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

2014-11-21: Tim C. Lueth, Professor at Technische Universität München, Germany (URL: http://www.SG-Lib.org) - Last Change: 2019-06-11

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: Creation of Kinematic Chains and Robot Structures
- Tutorial 36: Creating a Patient-Individual Arm-Skin Protector-Shell
- Tutorial 37: Dimensioning of STL Files and Surface Data
- Tutorial 38: Some more solid geometry modelling function
- Tutorial 39: HEBO Modules robot design
- Tutorial 40: JACO Robot Simulation and Control
- Tutorial 41: Inserting Blades, Cuts and Joints into Solid Geometries
- Tutorial 42: Performing FEM Stress and Displacement Analysis and Structural Optimization of Solids
- Tutorial 43: Performing FEM Structural Optimization (CAO) and Topological Optimization (SKO) of Solids
- Tutorial 44: Creation of solids and kinematics from 3D curves and transformation matrices
- Tutorial 45: Creation of Solids using the SG-Coder - SGofCPLcommand
- Tutorial 46: Creating Fischertechnik compatible gear boxes using SGofCPLcommand
- Tutorial 47: Creating four-joints by 3 pose synthesis

Motivation for this tutorial: (Originally SolidGeometry 1.6 required)

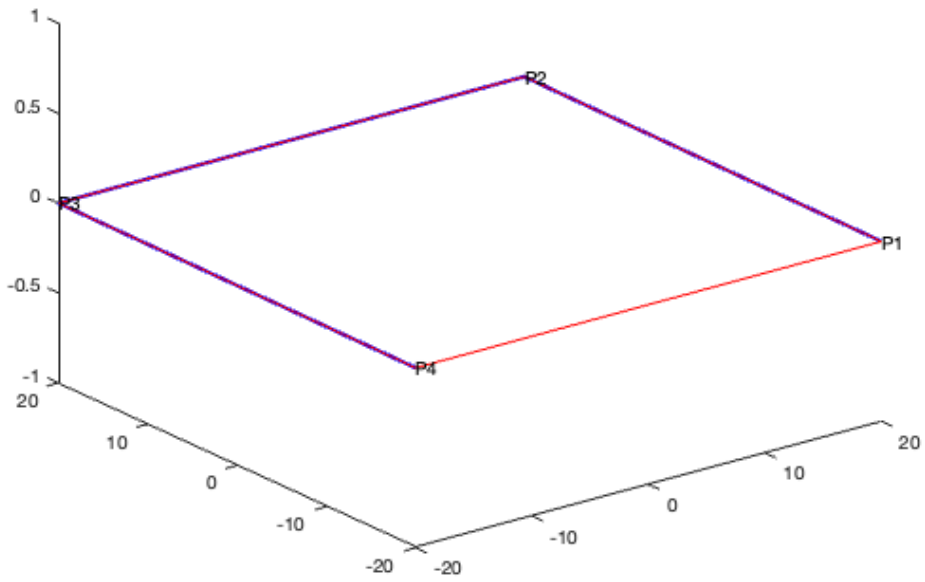
2. Moving and rotating point lists (PL) and closed polygon lists (CPL)

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

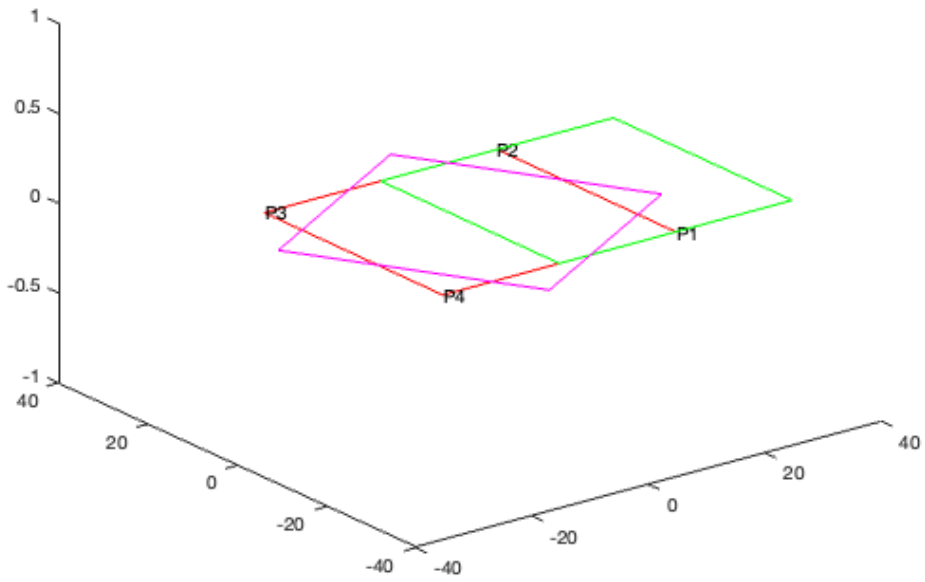
```
close all;
PLA=PLcircle(20*sqrt(2),4);             % Generate a circle with 4 point, i.e. square
PLplot(PLA,'b',2);                      % Plots the points in blue
CPLplot(PLA,'r');                       % Plots the closed polygon in red
textVL (PLA);                           % Plots point descriptors
```



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
```



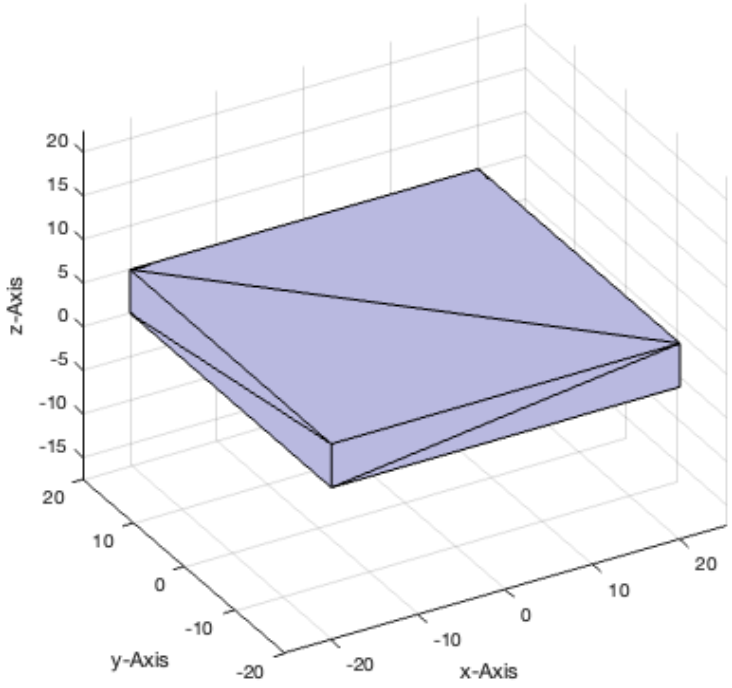
3. Simple extrusion of point lists (PL/CPL) to design 2½D solids

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);
```

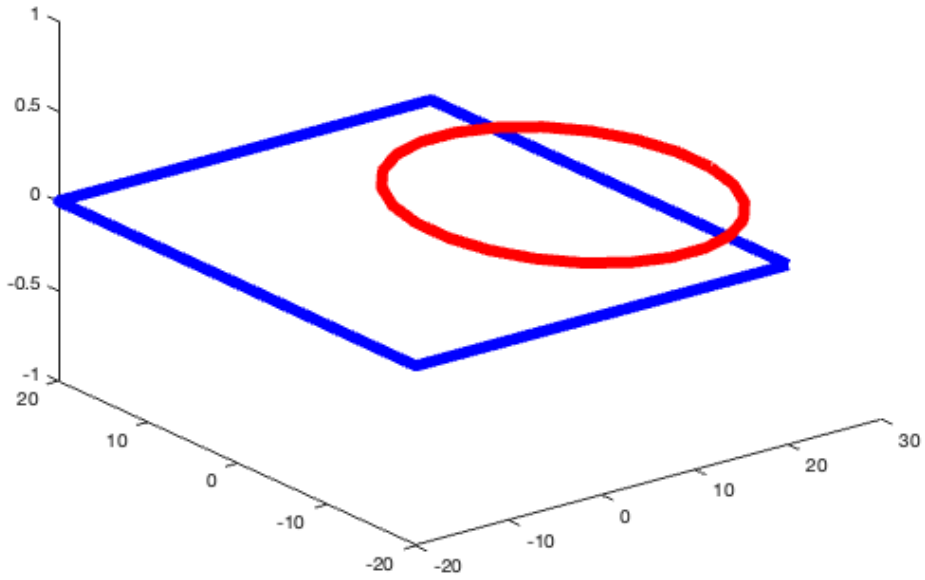
```
ans =
  Patch with properties:

    FaceColor: 'flat'
    FaceAlpha: 1
    EdgeColor: [0 0 0]
    LineStyle: '-'
        Faces: [12×3 double]
     Vertices: [8×3 double]

  Use GET to show all properties
```



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

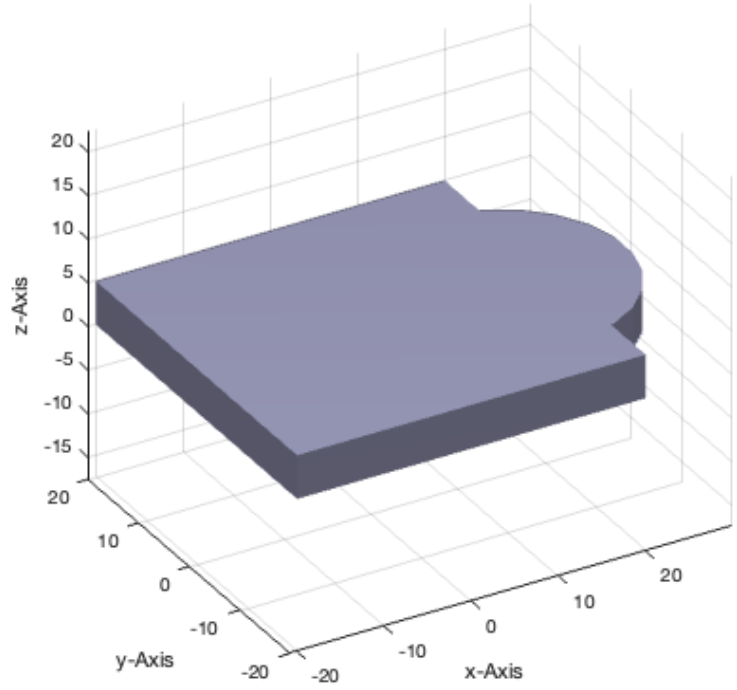
```
close all;
PLA=PLcircle(20*sqrt(2),4);
PLB=PLcircle(10*sqrt(2),24);  PLB=PLtransP(PLB,[15 0]);
CPLplot(PLA,'b',6); CPLplot(PLB,'r',6);
```



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')
```

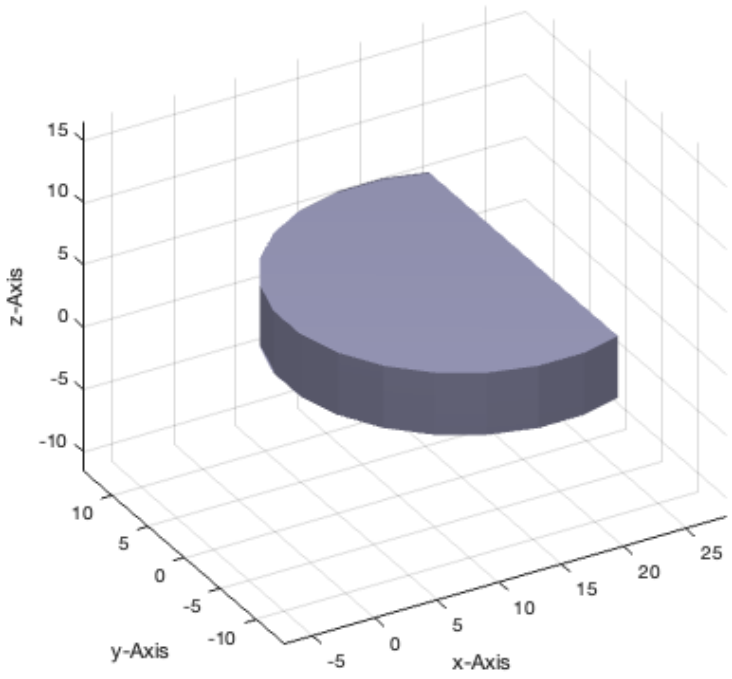
```
ans =
  Patch with properties:

    FaceColor: 'flat'
    FaceAlpha: 1
    EdgeColor: [0 0 0]
    LineStyle: '-'
        Faces: [60×3 double]
     Vertices: [32×3 double]

  Use GET to show all properties
WRITING STL FILE /Users/timlueth/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')
```

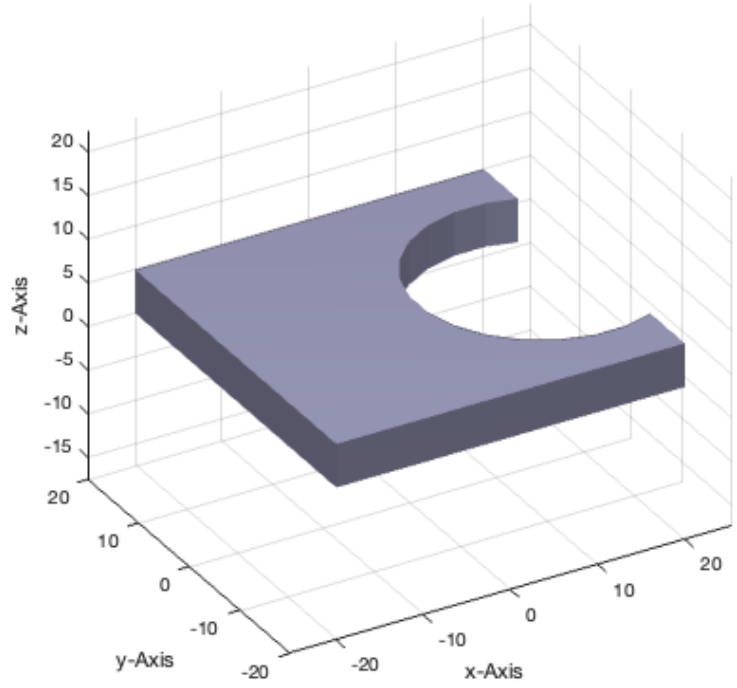
```
ans =
  Patch with properties:

    FaceColor: 'flat'
    FaceAlpha: 1
    EdgeColor: [0 0 0]
    LineStyle: '-'
        Faces: [60×3 double]
     Vertices: [32×3 double]

  Use GET to show all properties
WRITING STL FILE /Users/timlueth/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')
```

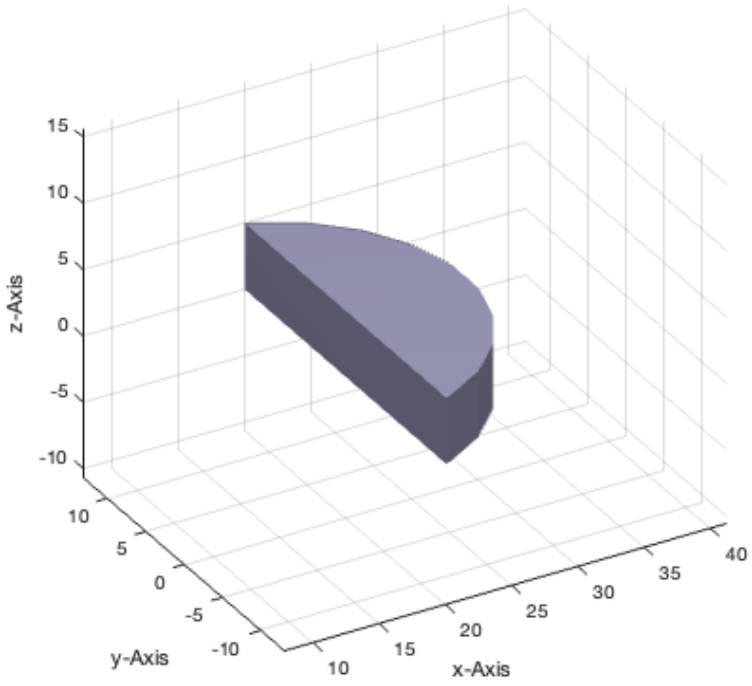
```
ans =
  Patch with properties:

    FaceColor: 'flat'
    FaceAlpha: 1
    EdgeColor: [0 0 0]
    LineStyle: '-'
        Faces: [76×3 double]
     Vertices: [40×3 double]

  Use GET to show all properties
WRITING STL FILE /Users/timlueth/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')
```

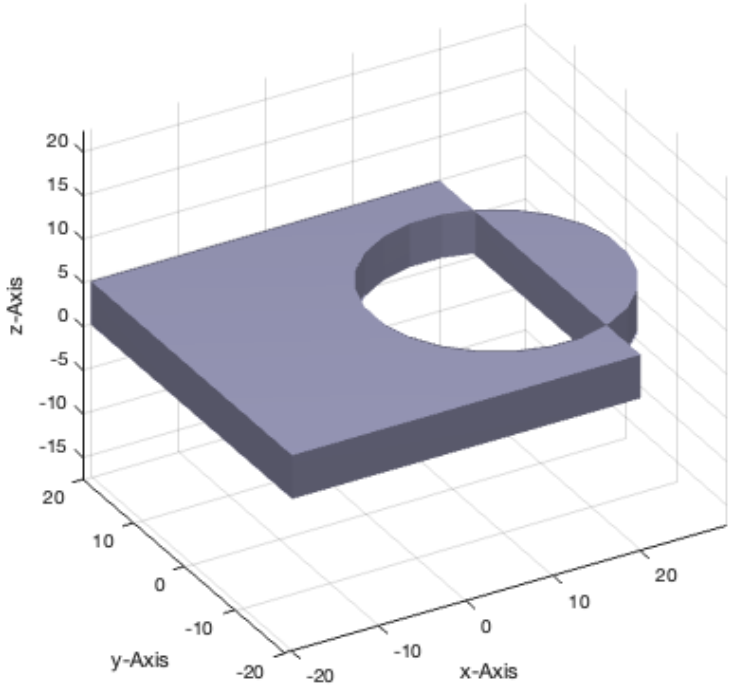
```
ans =
  Patch with properties:

    FaceColor: 'flat'
    FaceAlpha: 1
    EdgeColor: [0 0 0]
    LineStyle: '-'
        Faces: [44×3 double]
     Vertices: [24×3 double]

  Use GET to show all properties
WRITING STL FILE /Users/timlueth/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);
```

```
ans =
  Patch with properties:

    FaceColor: 'flat'
    FaceAlpha: 1
    EdgeColor: [0 0 0]
    LineStyle: '-'
        Faces: [120×3 double]
     Vertices: [60×3 double]

  Use GET to show all properties
WRITING STL FILE /Users/timlueth/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.  (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 07:10:40!
Executed 03-Oct-2023 07:10:42 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 2014-11-21*
- *Mattias Traeger, executed and published on 64 Bit PC using Windows with Matlab 2014b on 2014-11-21*

Published with MATLAB® R2023a