CREATE STL WITH SKETCHUP FOR SOLIDO / TRISCO

1. Introduction

This document describes how to generate .STL files with SketchUp and use them for SOLIDO or TRISCO.

2. STL files

A STL file contains oriented triangles. Each triangle is defined by the coordinates of the three vertices and by the coordinates of the unit normal to the triangle face (which defines the front side or external surface of the triangle).

All triangles in a STL file of a valid 3D object (as required by SOLIDO) form the watertight surface of the object. The triangle unit normals must point outwards against the object.

The STL file can be in text format or in binary format.

In the STL file format there is no scale definition from the coordinate unit to a length unit (e.g. m, mm or cm).

SOLIDO can read STL in text or binary format, but writes STL (after a coordinate transformation) only in binary format.

3. SketchUp STL plugin

SketchUp can not fully handle .STL files by default.

Therefore, install first a Plugin in SketchUp to handle .STL files (e.g. " STL import & Export 2.2.0").

SketchUp is not a solid modeler, but a drawing program in 3D based on faces.

The user should take care that the exported faces are the correct (watertight) surface of a 3D object.

Interesting Plugin tools for SketchUp which may help detecting errors in a STL file (e.g. internal faces, non-watertight faces) are "Solid Inspector" and "CleanUp".

4. Draw object in SketchUp

The faces of each material and boundary condition must be exported to a separate STL file (as required in SOLIDO, with reference to a colour with index between 0 and 255). It is a good practice to put the faces of each material and boundary condition into a separate layer tag (with layer tag name containing the colour index number e.g. 136_concrete).

For each layer tag select all faces and put them into a group (using the command *Edit > Make Group*) to fix the geometry and disconnect it from the faces of other layers (which faces may share the same vertices).

The faces of a STL object may not contain any internal face inside the object (remove internal face for example with the Plugin "Solid Inspector").

All faces of the STL object must have their front side (with white colour) facing outwards of the object. Use *Reverse Faces* from the context menu (click right mouse button) if the back side (with blue colour) appears on the external object surface.

Any back face on an external surface of an object must be reversed.

To display the geometry in its tag colours instead of the default face colours (white front, blue back), enable Color by Tag (*Default Tray > Tags > Color by Tag*).

5. Export STL file

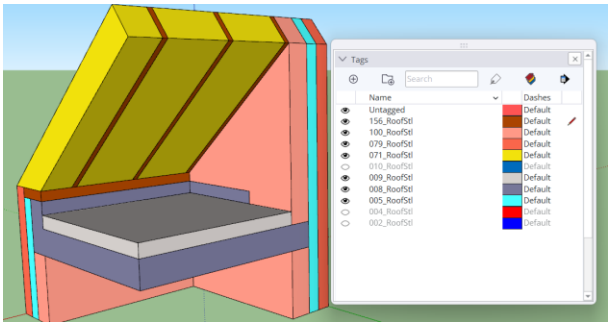
SOLIDO requires a separate STL file per material.

Select separately each material group in SketchUp and export as STL file ("Export STL…" and check "export current selection"). Repeat this for every material group to generate the separate STL files.

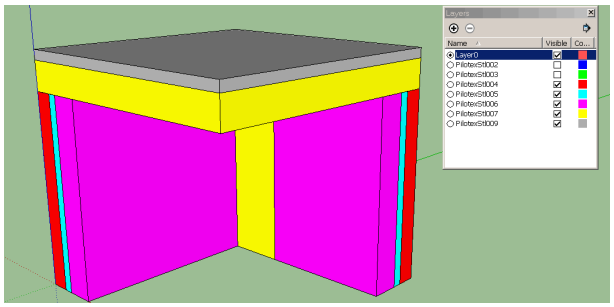
Now the STL files (in text format) are ready for use in SOLIDO.

See the SOLIDO data file RoofStl.sld for an example of object description with STL files, created in SketchUp (RoofStl.skp)

View in SketchUp of RoofStl.skp (coloured by layer).



RoofStl.skp

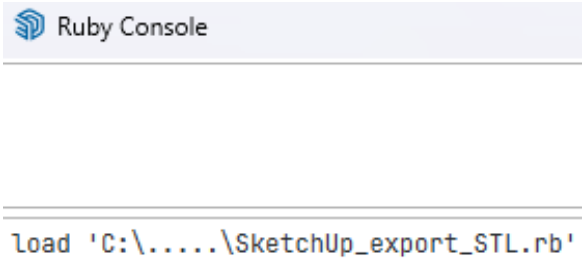


PilotexStl.skp

6. Export STL files in one step

Once the 3D model has been created as described above, all STL files can be exported in one step via the Ruby Console (*Extensions > Developer > Ruby Console*). For this purpose, the script file SketchUp_export_STL.rb is loaded into the Ruby Console.

To do so, SketchUp_export_STL.rb is right-clicked in File Explorer and Copy as path is selected. The copied path is then entered in the Ruby Console after the load command, in one of the forms shown below.



Note that the path is copied in the following form:

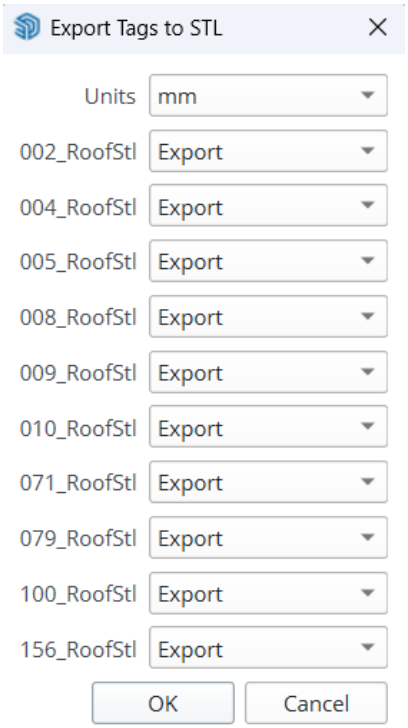
"C:\Users\.....\SketchUp_export_STL.rb"

It must be entered in the Ruby Console in one of the following forms:

load "C:/Users/...../SketchUp_export_STL.rb" or load 'C:\Users\.....\SketchUp_export_STL.rb'

Therefore, the backslashes must be replaced with forward slashes, or the path must be enclosed in single quotes.

The script exports each layer tag in the model as a separate STL file. Tags that should not be exported can be deselected. In addition, the unit of the model must be specified so that the dimensions are correct in SOLIDO. If the model is drawn in mm, mm should be selected, and the grid unit in SOLIDO should also be set to mm when the files are loaded, so that the units are consistent across applications.



After OK is clicked, a destination folder for the STL files must be selected.

7. Create TRISCO data with SketchUp (via STL files loaded into SOLIDO)

SketchUp is very well suited to create a rectangular geometry, as required in TRISCO.

Create STL files as described above, which are loaded in SOLIDO .

Then create a minimum grid (*Grid > Minimum Grid…*) to initialise the Grid window.

Insert all STL coordinates as grid coordinates (*Grid > Insert STL Coordinates…*).

Fill the STL blocks (*Image > Fill Blocks*) and rasterise using the grid coordinates (*Image > Fill Grid*).

Now the rasterised object can be saved as TRISCO file (*File > Save As…* with type: TRISCO data file).

In TRISCO rectangular blocks allow to use a grid corresponding to the block coordinates.

For rectangular blocks the three dimensions (X, Y and Z) have less interaction with one another, and the grid definition is less critical than for arbitrary shapes in SOLIDO (where node fitting interacts more closely with the three dimensions).

View PilotexStl.skp in SketchUp (coloured by layer tag).

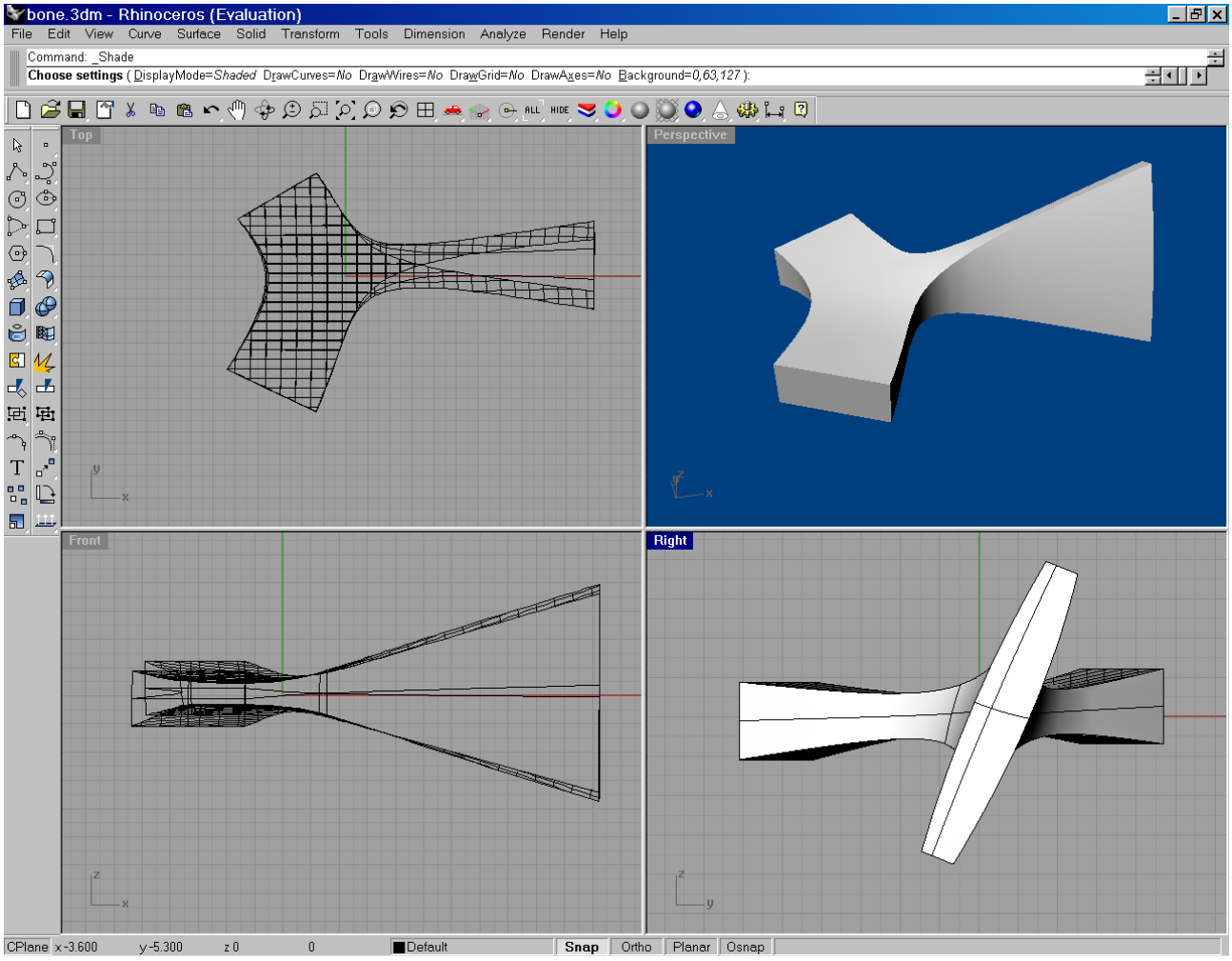
See the SOLIDO(PilotexStl.sld) and TRISCO(PilotexStl.trc) data

CREATE STL WITH THE PROGRAM RHINO FOR SOLIDO / TRISCO

1. Introduction

This short information note shows the principles of how the Physibel program SOLIDO uses solid objects created in a CAD program such a Rhinoceros (Robert McNeel & Associates).

2. Program Rhinoceros

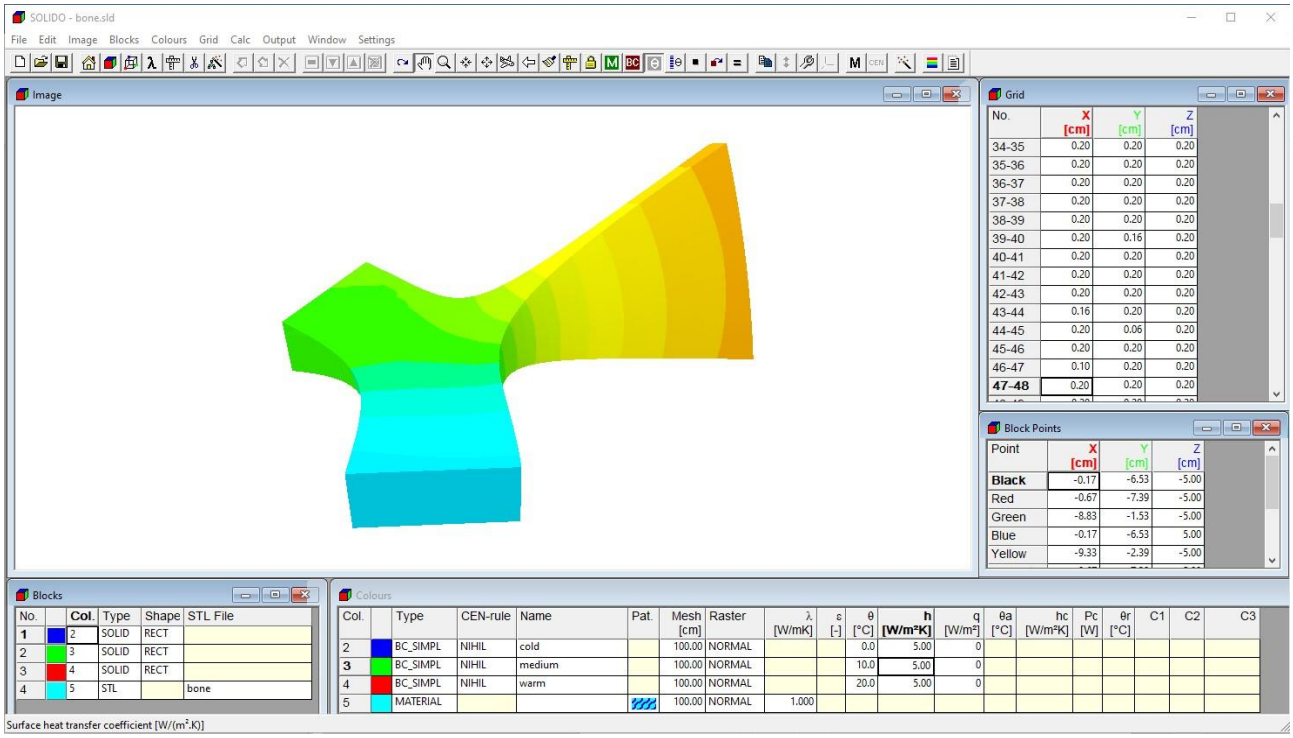


A sample file bone.3dm containing a "solid" was downloaded from http://laminadesign.com/procbo.html and opened in the program Rhinoceros.

Use >File > Save as > Save as type 'Stereolithography *.STL > bone.stl > tolerance OK > STL export option ASCII.

Important: The objects created in the program Rhinoceros have to be of the type solid.

3. Program SOLIDO



In the program SOLIDO, the file bone.stl was imported in the Blocks window.

3 additional blocks were defined for 3 boundary conditions at the edges of the bone-object.

A grid with a maximum grid size of 0.2 was created. The number of nodes amounts to 18 304.

The temperature field is shown in the picture above.

Important:

- At least one STL file is required pro material.
- Blocks (including STL blocks) can be copied and transformed in SOLIDO.
- Thin layers (for example metal sheet) that are not perpendicular to one of the axes (X,Y,Z) have to be avoided. Such layers require a relatively fine grid or a large number of nodes.