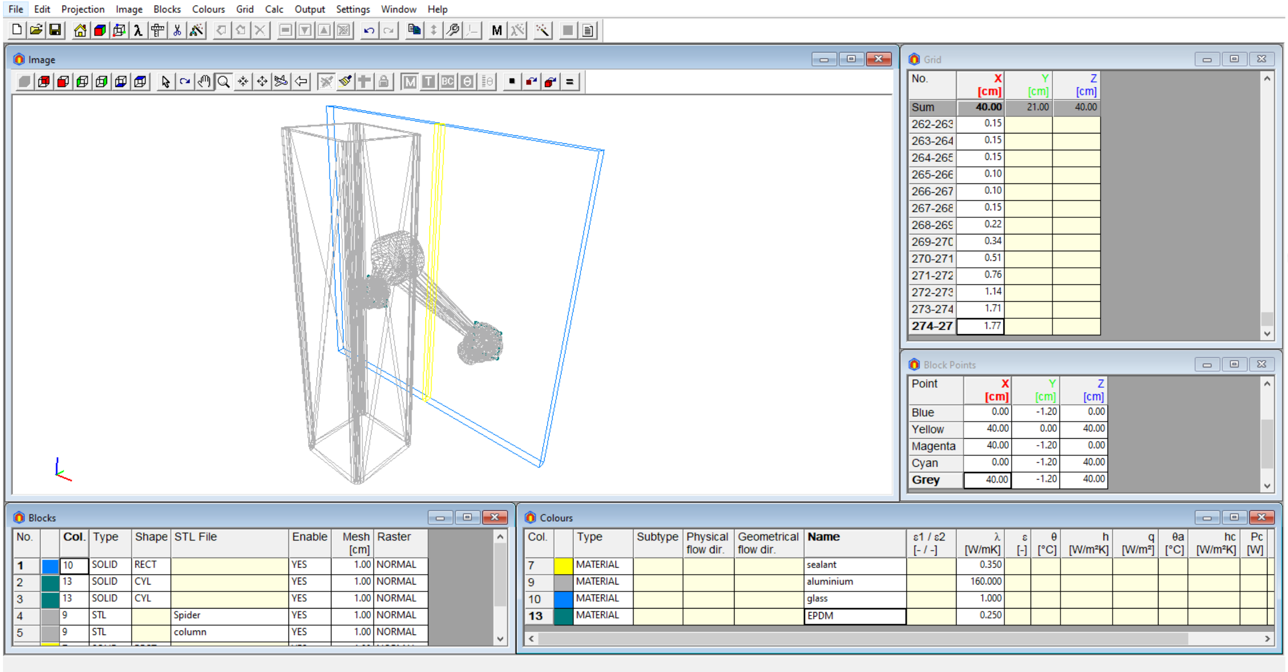
The position of the black block point (which is the anchor point for these rounded shapes) can be chosen to be any block vertex (see B.4.9) and clicking one of the 8 block vertices in the **Image window**.

STL file

A block can refer to an external STL file. The file name must be entered in the corresponding STL field by double clicking in this field.
A File Open dialog box appears, in which the STL file can be selected. By holding the <Ctrl> key, multiple STL files can be selected for loading. If multiple STL files are loaded, a new block for each STL file will be added in the **Blocks window**.
If the STL file has a name with prefix of the format "xxx_name", it forces a colour number (xxx = 3 digits, from 000 up to 255) to be used for the block.
The STL file(s) must be located in the same directory as the SOLIDO data file (with extension .sld).

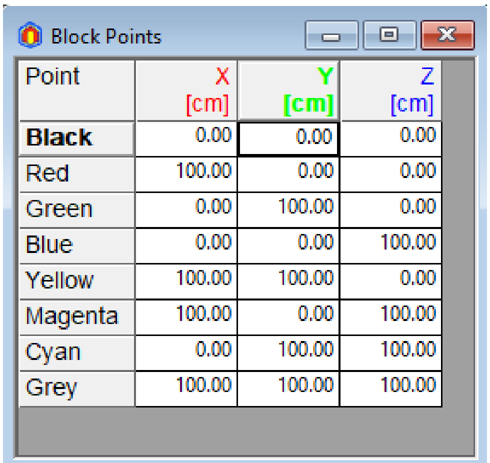
STL files ("Stereolithography Tessellation Language" format) can be exported from most CAD systems with 3D solid modelling (e.g. AutoCAD).

A STL file contains a boundary representation with triangles of a (composite) solid. In most CAD systems the user can specify a deviation tolerance with respect to the exact model, in order to minimise the number of triangles.

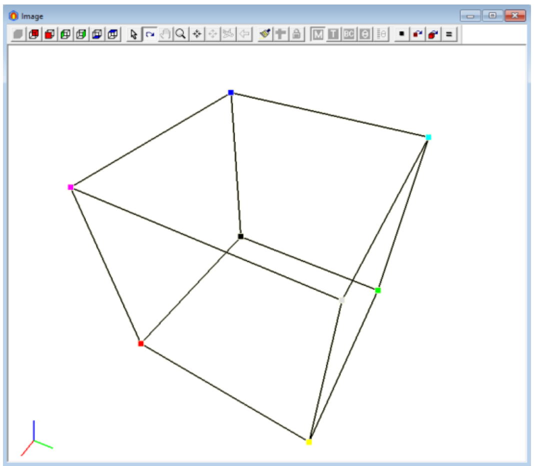


B.6. Block Points window

Menu command: *Edit → Block Points*



Point	X [cm]	Y [cm]	Z [cm]
Black	0.00	0.00	0.00
Red	100.00	0.00	0.00
Green	0.00	100.00	0.00
Blue	0.00	0.00	100.00
Yellow	100.00	100.00	0.00
Magenta	100.00	0.00	100.00
Cyan	0.00	100.00	100.00
Grey	100.00	100.00	100.00



The **Block Points window** is the input window for the coordinates of the colour-marked vertices of the active block in the **Blocks window**. The block point names in the first column refer to the colours of the points in the **Image window**.

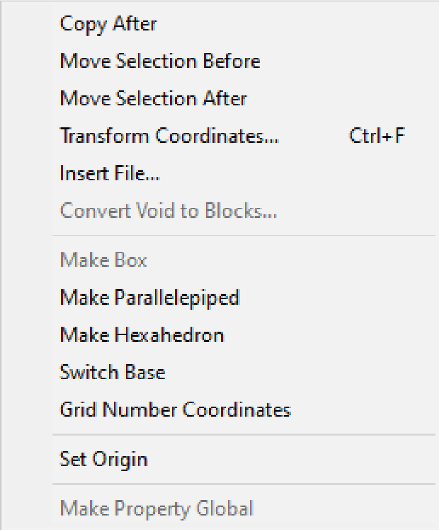
<u>Block types</u>

A block can have one of three different types, as defined in the **Blocks window**.

- SOLID: the block is defined by two, four or eight free block points in the **Block Points window** (see B.7.7) with editable coordinates.
- STL: the block is defined by a STL file, the bounding box of which is defined by two points in the **Block Points window** with non-editable coordinates.

– POINT: the block is defined by a single block point in the **Block Points window** with editable coordinates. This point is the seed of a seed fill during the rasterisation (see B.4.4).

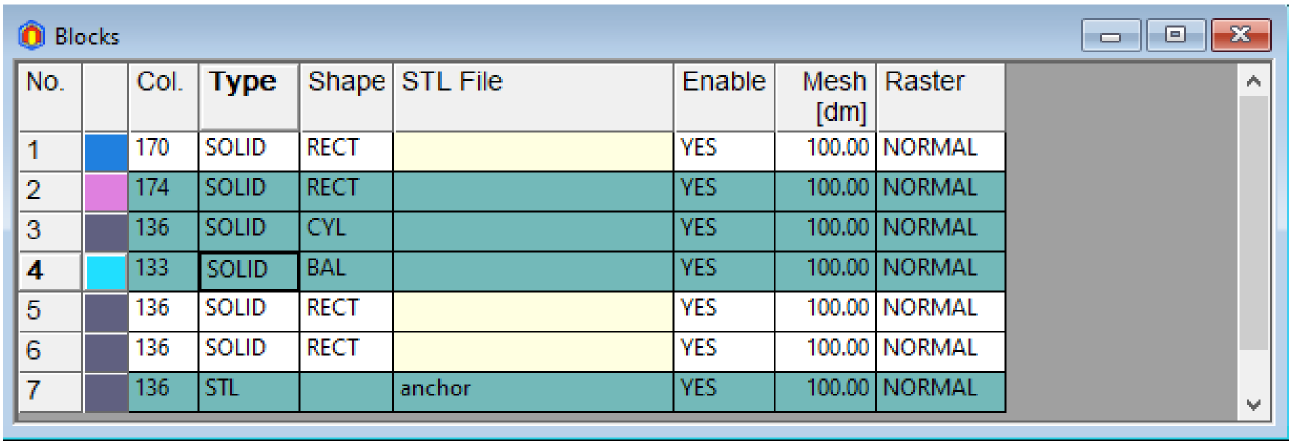
B.7. Block functions



B.7.1. Insert block

- *Edit → Insert Before*
  Insert a new block before the active block.
- *Edit → Insert After*
  Insert a new block after the active block.

B.7.2. Block selections



Individual blocks can be selected to perform global actions (delete, copy, move, transform).
The selected blocks are marked in turquoise in the **Blocks window**.
Blocks can be selected by clicking (left mouse button) the row number. Holding the <Shift> and <Ctrl> keys, a selection range can be defined.
Alternatively, menu functions can be used to select blocks.

<u>Block selection functions:</u>

- *Edit → Switch Select*
  Switch on/off the selection for the active block.

- *Edit → Switch Select Downward*
  Switch on/off the selection of the active block and all next blocks with equal selection mode.
- *Edit → Switch Select Upward*
  Switch on/off the selection of the active block and all previous blocks with equal selection mode.
- *Edit → Select All*
  Select all blocks.
- *Edit → Deselect All*
  Deselect all blocks.

B.7.3. Delete, copy, move

- *Edit → Delete*
  Delete selected block(s), otherwise (in absence of a block selection) delete the active block.

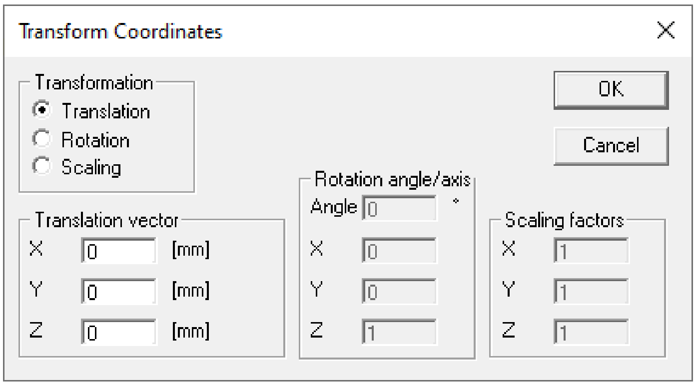
- *Blocks → Copy After*
  Copy selected block(s) to a new position in the blocks list.
  If there is no block selection, the active block is copied after itself.
  If there is a block selection, the selected blocks are copied immediately after the active block.
  After the copy operation the copied blocks become the new block selection.

- *Blocks → Move Selection Before*
  Move selected block(s) to a new position in the blocks list.
  If there is no block selection, this function is disabled
  If there is a block selection, the selected blocks are moved immediately before the active block.
  After the move operation the selected blocks remain selected.

- *Blocks → Move Selection After*
  Move selected block(s) to a new position in the blocks list.
  If there is no block selection, this function is disabled
  If there is a block selection, the selected blocks are moved immediately after the active block.
  After the move operation the selected blocks remain selected.

B.7.4. Transform coordinates

Menu command: *Blocks → Transform Coordinates…*



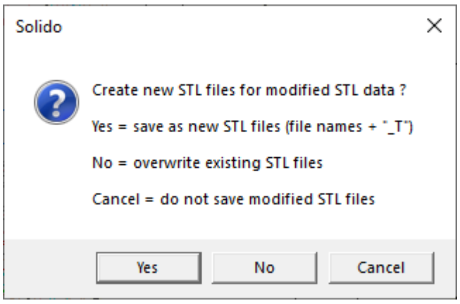
The block points of the selected block(s) are transformed in accordance to the given parameters.

Types of transformation:

- *Translation*: over a given translation vector.
- *Rotation*: over a given angle around a rotation axis through the origin of the coordinate system and a given point, defined by the X, Y and Z coordinates.
  The rotation angle is positive in counterclockwise direction, viewed along the rotation axis from the given point to the origin.
  A rotation always modifies the transformed blocks to hexahedra (B.7.7).
- *Scaling*: of the X, Y and Z coordinates by given scaling factors.
  (The origin of the coordinate system is the centre of the scaling.)
  A mirror transformation can be executed by using a negative scaling factor.
  E.g. mirror X corresponds to the scaling factors -1, 1, 1.

Transformation of a STL block

For each block of type STL (referring to an external STL file), transforming the coordinates of the vertices changes the STL file data.
Whenever the user saves the SOLIDO data file (.sld), the option is therefore given to either save the changes to the STL file(s) or save the changed STL file(s) with a different name. It is good practice to save the SOLIDO data file after every transformation of blocks of type STL. When the existing STL file is overwritten (when using the same file name), the old coordinates can not be restored (e.g. when using *Edit → Undo*).



B.7.5. Insert SOLIDO file

Menu command: *Blocks → Insert File...*

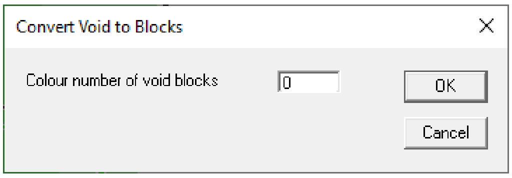
The blocks of a selected SOLIDO file are inserted in the current SOLIDO document at the current position in the block list. Also the corresponding block point coordinates are loaded (into the **Block Points window**), and the colour definitions of not yet defined colours (into the **Colours window**).
This function allows to build an object using pre-defined parts, saved in separate SOLIDO files.

B.7.6. Detect void (adiabatic) spaces within rasterised object

Menu command: *Blocks → Convert Void to Blocks...*

Input: *rasterised object* (i.e. after *Image → Fill Grid (Rasterise)*, B.4.4) *or fitted object* (i.e. after *Image → Fit Grid*, B.4.5)

This function is only enabled if there are void spaces (not covered by object or boundary condition blocks) present in the voxel space. Void spaces are not directly visible in the **Image window**, but this function makes them visible.
Choose a colour number not yet present in the object or boundary conditions.

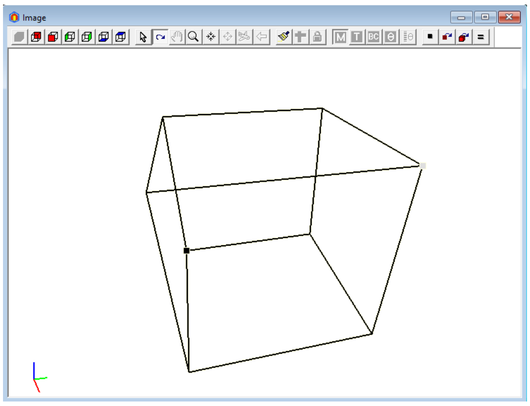


Each void voxel zone is converted to a block of type POINT with the chosen (new) colour number. Because the new blocks are of type POINT, their outlines are not visible when the **Blocks window** is active. To visualize the outline of the void spaces in the **Image window** (if the option *Image → Hide Selection Outline* is switched off), activate the new colour in the **Colours window** (use type = MATERIAL).
Void spaces should normally be avoided in the voxel space, because they imply adiabatic boundary conditions for all adjacent object surfaces.

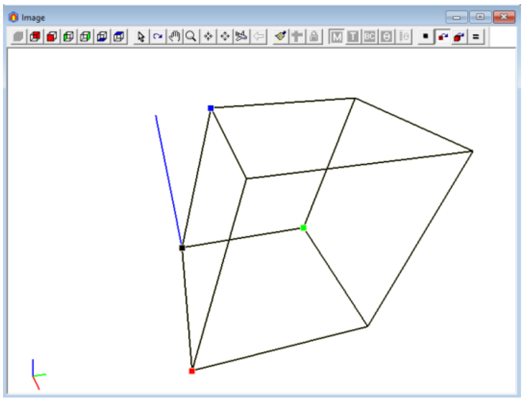
B.7.7. Block forms: box, parallelepiped, hexahedron

Blocks of type SOLID can have different forms dependent on the number of free block points. The form can be altered using commands in the Blocks submenu from the **Block Points window** as active window.

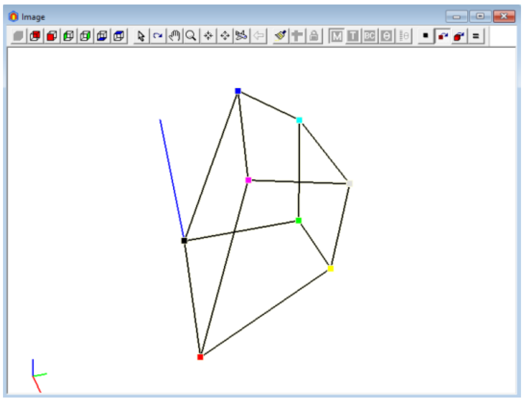
– Box: defined by two points (black and grey), set with the command *Blocks → Make Box*.
  A box is bounded by faces through the black and grey points, parallel to the coordinate planes XY, XZ and YZ.
  A new block that is added to the list of blocks in the **Blocks window** is always a box: a cube with black point in the origin (0,0,0) and grey point in (100,100,100).



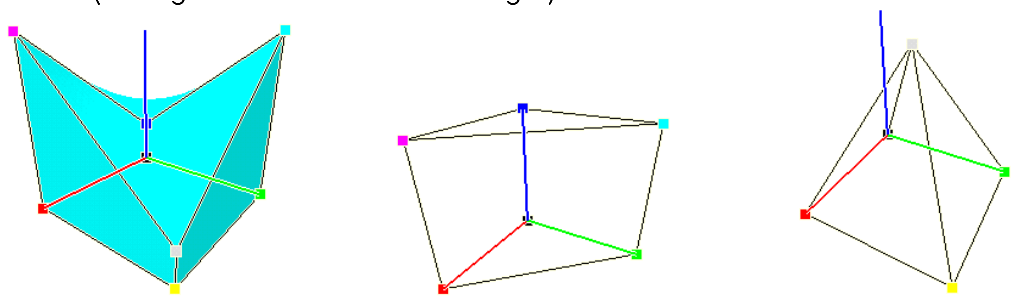
– Parallelepiped: defined by four points (black, red, green and blue), set with the command *Blocks → Make Parallelepiped*.
  A parallelepiped is bounded by three intersecting sets of two parallel faces (as in the figure below).



– Hexahedron: defined by eight points, set with the command *Blocks → Make Hexahedron*. A hexahedron is bounded by six faces, each face defined by four (not necessarily coplanar) points (as in the figure below).



If the 4 vertices of a solid face are not coplanar, a quadratic surface (hyperbolic paraboloid or saddle surface) is created. In this case the triangulation mesh size must be small enough (see Mesh column in the **Blocks window**, B.5) to get a mesh of small triangles, which yields a better approximation of the quadratic surface (see figure below left).
Block points can be made coincident to create e.g. a prism with triangular base or a pyramid (see figure below middle and right).



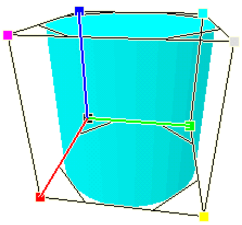
By altering the block points, it is possible to manipulate a SOLID geometry so that is has self-intersecting block faces. Note however that this is not a valid geometrical input for the thermal calculation.

B.7.8. Switch base

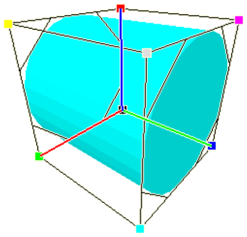
Active window: **Block Points window**
Menu command: *Blocks → Switch Base*

This command automatically upgrades a *box* to a *parallelepiped* and the order of block points is switched in a such way that the base face (containing the black, red and green points) is switched to a different block face.

Before switch base:



After switch base:



B.7.9. World coordinates / grid number coordinates

The block point coordinates in the **Block Points window** are given either with respect to the world coordinate system (a right-handed cartesian coordinate system, with unit defined in *Settings → Coordinates...*, A.9), or with respect to the calculation grid (as defined in the **Grid window**).

For the latter case, the successive calculation grid planes in each direction X, Y and Z have successive grid number coordinates 0,1,2,... These numbers can be used as coordinates of block points in the **Block Points window**, which are called *grid number coordinates.* Also non-integer numbers (i.e. floating point numbers) can be used, which refer to interpolated positions between the adjacent grid plane positions. (Non-integer numbers are not possible in TRISCO).

To switch between both coordinate systems, use the menu command *Blocks → Grid Number Coordinates*. The selected coordinate system is used for point coordinates in the **Block Points window**, the **Areas window** and the **U Values window**.

The coordinate system selection is saved in the SOLIDO data file.

A new SOLIDO document starts in the world coordinate system.

Imported TRISCO or Trisco2D data files operate in grid number coordinates. This is because in the program TRISCO and Trisco2D all point and block coordinates are directly expressed in (integer) grid number coordinates (see B.2.4 and B.2.5).

B.7.10. Set origin

Active window: **Block Points window**
Menu command: *Blocks → Set Origin*

Moves the origin of the world coordinate system to the active block point in the **Block Points window**. All point coordinates are adapted accordingly.

Changing the origin transforms the coordinates of the vertices of the STL file(s). When the user saves the SOLIDO data file (.sld), the option is therefore given to either save the changes to the STL file(s) or save the changed STL file(s) with a different name (see B.7.4).

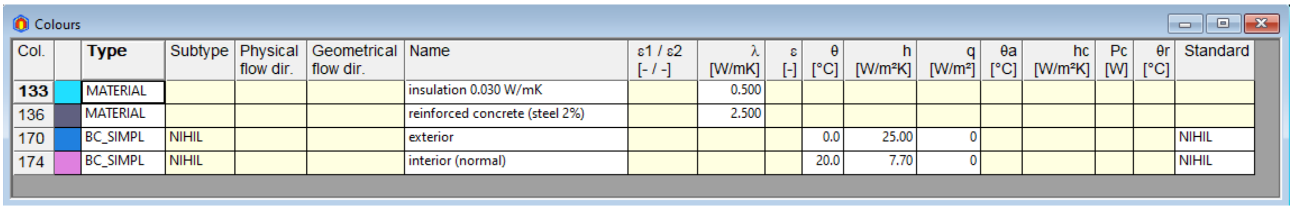
B.7.11. Make property global

Menu command: *Blocks → Make Property Global*

This command copies the active cell value to all other cells in the same column of the **Blocks window**.

B.8. Colours window

Menu command: *Edit → Colours*



Colours

Col.		Type	Subtype	Physical flow dir.	Geometrical flow dir.	Name	ε1 / ε2 [- / -]	λ [W/mK]	ε [-]	θ [°C]	h [W/m²K]	q [W/m²]	θa [°C]	hc [W/m²K]	Pc [W]	θr [°C]	Standard
133		MATERIAL				insulation 0.030 W/mK		0.500									
136		MATERIAL				reinforced concrete (steel 2%)		2.500									
170		BC_SIMPL	NIHIL			exterior				0.0	25.00	0					NIHIL
174		BC_SIMPL	NIHIL			interior (normal)				20.0	7.70	0					NIHIL

The **Colours window** is the input window for the definition of <u>(thermal) properties of materials and surface boundary conditions</u>. The colour properties are linked via the colour index numbers (between 0 and 255) to the geometric blocks defined in the **Blocks window** (B.5).

<u>Fields per colour</u>

The fields are editable (white background) or not (light yellow background) dependent on the chosen settings in other fields.

Column	Value	Description
Col.	0-255	Colour index number (referenced from e.g. **Blocks window**)
-	Colour	Visual colour from colour palette (default values defined in the file MatColor.rgb). To edit: press <Enter> in this field, or double click with the left mouse button, which opens a Colour dialog box.
Type	MATERIAL	Material (with user defined thermal conductivity) The material name and λ-value can be loaded from a material database (see B.9.5).
	EQUIMAT	Equivalent material defined by a single equivalent thermal conductivity (e.g. air cavity). The equivalent conductivity is typically calculated from standard rules (C.1).
	BC_SIMPL	Simplified surface boundary condition, defined by a global surface heat transfer coefficient (for combined convection and radiation), environmental temperature and possible additional surface heat flux
	BC_FRE_S	Enclosure with unknown (free-floating) internal temperature, possible additional internal heat power, simplified surface heat transfer defined by a global surface heat transfer coefficient and possible additional surface heat flux.
	BC_FREE	*(Only available with <u>RADCON module</u>)* Surface boundary condition modelled as enclosure with unknown (free-floating) air temperature, radiation between the surfaces (using the infrared emissivities of the adjacent materials) and convective heat transfer coefficient.
	BC_SKY	*(Only available with <u>RADCON module</u>)* Surface boundary condition modelled as an environment with defined air and radiation temperature (the sky) and convective heat transfer coefficient.

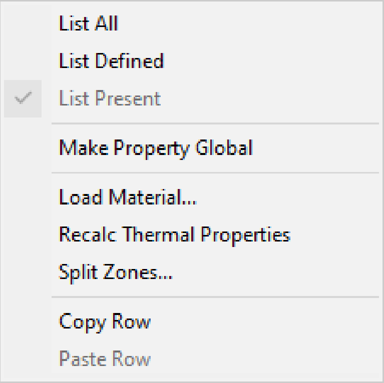
	BC_NOSKY	*(Only available with <u>RADCON module</u>)* Surface boundary condition modelled as an environment with defined air temperature, no radiation to the environment (no sky) and convective heat transfer coefficient.
	TRANSMAT	*(Only available with <u>RADCON module</u>)* Transparent material defined by thermal conductivity, and radiation between the surfaces (using the infrared emissivities of the adjacent materials).
Standard (last column)	NIHIL	no automatic calculation of thermal properties.
	EN10077	Standard EN ISO 10077 used for the automatic calculation of λ_{eq} or h_c (cf. section C.1).
	EN6946	Standard EN ISO 6946 used for the automatic calculation of λ_{eq} or h_c (cf. section C.1).
Subtype	NIHIL	No automatic calculation of thermal property
	Other	Automatic calculation of thermal properties (see C.1) using *Colours → Recalc Thermal Properties* (see 0)
Physical Flow Dir.	HOR	Horizontal heat flow.
	UP	Upward heat flow.
	DOWN	Downward heat flow.
Geom. Flow Dir.	X	Heat flow parallel to the grid X axis (see B.10)
	Y	Heat flow parallel to the grid Y axis (see B.10)
	Z	Heat flow parallel to the grid Z axis (see B.10)
	Xy	Heat flow parallel to the grid X axis and cavity width parallel to the grid Y axis (see B.10)
	Xz	Heat flow parallel to the grid X axis and cavity width parallel to the grid Z axis (see B.10)
	Yx	Heat flow parallel to the grid Y axis and cavity width parallel to the grid X axis (see B.10)
	Yz	Heat flow parallel to the grid Y axis and cavity width parallel to the grid Z axis (see B.10)
	Zx	Heat flow parallel to the grid Z axis and cavity width parallel to the grid X axis (see B.10)
	Zy	Heat flow parallel to the grid Z axis and cavity width parallel to the grid Y axis (see B.10)
Name	Text	Name for material or boundary condition
ε1 / ε2 [- / -]	Number	Infrared emissivities (values between 0.001 and 1) at warm and cold sides of cavity (type EQUIMAT).
λ [W/(m.K)]	Number(1)	thermal conductivity of material (MATERIAL), equivalent material (EQUIMAT) or transparent material (TRANSMAT).
ε [-]	Number	Infrared emissivity (value between 0 and 1) of material surfaces (MATERIAL, EQUIMAT) adjacent to boundary condition or cavity with radiative heat transfer (with RADCON module)
θ [°C]	Number	Environmental temperature (BC_SIMPL).
h [W/(m^2.K)]	Number	Global surface heat transfer coefficient (BC_SIMPL, BC_FRE_S).
q [W/m^2]	Number	Surface heat flux (in W/m^2) entering adjacent materials (BC_SIMPL, BC_FRE_S, BC_FREE, BC_SKY, BC_NOSKY)
θ_a [°C]	Number	Air temperature (in °C) (BC_SKY, BC_NOSKY)

h_c [W/(m^2.K)]	Number[1]	Convective heat transfer coefficient (in W/(m^2.K)) (BC_FREE, BC_SKY, BC_NOSKY)
P_c [W]	Number	Convective power (in W) dissipated in the environment (BC_FRE_S, BC_FREE)
θ_r [°C]	Number	Radiation temperature (in °C) (BC_SKY)

[1] As long as the *fitted object* (i.e. after *Image → Fit Grid*, see B.4.5) is not yet available, the required geometric measures for automatically calculated thermal properties (see C.1) are not known and no value (-) is shown in the **Colours window**.

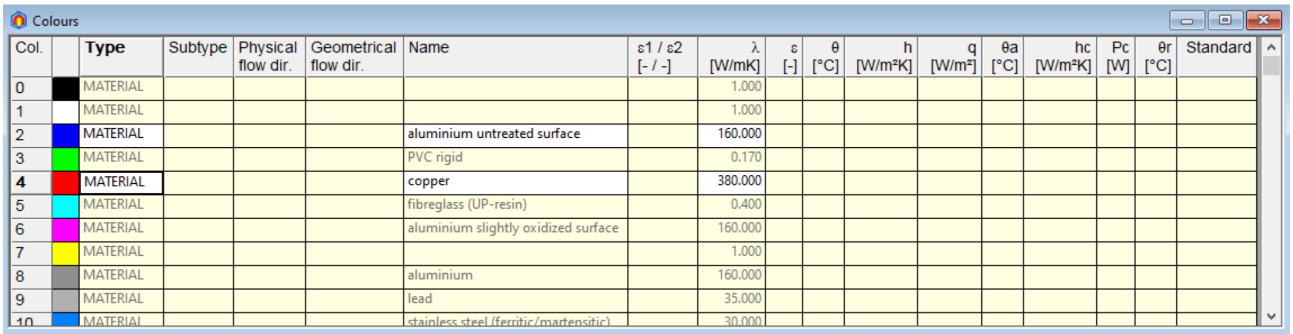
B.9. Colour functions

Colours submenu:



B.9.1. List all colours

Menu command: *Colours → List All*


Colours

Col.		Type	Subtype	Physical flow dir.	Geometrical flow dir.	Name	ε1 / ε2 [- / -]	λ [W/mK]	ε [-]	θ [°C]	h [W/m²K]	q [W/m²]	θa [°C]	hc [W/m²K]	Pc [W]	θr [°C]	Standard
0		MATERIAL						1.000									
1		MATERIAL						1.000									
2		MATERIAL				aluminium untreated surface		160.000									
3		MATERIAL				PVC rigid		0.170									
4		MATERIAL				copper		380.000									
5		MATERIAL				fibreglass (UP-resin)		0.400									
6		MATERIAL				aluminium slightly oxidized surface		160.000									
7		MATERIAL						1.000									
8		MATERIAL				aluminium		160.000									
9		MATERIAL				lead		35.000									
10		MATERIAL				stainless steel (ferritic/martensitic)		30.000									

This command lists all colour definitions for colour number 0 up to 255.
Colours not referenced from the **Blocks window** are not editable (except for the colour values in the second column).
This list may be useful in choosing new colour numbers as colour references in the **Blocks window**.

B.9.2. List defined colours

Menu command: *Colours → List Defined*

This command lists all defined colours, i.e. all present colours (referenced from the **Blocks window**), the colour definitions from the colour database (F.2) and from files previously opened in the current SOLIDO session, which are still present in memory.
Colours not referenced from the **Blocks window** are not editable (except for the colour values in the second column).

B.9.3. List present colours

Menu command: *Colours → List Present*

This command lists the colours referenced from the blocks in the **Blocks window**.
This is the default setting when a new project file is opened.

B.9.4. Make property global

Menu command: *Colours → Make Property Global*

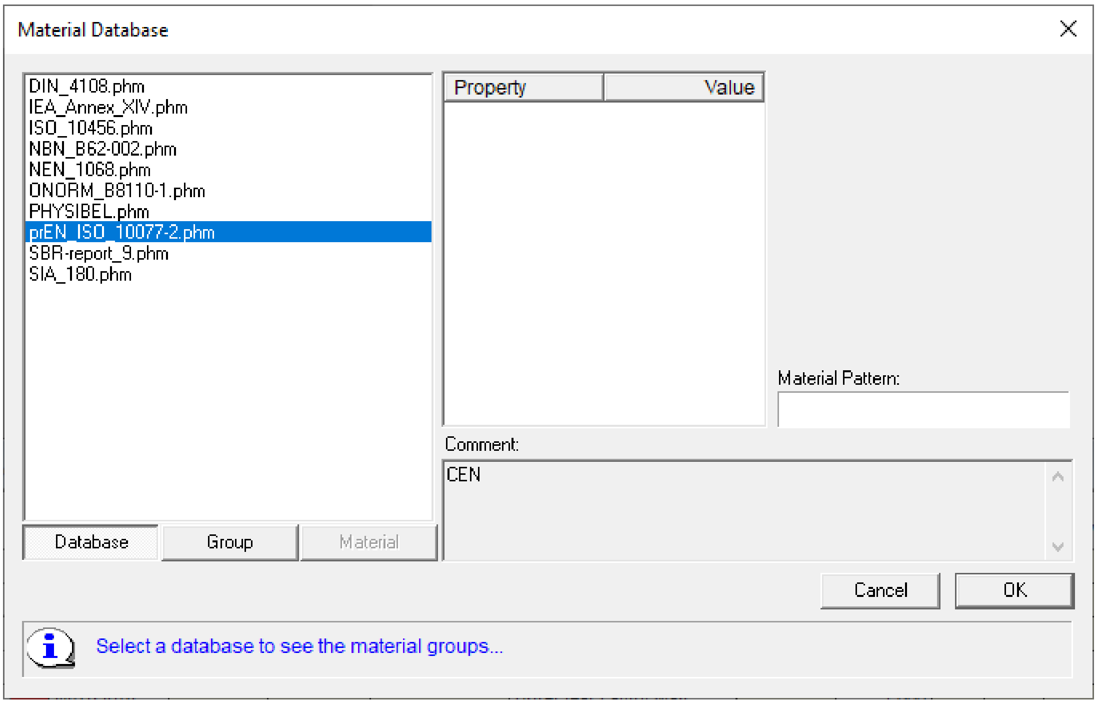
This command copies the active cell value to all other cells in the same column of the **Colours window**.
If the value should only be copied to a limited number of other colours, then use the functions *Edit → Copy Value* and *Edit → Paste Value* (see A.10).

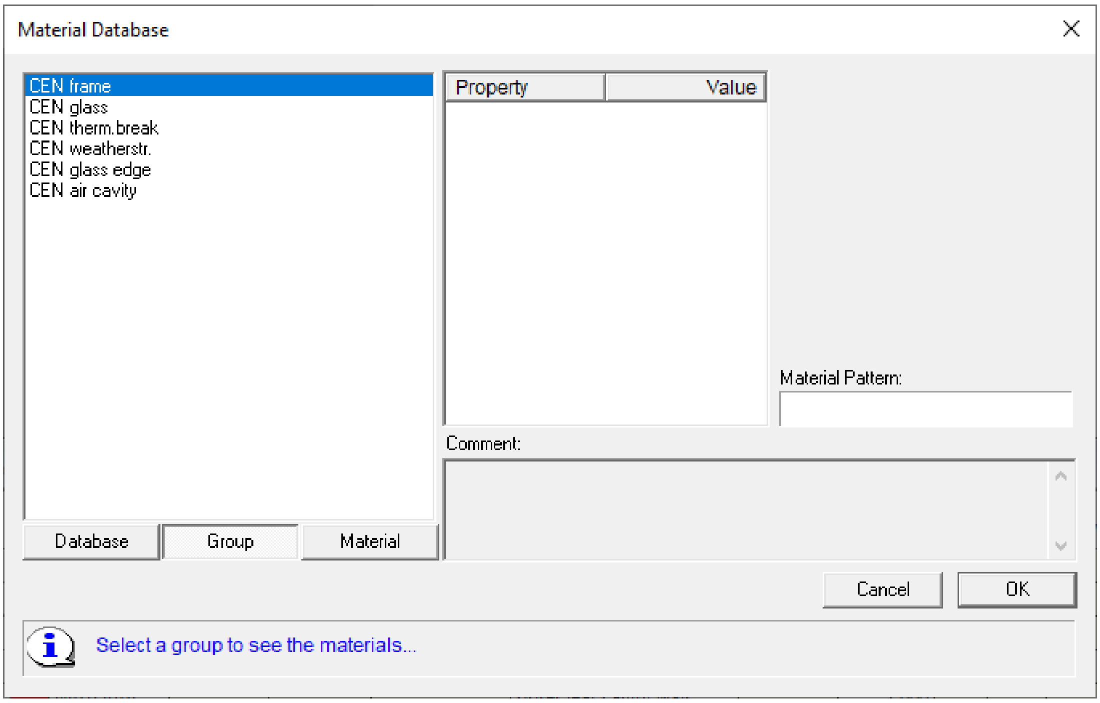
B.9.5. Load material properties

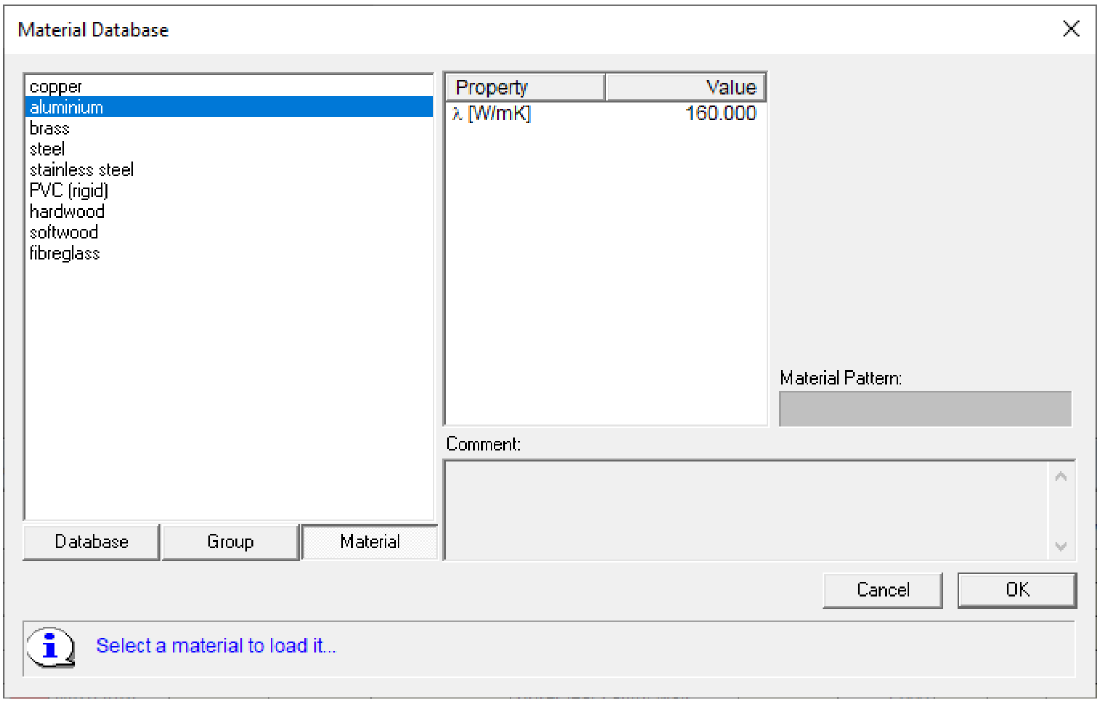
Menu command: *Colours → Load Material...*

The Material Database dialog box allows to select a material from a group of materials in a selected database.
The available databases contain the thermal conductivities of different materials according to several European standards, and are stored in the files with extension .phm in the data folder ...\Physibel\Materials (see A.7). The files can be edited using the program MatEdit.exe in the same folder.

After selecting a material and pressing the OK button, the corresponding properties are loaded for the active colour number.







B.9.6. Recalculation of thermal properties

Menu command: *Colours → Recalc Thermal Properties*
Input: *fitted object* and calculated solution

This function allows to update the automatically calculated thermal properties depending on the solution temperatures if a solution is available (after a system calculation or after reading an existing solution from disk).

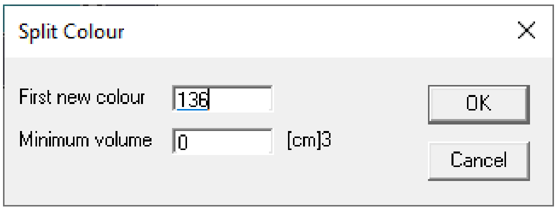
See section C.1 for more information.

B.9.7. Split separated voxel zones

Menu command: *Colours → Split Zones...*
Input: *rasterised object*

This function detects all separated voxel regions for a given colour starting from the *rasterised object.* To each separated voxel region a new colour is attributed. The thermal properties for the new colours are copied from the original colour. New blocks are added at the end of the blocks list in the **Blocks window**. The new blocks get the type POINT. The corresponding block point coordinates are automatically calculated from the rasterised geometry.

A dialog box is opened in which a colour number must be entered as start colour ("first new colour") for the split colours.



In the voxel space of the *rasterised object* each connected region of voxels having the same colour as the active colour (where the active input cell is located in the **Colours window**) is detected. When the volume (expressed in world coordinates) of this region is greater than or equal to the *minimum volume* (parameter as defined in the Split Colour dialog box), then all voxels of this region get a new colour, which is not yet present in the current voxel space. The colour number (from 0 up to 255) of this new colour is the lowest not yet occupied number starting from the *first new colour* number in the Split Colour dialog box. (When the number 255 is reached, the search continues with the number 0). An error message appears when there are not enough free colours available.
New blocks of type POINT are added to the block list in the **Blocks window**. The coordinates of the corresponding (black) block points are automatically calculated within the split voxel regions.

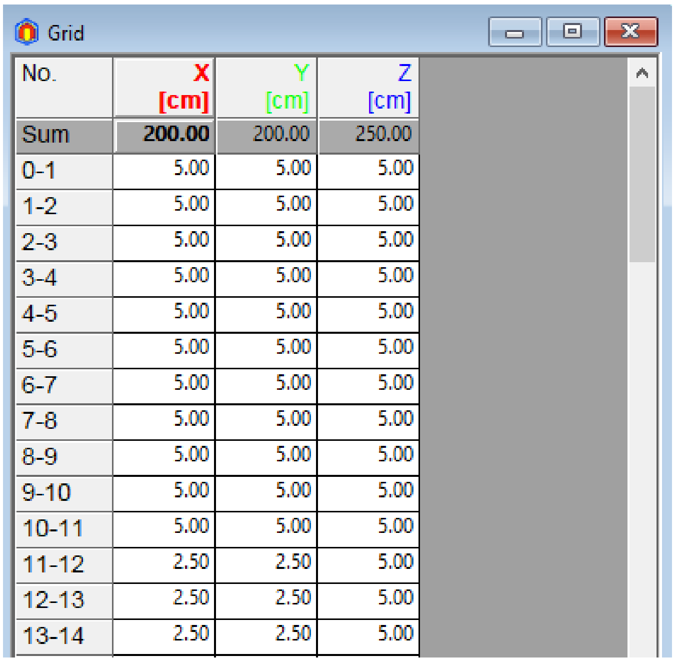
The function is useful for colours of type BC_FRE_S, BC_FREE, BC_SKY, BC_NOSKY or TRANSMAT, which require single-zone colours to be valid.

B.9.8. Copy - paste

The functions *Colours → Copy Row* and *Colours → Paste Row* allow to copy all properties from one colour number to another colour number.

B.10. Grid window

Menu command: *Edit → Grid*



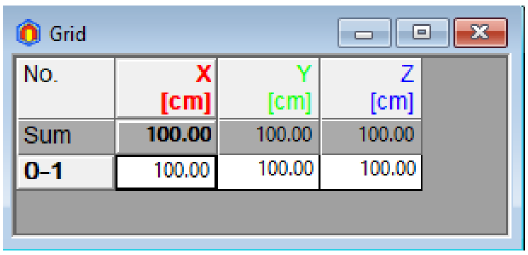
No.	X [cm]	Y [cm]	Z [cm]
Sum	200.00	200.00	250.00
0-1	5.00	5.00	5.00
1-2	5.00	5.00	5.00
2-3	5.00	5.00	5.00
3-4	5.00	5.00	5.00
4-5	5.00	5.00	5.00
5-6	5.00	5.00	5.00
6-7	5.00	5.00	5.00
7-8	5.00	5.00	5.00
8-9	5.00	5.00	5.00
9-10	5.00	5.00	5.00
10-11	5.00	5.00	5.00
11-12	2.50	2.50	5.00
12-13	2.50	2.50	5.00
13-14	2.50	2.50	5.00

The **Grid window** is the input window for grid mesh sizes of a rectangular 3D grid, used in solving the thermal system.

A mesh is defined as the distance between two successive grid planes in any one dimension of the 3D orthogonal grid.

The grid is not necessarily directly linked to the object, as defined in the **Blocks window**. The grid can be defined independently from the object coordinates. However to achieve good calculation results the grid must approximate closely the object geometry. To this end a set of grid functions is available, and an automatic fitting of grid nodes to the object surfaces is used during the system calculation.

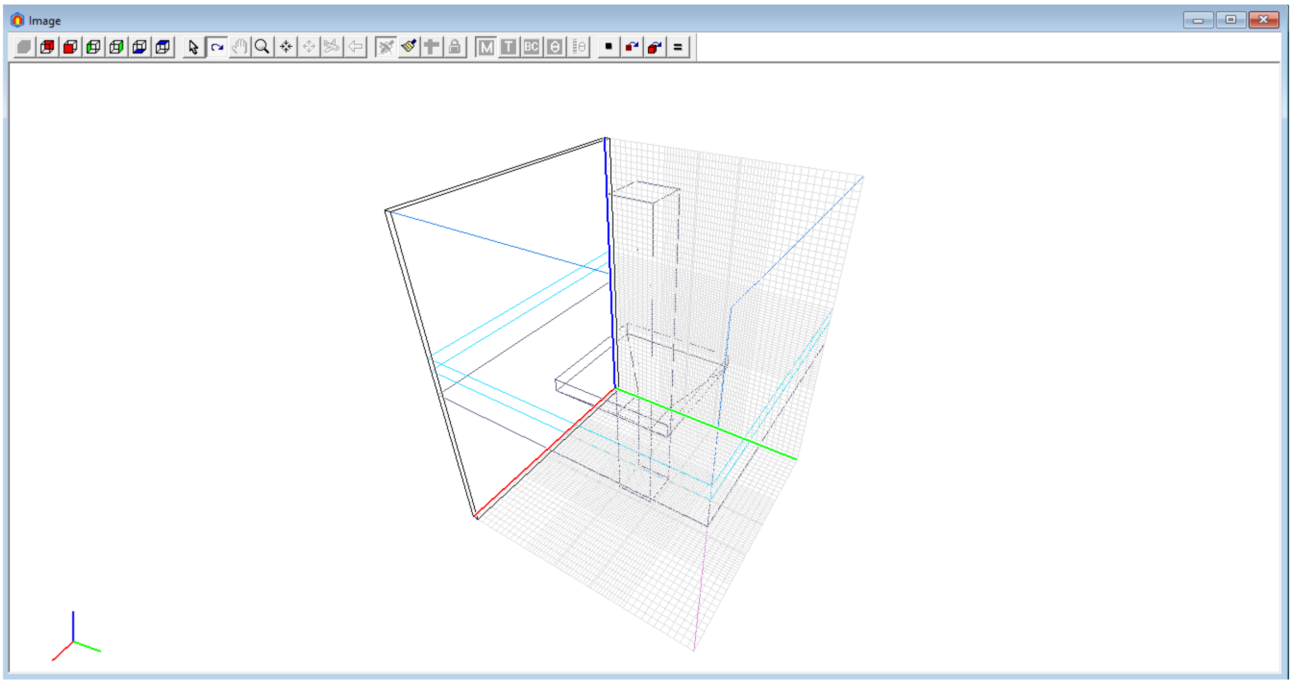
Initially for a new SOLIDO document, the grid origin equals the object origin, and only 1 grid mesh of fixed size is present in each direction X, Y and Z.



No.	X [cm]	Y [cm]	Z [cm]
Sum	100.00	100.00	100.00
0-1	100.00	100.00	100.00

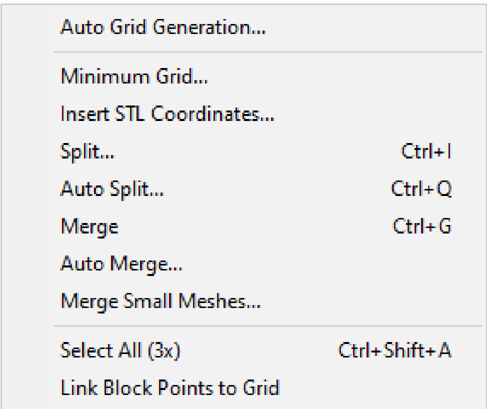
Using grid functions this grid can be refined and modified. Each individual grid mesh size can be edited in the **Grid window**.

The active grid mesh in relation to the object blocks is highlighted in the **Image window**.



B.11. Grid functions

Grid submenu:



B.11.1. Grid mesh selections

One or more grid meshes can be selected to perform global actions (split, merge).
The selected meshes are marked with a turquoise background colour in the **Grid window**.

Meshes can be selected by clicking (left mouse button) the cell while holding the <Ctrl> key. A selection range can be defined by holding the <Shift> key. All meshes within the same column ΔX or ΔY can by selected by clicking (left mouse button) the column header.
Alternatively, menu functions can be used to select meshes.

<u>From the Edit submenu:</u>

- *Edit → Switch Select*
  Switch on/off the selection for the active mesh.
- *Edit → Switch Select Downward*
  Switch on/off the selection of the active mesh and all next meshes with equal selection mode.

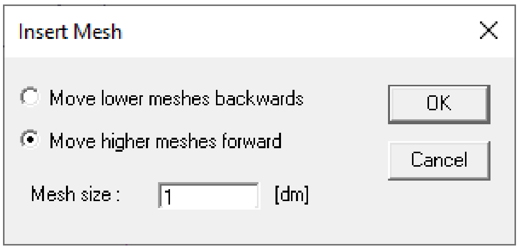
- *Edit → Switch Select Upward*
  Switch on/off the selection of the active mesh and all previous meshes with equal selection mode.
- *Edit → Select All*
  Select all meshes on the same axis as the active mesh.
- *Edit → Deselect All*
  Deselect all meshes.

<u>From the Grid submenu:</u>

- *Grid → Select All (3x)*
  Select all meshes (on all three axes).

B.11.2. Insert mesh

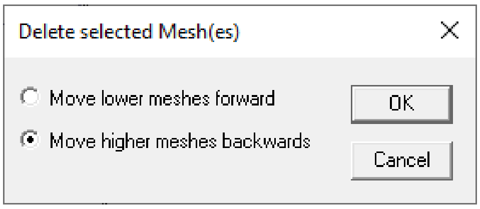
Menu commands:
Edit → Insert Before (to insert a new mesh before the active mesh)
Edit → Insert After (to insert a new mesh after the active mesh)



Insert a new mesh with given mesh size.
The lower meshes are either moved backwards, or the higher meshes are moved forward.
This is not directly visible in the **Grid window** (because no absolute coordinates are shown), but the effect can be seen in the **Image window** (showing the grid lines) and the **Grid Coordinates window** (showing absolute coordinates).

B.11.3. Delete mesh(es)

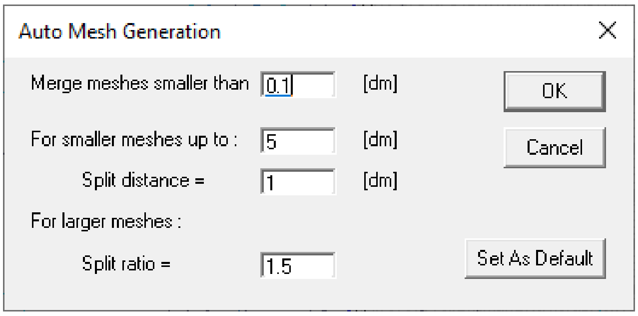
Menu commands: *Edit → Delete*



Delete the active mesh.
The lower meshes are either moved forward, or the higher meshes are moved backwards.
This is not directly visible in the **Grid window** (because no absolute coordinates are shown), but the effect can be seen in the **Image window** (showing the grid lines) and the **Grid Coordinates window** (showing absolute coordinates).

B.11.4. Automatic grid generation

Menu command: *Grid → Auto Grid Generation...*



This function automatically calculates a grid which is directly ready for a system calculation of node temperatures. The parameters correspond to parameters of the underlying grid functions:

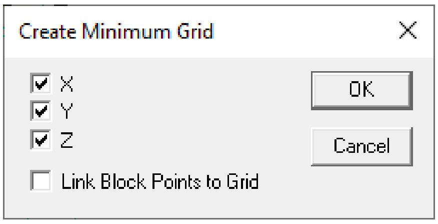
- Merge meshes smaller than a given value: see *Grid → Merge Small Meshes…* (cf. section B.11.11).
- Split distance (for smaller meshes): see split distance in *Grid → Auto Split…* (cf. section B.11.8).
- Split ratio (for larger meshes): see split ratio in *Grid → Auto Split…* (cf. section B.11.8).

The function executes the following grid functions sequentially:

1. *Grid → Minimum Grid…* to get the surrounding box of the global object (cf. section B.11.5).
2. *Grid → Insert STL Coordinates…* to get the dominant coordinates of the STL files (cf. section B.11.6).
3. *Grid → Merge Small Meshes…* to eliminate very small meshes (cf. section B.11.11).
4. Rasterising in multiples of the Auto Split (cf. section B.11.8) split distance, depending on the gradient of the surfaces of triangulated object. This automated step is not available as a separate grid function in the program menu. Alternatively – though less efficient –, the functions *Grid → Select all (3x)* (cf. section B.11.1) and *Grid → Split…* (cf. section B.11.7) with a maximum ‘Mesh width’ equal to the split distance for smaller meshes defined, can be applied manually for this step.
5. *Grid → Merge Small Meshes…* to eliminate very small meshes (cf. section B.11.11).
6. *Grid → Auto Split…* using the split ratio for larger meshes, to obtain gradual increasing or decreasing meshes to obtain gradual temperature fields (cf. section B.11.8).

B.11.5. Minimum grid

Menu command: *Grid → Minimum Grid…*

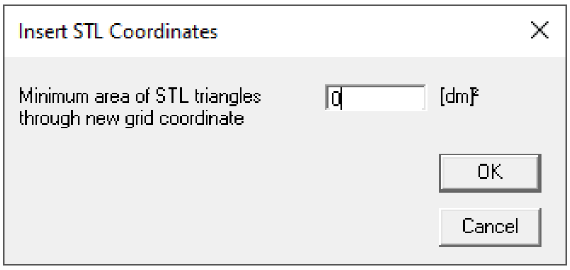


This function replaces the current grid by a *minimum grid* in the specified directions (X, Y and/or Z) based on the object geometry as defined in the **Blocks window** and **Block Points window**. The minimum grid meshes are determined by the coordinates of the block points of all blocks of type SOLID, and the limits of the global bounding box (useful for STL blocks).

The checkbox *Link Block Points to Grid* in the dialog box has the same purpose as the menu command *Grid → Link Block Points to Grid* (cf. section B.11.12).

B.11.6. Insert STL coordinates

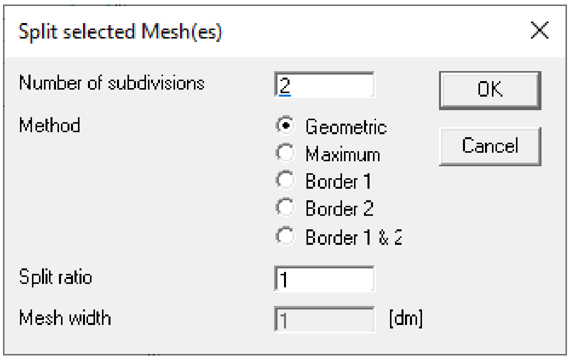
Menu command: *Grid → Insert STL Coordinates...*



Each STL file (associated with a STL block) consists of triangles, which together form the surface of the represented 3D object. The function *Grid → Insert STL Coordinates...* detects all STL triangles (of all STL blocks) that are orthogonal to a coordinate axis. Such a STL triangle has an equal X, Y or Z coordinate for all three vertices. That coordinate is inserted as new grid coordinate, if the total area of all STL triangles through that coordinate is large enough (i.e. greater than the value entered in the dialog box).

B.11.7. Split mesh(es)

Menu command: *Grid → Split...*



Split either the selected meshes or the active mesh (when no selection is defined) into smaller parts according to the given method.

<u>Method: Geometric</u>
Each mesh is split into the specified number of subdivisions using a geometric series (i.e. a series with a constant ratio between successive terms) with specified split ratio.

- When the split ratio = 1, then all created submeshes are equal.
- When the split ratio < 1, then the submeshes have decreasing mesh widths.
- When the split ratio > 1, then the submeshes have increasing mesh widths.

A negative or zero split ratio is not allowed.

<u>Method: Maximum</u>
All submeshes have a constant mesh width (as defined by the parameter *Mesh width* in the dialog box), except for the last submesh, for which the width equals the remainder after division. This last submesh can however never be smaller than 0.1 % of the maximum mesh width.

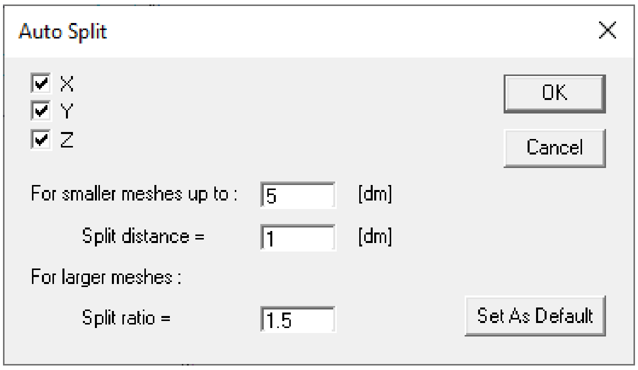
<u>Method: Border 1</u>
Each mesh is split into two parts, when the original mesh size is greater than the given mesh width (parameter Mesh width in dialog box). The first submesh (border 1) gets the size of the given mesh width, the second submesh gets the size of the remainder.

Method: Border 2
Each mesh is split into two parts, when the original mesh size is greater than the given mesh width (parameter Mesh width in dialog box). The second submesh (border 2) gets the size of the given mesh width, the first submesh gets the size of the remainder.

Method: Border 1 & 2
Calls successively: split using border 2, then split using border 1.
So each mesh is split into three parts (when the original mesh size is large enough): both outer submeshes get the size of the given mesh width, the middle submesh gets the size of the remainder.

B.11.8. Auto split

Menu command: *Grid → Auto Split...*



Split all meshes in the specified directions (X, Y, Z) into smaller parts.
For originally smaller meshes (up to a given mesh size), the mesh is split into submeshes having a given split distance, and possibly a last submesh with width equal to the remainder after division (cf. Method: Maximum in section B.11.7).
The other (larger) meshes are split using a geometric subdivision using a given split ratio (cf. Method: Geometric in section B.11.7) starting with the split distance used for smaller meshes at both ends of the original mesh.

B.11.9. Merge meshes

Menu command: *Grid → Merge*

Merge each group of consecutive selected meshes into a single mesh (the width of which equals the sum of the widths of the original meshes).
If no meshes are selected, the active mesh is merged with the next one.

B.11.10. Automatic merge

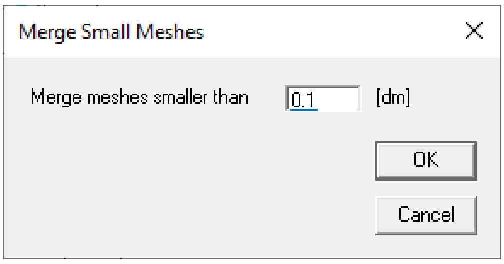
Menu command: *Grid → Auto Merge...*

This function merges all consecutive meshes that contain identical voxel layers in the rasterised object.

Remark: This function replaces the "most compact grid" function (as part of the "optimise grid" function) of SOLIDO v4.0w.

B.11.11. Merge small meshes

Menu command: *Grid → Merge Small Meshes...*



Each mesh smaller than the given value is merged with the previous or next mesh that is larger than that value.
If no adjacent larger mesh is found, the smaller meshes are maintained.

B.11.12. Link block points to grid

Menu command: *Grid → Link Block Points to Grid* (operates as an on/off switch).

When the switch is on, the block points are tied to the grid meshes. When a grid mesh size is modified (by typing a new value, or inserting or deleting a mesh), the object block points (in the **Block Points window**) are modified accordingly, thus changing the object geometry.

In addition to the object block points, the points defining the areas (in the **Areas window**, see B.17) and the U-values (in the **U Values window**, see B.18) are tied to the grid meshes when the switch is on. When a grid mesh size is modified (by typing a new value, or inserting or deleting a mesh), these points are thus modified accordingly. This behaviour is however limited to *defined areas* (i.e. with a surface area greater than 0 m^2) and *defined U-values* (i.e. at least 1 of the points not in the object origin).

Points located on the grid mesh planes remain positioned upon those planes after the modification operation. Points in between two grid mesh planes remain positioned at the same relative distance in between the two planes.

Because the object geometry changes when the grid is changed and the block points are tied to the grid, the point coordinates of the triangle vertices in a STL block are also modified. When the user saves the SOLIDO data file (.sld), the option is therefore given to either save the changes to the STL file(s) or save the changed STL file(s) with a different name (see B.7.4).

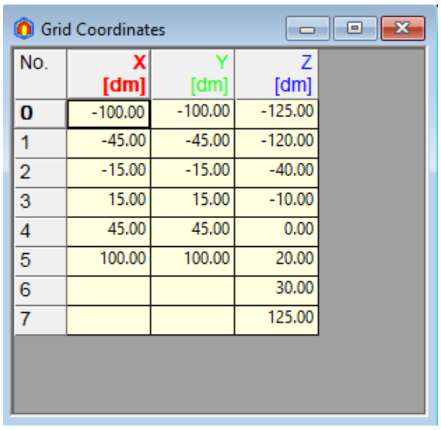
Using this function it is possible to change the thickness (in the direction of X, Y or Z) of a material layer by editing a grid mesh size without the need to change the individual block point coordinates.

Remark for TRISCO users

In TRISCO and Trisco2D the block points are always linked to the grid, because the block coordinates are expressed in (integer) grid number coordinates. When importing a TRISCO or Trisco2D file in SOLIDO with up to 100 blocks, the toggle *Grid → Link Block Points to Grid* is automatically set, to resemble the functionality of TRISCO (see B.2.4 and B.2.5).

B.12. Grid Coordinates window

Menu command: *Edit → Grid Coordinates*



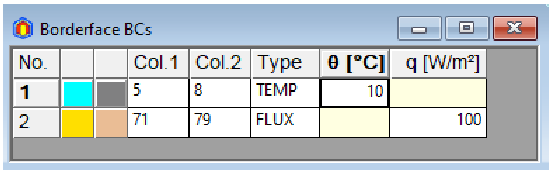
Grid Coordinates

No.	X [dm]	Y [dm]	Z [dm]
0	-100.00	-100.00	-125.00
1	-45.00	-45.00	-120.00
2	-15.00	-15.00	-40.00
3	15.00	15.00	-10.00
4	45.00	45.00	0.00
5	100.00	100.00	20.00
6			30.00
7			125.00

The **Grid Coordinates window** shows the world coordinates of the grid planes, as defined in the **Grid window** (where distances between the successive grid planes are given).
No editing is possible in the **Grid Coordinates window**.
The origin can be set with *Blocks → Set Origin* from the **Block Points window** (see B.7.10).

B.13. Borderface Boundary Conditions window

Menu command: *Edit → Borderface BCs*



Borderface BCs

No.			Col.1	Col.2	Type	θ [°C]	q [W/m²]
1			5	8	TEMP	10	
2			71	79	FLUX		100

The **Borderface Boundary Condition window** is the input window for the definition of a fixed temperature or fixed imposed heat flux in the interface between two colour regions.

The outline of the active borderface boundary condition (when the **Borderface Boundary Conditions window** is active) is shown only for the *rasterised object* and the *fitted object* (when the switch *Image →Hide Selection Outline* is off, see B.4.7).

Edit fields

Column	Value	Description
Col.1	0–255	Colour at one side of the boundary condition borderface
Col.2	0–255	Colour at the other side of the boundary condition borderface
Type	TEMP	Fixed temperature
	FLUX	Fixed heat flux
Θ [°]	Number	Temperature (in °C)
Q [W/m^2]	Number	Heat flux (in W/m^2)

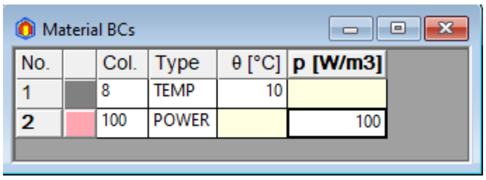
At least one of both colours must refer to a material (type MATERIAL, EQUIMAT or TRANSMAT).

Edit commands

- Insert a new borderface boundary condition: *Edit → Insert After*
- Delete the active borderface boundary condition: *Edit → Delete*

B.14. Material Boundary Conditions window

Menu command: *Edit → Material BCs*



Material BCs

No.		Col.	Type	θ [°C]	p [W/m3]
1		8	TEMP	10	
2		100	POWER		100

The **Material Boundary Conditions window** is the input window for the definition of a fixed temperature or a fixed heat power density within a material.

The outline of the active material boundary condition (when the **Material Boundary Conditions window** is active) is shown only for the *rasterised object* and the *fitted object* (when the switch *Image →Hide Selection Outline* is off, see B.4.7).

For a material boundary condition the total heat flow coming into the material is not output (in the **Text Output window**, see Chapter E).

<u>Edit fields</u>

Column	Value	Description
Col.	0-255	Index to colour of a material (type MATERIAL or EQUIMAT)
Type	TEMP	Fixed temperature
	POWER	Fixed heat power density
θ [°C]	Number	Temperature (in °C)
p [W/m^3]	Number	Heat power density (in W/m^3)

<u>Edit commands</u>

- Insert a new material boundary condition: *Edit → Insert After*
- Delete the active material boundary condition: *Edit → Delete*

B.15. Clip Object window

Menu command: *Edit → Clip Object*

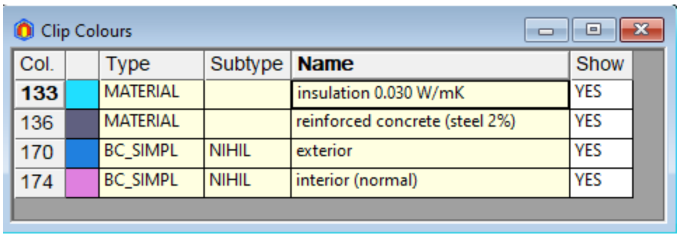
The **Clip Object window** is the input window for the definition of a clip box through the *rasterised object* or *fitted object* using grid plane numbers in the three directions X, Y and Z. In this way the inside of an object can be viewed. All image show options (show materials, show boundary conditions, show temperatures) are possible.
Object clipping is not possible for the *wireframe object* or the *triangulated object*.

The object clipping is also applied in the **Graphic Output window**.

Object clipping can be combined with colour clipping (see B.16).

B.16. Clip Colours window

Menu command: *Edit → Clip Colours*



Clip Colours

Col.		Type	Subtype	Name	Show
133		MATERIAL		insulation 0.030 W/mK	YES
136		MATERIAL		reinforced concrete (steel 2%)	YES
170		BC_SIMPL	NIHIL	exterior	YES
174		BC_SIMPL	NIHIL	interior (normal)	YES

The **Clip Colours window** is the input window for the definition of the visibility of blocks of a given colour in the **Image window**.

The input field *Show* can have the values YES or NO, which defines whether the correspondent colour is visible or hidden in the **Image window**.
In this way the inside of an object can be analyzed. All image show options (show triangulated object, show rasterised object, show fitted object, show materials, show boundary conditions, show temperatures) are possible.

The colour clipping is also applied in the **Graphic Output window**.

Colour clipping can be combined with object clipping (see 0).

B.17. Areas window

Menu command: *Edit → Areas*

Areas

No.	X1 [cm]	Y1 [cm]	Z1 [cm]	X2 [cm]	Y2 [cm]	Z2 [cm]	X3 [cm]	Y3 [cm]	Z3 [cm]	Area [m²]
1	40.00	40.00	0.00	40.00	140.00	0.00	40.00	140.00	100.00	1
2	40.00	40.00	0.00	140.00	40.00	0.00	140.00	40.00	100.00	1
3	40.00	40.00	100.00	140.00	40.00	100.00	140.00	140.00	100.00	1

The **Areas window** is the input window for the definition of areas of parallelograms in 3D space. These areas are used in the definition of derived thermal properties (cf. section 0).
A parallelogram can be defined by three vertices. The area of the parallelogram is twice the area of the triangle through the three vertices.

A fixed number of 3 areas can be defined in the **Areas window**.
The coordinates of 3 points per area are defined either in world coordinates (as default) or in *grid number coordinates* (see B.7.9). The corresponding parallelogram (obtained by mirroring the triangle against the longest side) is highlighted in the **Image window**.

B.18. U Values window

Menu command: *Edit → U values*
Input: *triangulated object*

U Values

No.	X1 [-]	Y1 [-]	Z1 [-]	X2 [-]	Y2 [-]	Z2 [-]	U [W/m²K]
1	0	0	0	0	0	65	0.337
2	23	23	0	23	23	65	0.221
3	15	18	0	15	18	65	0.265

The **U Values window** is the input window for the definition of thermal transmittances of (multi-layered) walls, which are used in the definition of derived thermal properties (cf. section 0).
A thermal transmittance (or U value) is defined by a line between two points, from one boundary condition to another boundary condition, along which the thermal resistances of the wall layers are calculated. The wall layers are bounded by successive intersections of the line and the object surfaces from the *triangulated object* (i.e. after *Image → Fill Blocks (Triangulate)*, B.4.3).

The definition line of a U value must meet the following rules, otherwise U = 0.000 W/(m^2.K):

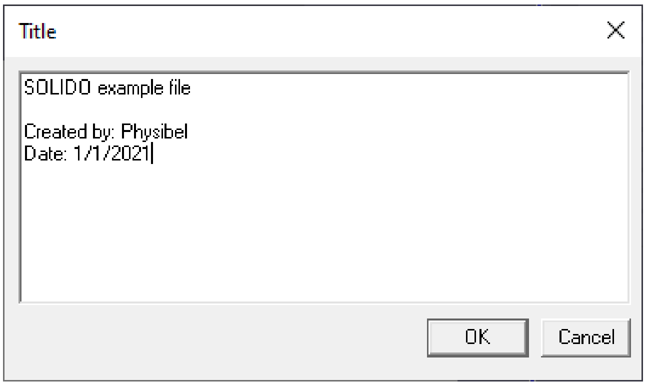
– The *triangulated object* must be available.
– The end points of the definition line must lie in (or touch) different boundary conditions of type BC_SIMPL or BC_FRE_S, with surface heat transfer coefficient h > 0 W/(m^2.K).

– The line must only cross (in an unambiguous way) material layers, and no other boundary conditions.

A fixed number of 3 U values can be defined in the **U Values window**.
The coordinates of both end points of the line defining a U value, are either in world coordinates (as default) or in *grid number coordinates* (see B.7.9).

B.19. Notes

Menu command: *Edit → Notes...*



In the Notes dialog box the user can enter any text as comments on the current project file.
The text is saved in the SOLIDO data file (with extension .sld).

B.20. Hints for TRISCO users

Some useful pointers for users familiar with TRISCO who (are starting to) use SOLIDO.

On the compatibility between TRISCO and SOLIDO:

– TRISCO data can be imported in SOLIDO using *File → Import Data...* (cf. section B.2.3)
– When importing a TRISCO model, the defined Areas and U values are also imported. The U values are available if the *triangulated object* is available (i.e. after *Image → Fill Blocks (Triangulate)*, B.4.3).
In order to be recognised in the more complex geometry of SOLIDO, it is advised to avoid defining the U values in TRISCO along a line on the model borders.
– Rectangular SOLIDO blocks can be exported into a TRISCO data file (using *File →Save As...*). A minimum grid is created for the TRISCO data (cf. section B.2.7).
– Point coordinates of block vertices are normally expressed in world coordinates in SOLIDO, independent of the calculation grid. There is a function to convert all coordinates to *grid number coordinates* (*Blocks → Grid Number Coordinates* from the **Block Points window**, see B.7.9), as used in TRISCO. When importing TRISCO data, this option is automatically chosen (up to 100 blocks).

Important differences in functionality:

– The coordinate unit (or scale) is defined via *Settings → Coordinates...* (see A.9).
The coordinate origin can be moved via *Blocks → Set Origin* (see B.7.10) from the **Block Points window**.
– In SOLIDO the grid number coordinates are not restricted to integer values, but also non-integer values are possible to reference interpolated values.
– The calculation grid in SOLIDO is not necessarily linked to the block vertices, as in TRISCO (through the use of grid number coordinates). A program toggle (*Grid → Link Block Points*

to Grid from the **Grid window**, see B.11.12) allows to have the same link as in TRISCO. This option is automatically set when importing a TRISCO file (up to 100 blocks).

- SOLIDO requires the following successive object models prior to a thermal calculation: *wireframe object, triangulated object, rasterised object, fitted object*.
  After each edit action which affects the object geometry, the **Image window** shows the modified object as a *wireframe object*. The other object models are automatically cleared and should be recalculated.
  A program setting ("automatic fill blocks" via *Settings → Fill Option...*, see A.9) allows to calculate and show immediately the *triangulated object*, which is identical to the view mode in TRISCO. This option is automatically set when importing a TRISCO file.
- There is no **Minimum Grid window** or **Node BCs window** in SOLIDO.

CHAPTER C CALCULATION

C.1. Automatic calculation of thermal properties

In the **Colours window** colour types and subtypes can be selected which contain an automatic calculation of thermal properties in accordance to international standards (equivalent thermal conductivity or convective heat transfer coefficient).

During the input of new data the calculation is based on the default temperature values from the relevant standards, or on the default temperature difference across airspace as defined by the user in the Calculation Parameters dialog box (see C.1.5) if the calculated thermal property is not standardized.
The required geometric measures for the automatically calculated thermal properties are derived from the *fitted object* (i.e. after *Image → Fit Grid*, see B.4.5). If the *fitted object* is not yet available, no value is shown in the **Colours window**.

During a system calculation the thermal properties are recalculated based on the real temperature values, obtained by the global thermal calculation (when requested by calculation parameter "Recalculation of thermal values (before each iteration cycle)", see C.2). The non-linear calculation is split in several *iteration cycles*. Each iteration cycle solves a linear system. The linear system is based on updated thermal properties derived from the temperatures available from the solution of the previous iteration cycle. The recalculated thermal properties are updated in the **Colours window** at the start of each new iteration cycle.

After a system calculation (or after reading an existing solution from disk) the thermal properties can be updated using the last available temperature solution with the function *Colours → Recalc Thermal Properties*. With a good solution the values will not change significantly.

An existing system solution loaded in memory is cancelled after each edit operation (which alters the geometric, grid or thermal input data). The thermal properties are then recalculated again based on the default temperature values.

C.1.1. Colour type EQUIMAT

Colour modelled as *equivalent material* (with *single equivalent thermal conductivity* λ_{eq} for combined convective and radiation heat transfer)

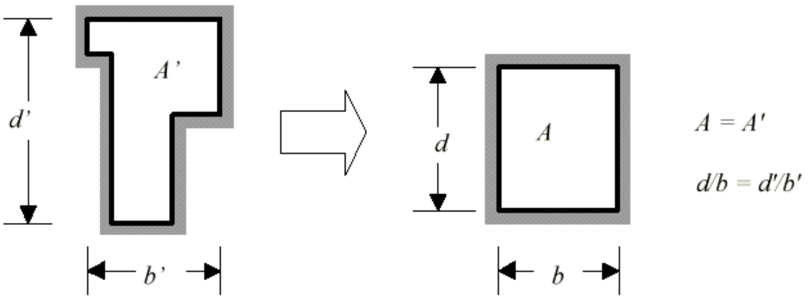
Type	*Subtype*	*Meaning*	*Standard*
EQUIMAT	NIHIL	Thermal conductivity defined by user	
	CAVITY	Unventilated (internal) air cavity with given heat flow direction	EN ISO 10077 EN ISO 6946
	CAVITY_E	Ventilated (external) air cavity with given heat flow direction	EN ISO 10077
	LAYER	Unventilated air layer with given heat flow direction	EN ISO 6946

Characteristic dimensions
Consider a characteristic 2D section of the air cavity, containing the heat flow direction (parallel to a coordinate axis) and the smaller of the other two directions of the cavity. The characteristic

dimensions in the 2D section are called *depth* (parallel to the heat flow direction) and *width* (orthogonal to the heat flow direction).

If the air cavity is not a rectangular parallelepiped (or cuboid) shape, the depth and width are calculated from an equivalent rectangular parallelepiped with the same aspect ratio:
d' = size of surrounding box of air cavity in depth direction
b' = size of surrounding box of air cavity in width direction
V' = volume of surrounding box of air cavity
V = real volume of air cavity
$d = d' \cdot s$
$b = b' \cdot s$
$s = \sqrt{\frac{V}{V'}}$

If the air cavity has a uniform length in the third dimension (perpendicular to depth and width), then the calculation corresponds to the 2D calculation rule in EN ISO 10077-2.



The calculated dimensions *d* and *b* are further used in the formulas below.

Equivalent thermal conductivity of unventilated air cavity (subtype CAVITY)
$\lambda_{eq} = (h_c + h_r) \cdot d$

- h_c = convective heat transfer coefficient (from warm surface to cold surface)
  if b < 0.005 m (for EN ISO 10077), then
  $h_c = \frac{C_1}{d}$
  if b ≥ 0.005 m (for EN ISO 10077) or for any b (EN ISO 6946), then
  $h_c = max\left\{\frac{C_1}{d}, C_2 \cdot (\Delta\theta_{ss})^{C_3}\right\}$
  $\Delta\theta_{ss}$ = maximum surface temperature difference in air cavity ($\Delta\theta_{ss}$ ≥ 5°C for EN ISO 6946) or $\Delta\theta_{ss} = \Delta\theta_{ss,default}$ (see C.2)

Standard	*Heat flow direction*	C_1	C_2	C_3	$\Delta\theta_{ss,default}$
EN ISO 10077-2	horizontal	0.025	0.73	0.333333	10°C
EN ISO 6946	horizontal	0.025	0.73	0.333333	5°C
	upward	0.025	1.14	0.333333	5°C
	downward	0.025	$0.09 \cdot d^{-0.44}$	0.187	5°C

- h_r = radiative heat transfer coefficient
  EN ISO 10077: $h_r = 4 \cdot 5.67 \cdot 10^{-8} \cdot (\theta_m + 273.16)^3 \cdot E \cdot F$
  EN ISO 6946: $h_r = 4 \cdot 5.67 \cdot 10^{-8} \cdot (\theta_m + 273.16)^3 / \left(\frac{1}{E} + \frac{1}{F} - 1\right)$

$$E = \frac{1}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1}$$

$$F = \left(1 + \sqrt{1 + \left(\frac{d}{b}\right)^2} - \frac{d}{b}\right) \Big/ 2$$

$$\theta_m = \frac{\theta_{s,min} + \theta_{s,max}}{2}$$

$\theta_{s,min}$ = minimum surface temperature in air cavity [°C]
$\theta_{s,max}$ = maximum surface temperature in air cavity [°C]
or θ_m = 10°C (default mean temperature, see **Error! Reference source not found.**)
ε_1, ε_2 = emissivities at warm and cold sides of air cavity (default values: ε_1 = ε_2 = 0.9)

<u>Equivalent thermal conductivity of slightly ventilated air cavity</u> (subtype CAVITY_E (EN ISO 10077))
An air cavity is considered to be slightly ventilated when there is an open connection to an exterior or interior boundary condition with a slit between 2 and 10 mm, and $d \geq b$.
λ_{eq} = 2 · (λ_{eq} of unventilated air cavity).

<u>Equivalent thermal conductivity of air layer</u> (subtype LAYER (EN ISO 6946))
In the formula above for subtype CAVITY set d/b = 0 for h_r.

<u>Adiabatic boundaries</u>
An adiabatic boundary (i.e. a boundary through which there is no heat transfer) can be considered as a symmetry axis (with symmetric heat flow pattern).
When there is an adiabatic boundary at one side of the width direction, the equivalent width *b* is automatically doubled for subtype CAVITY.
When there are adiabatic boundaries at both sides of the width direction, the equivalent width *b* is automatically considered to be infinite for subtype CAVITY (so *d/b* = 0, as for subtype LAYER).

C.1.2. Colour type BC_SIMPL or BC_FRE_S

Purpose: simplified surface boundary condition (using a global surface heat transfer coefficient). For BC_SIMPL, the environment temperature is defined by the user. For BC_FRE_S, the environment temperature is unknown (free-floating).

Type	*Standard*	*Subtype*	*Physical heat flow*	*Meaning*
BC_SIMPL		NIHIL		Global surface heat transfer coefficient h and temperature θ defined by user
	EN ISO 10077	HI_NORML	horizontal	Normal h at interior surface: $h = 7.7\ W/(m^2.K)$, $\theta_i = 20°C$
		HI_REDUC	horizontal	Reduced h at interior surface: $h = 5\ W/(m^2.K)$, $\theta_i = 20°C$
		HE		h at exterior surface: $h = 25\ W/(m^2.K)$, $\theta_e = 0°C$
		HI_CAV_E	horizontal	Slightly ventilated air cavity: $h = 3.33\ W/(m^2.K)$, $\theta_i = 20°C$
		HE_CAV_E	horizontal	Slightly ventilated air cavity: $h = 3.33\ W/(m^2.K)$, $\theta_e = 0°C$
	EN ISO 6946	HI	horizontal	h at interior surface: $h = 7.7\ W/(m^2.K)$
			upward	h at interior surface: $h = 10\ W/(m^2.K)$
			downward	h at interior surface: $h = 5.9\ W/(m^2.K)$
		HE		h at exterior surface: $h = 25\ W/(m^2.K)$

C.1.3. Colour type BC_FREE

(This type requires the RADCON module, see Chapter G)

Purpose: enclosure with unknown (free-floating) air temperature to be calculated from convective heat balance and radiation based on view factors.

Type	*Subtype*	*Meaning*
BC_FREE	NIHIL	Convective surface heat transfer coefficient h_c defined by user
	CAVITY	Air cavity with given heat flow direction
	CONVEC	Large air space, without preferential heat flow direction*

*typically larger than 0.3m

<u>CAVITY</u>:
The CAVITY subtype is to be used when a global heat flow direction is assumed (surface-to-surface heat transfer with an air temperature in between temperature of 'hot' and 'cold' surface temperatures, e.g. typically used for a cavity in a façade).
The characteristic dimensions of the air cavity, *d* (= depth, parallel to heat flow) and *b* (= width), are calculated as for colour type EQUIMAT (see above).
h_c = convective surface heat transfer coefficient (surface to air)

$$h_c = 2 \cdot max\left\{\frac{C_1}{d}, C_2 \cdot (\Delta\theta_{ss})^{C_3}\right\}$$

$\Delta\theta_{ss}$= maximum surface temperature difference in air cavity
or $\Delta\theta_{ss}$ = calculation parameter "Default temperature difference across airspace" (see C.2)

($\Delta\theta_{ss} \geq 5$°C)

Heat flow direction	*C_1*	*C_2*	*C_3*
horizontal	0.025	0.73	0.333333
upward	0.025	1.14	0.333333
downward	0.025	$0.09 \cdot d^{-0.44}$	0.187

<u>CONVEC:</u>
The subtype CONVEC must be used when no global heat flow direction can be assumed in the enclosure (surface-to-air heat transfer with an air temperature higher or lower than surface temperatures, e.g. typically used for a room).
h_c= convective surface heat transfer coefficient (surface to air)
$h_c = 2 \cdot C_2 \cdot (\Delta\theta_{sa})^{C_3}$
$\Delta\theta_{sa}$= greatest difference between air temperature and surface temperature
$\Delta\theta_{sa} = \max\left(|\theta_a - \theta_{s,min}|, |\theta_a - \theta_{s,max}|\right)$
θ_a = air temperature
$\theta_{s,min}$ = minimum surface temperature in contact with given boundary condition colour
$\theta_{s,max}$ = maximum surface temperature in contact with given boundary condition colour
or $\Delta\theta_{sa}$= calculation parameter "Default temperature difference across airspace" (see C.2)

Heat flow direction	*C_2*	*C_3*
horizontal	0.73	0.333333
upward	1.14	0.333333
downward	0.09	0.187

C.1.4. Colour type BC_SKY or BC_NOSKY

(This type requires the RADCON module, see Chapter G)

Purpose: environment with known air temperature and either user defined radiation temperature (BC_SKY) and radiation based on view factors or no radiation exchange (BC_NOSKY).

Type	*Subtype*	*Meaning*
BC_SKY or BC_NOSKY	NIHIL	Convective surface heat transfer coefficient h_c defined by user
	CONVEC	Natural convection surface-air (calculation of h_c: cf. above BC_FREE - CONVEC, see C.1.3)

C.1.5. Colour type TRANSMAT

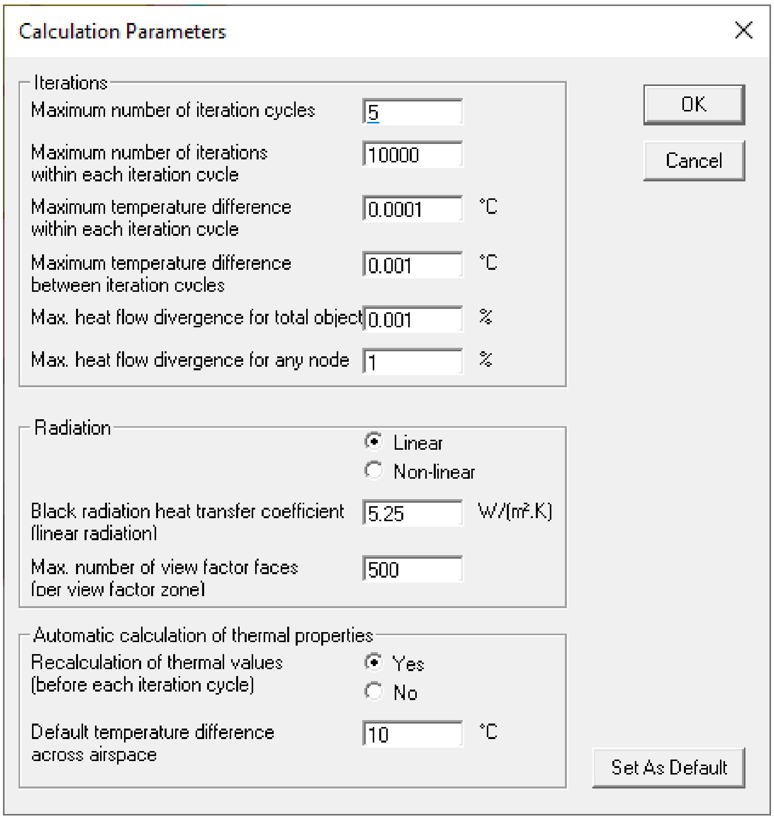
(This type requires the RADCON module, see Chapter G)

Purpose: air cavity modelled as *transparent material* (radiation based on view factors, cf. radiosity method (EN ISO 10077))

Type	*Subtype*	*Meaning*
TRANSMAT	NIHIL	Thermal conductivity defined by user
	CAVITY	Unventilated air cavity (calculation of λ_{eq}: cf. above EQUIMAT, CAVITY with h_r = 0, because radiation is calculated using view factors)

C.2. Calculation parameters

Menu command: *Edit → Calc Parameters...*



The calculation parameters control the thermal system calculation (*Calc → Calc System*).
The factors that are not relevant to the current data file are not editable (disabled input fields in grey).

<u>Iteration cycles</u>
A non-linear problem (e.g. radiation or due to temperature dependent properties) is solved by splitting the problem in a sequence of linear problems.
Each linear problem is called an *iteration cycle*.
At the start of each iteration cycle the non-linear thermal properties are updated based on the calculation results of the previous iteration cycle to achieve renewed linear properties that can be used in the linear problem.
The iteration cycle is then solved using an iterative method.
At the end of the iteration cycle the new solution is compared to the solution of the previous iteration cycle to check the convergence.
Parameters:

- *Maximum number of iteration cycles*: this is a limit on the total number of iteration cycles (needed when there is no convergence).
  Normally 5 iteration cycles are sufficient to solve most non-linear systems with adequate precision.

- *Maximum temperature difference between iteration cycles*: this is the maximum allowed temperature difference in any node between the previous and the current iteration cycle solution.
  When the temperature differences in all nodes are smaller, then the iteration cycles are stopped.

Iteration stopping criteria (within each iteration cycle)

- *Maximum number of iterations*: this is a limit on the total number of iteration steps (needed when the other stopping criteria fail).
- *Maximum temperature difference*: this is the maximum allowed temperature difference in any node between the previous and the current iteration step.
- *Maximum heat flow divergence for total object*: this is the maximum allowed relative energy balance error for the total object (as union of all elementary control volumes).
  (ISO 10211 requires max. 0.01 %)
  The relative energy balance error equals the absolute value of the sum of all elementary heat flows (positive and negative) divided by half the sum of the absolute values of all these heat flows. The theoretical maximum value is 200 %. This happens e.g. for an object with only incoming heat flows and no outgoing heat flows.
- *Maximum heat flow divergence for any node*: this is the maximum allowed relative energy balance error for the worst elementary control volume.
  (Sometimes this condition is too severe; in that case enter 200 %, or better 200.1 % to avoid problems due to numerical rounding errors.

The iteration stops when the last three conditions are satisfied or when the *maximum number of iterations* is reached.

Radiation (requires RADCON module, see Chapter G)
The radiation between two black surfaces is expressed as:
$q_r = F_{ij} \cdot h_{rb} \cdot (\theta_i - \theta_j)$
q_r= radiation heat flux [W/m2]
F_{ij}= view factor from surface i to surface j (= fraction of radiation leaving surface i that is intercepted by surface j)
h_{rb}= black radiation heat transfer coefficient [W/(m2.K)]
θ_i = temperature of surface i [°C]
θ_j = temperature of surface j [°C]
The radiation between grey surfaces, having an emissivity (as defined in the **Colours window**), is derived from the radiation between the black surfaces by adding supplementary resistances between the grey and black surface nodes.
Radiation calculation parameters:

- *Linear radiation*: the black radiation heat transfer coefficient is constant (and defined as a calculation parameter).
- *Non-linear radiation*: the black radiation heat transfer coefficient is locally adjusted in function of the surface temperatures, before each new iteration cycle.
- *Black radiation heat transfer coefficient*: parameter used for linear radiation or when no surface temperatures are available (i.e. for first iteration cycle in case of non-linear radiation).
  Normal value: 5.1 W/(m^2.K) (for mean temperature = 10°C).

The view factors calculation between elementary surfaces is affected by the following parameters:

- *Smallest accepted view factor*: calculated view factors that are smaller than this parameter, are ignored (rounded off to 0).
- *Maximum number of view factor faces (per view factor zone)* : calculation grid radiative surfaces are clustered into larger view factors if the number of surfaces in a view factor

zone exceeds this parameter. In general, the *view factor grid* may be less fine than the *system grid* without loss of accuracy. Very small grid meshes do not bring significantly better results when the surfaces are at a relatively large distance from one another, although the calculation time is much larger (see C.3).

Automatic calculation of thermal properties

- *Recalculation of thermal values*: yes or no.
  If yes: the automatically calculated thermal properties in the **Colours window** (equivalent thermal conductivity and convective heat transfer coefficient, see C.1) are recalculated before each new iteration cycle.
- *Default temperature difference across airspace*: parameter used in automatic calculation of thermal properties when no temperature solution and no default temperature difference according to the relevant standard are available (see C.1).

C.3. View factors calculation (requires RADCON module)

Menu command: *Calc → Calc Viewfactors*
Input: *fitted object*

The command is only enabled when one or more view factor based zones (colour type BC_FREE, BC_SKY, BC_NOSKY or TRANSMAT) are present.
The view factors calculation is automatically included in a thermal system calculation (C.4), but can also be executed separately, to get the global view factors between all different colour surfaces in the **Text Output window** (see E.4.4).

First the following preliminary steps are automatically executed (if not yet available):

- the object blocks are filled (as with *Image → Fill Blocks (Triangulate)*, see B.4.3) to obtain the *triangulated object*
- the object is rasterised using the grid (as with *Image → Fill Grid (Rasterise)*, B.4.4) to obtain the *rasterised object*
- the surface nodes are fitted (as with *Image → Fit Grid*, B.4.5) to obtain the *fitted object*

The view factors calculation is based on the object surfaces of the *fitted object* bounding the zones of colour type BC_FREE, BC_SKY, BC_NOSKY and TRANSMAT.

Zones
The view factor boundary condition blocks (of colour type BC_FREE, BC_SKY, BC_NOSKY, TRANSMAT) are classified in separate zones. Adjacent boundary condition blocks of different colours must have the same type and are combined in the same view factor zone.
The same colour (of a view factor BC) is not allowed to occur in different zones (otherwise an error message appears when starting the calculation).

View factor between two surfaces
For two small surfaces at a large distance the following formula is used:

$$F_{ij} = \frac{A_j \cos\theta_i \cos\theta_j}{\pi\ R^2}$$

F_{ij} = view factor from surface i to surface j (= fraction of radiation leaving surface i that is intercepted by surface j)
A_j = area of surface j
θ_i = angle between normal vector of surface i and line between gravity centres of surfaces i and j
θ_j = angle between normal vector of surface j and line between gravity centres of surfaces i and j
R = distance between gravity centres of surfaces i and j

Visibility test

Other surfaces may be located in between both view factor surfaces and obstruct their mutual visibility. If the line between the gravity centres of both small surfaces intersects any other surface, then we assume $F_{ij} = 0$.

Subdivision of large surfaces

Large surfaces are automatically subdivided into smaller triangles to achieve a better accuracy.

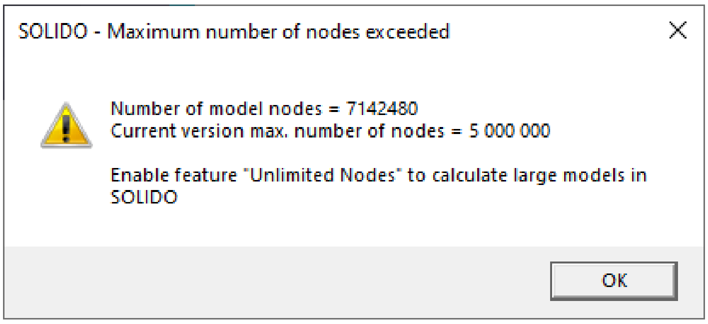
Clustering of very small surfaces

Very small surfaces are automatically clustered into larger surfaces, based on orientation, area and proximity, to achieve a higher calculation efficiency.
The total number of surfaces (after clustering) per zone is a calculation parameter (see C.2), to limit the number of view factor calculations. Between all surface clusters the view factors are calculated, based on the formula above, applied to the gravity centres of each surface cluster and the apparent cluster surface areas.
The calculation of surface clusters is proportional to the number of the grid surfaces in the *fitted object* bounding the view factor zones.
The view factor between two individual grid surfaces of the *fitted object* can be derived from the view factor between the correspondent surface clusters, using surface weighted interpolation.

C.4. System calculation

Menu command: *Calc* → *Calc System*

For a standard version of SOLIDO, the number of nodes is limited to 5 000 000. When the calculation of a model with a higher number of nodes is started, a warning appears and the calculation is not performed. The add-on feature 'Unlimited Nodes' removes this limitation on the number of nodes.



Input:

- Object geometry (defined in **Blocks window**, **Block Points window**, Coordinates dialog box)
- Thermal properties (defined in **Colours window**)
- Calculation grid (defined in **Grid window**)
- Calculation parameters (defined in Calculation Parameters dialog box)

Before starting a thermal calculation it is good practice to verify the presence and correctness of the boundary conditions on the object surfaces of the *fitted object* in the **Image window** using the option *Image* → *Show BCs* (B.4.7).

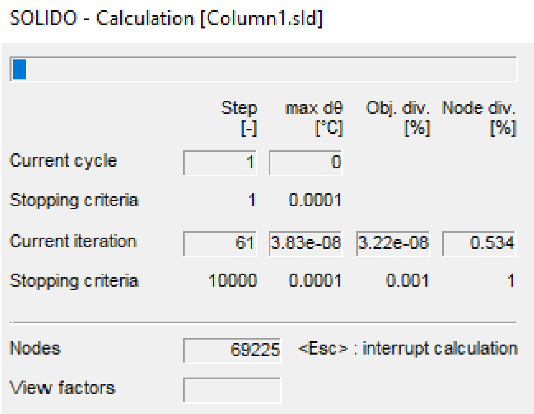
Output:

- Solution file(s): having the same file name as the data file and stored in the same directory. The node temperatures are saved in a file with extension .sol.

Heat flux values calculated using the RADCON module are saved in a file with extension .flx.
- Surface temperatures on *fitted object* are displayed in the **Image window**
- Recalculated automatically calculated thermal properties in **Colours window** (if requested)

<u>Calculation procedure:</u>
1. The input data are saved on disk (in a data file with extension .sld)
2. Calculation of *triangulated object* (if not yet done using *Image → Fill Blocks (Triangulate)*, see B.4.3)
3. Calculation of *rasterised object* (if not yet done using *Image → Fill Grid (Rasterise)*, see B.4.4)
4. Calculation of *fitted object* (if not yet done using *Image → Fit Grid*, see B.4.5)
5. View factors calculation (if required)
6. System solution: a non-linear system is iteratively solved by a sequence of iteration cycles. Each iteration cycle contains the solution of a linearised system:
   1. If required, recalculation of automatically calculated thermal properties based on last available node temperatures, or default temperatures (when no temperature solution available) (see C.1)
   2. Building of linear system of energy balance equations for control volumes around the individual system nodes
   3. Iterative solution of linear system until stopping criteria are fulfilled

The calculation progress can be followed in the calculation monitor.



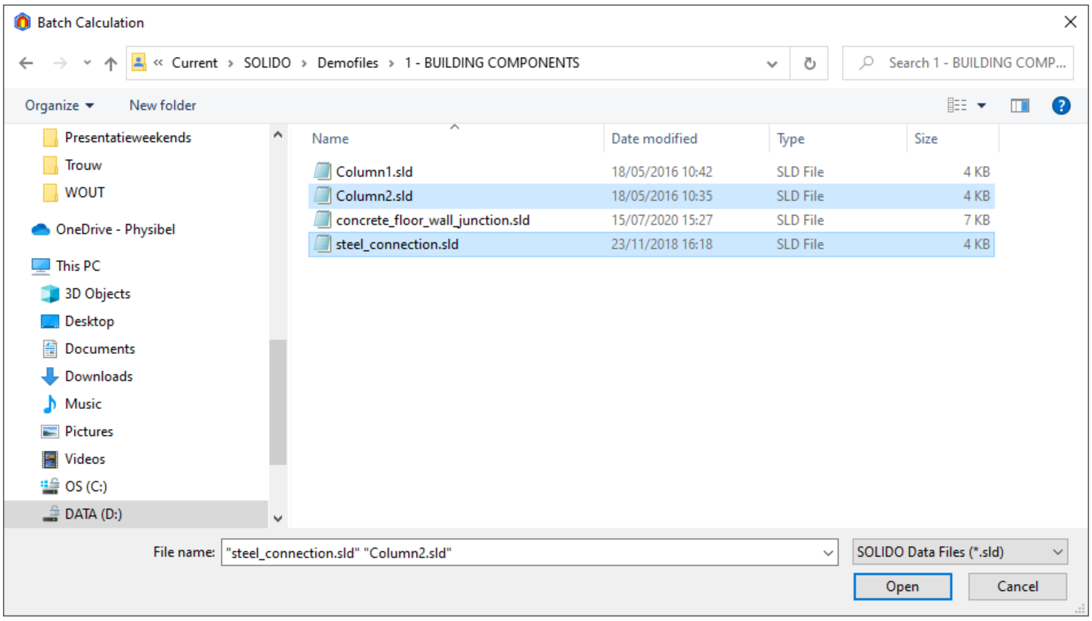
The calculation can be interrupted by pressing the <Esc> key. When the iteration process has already started, the last obtained temperatures are saved in a solution file (extension .sol). When restarting the calculation, these intermediate temperatures are reloaded and the iterative calculation procedure is continued.

C.5. Batch calculation

Menu command: *File → Batch Calculation…*

A group of several SOLIDO data files is launched at one time to be calculated one after the other. Select the data files in the same data folder. It is not possible to select data files located in different directories. Use the Ctrl-key or the Shift-key to select more than one file.

At the end of the calculation for each data file the calculation results are written in a text file (FileName_RES.txt, with FileName the name of the data file).



C.6. Command line program execution

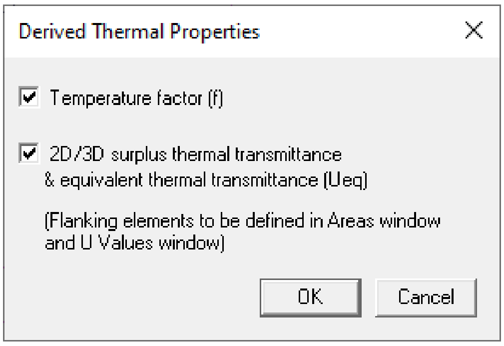
SOLIDO can be started from the command line prompt with a data file path (including directory and file extension .sld) as parameter. Then SOLIDO is opened for this data file.
If the switch "/Automated" is added, the calculation is immediately started for that data file (as in a batch calculation).

Example of command line launch:

Run → "c:\...\Physibel\SOLIDO15\SOLIDO5.exe" "c:\...\Physibel\SOLIDO5\Demofiles\1 – BUILDING COMPONENTS\Column1.sld" /Automation

C.7. Derived thermal properties

Menu command: *Edit → Derived Thermal Properties...*



This dialog box defines which derived thermal properties should be added to the calculation results in the **Text Output window** (see Chapter E). The derived thermal properties are derived from the calculation solution. The desired properties must be checked in the dialog box.

C.7.1. Temperature factor

The temperature factor is a dimensionless expression of the minimum internal surface temperature, in order to assess the risk of surface condensation. It can only be evaluated if two boundary conditions with different temperatures are present: the external temperature and the internal temperature.

The evaluation formula (cf. standards ISO 10211 and EN ISO 13788) is:

$$f = \frac{\theta_{si,min}}{\theta_i - \theta_e}$$

f = temperature factor [-]
θ_i = internal boundary condition temperature [°C]
θ_e = external boundary condition temperature [°C]
$\theta_{si,min}$ = minimum internal surface temperature [°C]

The output also reports the internal heat transfer coefficient h_i [W/(m^2.K)] and internal surface resistance R_{si} [m^2.K/W] where $\theta_{si,min}$ applies.

C.7.2. 2D/3D surplus thermal transmittance & equivalent thermal transmittance

These properties can only be reported if two boundary conditions with different temperatures are present and at least one area defined in the **Areas window** does not equal zero.

The 2D/3D surplus thermal transmittance is a correction term for the 2D and 3D influences in a thermal bridge when calculating the heat flow (per unit temperature difference between both environments) from 1D calculations. For a 2D object the definition corresponds to the *linear thermal transmittance.*

The equivalent thermal transmittance is the U value of an 1D element, that is thermally equivalent to the calculated object, with the same heat flow going through that element for a user defined reference wall area.

The applied formulas (cf. norm ISO 10211) are:

$$\Delta L = \frac{\Phi}{\theta_i - \theta_e} - U_1 * A_1 - U_2 * A_2 - U_3 * A_3$$

$$U_{eq} = \frac{\Phi}{(\theta_i - \theta_e) * (A_1 + A_2 + A_3)}$$

ΔL = 2D/3D surplus thermal transmittance [W/K]
U_{eq} = equivalent thermal transmittance [W/(m^2.K)]
Φ = total heat flow through object [W]
θ_i = internal boundary condition temperature [°C]
θ_e = external boundary condition temperature [°C]
U_1, U_2, U_3 = 1D thermal transmittances of flanking elements [W/(m^2.K)]
A_1, A_2, A_3 = areas of flanking elements [m^2]

The areas and U values of the flanking elements must be defined in the **Areas window** (see B.17) and **U Values window** (see B.18).

In case of a 2D problem the linear thermal transmittance or psi-value Ψ [W/(m.K)] is obtained by dividing ΔL by the single mesh width of the third dimension.

C.7.3. Example of text output

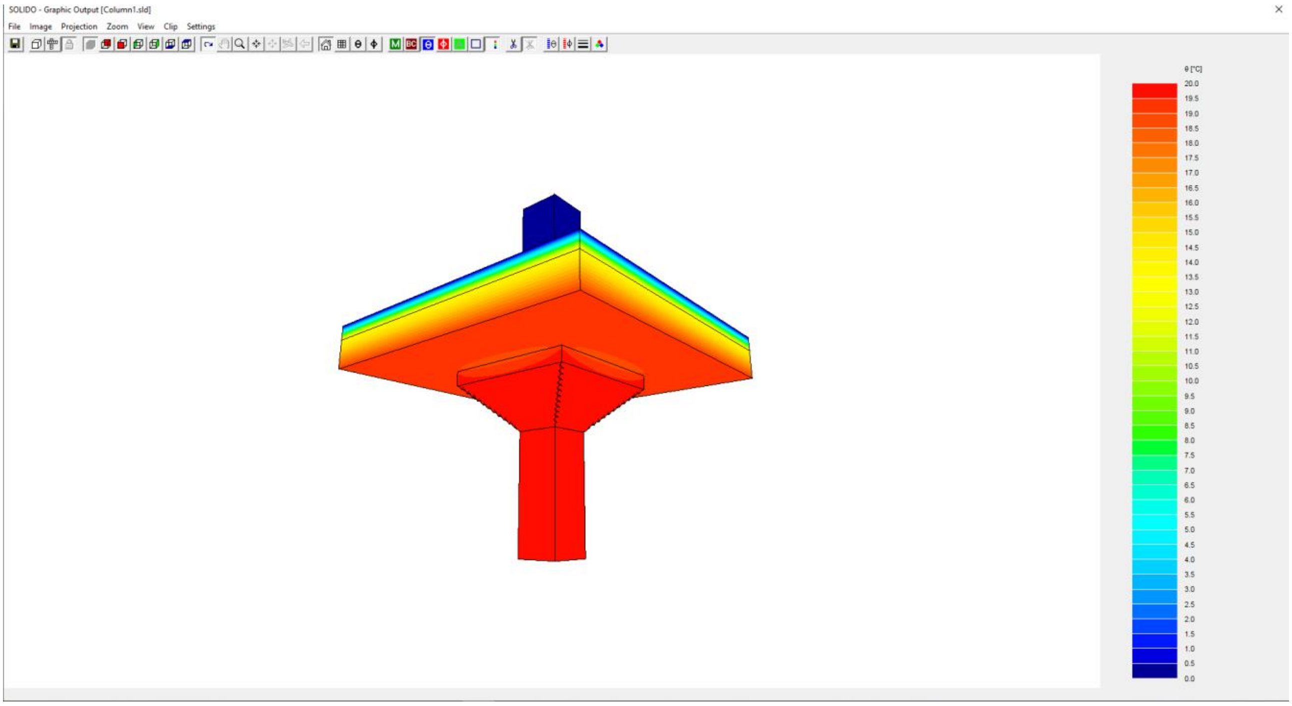
Below is an example of additional output of calculation results, corresponding to derived thermal properties (see Chapter E).

```
Temperature factor (EN ISO 10211-2) = 0.451
  hi = 5.00 W/(m².K)
  Rsi = 0.20 m².K/W

2D/3D surplus thermal transmittance
```

```
dL = Q/(ti-te) - U1*A1 - U2*A2 - U3*A3 = 1.906 W/K
Equivalent thermal transmittance
Ueq = Q/((ti-te)*(A1+A2+A3)) = 0.973 W/(m².K)
  Q = 58.368 W
  ti = 20.00°C
  te = 0.00°C
  U1 = 0.000 W/(m².K)
  A1 = 1.00 m²
       X1=40.00 Y1=40.00 Z1=0.00
       X2=40.00 Y2=140.00 Z2=0.00
       X3=40.00 Y3=140.00 Z3=100.00
  U2 = 0.504 W/(m².K)
       X1=120.00 Y1=10.00 Z1=20.00
       X2=120.00 Y2=40.00 Z2=20.00
  A2 = 1.00 m²
       X1=40.00 Y1=40.00 Z1=0.00
       X2=140.00 Y2=40.00 Z2=0.00
       X3=140.00 Y3=40.00 Z3=100.00
  U3 = 0.508 W/(m².K)
       X1=120.00 Y1=120.00 Z1=100.00
       X2=120.00 Y2=120.00 Z2=120.00
  A3 = 1.00 m²
       X1=40.00 Y1=40.00 Z1=100.00
       X2=140.00 Y2=40.00 Z2=100.00
       X3=140.00 Y3=140.00 Z3=100.00
```

D.1.Graphic Output window

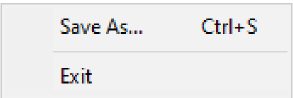
Menu command: *Output → Graphic Output*
A new full-screen window with its own menu, toolbar and status bar are displayed. To return to the main SOLIDO window, choose *File → Exit*



No viewpoint depending object shading is used in the **Graphics Output window** as is done in the **Image window**. All colours are flat and are explained in the legend bar at the right.
Object depth is suggested with contour lines.
The status bar shows (dependent on the view option) the material colour, boundary condition colour, temperature or heat flux under the mouse cursor.

D.2. File commands

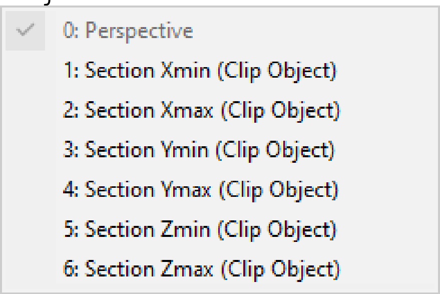
File submenu:



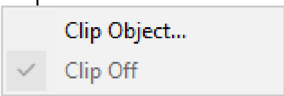
Menu command: *File → Save As...*
Save the graphic output as a PNG file, for which the image size and font size of legend text are defined via the image size settings (see 0).

D.3. Projections

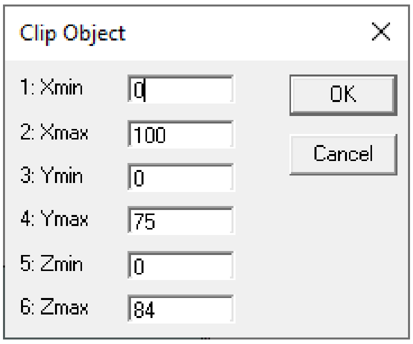
Projection submenu



Clip submenu



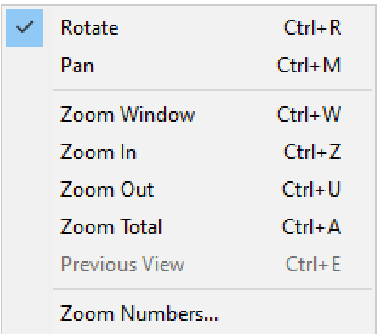
- *Projection* → *Perspective*: same 3D perspective view as in **Image window** (see B.3.2).
- *Projection* → *Section ...*: section view through object along grid coordinate.
  The clip object boundaries as defined in either the **Clip Object window** from the main window (cf. section B.15) or the Clip Object dialog box (*Clip* → *Clip Object...* from the **Graphics Output window**) are used as section definitions.
  The view directions for the sections through the minimum and maximum clip coordinates are facing each other (i.e. viewing towards the clipped object).

  By editing the clip coordinates, cross sections through all grid planes are possible.
  If a clipping plane does not intersect the object, then the corresponding section view is empty.
  Reset all clip coordinates using *Clip* → *Clip Off*.

D.4. Zoom commands

Zoom submenu



The zoom commands are identical to those of the **Image window** in the main menu (see B.4.2). However a mouse drag movement in rotate or pan mode works not dynamically as in the **Image window** (because the Graphic Output window does not make use of OpenGL hardware

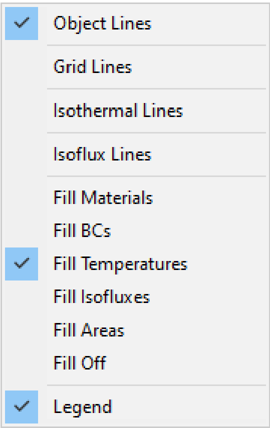
acceleration). A rubber-band vector is drawn, and after releasing the left mouse button the new view is painted.
Zooming in and out is also possible via scrolling forward and backward of the mouse wheel.

D.5. Image commands

Image → Show Triangulated Object
Image → Show Rasterised Object
Image → Show Fitted Object

D.6. View commands

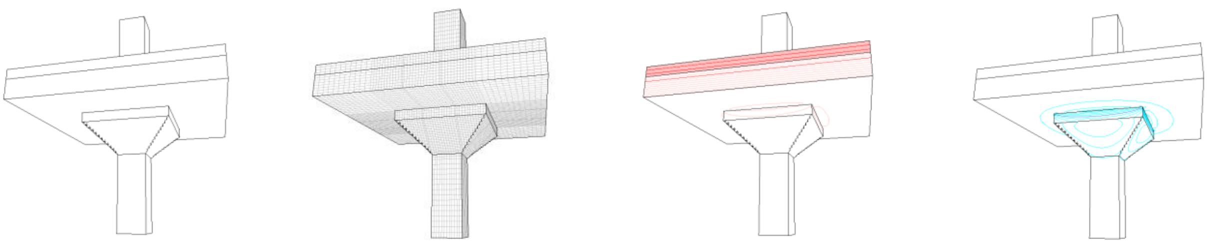
View submenu:



A check mark in front of a command shows that the option is enabled.
Any change in the view options is immediately reflected in the image.

The first 4 options can be switched on or off independently:

- *View → Object Lines*: show object contour lines.
- *View → Grid Lines*: show grid lines on object surfaces.
  For the *triangulated object* the mesh lines refer to the triangulation of the surfaces of the solid blocks and STL blocks (in accordance to the mesh sizes in the **Blocks window**).
  For the *rasterised object* and *fitted object* the mesh lines refer to the grid definition in the **Grid window**.
- *View → Isothermal Lines*: show isothermal lines conform to the temperature range parameters (see D.7).
- *View → Isoflux Lines*: show isoflux lines conform to the flux range parameters (see D.7).

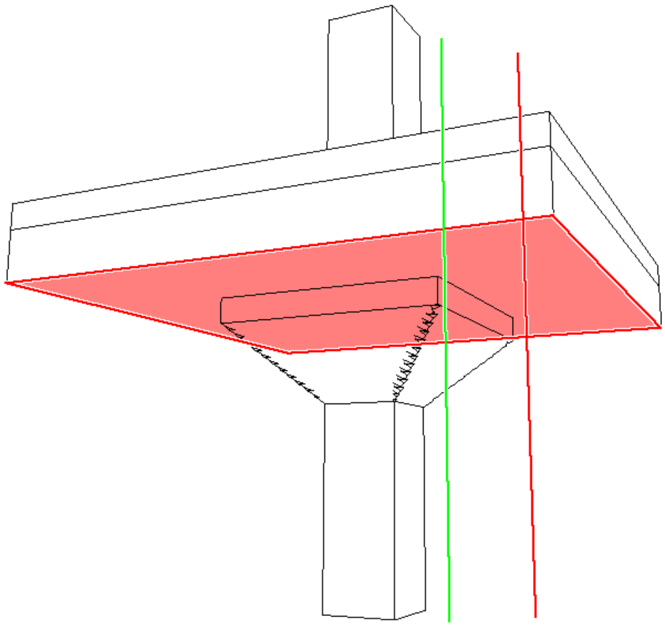


One of the following fill modes is to be selected:

- *View → Fill Materials*: fill material surfaces using the object block colours (as defined in the **Colours window**).
- *View → Fill BCs*: fill material surfaces using the adjacent surface boundary condition block colours (as defined in the **Colours window**).

If no surface boundary condition is active on a material surface (i.e. an implicit adiabatic boundary condition is assumed), then the fill colour is light yellow.

- *View → Fill Temperatures*: fill material surfaces using temperature colours conform to the temperature range parameters (see D.7) and of the selected thermal palette (see A.9 and D.7).
- *View → Fill Isofluxes*: fill material surfaces using isoflux colours conform to the flux range parameters (see D.7).
- *View → Fill Areas:* show areas (as defined in **Areas window**) and lines along which U values are determined (as defined in **U Values window**). Each couple of corresponding area and U value (as in the formulas of derived thermal transmittances) has a different colour (red, green or blue).



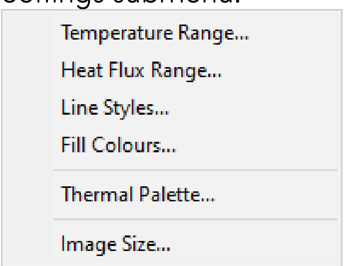
- *View → Fill Off*: switch off filling.

The legend can be shown or hidden with:

- *View → Legend*

D.7. Settings

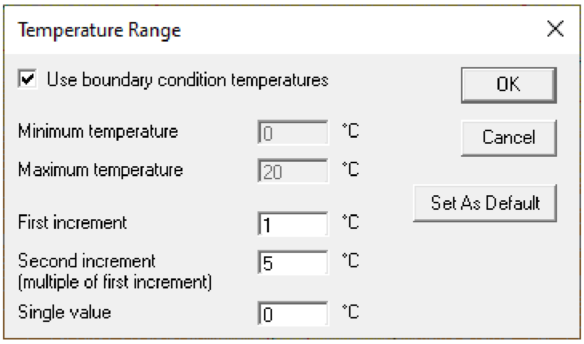
Settings submenu:



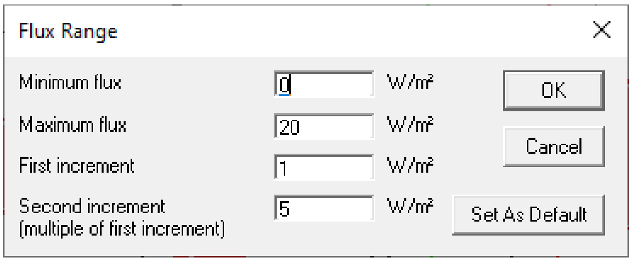
Settings → Temperature Range...
The parameters in this dialog box are synchronised with those in the **Image window** (see B.4.7).

- The option "Use boundary condition temperatures" initialises the minimum and maximum temperatures from the boundary condition temperatures of types BC_SIMPL present in the **Colours window**.
- Minimum temperature [°C] in graphic output.
- Maximum temperature [°C] in graphic output.

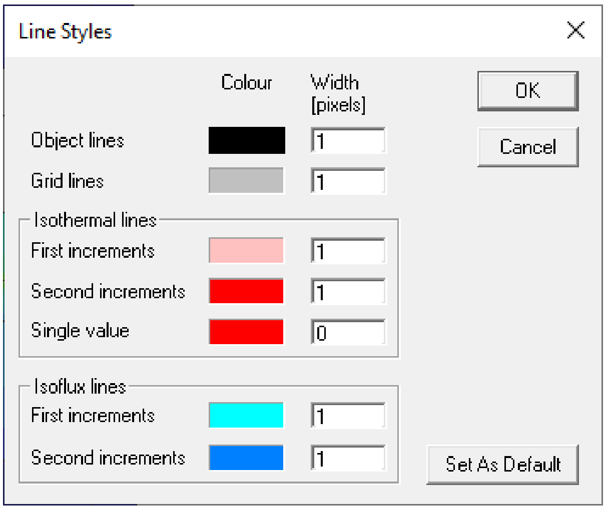
– First (or minor) increment [°C].
– Second (or major) increment [°C] (must be a multiple of first increment).
  The second increment is only used with isothermal lines.
– Single temperature value [°C] (e.g. dew point).



Settings → Heat Flux Range...
To define the minimum and maximum flux and flux increments for isoflux lines and isoflux fill colours.
The second flux increment is only used for drawing enhanced isoflux lines.
The second increment must be an integer multiple of the first increment.



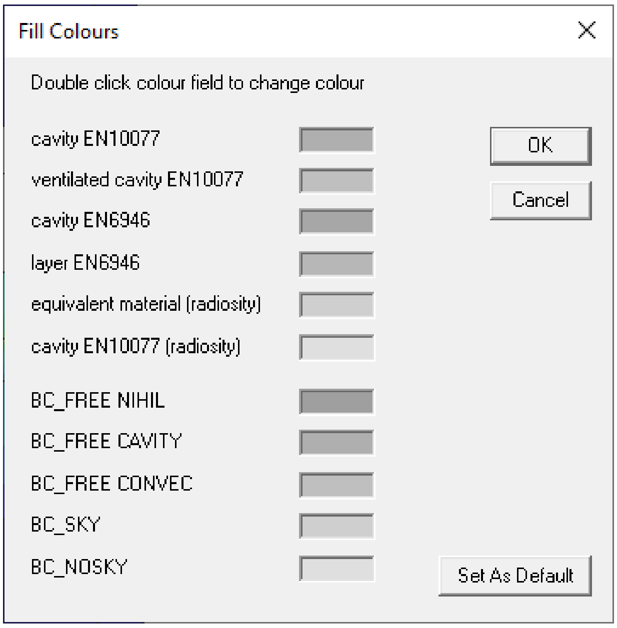
Settings → Line Styles...
To define the line colour and line thickness of the object lines, grid lines, isothermal lines and isoflux lines.



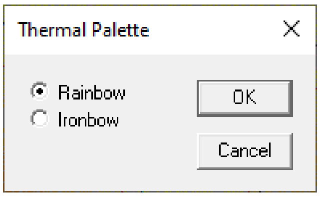
A colour field can be edited by double clicking on it with the left mouse button, which opens a Colour dialog box in which the colour can be defined.
The width of isothermal lines, isoflux lines and heat flow lines can be set to 0 pixels to hide them. This is useful when e.g. only the second isothermal lines are desired.

Settings → Fill Colours…

Colours with automatically calculated properties (see C.1) are clustered thematically in the **Graphic Output window** according to the categories of the Fill Colours dialog box, to avoid a potentially long list of colours of the same type (and subtype and standard) with small variations in the calculated thermal property. The fill colour of the clusters can be edited by double clicking on it with the left mouse button, which opens a Colour dialog box in which the colour can be defined.

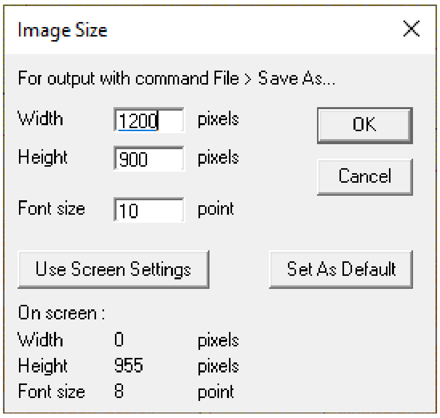


Settings → Thermal palette…



The selection of the thermal palette is synchronised with that selected in the main window (see A.9A.9).

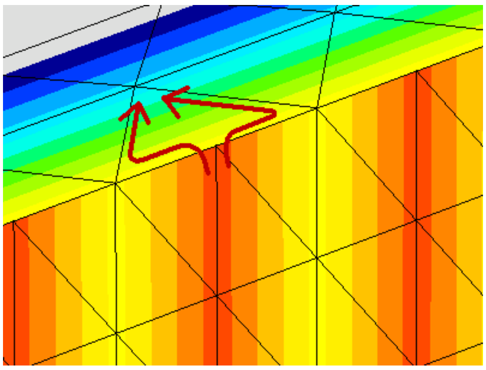
Settings → Image Size...
Set image size for saving (see D.2).



- The font size for text must be chosen in proportion to the image size.
- The file format is always PNG.

D.8. Local surface temperature smoothing

The calculation is based on the energy balance per control volume around each system node. The surface temperatures can exceed the minimum and maximum temperatures of the boundary conditions in certain nodes, because the calculation method is based on *mean* surface temperatures.
Also rather important local temperature differences can be found on the surface of a single grid cell, where theoretically a uniform temperature would be expected (e.g. due to unidirectional heat transfer). This can occur when that surface is not parallel to a coordinate plane. In that case the *rasterised object* shows a stepwise surface, containing inner and outer corner nodes. In the *fitted object* (i.e. after projection of the surface nodes to the original slanted surface) the inner and outer corner nodes become coplanar (cf. figure below). However the heat flow entering the control volume of an originally outer corner node (cf. red arrows in figure below) must first travel along the surface, before turning into the material, because the outer corner vertex is only connected to other surface nodes and not directly to a node inside the material. After the system solution SOLIDO applies an automatic surface filter to smoothen these local temperature differences.



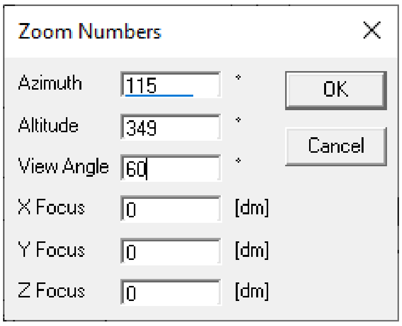
D.9. Graphic output hints

D.9.1. Interrupt drawing

Press the <Esc> key to interrupt and stop the calculation of a drawing.

D.9.2. Image reproduction

Make a note of the zoom numbers (see D.4) to be able to reproduce the same view at a later moment.
If possible, use rounded zoom numbers.

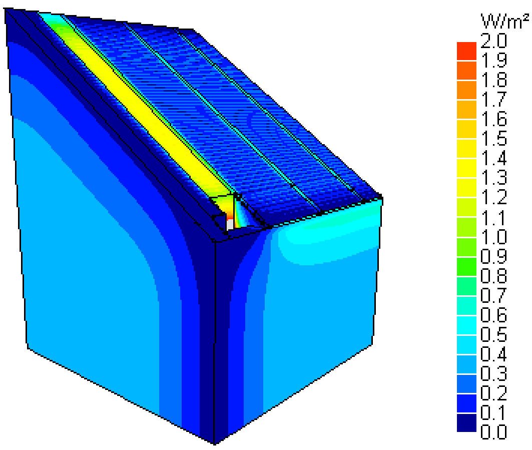


D.9.3. Object model

In general the object lines are cleaner for the *triangulated object* than for the *fitted object* (see D.5).
The triangulation mesh sizes in the **Colours window** must be of the same order of size as the grid mesh sizes in the **Grid window** to achieve a reliable result of the temperature and isoflux image.

D.9.4. Visualise thermal transmittances

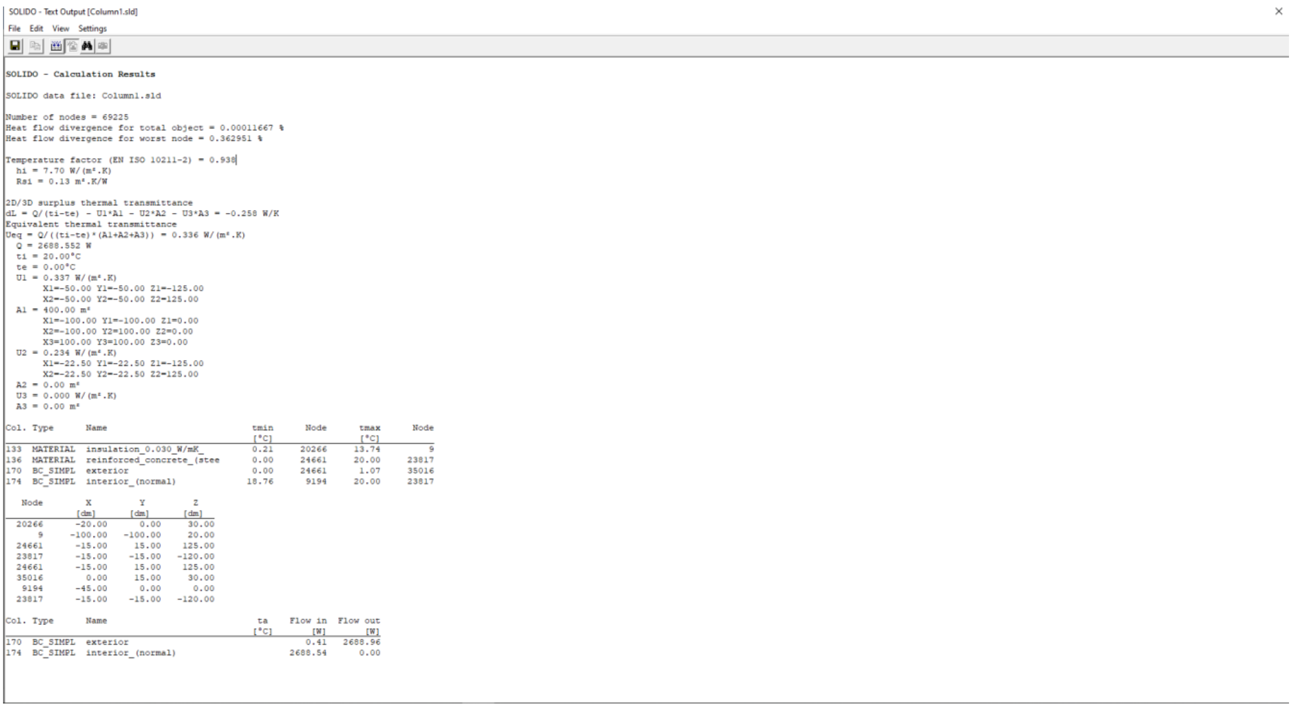
A thermal transmittance (or U value) is expressed in W/(m^2.K).
Fluxes through the object surfaces are expressed in W/m^2.
When there is a temperature difference of 1°C between the indoor and outdoor environment, the isofluxes on a graphic output thus show the local thermal transmittances on the object surfaces.



CHAPTER E TEXT OUTPUT

E.1. Text Output window

Menu command: *Output → Text Output*
A new full-screen window with its own menu and toolbar are displayed. To return to the main SOLIDO window, choose *File → Exit*



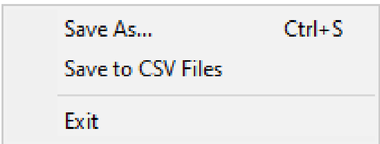
SOLIDO - Text Output [Column1.sld]

File Edit View Settings

```
SOLIDO - Calculation Results

SOLIDO data file: Column1.sld

Number of nodes = 69225
Heat flow divergence for total object = 0.00011667 %
Heat flow divergence for worst node = 0.362951 %

Temperature factor (EN ISO 10211-2) = 0.938
  hi = 7.70 W/(m².K)
  Rsi = 0.13 m².K/W

2D/3D surplus thermal transmittance
dL = Q/(ti-te) - U1*A1 - U2*A2 - U3*A3 = -0.258 W/K
Equivalent thermal transmittance
Ueq = Q/((ti-te)*(A1+A2+A3)) = 0.336 W/(m².K)
  Q = 2688.552 W
  ti = 20.00°C
  te = 0.00°C
  U1 = 0.337 W/(m².K)
       X1=-50.00 Y1=-50.00 Z1=-125.00
       X2=-50.00 Y2=-50.00 Z2=125.00
  A1 = 400.00 m²
       X1=-100.00 Y1=-100.00 Z1=0.00
       X2=-100.00 Y2=100.00 Z2=0.00
       X3=100.00 Y3=100.00 Z3=0.00
  U2 = 0.234 W/(m².K)
       X1=-22.50 Y1=-22.50 Z1=-125.00
       X2=-22.50 Y2=-22.50 Z2=125.00
  A2 = 0.00 m²
  U3 = 0.000 W/(m².K)
  A3 = 0.00 m²

Col. Type      Name                          tmin     Node     tmax     Node
                                             [°C]              [°C]
133  MATERIAL  insulation_0.030_W/mK_        0.21    20266    13.74        9
136  MATERIAL  reinforced_concrete_(stee     0.00    24661    20.00    23817
170  BC_SIMPL  exterior                      0.00    24661     1.07    35016
174  BC_SIMPL  interior_(normal)            18.76     9194    20.00    23817

   Node        X         Y         Z
             [dm]      [dm]      [dm]
  20266    -20.00      0.00     30.00
      9   -100.00   -100.00     20.00
  24661    -15.00     15.00    125.00
  23817    -15.00    -15.00   -120.00
  24661    -15.00     15.00    125.00
  35016      0.00     15.00     30.00
   9194    -45.00      0.00      0.00
  23817    -15.00    -15.00   -120.00

Col. Type      Name                            ta   Flow in  Flow out
                                             [°C]       [W]       [W]
170  BC_SIMPL  exterior                                0.41   2688.96
174  BC_SIMPL  interior_(normal)                    2688.54      0.00
```

E.2. File commands

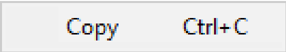
File submenu



- *File → Save As...*
  Save the whole text in the current **Text Output window** into a Microsoft Word document (extension .doc) on disk.
- *File → Save to CSV Files*
  Save the text output to CSV files for use in a spreadsheet program (e.g. Microsoft Excel).
- *File → Exit*
  Return to the main SOLIDO window.

E.3. Edit command

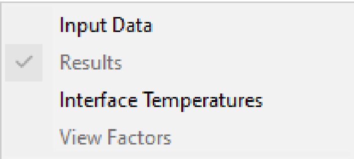
Edit submenu



Edit → Copy
Copy the selected text (selected using the mouse or the arrow keys with the <Shift> key held down) into the clipboard (so that it can be pasted in an external text editor).

E.4. View options

View submenu



E.4.1. Input data

Menu command: *View → Input Data*

Output

- Document notes (see B.19)
- SOLIDO data file name
- Grid data: grid mesh widths along X,Y and Z
- Blocks: colour, type, shape, triangulation mesh size, raster, block point coordinates, STL file names
- Colours: all present colours with colour type and thermal properties
- Borderface boundary conditions
- Material boundary conditions
- Calculation parameters

Example

SOLIDO - Input Data

```
SOLIDO data file: Column1.sld

GRID
No.                X          Y          Z
                 [dm]       [dm]       [dm]
0-1               5.00       5.00       5.00
1-2               5.00       5.00       5.00
2-3               5.00       5.00       5.00
3-4               5.00       5.00       5.00
4-5               5.00       5.00       5.00
5-6               5.00       5.00       5.00
6-7               5.00       5.00       5.00
7-8               5.00       5.00       5.00
8-9               5.00       5.00       5.00
9-10              5.00       5.00       5.00
10-11             5.00       5.00       5.00
11-12             2.50       2.50       5.00
12-13             2.50       2.50       5.00
13-14             2.50       2.50       5.00
14-15             2.50       2.50       5.00
15-16             2.50       2.50       5.00
16-17             2.50       2.50       2.50
17-18             2.50       2.50       2.50
18-19             2.50       2.50       2.50
19-20             2.50       2.50       2.50
```

20-21	2.50	2.50	2.50
21-22	2.50	2.50	2.50
22-23	1.25	1.25	2.50
23-24	1.25	1.25	2.50
24-25	2.50	2.50	2.50
25-26	2.50	2.50	2.50
26-27	2.50	2.50	2.50
27-28	2.50	2.50	2.50
28-29	2.50	2.50	2.50
29-30	2.50	2.50	2.50
30-31	2.50	2.50	2.50
31-32	2.50	2.50	2.50
32-33	2.50	2.50	2.50
33-34	2.50	2.50	2.50
34-35	2.50	2.50	2.50
35-36	2.50	2.50	2.50
36-37	1.25	1.25	2.50
37-38	1.25	1.25	2.50
38-39	2.50	2.50	2.50
39-40	2.50	2.50	2.50
40-41	2.50	2.50	2.50
41-42	2.50	2.50	2.50
42-43	2.50	2.50	2.50
43-44	2.50	2.50	2.50
44-45	2.50	2.50	2.50
45-46	2.50	2.50	2.50
46-47	2.50	2.50	5.00
47-48	2.50	2.50	5.00
48-49	2.50	2.50	5.00
49-50	5.00	5.00	5.00
50-51	5.00	5.00	5.00
51-52	5.00	5.00	5.00
52-53	5.00	5.00	5.00
53-54	5.00	5.00	5.00
54-55	5.00	5.00	5.00
55-56	5.00	5.00	5.00
56-57	5.00	5.00	5.00
57-58	5.00	5.00	5.00
58-59	5.00	5.00	5.00
59-60	5.00	5.00	5.00
60-61			5.00
61-62			5.00
62-63			5.00
63-64			5.00
64-65			5.00
Sum	200.00	200.00	250.00

BLOCKS

No.	Col.	Type	Shape	Mesh	Raster	Point [dm]	X [dm]	Y [dm]	Z [dm]
1	170	SOLID	RECT	100.00	NORMAL	Black	-100.00	-100.00	30.00
						Grey	100.00	100.00	125.00
2	174	SOLID	RECT	100.00	NORMAL	Black	-100.00	-100.00	0.00
						Grey	100.00	100.00	-125.00
3	136	SOLID	RECT	100.00	NORMAL	Black	-100.00	-100.00	0.00
						Grey	100.00	100.00	20.00
4	133	SOLID	RECT	100.00	NORMAL	Black	-100.00	-100.00	20.00
						Grey	100.00	100.00	30.00
5	136	SOLID	RECT	100.00	NORMAL	Black	-15.00	-15.00	-120.00
						Grey	15.00	15.00	125.00
6	136	SOLID	RECT	100.00	NORMAL	Black	-45.00	-45.00	0.00
						Grey	45.00	45.00	-10.00
7	136	SOLID	RECT	100.00	NORMAL	Black	-45.00	-45.00	-10.00
						Red	-45.00	45.00	-10.00
						Green	45.00	-45.00	-10.00
						Blue	-15.00	-15.00	-40.00
						Yellow	45.00	45.00	-10.00
						Magenta	-15.00	15.00	-40.00
						Cyan	15.00	-15.00	-40.00
						Grey	15.00	15.00	-40.00

COLOURS

Col.	Type	Subtype	Phys. flow	Geom. flow	Name	eps1 / eps2 [- / -]
133	MATERIAL				insulation_0.030_W/mK_	
136	MATERIAL				reinforced_concrete_(stee	
170	BC_SIMPL	NIHIL			exterior	
174	BC_SIMPL	NIHIL			interior_(normal)	

Col.	lambda	eps	t	h	q	ta	hc	Pc	tr	Standard

	[W/mK]	[-]	[°C]	[W/m^2K]	[W/m^2]	[°C]	[W/m^2K]	[W]	[°C]	
133	0.500									
136	2.500									
170			0.0	25.00	0					NIHIL
174			20.0	7.70	0					NIHIL

Calculation parameters

```
Maximum number of iterations = 10000
Maximum temperature difference = 0.0001°C
Maximum heat flow divergence for total object = 0.001 %
Maximum heat flow divergence for worst node = 1 %
```

E.4.2. Calculation results

Menu command: *View → Results*

Output

- SOLIDO data file name
- Number of system nodes
- Iteration quality numbers for last executed iteration cycle:
  heat flow divergence for total object and for worst node
- List of minimum and maximum temperatures (and corresponding coordinates) per colour
- List of total heat flows (in and out of object) per surface boundary condition.
  Calculated air temperatures for surface boundary conditions of type BC_FRE_S and BC_FREE
- Derived thermal properties (see C.7)

Example

```
SOLIDO - Calculation Results

SOLIDO data file: Column1.sld

Number of nodes = 69225
Heat flow divergence for total object = 0.000116671 %
Heat flow divergence for worst node = 0.36295 %

Temperature factor (EN ISO 10211-2) = 0.938
  hi = 7.70 W/(m².K)
  Rsi = 0.13 m².K/W

2D/3D surplus thermal transmittance
dL = Q/(ti-te) - U1*A1 - U2*A2 - U3*A3 = -0.258 W/K
Equivalent thermal transmittance
Ueq = Q/((ti-te)*(A1+A2+A3)) = 0.336 W/(m².K)
  Q = 2688.552 W
  ti = 20.00°C
  te = 0.00°C
  U1 = 0.337 W/(m².K)
       X1=-50.00 Y1=-50.00 Z1=-125.00
       X2=-50.00 Y2=-50.00 Z2=125.00
  A1 = 400.00 m²
       X1=-100.00 Y1=-100.00 Z1=0.00
       X2=-100.00 Y2=100.00 Z2=0.00
       X3=100.00 Y3=100.00 Z3=0.00
  U2 = 0.234 W/(m².K)
       X1=-22.50 Y1=-22.50 Z1=-125.00
       X2=-22.50 Y2=-22.50 Z2=125.00
  A2 = 0.00 m²
  U3 = 0.000 W/(m².K)
  A3 = 0.00 m²
```

Col.	Type	Name	tmin [°C]	Node	tmax [°C]	Node
133	MATERIAL	insulation_0.030_W/mK_	0.21	20266	13.74	9
136	MATERIAL	reinforced_concrete_(stee	0.00	24661	20.00	23817
170	BC_SIMPL	exterior	0.00	24661	1.07	35016
174	BC_SIMPL	interior_(normal)	18.76	9194	20.00	23817

Node	X [dm]	Y [dm]	Z [dm]
20266	-20.00	0.00	30.00
9	-100.00	-100.00	20.00
24661	-15.00	15.00	125.00
23817	-15.00	-15.00	-120.00
24661	-15.00	15.00	125.00
35016	0.00	15.00	30.00
9194	-45.00	0.00	0.00
23817	-15.00	-15.00	-120.00

Col.	Type	Name	ta [°C]	Flow in [W]	Flow out [W]
170	BC_SIMPL	exterior		0.41	2688.96
174	BC_SIMPL	interior_(normal)		2688.54	0.00

<u>Remark on listed heat flows</u>

The incoming and outgoing heat flows per surface boundary condition colour, as listed in the calculation results, equal the sums of elementary incoming and outgoing heat flows for all control volumes around the system nodes touching that surface boundary condition colour. This is done this way because the system solution is also based on energy balances containing these heat flows through the control volumes.

Sometimes a system node belongs to more than one surface boundary condition colour. This happens when two or more surface boundary condition colours touch each other. SOLIDO requires that those surface boundary condition colours must have the same type (BC_SIMPL, BC_FRE_S, BC_FREE, BC_SKY, BC_NOSKY, TRANSMAT), otherwise an error message appears when starting the calculation.
In that case the control volume of such a system node is attributed to one of the adjacent surface boundary condition colours.
This implies that some of the total heat flows can be underestimated and other overestimated. However the sums of the incoming and outgoing heat flows of all adjacent surface boundary condition colours are always correct. The underestimation or overestimation error can be decreased by applying smaller grid mesh widths at the contact nodes of the different surface boundary condition colours.

E.4.3. Interface temperatures

Menu command: *View → Interface Temperatures*

Shows a list of temperature values (minimum temperature, maximum temperature, area weighted average temperature) in the interfaces between the different colours.

<u>Example</u>

SOLIDO - Interface Temperatures

SOLIDO data file: Pipe.sld

Col.1	Col.2	tmin [°C]	tmax [°C]	tmean [°C]
2	5	15.01	16.39	15.35
2	7	0.46	16.39	2.49
4	5	15.10	17.26	15.83
5	7	16.36	17.22	16.95

E.4.4. View factors

Menu command: *View → View Factors*

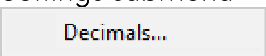
Shows global view factors between each pair of surfaces of different colours.

<u>Example</u>

```
SOLIDO - View Factors

SOLIDO data file: ViewObst2.sld

From     To      View
col.     col.    factor
  2       3      0.1242
  2       4      0.1139
  2       7      0.1951
  3       2      0.1662
  3       3      0.0000
  3       4      0.1662
  3       7      0.1662
  4       2      0.1139
  4       3      0.1242
  4       7      0.1928
  7       2      0.1951
  7       3      0.1242
  7       4      0.1928
```

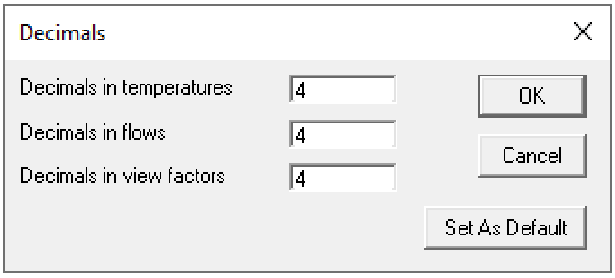
E.5. Settings

Settings submenu



E.5.1. Output precision

Menu command: *Settings → Decimals...*



CHAPTER F SOLIDO TEMPLATE FILES

F.1. General

SOLIDO makes use of template files containing default information for a number of functions. These template files are saved in the application data folder (see A.7) of the user account on the computer where SOLIDO is installed. The path to the application data folder is C:/Users/*username*/Appdata/Roaming/Physibel/SOLIDO5 (AppData is a hidden folder by default in Windows).

The template files used by SOLIDO are:

- ColourDatabase.sld (F.2)

The use of these template files in different SOLIDO functions has been outlined throughout the manual. In the following sections, handling and customization of these template files is briefly discussed.

F.2. Colour database

The colour database file <u>ColourDatabase.sld</u> is saved in the application data folder (C:/Users/*username*/Appdata/Roaming/Physibel/SOLIDO5). The colour database contains predefined material and boundary condition properties and is loaded at every SOLIDO start-up (see B.1).

It is advisable to add frequently used materials and boundary conditions to the colour database. These predefined colours of the colour database are then available for new colours in any document.
The command *File → Open Colour Database...* (see B.2.3) allows to modify the 256 available predefined colours of the colour database. After editing the colour data, the file should be saved (*File → Save*) to adapt the colour database on disk.

Alternatively, if a user wants to save a custom file as colour database, this is possible by saving it with the name ColourDatabase.sld in the application data folder (*File → Save as...*) thus overwriting the default file. In the application data folder, the default colour database is present as ColourDatabase_PhysibelDefault.sld. This file can be saved as ColourDatabase.sld to restore the original colour database.

The ColourDatabase.sld file replaces the Solido.sld file in older versions of SOLIDO. Users who had been working with a customised Solido.sld file, can thus easily transfer to the new system by opening Solido.sld and saving it with the name ColourDatabase.sld in the application data folder.

CHAPTER G RADCON MODULE

G.1. Purpose

The RADCON module for SOLIDO is a program add-on module to simulate more realistically the heat transfer between material surfaces and their adjacent environments.
The heat transfer has the following components:

- Radiation: based on view factors and material surface emissivities.
- Convection: based on a convective heat transfer coefficient applied to the contact surface between material and environment.
  For natural convection the convective heat transfer coefficient can be calculated in function of the maximum temperature difference between surface temperature and air temperature.
- Conduction: in a (transparent) material with a fixed thermal conductivity coefficient.

The RADCON module offers a physically more correct alternative for:

- the global surface heat transfer coefficient.
- the equivalent thermal conductivity of an air cavity.

G.2. Implementation

Without the RADCON module, SOLIDO has the following colour types (defined in the **Colours window**, see B.8 and C.1 for more information):

- MATERIAL
- EQUIMAT
- BC_SIMPL
- BC_FRE_S

With the RADCON module, SOLIDO has the following additional colour types:

- BC_FREE for an enclosure (normally fully surrounded by material surfaces, e.g. a cavity or a room) with unknown air temperature (free-floating air temperature), known surface heat flux (positive or negative), known (or calculated) convective heat transfer coefficient and known injected convective heat power (positive or negative).
  The unknown air temperature is calculated from the convective heat balance equation:
  Σ (convective heat transfer from environment to surface) = injected convective heat power.
  The radiation between the different surfaces bordering the environment are calculated based on view factors. When the environment is not completely surrounded by material surfaces, the open end is considered to be a symmetry plane for heat transfer. The symmetry plane is simplified by use of an additional *radiation mirror node*. In that way the environment is made closed for radiation heat transfer: i.e. for each elementary surface the sum of view factors equals 1 (taking into account the radiation mirror node for the rest view factor). The radiation heat transfer is somewhat underestimated compared to the real cavity (with double volume of the environment bounded by the symmetry plane).
  Therefore it is recommended to use fully enclosed environments.
- BC_SKY for an environment (normally not fully surrounded by material surfaces, e.g. in contact to the sky) with known air temperature, known radiation temperature, known surface heat flux (positive or negative) and known (or calculated) convective heat transfer coefficient.
  This implies there can be a resultant radiation heat flow from the environment to the material surfaces (using view factors) and a resultant convective heat flow from the environment to the material surfaces.

- **BC_NOSKY** for an environment with known air temperature, known surface heat flux (positive or negative) and known (or calculated) convective heat transfer coefficient.
  There is no radiation to the environment (i.e. no sky radiation). The surfaces bordering the environment exchange radiation based on view factors. In general the sum of view factors from an elementary surface to the other surfaces is less than 1.
  The known air temperature implies there can be a resultant convective heat flow from the environment to the material surfaces.
- **TRANSMAT** for a transparent material with known (or calculated) thermal conductivity (normally of still air or another gas).
  Additionally to the material conduction there is radiative heat transfer between the adjacent material surfaces (using the emissivity of the adjacent materials).
  If the environment (normally a small cavity) modelled as transparent material is not fully surrounded by material surfaces at the outer border of the geometry, then this border is considered as an adiabatic surface with emissivity 0.

If there is convection within the cavity, the thermal conductivity for still gas should be multiplied by the Nusselt number. This is done automatically for subtype CAVITY (see C.1.5).

With RADCON the colour types MATERIAL and EQUIMAT also have a user defined surface emissivity, which is applied to the radiative heat transfer in all RADCON boundary conditions bordering the material.

G.3. Formulas

Combined convective and radiative heat transfer (BC_SIMPL, BC_FRE_S)
The global (combined convective and radiative) heat flow is:
$Q = h \cdot (\theta_s - \theta_{bc}) \cdot A$
Q = global heat flow [W]
h = global heat transfer coefficient [W/(m^2.K)]
θ_s = surface temperature [°C]
θ_{bc} = ambient temperature [°C]
A = surface area [m^2]
In this formula the ambient temperature is not defined precisely. It can be the air temperature, the dry resultant temperature or an interpolated value. Depending on that choice, an appropriate value for the global heat transfer coefficient must be chosen.

Convective heat transfer (BC_FREE, BC_SKY, BC_NOSKY)
The convective heat flow is:
$Q_c = h_c \cdot (\theta_s - \theta_a) \cdot A$
Q_c = convective heat flow [W]
h_c = convective heat transfer coefficient [W/(m^2.K)]
θ_s = surface temperature [°C]
θ_a = air temperature [°C]
A = surface area [m^2]
The convective heat transfer coefficient h_c can be calculated based on the given subtype (cf. section **Error! Reference source not found.** and **Error! Reference source not found.**).

Black radiation (BC_FREE, BC_SKY, BC_NOSKY, TRANSMAT)
The radiative heat flow between two black surfaces is:
$Q_{ij} = A_i \cdot F_{ij} \cdot h_{rb} \cdot (\theta_{sbi} - \theta_{sbj})$
Q_{ij} = radiative heat flow between black surface i and black surface j [W]
A_i = area of surface i [m^2]
F_{ij} = view factor from surface i to surface j (between 0 and 1)
h_{rb} = black radiation heat transfer coefficient [W/(m^2.K)]

θ_{sbi} = black surface temperature of surface i [°C]
θ_{sbj} = black surface temperature of surface j [°C]

For linear radiation the black radiation heat transfer coefficient is constant, defined as a calculation parameter (see C.2). The value depends on the temperature range of the problem considered. Section G.4 contains a table with the values of h_{rb}, depending on the mean temperature θ_m [°C] of the surfaces i and j and the temperature difference between the surfaces $\Delta\theta_{ij}$ [°C]. The table shows that the h_{rb}-value mainly depends on the mean surface temperature θ_m . The temperature difference between the surfaces has a minor influence.
The default value is a black radiation heat transfer coefficient h_{rb} = 5.25 W/(m^2.K) (assuming θ_m = 12°C), but the user can change this value in the Calculation Parameters dialog box.

For non-linear radiation:
$h_{rb} = \sigma \cdot (T_{sbi}^2 + T_{sbj}^2) \cdot (T_{sbi} + T_{sbj})$
σ = 5.67 10^{-8} W(m^2.K^4) (Stefan-Boltzmann constant)
T_{sbi} = θ_{sbi} + 273.16 [K] (absolute temperature of black surface i)
T_{sbj} = θ_{sbj} + 273.16 [K] (absolute temperature of black surface j)
The black radiation heat transfer coefficients are recalculated at the start of each new iteration cycle based on the temperatures obtained in the last iteration cycle.

Grey radiation (BC_FREE, BC_SKY, BC_NOSKY, TRANSMAT)
Real surfaces are not black (emissivity ε = 1) but grey (emissivity ε < 1). In the RADCON module each (grey) surface is connected to a black surface node. All black surface nodes are connected to each other in a star-diagram.
The radiative heat flow between the real surface with given emissivity and the corresponding black surface is:
$Q_r = A \cdot \frac{\varepsilon}{1-\varepsilon} \cdot h_{rb} \cdot (\theta_{sb} - \theta_s)$
Q_r = radiative heat flow between grey surface and black surface [W]
A = surface area [m^2]
ε = surface emissivity (0 ≤ ε < 1)
h_{rb} = black radiation heat transfer coefficient [W/(m^2.K)]
θ_{sb} = black surface temperature [°C]
θ_s = (real) surface temperature [°C]
For non-linear radiation:
$h_{rb} = \sigma \cdot (T_{sb}^2 + T_s^2) \cdot (T_{sb} + T_s)$
T_{sb} = θ_{sb} + 273.16 [K]
T_s = θ_s + 273.16 [K]
The black radiation heat transfer coefficients are recalculated at the start of each new iteration cycle based on the temperatures obtained in the last iteration cycle.

G.4. Table of black radiation heat transfer coefficients

The following table shows the black radiation heat transfer coefficient.
The values are obtained using the formula:

$$h_{rb} = \sigma \cdot (T_i^2 + T_j^2) \cdot (T_i + T_j)$$

with σ = 5.67*10^{-8} W/(m^2.K^4) (Stefan-Boltzmann constant)

black radiation heat transfer coeff. h_{rb} [W/m^{2}K]		temperature difference between the surfaces i and j $\Delta\theta_{ij}$ [°C]										
		0.00	5.00	10.00	15.00	20.00	25.00	30.00	35.00	40.00	45.00	50.00
mean temp. θ_m [°C]	-20.00	3.68	3.68	3.68	3.68	3.69	3.69	3.69	3.70	3.70	3.71	3.72
	-15.00	3.90	3.90	3.90	3.91	3.91	3.91	3.92	3.92	3.93	3.93	3.94
	-10.00	4.13	4.13	4.13	4.14	4.14	4.14	4.15	4.15	4.16	4.16	4.17
	-5.00	4.37	4.37	4.37	4.38	4.38	4.38	4.39	4.39	4.40	4.40	4.41
	0.00	4.62	4.62	4.62	4.63	4.63	4.63	4.64	4.64	4.65	4.65	4.66
	5.00	4.88	4.88	4.88	4.88	4.89	4.89	4.90	4.90	4.91	4.91	4.92
	10.00	5.15	5.15	5.15	5.15	5.16	5.16	5.16	5.17	5.17	5.18	5.19
	15.00	5.43	5.43	5.43	5.43	5.43	5.44	5.44	5.45	5.45	5.46	5.47
	20.00	5.71	5.71	5.72	5.72	5.72	5.72	5.73	5.73	5.74	5.75	5.76
	25.00	6.01	6.01	6.01	6.02	6.02	6.02	6.03	6.03	6.04	6.05	6.05
	30.00	6.32	6.32	6.32	6.32	6.33	6.33	6.33	6.34	6.35	6.35	6.36
	35.00	6.64	6.64	6.64	6.64	6.64	6.65	6.65	6.66	6.66	6.67	6.68
	40.00	6.96	6.97	6.97	6.97	6.97	6.98	6.98	6.99	6.99	7.00	7.01
	45.00	7.30	7.30	7.31	7.31	7.31	7.32	7.32	7.33	7.33	7.34	7.35
	50.00	7.65	7.65	7.66	7.66	7.66	7.67	7.67	7.68	7.68	7.69	7.70

G.5. Table of emissivities

The table below contains values of the long-wave emissivity ε of some building materials taken from:

- H.C. Hottel, Heat Transmission, 1954
- J.S. Cammerer, Wärme- und Kälteschutz in der Industrie, 1962
- Stichting Bouwresearch, Eigenschappen van bouw- en isolatiematerialen, rapport 9, 1974

building material	long-wave emissivity at T = 300K ε [-]
black surface	1
aluminium, polished	< 0.07
aluminium, oxidised	0.11 à 0.3
brass, polished	0.038
copper, smoothed	0.09
copper, oxidised	0.78 - 0.79
cast iron	0.44
iron, oxidised and rusted red	0.61
lead, grey oxidised	0.28
tin	0.043 - 0.064
zinc, galvanised	0.23
zinc, oxidised	> 0.11
asbestos board	0.96
bitumen	0.92
brick, red, rough but no gross irregularities	0.92 - 0.93
concrete	0.88
glass	0.94
cement-tiles	> 0.85
roofing-tiles, red	0.85
roofing paper	0.91
rubber, hard and glossy	0.94
wood	0.86
aluminium painting	0.42
enamel, white	>0.90
metal paint, black	0.88
water	0.95 - 0.96
other materials (approximation) white, yellow, green, red, blue, grey, black	0.85 à 0.95

G.6. Tables for convective heat transfer coefficients

The convective heat flow density is calculated using the following formula:

$$q_c = h_c \cdot (\theta_a - \theta_s)$$

with q_c = convective heat flow density [W/m²]
h_c = convective heat transfer coefficient [W/(m².K)]
θ_a = air temperature [°C]
θ_s = surface temperature [°C]

Table 1 lists values of h_c for air cavities according to EN ISO 10077-2, depending on the temperature difference $\Delta\theta$ over the cavity and the depth d of the cavity. Above the bold line, the h_c-value depends only on the cavity depth (same h_c-values on each row), while underneath the bold line the h_c-value depends only on the temperature difference over the cavity (same h_c-values in each column). Above the bold line there is in fact only conduction (still air). Below the line natural convection is considered.

Table 1: convective heat transfer coefficient for air cavities (EN ISO 10077-2)

convective heat transfer coefficient h_c [W/(m².K)]		temperature difference over the cavity $\Delta\theta$ [°C]									
		0.500	1.000	1.500	2.000	3.000	4.000	5.000	6.000	8.000	10.000
depth of the cavity d [m]	0.001	50.000	50.000	50.000	50.000	50.000	50.000	50.000	50.000	50.000	50.000
	0.002	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000	25.000
	0.003	16.667	16.667	16.667	16.667	16.667	16.667	16.667	16.667	16.667	16.667
	0.004	12.500	12.500	12.500	12.500	12.500	12.500	12.500	12.500	12.500	12.500
	0.005	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000	10.000
	0.006	8.333	8.333	8.333	8.333	8.333	8.333	8.333	8.333	8.333	8.333
	0.007	7.143	7.143	7.143	7.143	7.143	7.143	7.143	7.143	7.143	7.143
	0.008	6.250	6.250	6.250	6.250	6.250	6.250	6.250	6.250	6.250	6.250
	0.009	5.556	5.556	5.556	5.556	5.556	5.556	5.556	5.556	5.556	5.556
	0.010	5.000	5.000	5.000	5.000	5.000	5.000	5.000	5.000	5.000	5.000
	0.011	4.545	4.545	4.545	4.545	4.545	4.545	4.545	4.545	4.545	4.545
	0.012	4.167	4.167	4.167	4.167	4.167	4.167	4.167	4.167	4.167	4.167
	0.013	3.846	3.846	3.846	3.846	3.846	3.846	3.846	3.846	3.846	3.846
	0.014	3.571	3.571	3.571	3.571	3.571	3.571	3.571	3.571	3.571	3.571
	0.015	3.333	3.333	3.333	3.333	3.333	3.333	3.333	3.333	3.333	3.333
	0.016	3.125	3.125	3.125	3.125	3.125	3.125	3.125	3.125	3.125	3.145
	0.017	2.941	2.941	2.941	2.941	2.941	2.941	2.941	2.941	2.941	3.145
	0.018	2.778	2.778	2.778	2.778	2.778	2.778	2.778	2.778	2.920	3.145
	0.019	2.632	2.632	2.632	2.632	2.632	2.632	2.632	2.653	2.920	3.145
	0.020	2.500	2.500	2.500	2.500	2.500	2.500	2.500	2.653	2.920	3.145
	0.025	2.000	2.000	2.000	2.000	2.106	2.318	2.497	2.653	2.920	3.145
	0.030	1.667	1.667	1.671	1.839	2.106	2.318	2.497	2.653	2.920	3.145
	0.035	1.429	1.460	1.671	1.839	2.106	2.318	2.497	2.653	2.920	3.145
	0.040	1.250	1.460	1.671	1.839	2.106	2.318	2.497	2.653	2.920	3.145
	0.045	1.159	1.460	1.671	1.839	2.106	2.318	2.497	2.653	2.920	3.145
	0.050	1.159	1.460	1.671	1.839	2.106	2.318	2.497	2.653	2.920	3.145
	0.060	1.159	1.460	1.671	1.839	2.106	2.318	2.497	2.653	2.920	3.145
	0.070	1.159	1.460	1.671	1.839	2.106	2.318	2.497	2.653	2.920	3.145
	0.080	1.159	1.460	1.671	1.839	2.106	2.318	2.497	2.653	2.920	3.145
	0.090	1.159	1.460	1.671	1.839	2.106	2.318	2.497	2.653	2.920	3.145
	0.100	1.159	1.460	1.671	1.839	2.106	2.318	2.497	2.653	2.920	3.145

Table 2 lists values of h_c for exterior surfaces, depending on the air velocity v and the temperature difference $\Delta\theta$ between surface and environment (from "Informatiemap voor bouwfysici", 1991).

Table 2: the convective heat transfer coefficient for exterior environments

convective heat transfer coefficient h_c [W/(m^2.K)]		temperature difference object –environment $\Delta\theta$ [°C]			
		5	10	20	40
air velocity v [m/s]	0.5	7	8	9	11
	1	9	10	11	13
	2	12	13	15	16
	3	19	20	21	23
	4	25	26	27	28
	5	30	31	32	34

The following formulas define the convective heat transfer coefficient for interior environments (from "Element 29: Wärmeschutz und Energie im Hochbau", 1990). Table 3 lists the h_c-values according to these formulas.

$h_c = 1.31 \cdot \sqrt[3]{\Delta\theta}$ for vertical constructions (walls) with a horizontal heat flow

$h_c = 1.77 \cdot \sqrt[4]{\Delta\theta}$ for vertical constructions (window) with a horizontal heat flow

$h_c = 1.52 \cdot \sqrt[3]{\Delta\theta}$ for horizontal constructions (walls) with an upgoing heat flow

$h_c = 0.59 \cdot \sqrt[4]{\Delta\theta}/L$ for horizontal constructions (walls) with a downgoing heat flow

with h_c = convective heat transfer coefficient [W/(m^2.K)]
$\Delta\theta$ = temperature difference object - environment [°C]
L = ½ (length x width) of the construction [m^2]

Table 3: the convective heat transfer coefficient for interior environments

convective heat transfer coefficient h_c [W/(m^2.K)]		vertical constructions horizontal heat flow		horizontal constructions	
		walls	windows	upwards heat flow	downwards heat flow *
temperature difference object - environment $\Delta\theta$ [°C]	1	1.31	1.77	1.52	0.39
	2	1.65	2.10	1.92	0.47
	3	1.89	2.33	2.19	0.52
	4	2.08	2.50	2.41	0.56
	5	2.24	2.65	2.60	0.59
	6	2.38	2.77	2.76	0.62
	7	2.51	2.88	2.91	0.64
	8	2.62	2.98	3.04	0.66
	9	2.72	3.07	3.16	0.68
	10	2.82	3.15	3.27	0.70
	11	2.91	3.22	3.38	0.72
	12	3.00	3.29	3.48	0.73
	13	3.08	3.36	3.57	0.75
	14	3.16	3.42	3.66	0.76
	15	3.23	3.48	3.75	0.78

* using L = 5 m^2