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

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

Contents

- 1. Summary of the already explained aspects of the VLFL-Toolbox
- 2. Create a sample solid for this exercise
- 3. Analyze a slice plane through a solid geoemtry
- 4. Cutting and separating a solid geometries in two parts
- 5. Cutting as useful tool for the ending of complex shaped geoemtries
- Final remarks on toolbox version and execution date

1. Summary of the already explained aspects of the VLFL-Toolbox

In examples 1-7 we've learned about:

- *Point lists* (PL), *Vertex lists* (VL), *Facet lists* (VL), *Edge lists* (EL)
- Import, and export of *STL-File* data, Conversion of text strings into solid volumes
- *Closed polygon lists* (CPL) for boolesche operations of contours
- 2.5D extrusion of closed polyogon list (CPL) to create *Solid Geometries* (SG)
- 2.5D rotation of closed polyogon list (CPL) to create *Solid Geometries* (SG)
- Spatial transformation, positioning, alignment and merging of solid geometry sets

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.

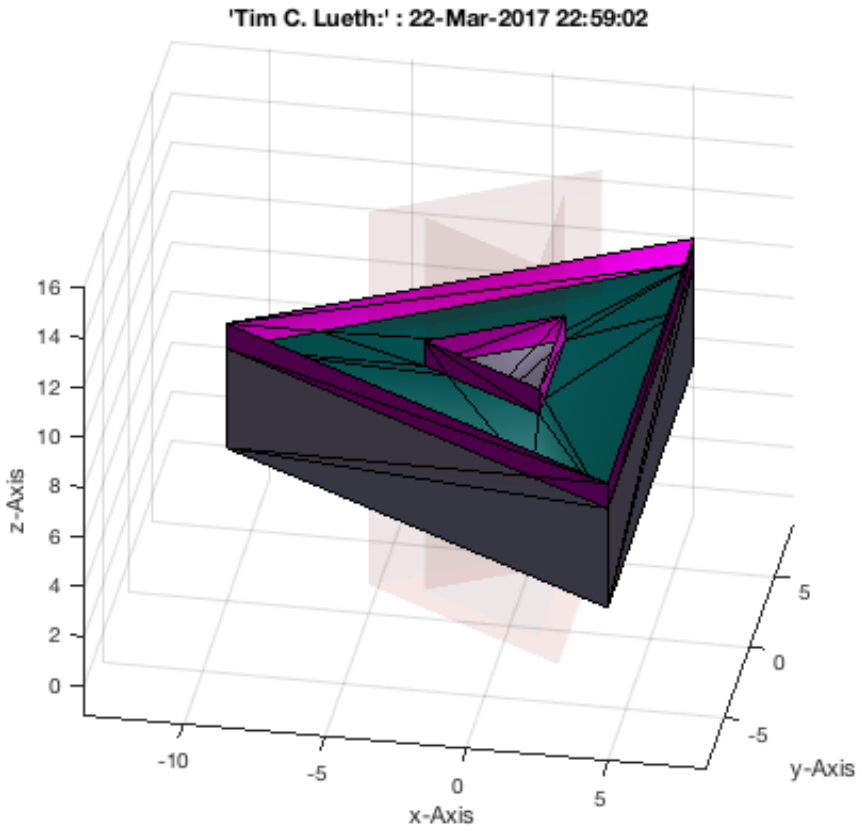
```
close all
SGsample(7);
A=SGsample(7);
```



3. Analyze a slice plane through a solid geoemtry

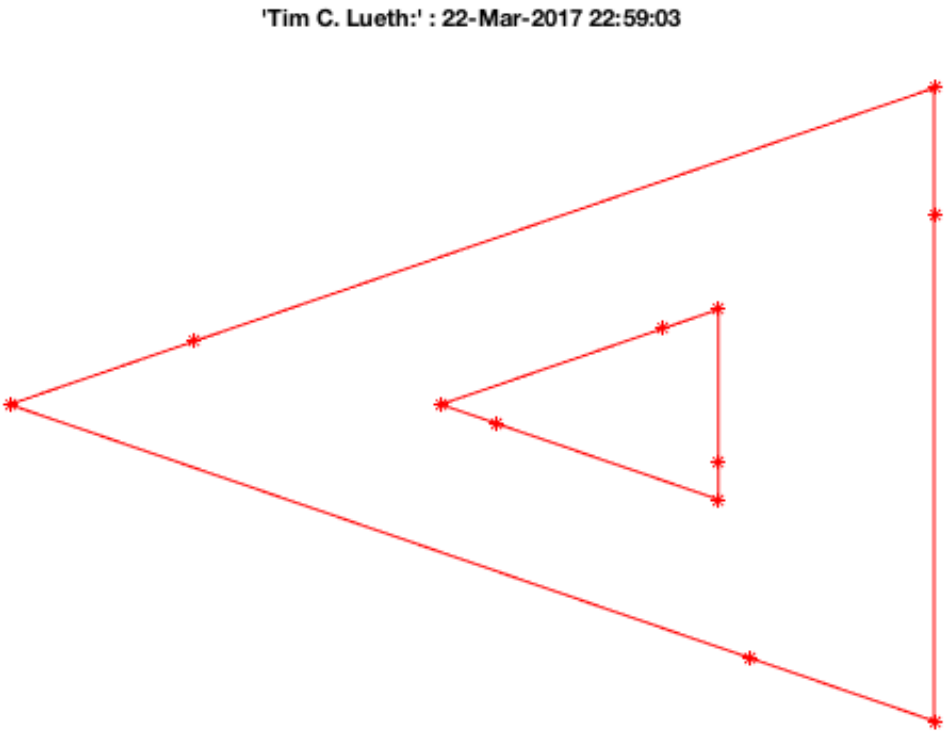
Slicing at a specified z-coordinates is a more complex procedure than expected if several solids are processed that can penetrate each other. By slicing a single solid, the crossed facets/triangles are separated into 2 upper and lower parts that will lead to 2 lower and 1 upper facets or 1 lower and 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);
```



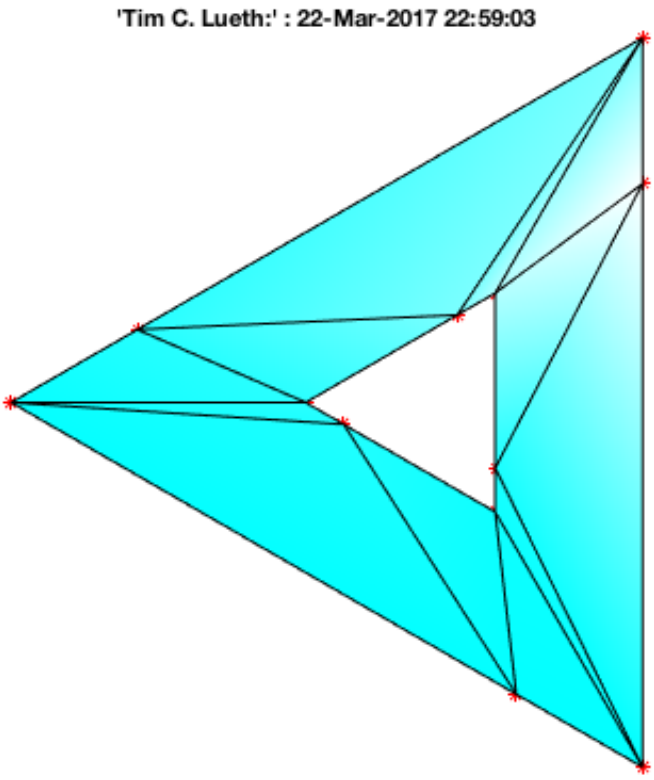
It is also possible just to show the cutting edges of the cutting contour

```
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');
```



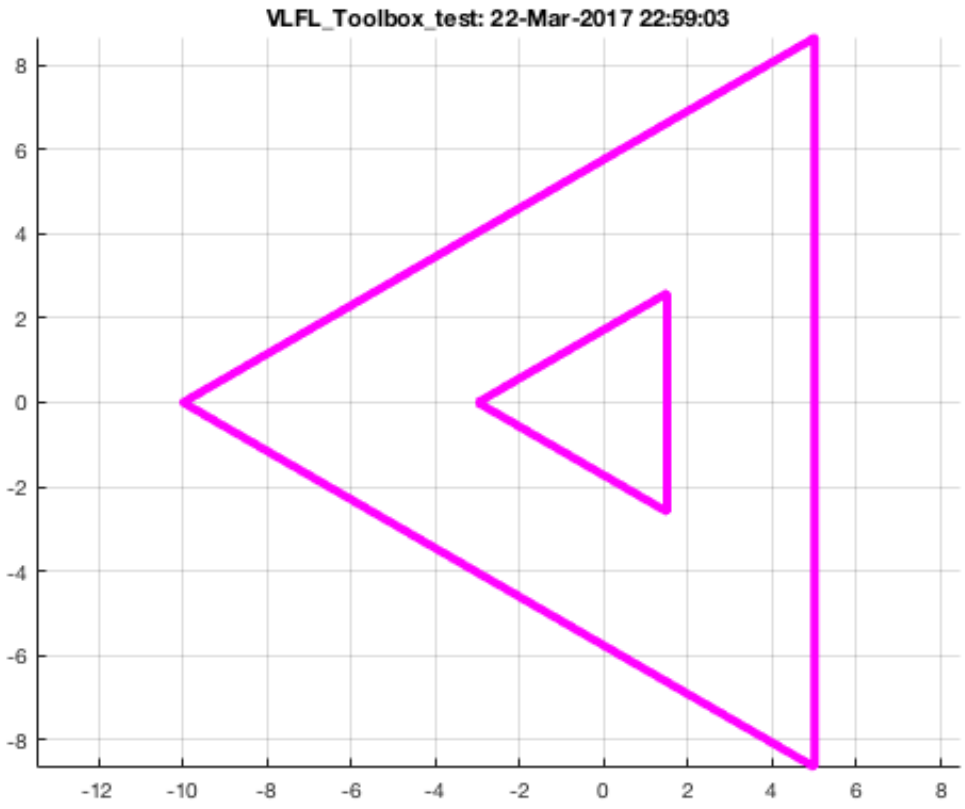
Often we want directly getting a closed contour of a slice.

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

```
warn =

  logical

   0
```



The output parameter warns if a ambiguous cutting result exists

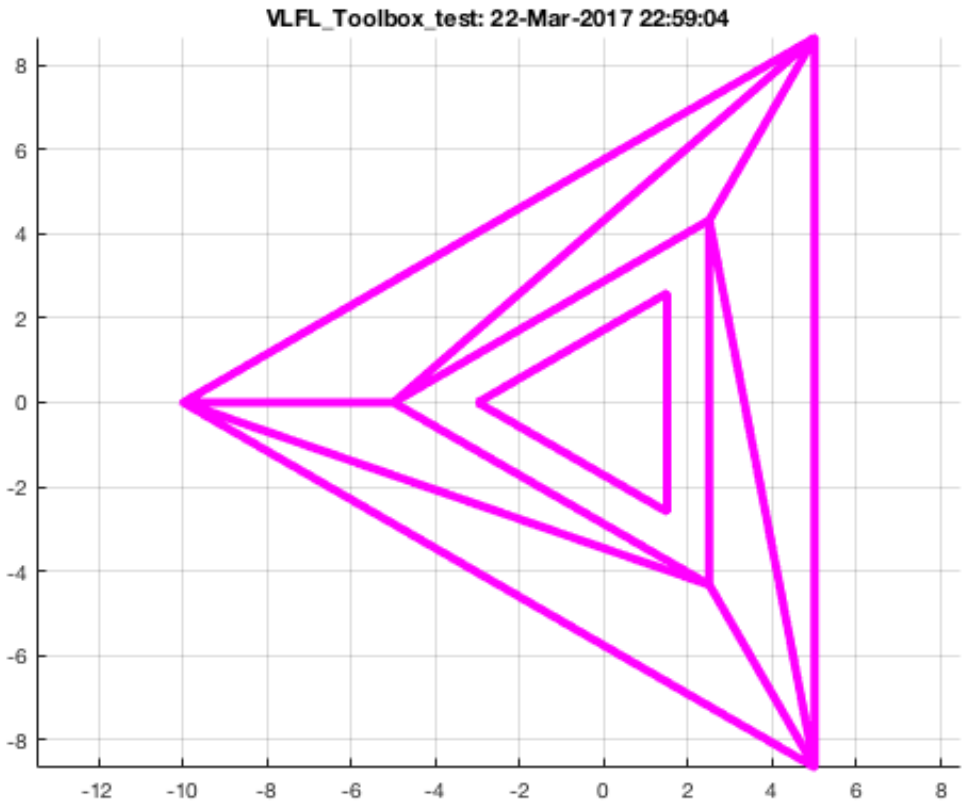
```
CPLofSGslice(A,10); [CPL,warn]=CPLofSGslice(A,10); warn
```

```
Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free

warn =

  logical

   1
```



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);
```

'Tim C. Lueth:' : 22-Mar-2017 22:59:05



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 =

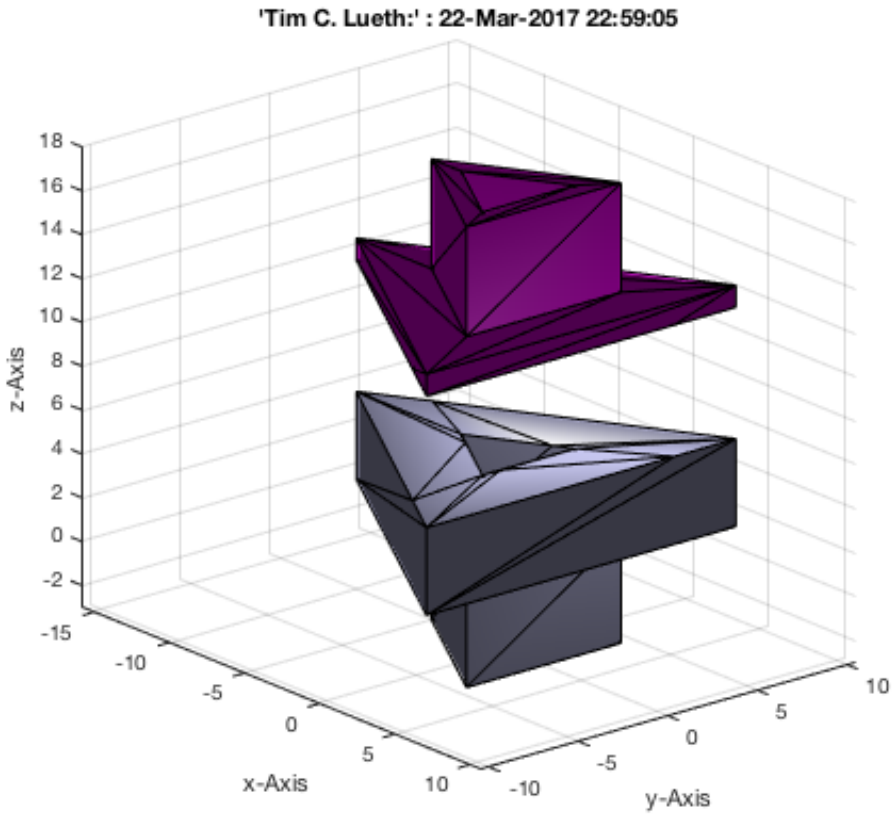
  struct with fields:

    VL: [27×3 double]
    FL: [54×3 double]


U =

  struct with fields:

    VL: [27×3 double]
    FL: [54×3 double]
```

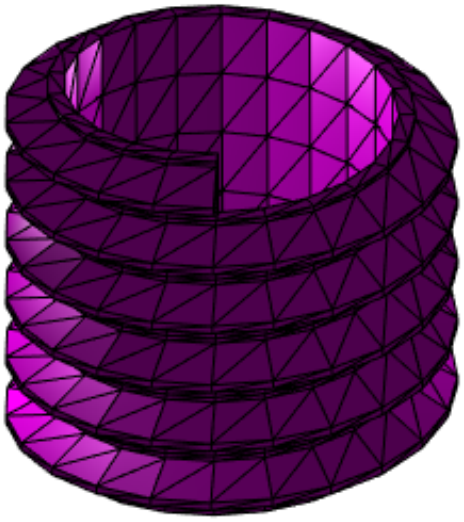


5. Cutting as useful tool for the ending of complex shaped geoemtries

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:

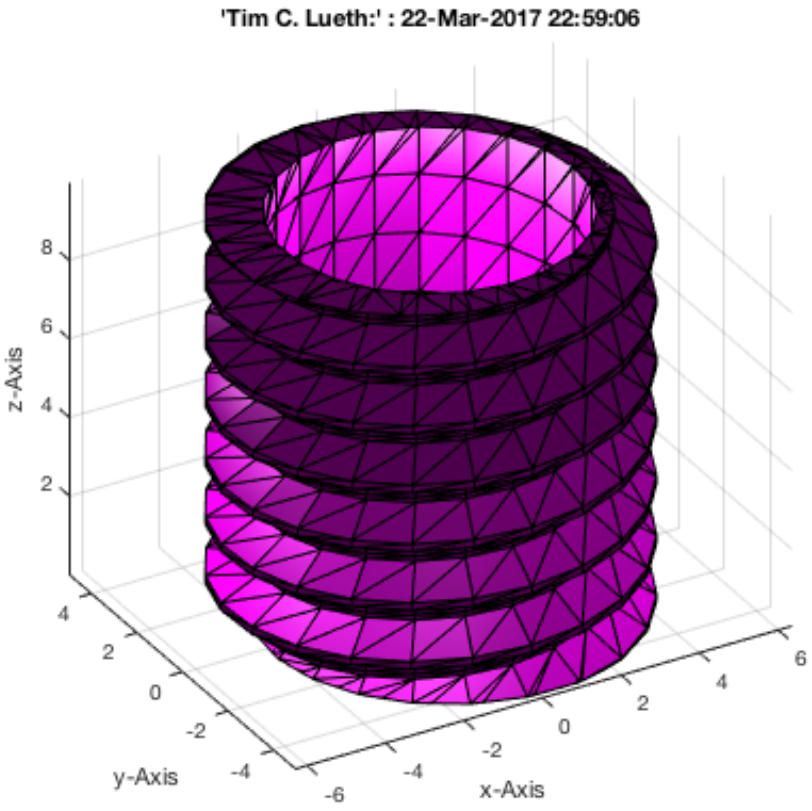
```
VLFLfigure;
SGthread (10,10,[],[],'C'); view (-30,30);
% [A,b,c]=SGthread (10,10);
```

'Tim C. Lueth:' : 22-Mar-2017 22:59:06



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. 3.6 (2017-Mar-23), is for limited non commercial educational use only!
Licensee: 3.6 Tim Lueth for academic and education 05-Mar-2017 !
Please contact Tim Lueth, Professor at TU Munich, Germany!
WARNING: This VLFL-Lib (Rel. 3.6) license will exceed at 18-Mar-2018!
Executed 22-Mar-2017 22:59:09 by 'lueth' using Matlab 9.1.0.441655 (R2016b) on a MACI64
======================================== Used Matlab products: ============================
==========
map_toolbox
matlab
simmechanics
simscape
simulink
================================================================================================
==========
```

- *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® R2016b