Tutorial 08: Slicing, Closing, Cutting and Separation of Solid Geometries

2015-08-06: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de) - Last Change: 2017-07-07

Contents

Complete List of all Tutorials with Publishable MATLAB Files of this Solid-Geoemtries Toolbox

The following topics are covered an explained in the specific tutorials:

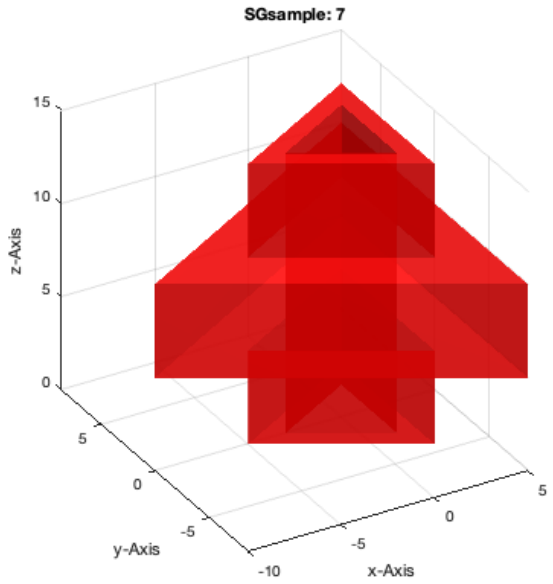
- Tutorial 01: First Steps Using the VLFL-Toolbox for Solid Object Design
- Tutorial 02: Using the VLFL-Toolbox for STL-File Export and Import
- Tutorial 03: Closed 2D Contours and Boolean Operations in 2D
- Tutorial 04: 2½D Design Using Boolean Operators on Closed Polygon Lists (CPL)
- Tutorial 05: Creation, Relative Positioning and Merging of Solid Geometries (SG)
- Tutorial 06: Relative Positioning and Alignment of Solid Geometries (SG)
- Tutorial 07: Rotation of Closed Polygon Lists for Solid Geometry Design
- Tutorial 08: Slicing, Closing, Cutting and Separation of Solid Geometries
- Tutorial 09: Boolean Operations with Solid Geometries
- Tutorial 10: Packaging of Sets of Solid Geometries (SG)
- Tutorial 11: Attaching Coordinates Frames to Create Kinematik Models
- Tutorial 12: Define Robot Kinematics and Detect Collisions
- Tutorial 13: Mounting Faces and Conversion of Blocks into Leightweight-structures
- Tutorial 14: Manipulation Functions for Closed Polygons and Laser Cutting (SVG)
- Tutorial 15: Create a Solid by 2 Closed Polygons
- Tutorial 16: Create Tube-Style Solids by Succeeding Polygons
- Tutorial 17: Filling and Bending of Polygons and Solids
- Tutorial 18: Analyzing and modifying STL files from CSG modeler (Catia)
- Tutorial 19: Creating drawing templates and dimensioning from polygon lines
- Tutorial 20: Programmatically Interface to SimMechanics Multi-Body Toolbox
- Tutorial 21: Programmatically Convert Joints into Drives (SimMechanics)
- Tutorial 22: Adding Simulink Signals to Record Frame Movements
- Tutorial 23: Automatic Creation of a Missing Link and 3D Print of a Complete Model
- Tutorial 24: Automatic Creation of a Joint Limitations
- Tutorial 25: Automatic Creation of Video Titels, Endtitels and Textpages
- Tutorial 26: Create Mechanisms using Universal Planar Links
- Tutorial 27: Fourbar-Linkage: 2 Pose Syntheses and Linkage Export for 3D Printing
- Tutorial 28: Fourbar-Linkage: 3 Pose Syntheses and Linkage Export for 3D Printing
- Tutorial 29: Create a multi body simulation using several mass points
- Tutorial 30: Creating graphical drawings using point, lines, surfaces, frames etc.
- Tutorial 31: Importing 3D Medical DICOM Image Data and converting into 3D Solids
- Tutorial 32: Exchanging Data with a FileMaker Database
- Tutorial 33: Using a Round-Robin realtime multi-tasking system
- Tutorial 34: 2D Projection Images and Camera Coordinate System Reconstruction
- Tutorial 35: Collection of Ideas for Tutorials
- Tutorial 36: Creating a Patient-Individual Arm-Skin Protector-Shell

Motivation for this tutorial: (Originally SolidGeometry 2.4 required)

2. Create a sample solid for this exercise

Using the function SGsample it is possible to create samples for an experiment, to see all of them or to select one.

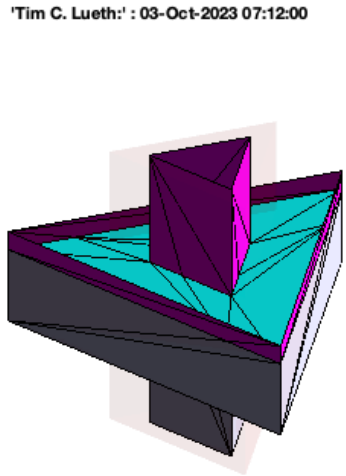
```
close all
SGsample(7);
A=SGsample(7);
```



3. Analyze a slice plane through a solid geoemtry

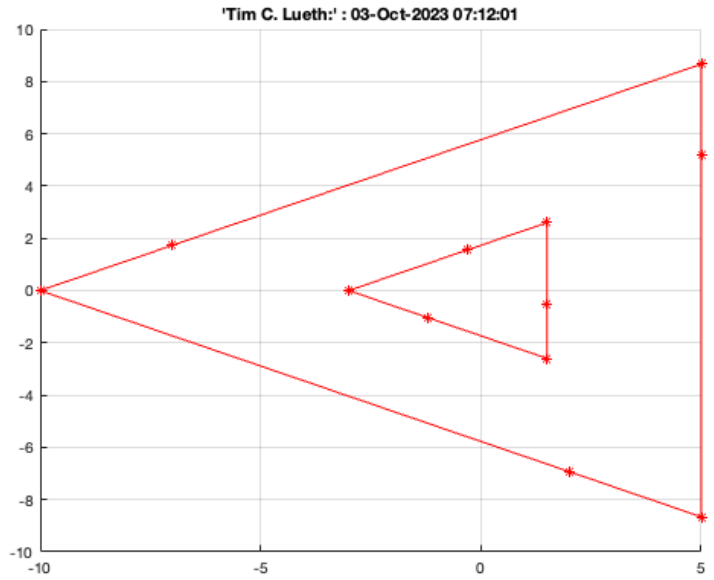
Slicing at a specified z-coordinates is a more complex procedure than expected if several solids are processed that can penetrate each other. By slicing a single solid, the crossed facets/triangles are separated into 2 upper and lower parts that will lead to 2 lower and 1 upper facets or 1 lower and 2 upper facets depending on how many edges are above or under the cutting plane. For slicing we use the function **SGslicer**. Be aware that it is not possible to slice surfaces without crossing edges (i.e. surfaces in the z_max or z_min plane)

```
SGslicer (A,9);
view (10,30);
```



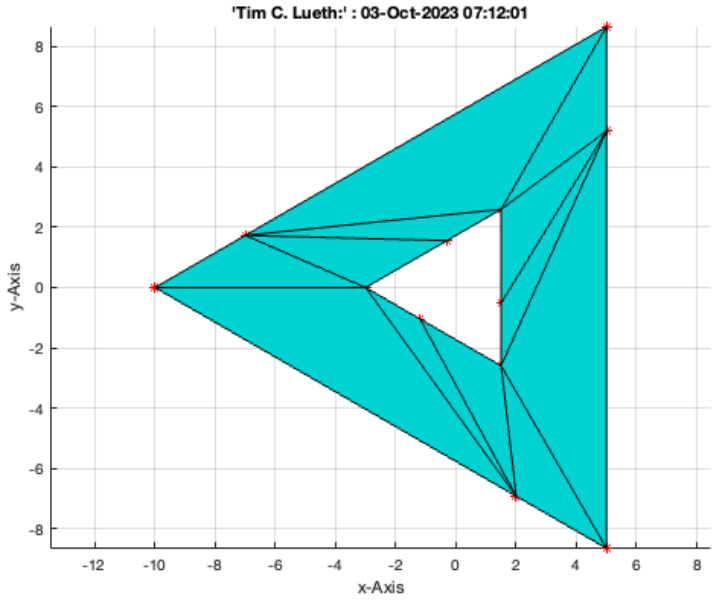
It is also possible just to show the cutting edges of the cutting contour

```
VLFLfigure;
TR2=SGslicer (A,9);
VLELplots(TR2.Points, TR2.Constraints);
```



The result of the slicing process is a delaunay triangulation of the cutting plane. It can be used as cover for closing the cutted solids.

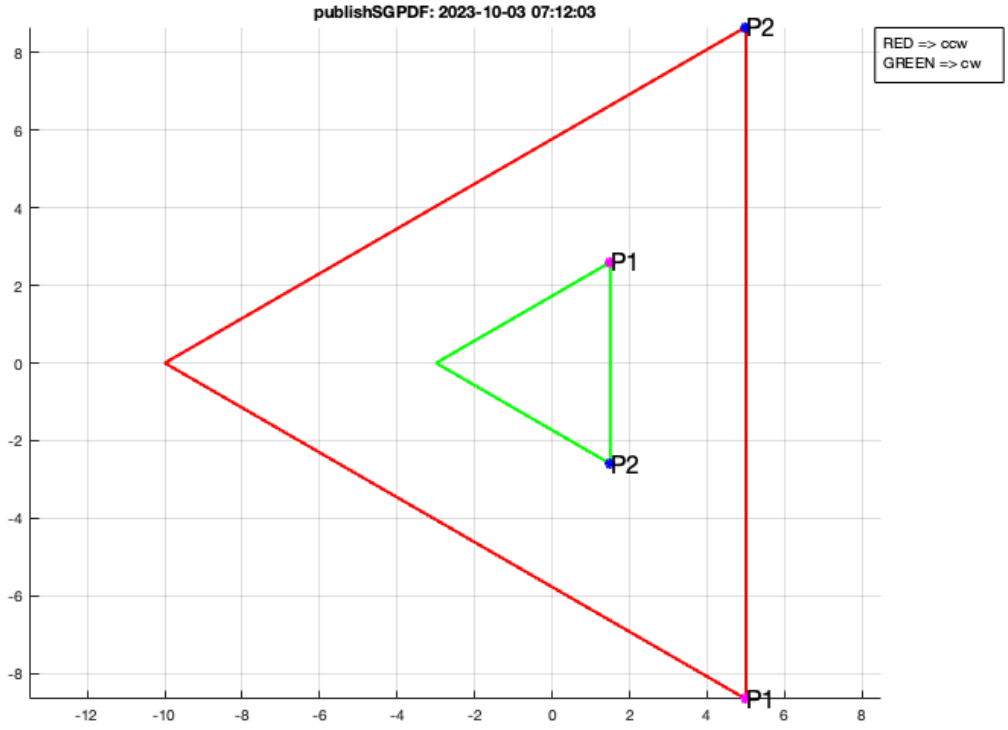
```
in=isInterior(TR2);
VLFLplots(TR2.Points, TR2.ConnectivityList(in,:),'c');
```



Often we want directly getting a closed contour of a slice.

```
CPLofSGslice(A,9); [CPL,warn]=CPLofSGslice(A,9); warn
```

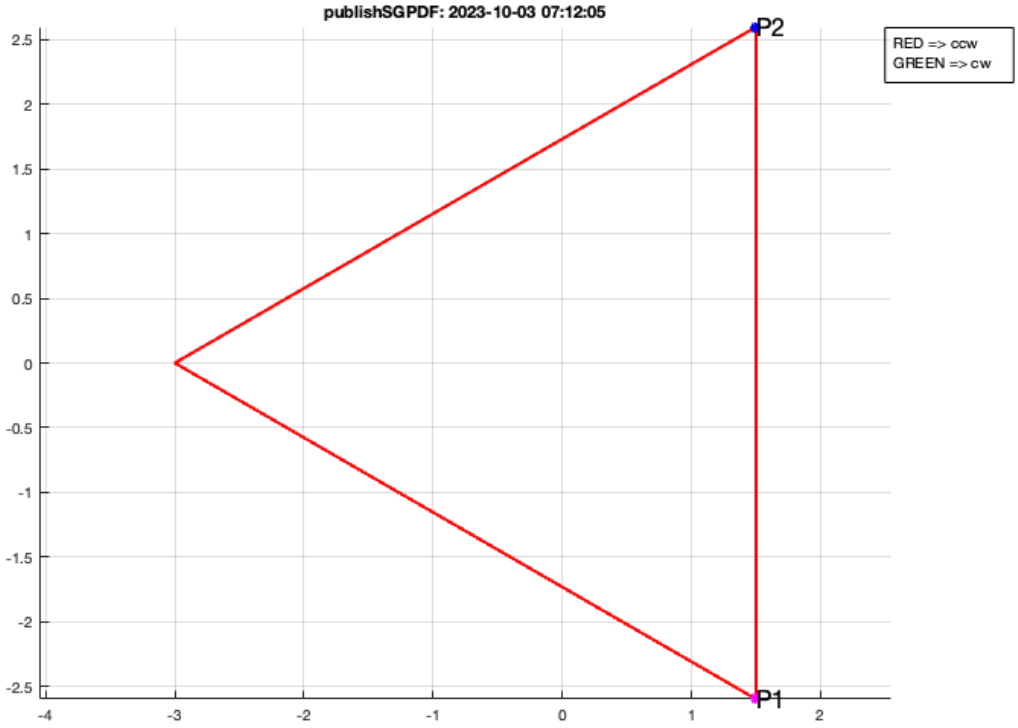
```
warn =
  logical
   0
```



The output parameter warns if a ambiguous cutting result exists

```
CPLofSGslice(A,10); [CPL,warn]=CPLofSGslice(A,10); warn
```

```
warn =
  logical
   0
```

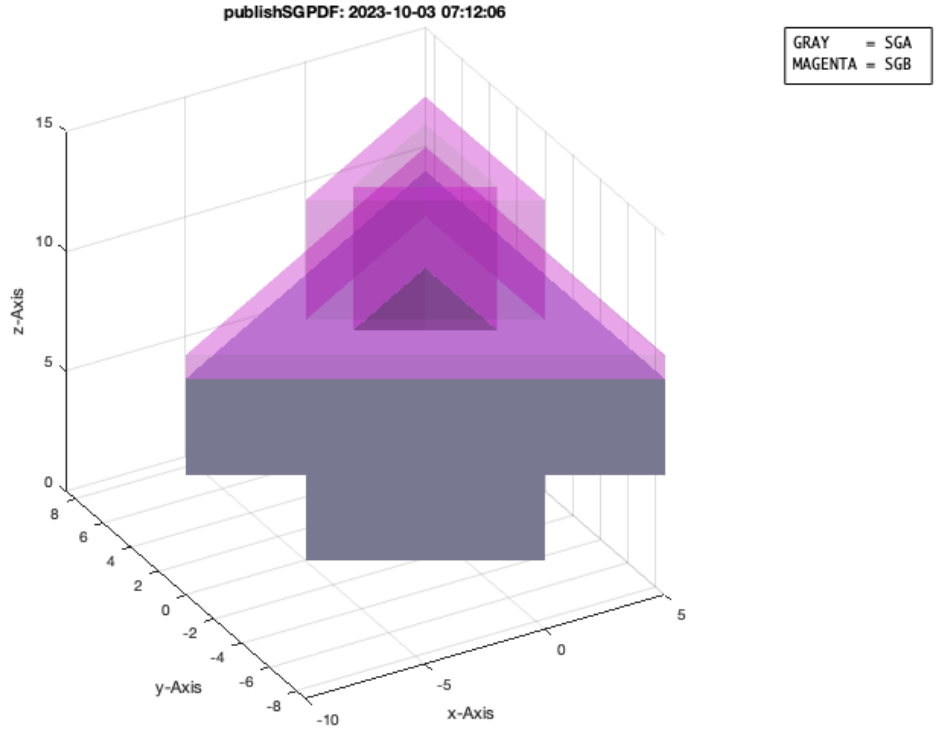


4. Cutting and separating a solid geometries in two parts

By using the output of SGslicer it is possible to create an upper and lower part of an object or even by two cutting plane to cut a part out of a larger obect. This is done by the function **SGcut**.

```
VLFLfigure;
SGcut(A,9);
```

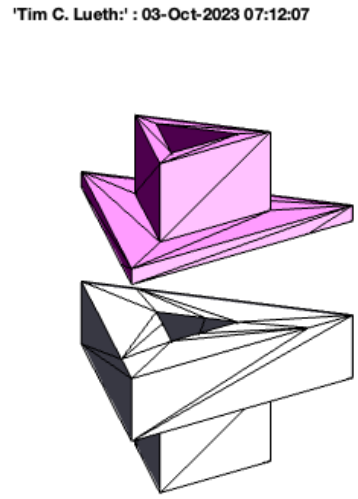
```
ans =
  Figure (1: AOI Matlab Solid Modeler app_2012_11_09) with properties:

      Number: 1
        Name: 'AOI Matlab Solid Modeler app_2012_11_09'
       Color: [1 1 0.9000]
    Position: [31 803 960 540]
       Units: 'pixels'

  Use GET to show all properties
```



The next figure shows a separation of the two part by moving the upper part upwards.

```
[L,U]=SGcut(A,9)
VLFLfigure;
SGplot(SGtransP(L,[0 0 -3]),'w');
SGplot(SGtransP(U,[0 0 +3]),'m');
view (50,20);
```

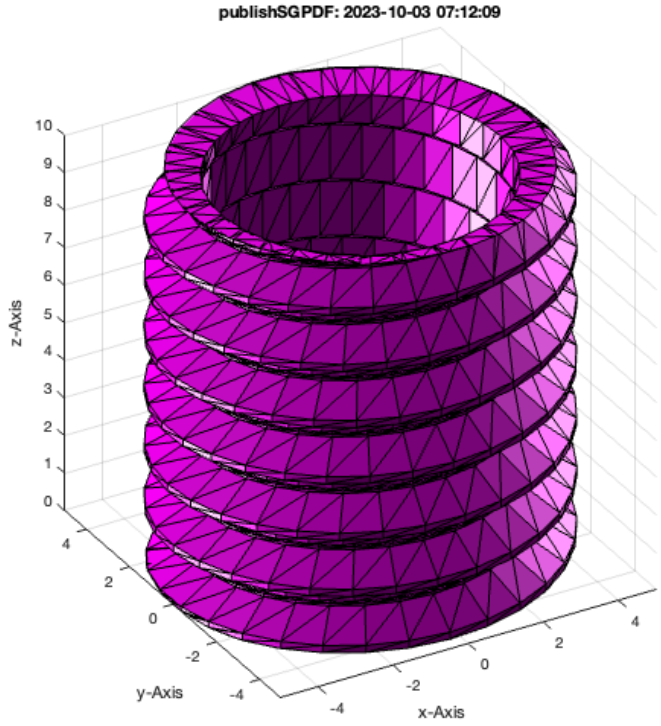
```
L =
  struct with fields:

    VL: [24×3 double]
    FL: [48×3 double]
U =
  struct with fields:

    VL: [24×3 double]
    FL: [48×3 double]
```



5. Cutting as useful tool for the ending of complex shaped geoemtries

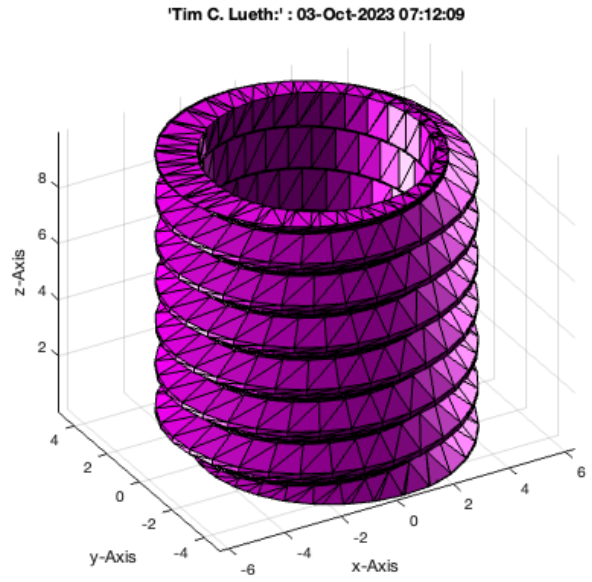
Some geometries such as screwnuts have specific geometries that have their origin in the manufacturing process of the threads. To create also similar shapes it is necessary to create a longer thread and to cut out the required length later:

```
VLFLfigure;
SGthread (10,10,[],[],'C'); view (-30,30);
% [A,b,c]=SGthread (10,10);
```



Now create a longer thread and cut out the required length later.

```
VLFLfigure;
A=SGthread (10,10+5+5,[],[],'C');
[~,B]=SGcut(A,[5.05 14.95]); B=SGtransP (B,[0 0 -5]);
SGplot(B,'m'); view (-30,30);
```



Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2023-Oct-03), is for limited non commercial educational use only!
Licensee: Tim Lueth (Development Version)!
Please contact Tim Lueth, Professor at TU Munich, Germany!
WARNING: This VLFL-Lib (Rel. ) license will exceed at 06-Jul-2078 07:12:10!
Executed 03-Oct-2023 07:12:12 by 'timlueth' on a MACI64 using Mac OSX 13.6 | R2023a Update 5 | SG-Lib 5.4
======================================== Used Matlab products: ========================================
database_toolbox
distrib_computing_toolbox
fixed_point_toolbox
image_toolbox
map_toolbox
matlab
optimization_toolbox
pde_toolbox
phased_array_system_toolbox
signal_blocks
signal_toolbox
simmechanics
simscape
simulink
statistics_toolbox
================================================================================================================
```

- *Tim Lueth, tested and compiled on OSX 10.7.5 with Matlab 2014b on 2015-06-08*
- *Christina Friedrich, executed and published on 64 Bit PC using Windows with Matlab 2015a on 2015-06-17*

Published with MATLAB® R2023a