# Exercise 04: 2½D Design Using Boolean Operators on Closed Polygon Lists (CPL)

2014-11-21: 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. Moving and rotating point lists (PL) and closed polygon lists (CPL)
- 3. Simple extrusion of point lists (PL/CPL) to design 2½D solids
- 4. Simple Design of 2½D solids by boolean operators for point lists (PL/CPL)
- 5. Unite both contours and extrusion: CPL=CPLpolybool('or',PLA,PLB)
- 6. Intersect both contours: CPL=CPLpolybool('and',PLA,PLB)
- 7. Substract contour B from A: CPL=CPLpolybool('-',PLA,PLB)
- 8. Substract contour A from B: CPL=CPLpolybool('-',PLB,PLA)
- 9. Exclusive or of contour A and B: CPLpolybool('xor',PLB,PLA)
- 10. Checking the solid volumes for 3D printing
- Final remarks on toolbox version and execution date


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

In example 1-3 we've learned about:

- Point lists (PL), vertex lists (VL), facet lists (VL) and edge lists (EL)
- Functions for import, check, and export of STL-Data
- Closed polygon lists (CPL) for boolesche operations of embedded contours
- Orientation of inner and outer contours for 3D extrusion
- Different strategies for surface tesselation (Delaunay,row-scanning)
- Functions for generating/extruding surfaces and surface bounded solid volumes
- Functions for displaying points, surfaces, volumes different colors and transparency
- Functions for turning, mirroring and spatial transformation of objects
- Functions for generation of text string as solid objects

## 2. Moving and rotating point lists (PL) and closed polygon lists (CPL)

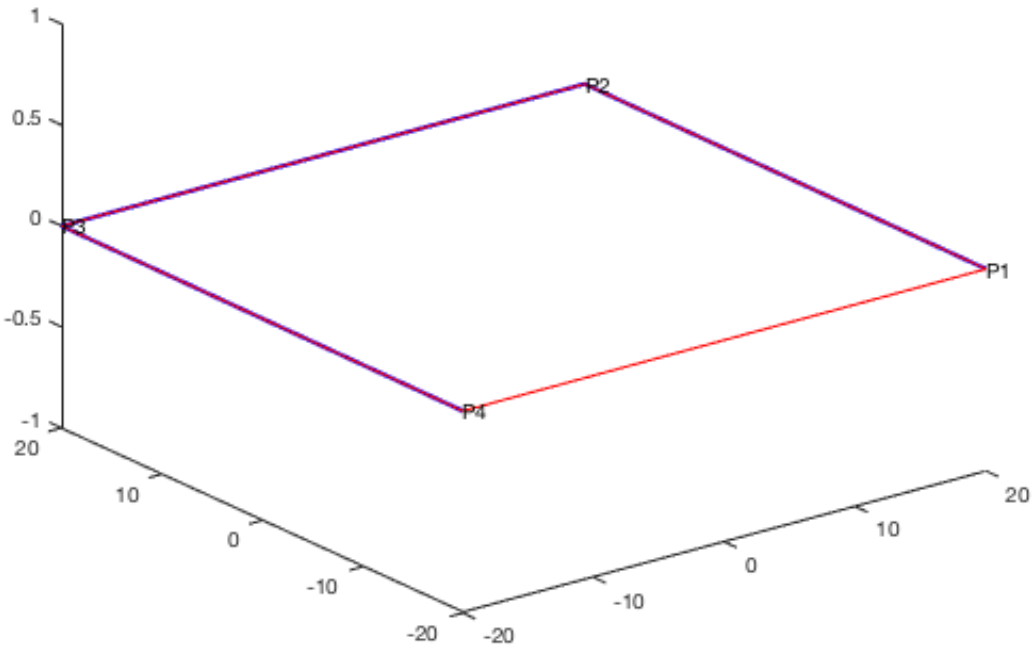
By using point lists (PL) and closed polybool lists (CPL) it is very convinient to design 2.5D objects. Here we see a first example to design a simple square by three function:

At the beginning we just plot simple point lists or closed polygon lists

- **PLplot** plots the point list as open contour
- **CPLplot** plots the point list as closed contour
- **textVL** plots descriptors at the points

```
close all;
PLA=PLcircle(20*sqrt(2),4);             % Generate a circle with 4 point, i.e. square
```

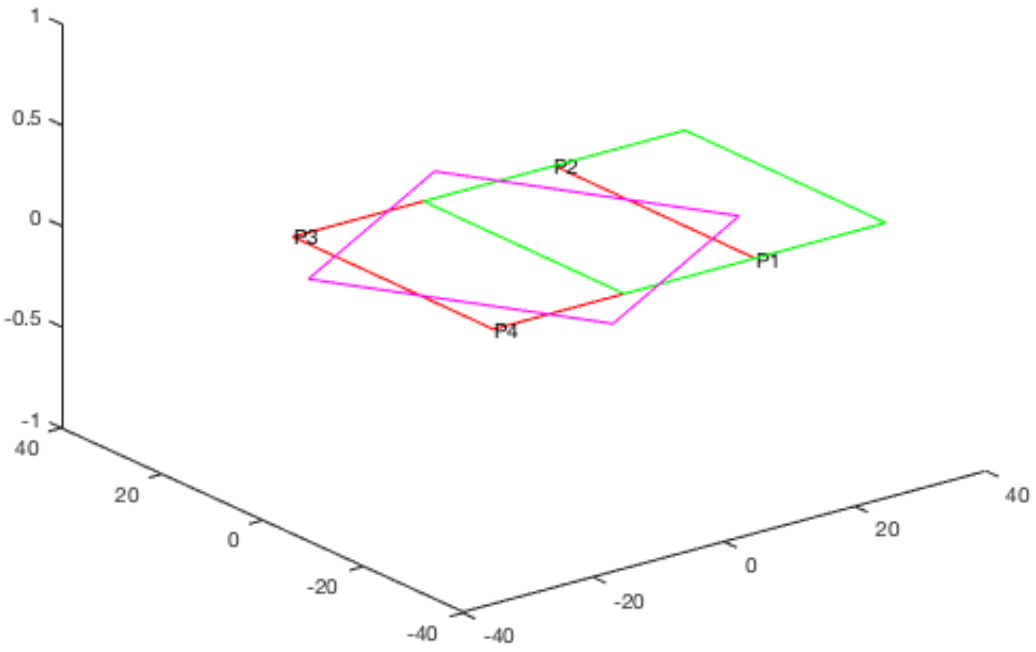
```
PLplot(PLA,'b',2);                      % Plots the points in blue
CPLplot(PLA,'r');                       % Plots the closed polygon in red
textVL (PLA);                           % Plots point descriptors
```




Next we move the square and rotate the square

- **PLtransP** moves a point list (PL) or closed polygon list(CPL)
- **PLtransR** rotates a point list (PL) or closed polygon list(CPL)

```
close all;
CPLplot(PLA,'r');                       % Plots the closed polygon in red
textVL (PLA);                           % Plots point descriptors
CPLplot(PLtransP(PLA,[20  0]),'g');     % Plot the moved polygon in green
CPLplot(PLtransR(PLA,rot(pi/6)),'m');   % Plot the rotated polygon in magenta
```

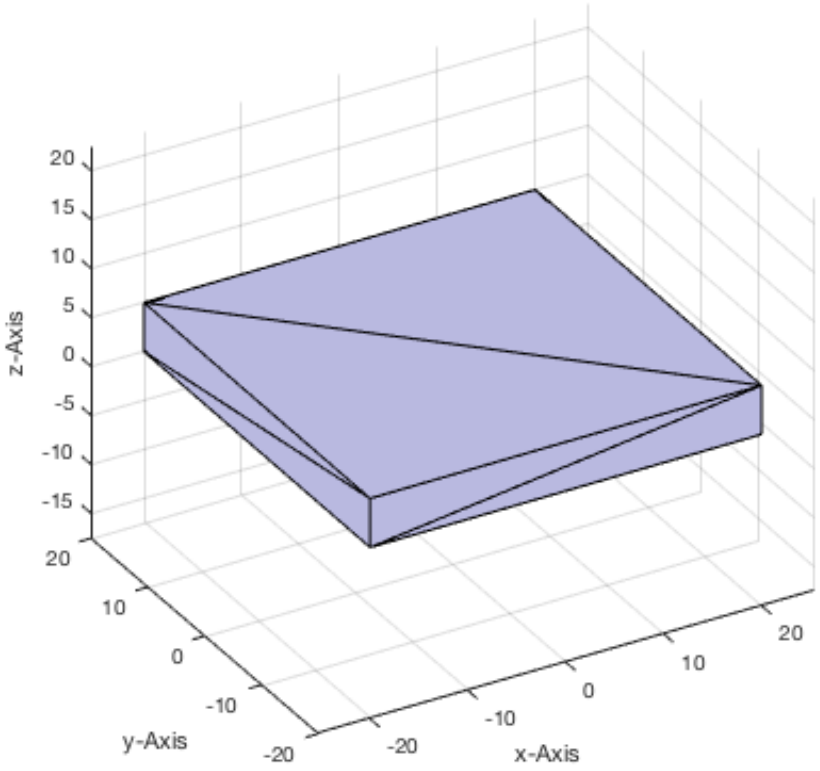



## 3. Simple extrusion of point lists (PL/CPL) to design 2½D solids

Next we extrude the square in 3D

- **VLFLofCPLz** extrudes point list (PL) or closed polygon list(CPL) in z

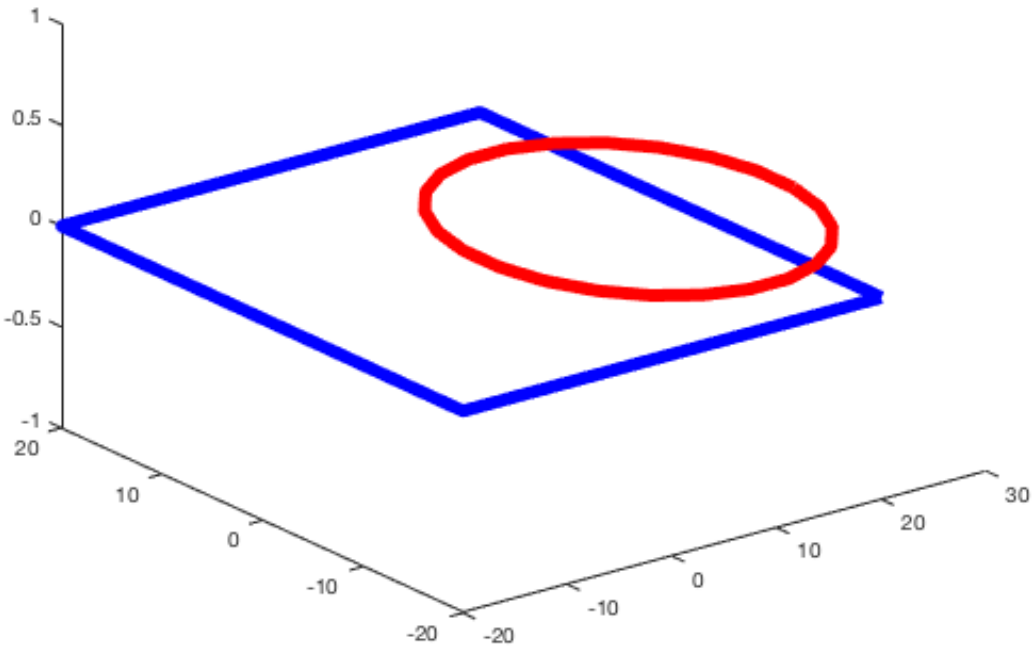
```
close all
[VL,FL]=VLFLofCPLz (PLA,5);
VLFLplot (VL,FL,'w'), axis equal, view(-30,30);
```




## 4. Simple Design of 2½D solids by boolean operators for point lists (PL/CPL)

In this example we start with two point list, a square and an octaedron

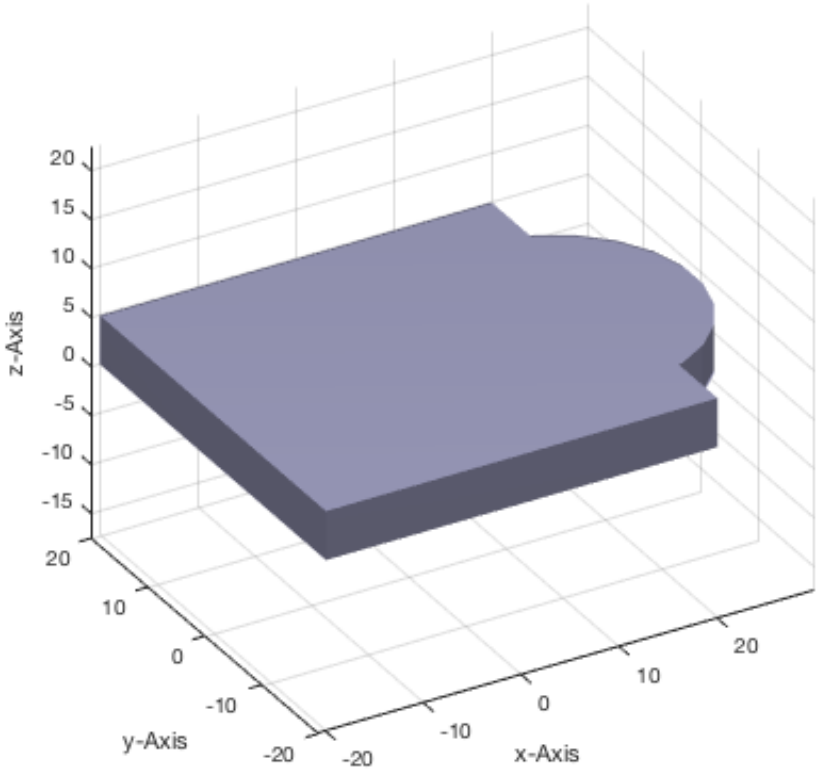
```
close all;
PLA=PLcircle(20*sqrt(2),4);
PLB=PLcircle(10*sqrt(2),24);  PLB=PLtransP(PLB,[15 0]);
CPLplot(PLA,'b',6); CPLplot(PLB,'r',6);
```




## 5. Unite both contours and extrusion: CPL=CPLpolybool('or',PLA,PLB)

```
close all
CPL=CPLpolybool('or',PLA,PLB); [VL,FL]=VLFLofCPLz(CPL,5);
VLFLplot (VL,FL,'w'), axis equal, view(-30,30); VLFLplotlight (1);
VLFLwriteSTL (VL,FL,'A','EXP04-unite')
```

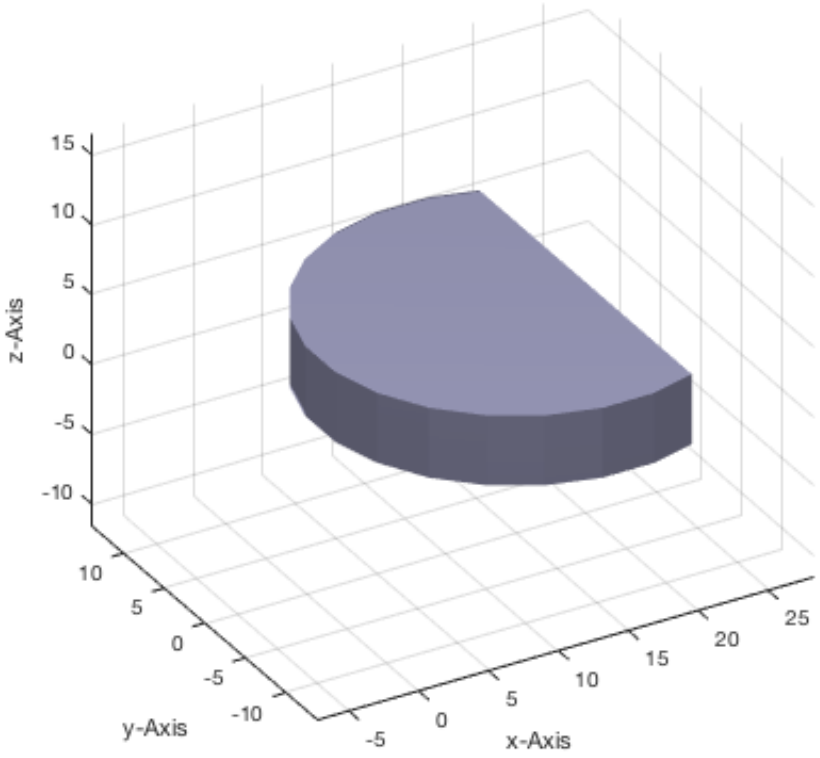
```
WRITING STL FILE /Users/lueth/Desktop/A.STL in ASCII MODE
completed.
```




## 6. Intersect both contours: CPL=CPLpolybool('and',PLA,PLB)

```
close all
CPL=CPLpolybool('and',PLA,PLB); [VL,FL]=VLFLofCPLz(CPL,5);
VLFLplot (VL,FL,'w'), axis equal, view(-30,30); VLFLplotlight (1);
VLFLwriteSTL (VL,FL,'A','EXP04-intersect')
```

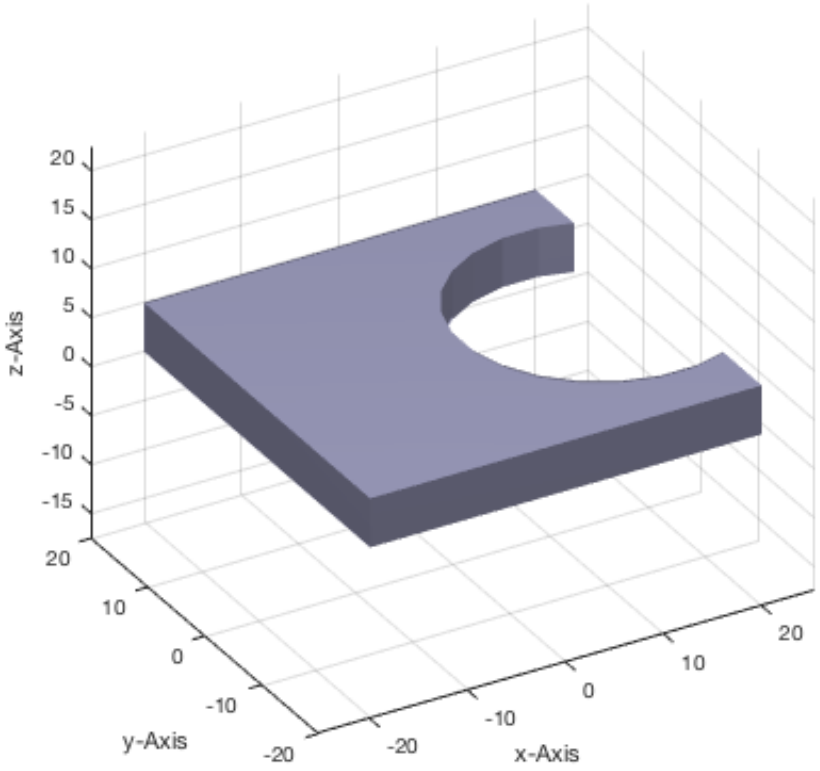
```
WRITING STL FILE /Users/lueth/Desktop/A.STL in ASCII MODE
completed.
```




## 7. Substract contour B from A: CPL=CPLpolybool('-',PLA,PLB)

```
close all
CPL=CPLpolybool('-',PLA,PLB); [VL,FL]=VLFLofCPLz(CPL,5);
VLFLplot (VL,FL,'w'), axis equal, view(-30,30); VLFLplotlight (1);
VLFLwriteSTL (VL,FL,'A','EXP04-AminusB')
```

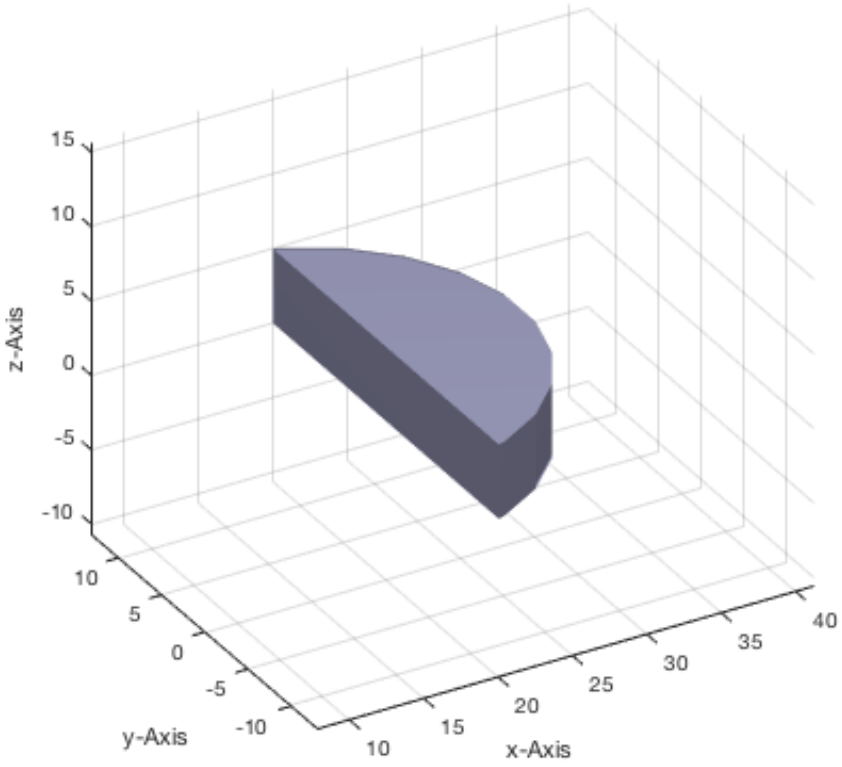
```
WRITING STL FILE /Users/lueth/Desktop/A.STL in ASCII MODE
completed.
```




## 8. Substract contour A from B: CPL=CPLpolybool('-',PLB,PLA)

```
close all
CPL=CPLpolybool('-',PLB,PLA); [VL,FL]=VLFLofCPLz(CPL,5);
VLFLplot (VL,FL,'w'), axis equal, view(-30,30); VLFLplotlight (1);
VLFLwriteSTL (VL,FL,'EXP04-AminusB')
```

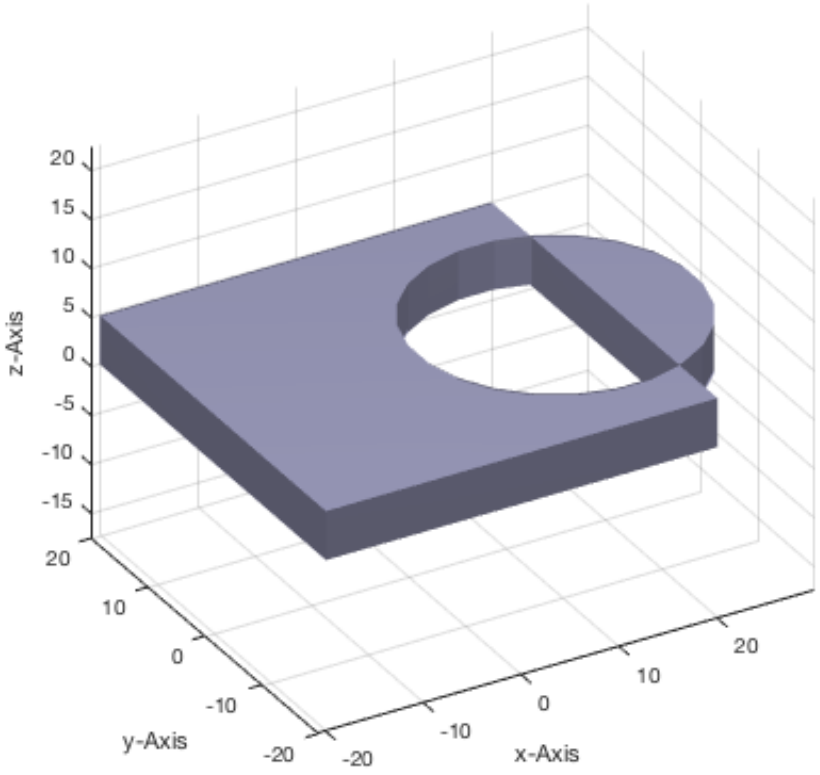
```
WRITING STL FILE /Users/lueth/Desktop/EXP04-AminusB.STL in ASCII MODE
completed.
```




## 9. Exclusive or of contour A and B: CPLpolybool('xor',PLB,PLA)

```
close all
CPL=CPLpolybool('xor',PLB,PLA); [VL,FL]=VLFLofCPLz(CPL,5);
VLFLplot (VL,FL,'w'), axis equal, view(-30,30); VLFLplotlight (1);
VLFLwriteSTL (VL,FL,'EXP04-AxorB')
% VLFLviewer(VL,FL);
```

```
Warning: Duplicate data points have been detected and removed.
 The Triangulation indices and Constraints are defined with respect to the
 unique set of points in DelaunayTri property X.
WRITING STL FILE /Users/lueth/Desktop/EXP04-AxorB.STL in ASCII MODE
completed.
```




## 10. Checking the solid volumes for 3D printing

During the last extrusion we got a warning from a Delaunay-triangulation during the extrusion function VLFLofCPLz. This is typically a warning that somehow the final part cannot be printed with a 3D printing process such as FDM,SLS,3DP etc. Here in this case, the result of xor were two parts that touch each other at two edges. Such a part cannot be printed. The reason behind is called non-manifold edge problem. There are also problems with non manifold points and non-manifold facets.

```
VLFLchecker (VL,FL);
```

```
VLFLchecker: 60 vertices and 120 facets.
    0 FACET PROBLEMS DETECTED (ERRORS)
    0 VERTEX PROBLEMS DETECTED (OBSOLETE WARNING)
    4 EDGE PROBLEMS DETECTED (NON MANIFOLD WARNING)
    0 SOLID/EDGE PROBLEMS DETECTED (OPEN SOLID WARNING)
```

## 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 22:58:20 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 2014-11-21*
- *Mattias Traeger, executed and published on 64 Bit PC using Windows with Matlab 2014b on 2014-11-21*

---

*Published with MATLAB® R2016b*