# Exercise 15: Create a Solid by 2 Closed Polygons

2015-10-03: 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. Basics on the creation of a solid between two planar contours in different height
- 3. Solid surface generation and the importance of start point of the contour (turning)
- 4. Solid surface generation and the importance of point assignment strategy
- 5. Solid surface generation for polygons with a small number of points
- 6. Two identical contours with (strictly) monotonic increasing point-center angle
- 7. Two contours of the same shape with different number of points
- 8. Two contours of the similar shape with different number of points
- 9. Two contours of the similar shape with different sum of boundary bending angle sum
- Final remarks on toolbox version and execution date


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

**The previous examples of this series covered the following topics:**

- Example 01: First Steps Using the VLFL-Toolbox for Solid Object Design
- Example 02: Using the VLFL-Toolbox for STL-File Export and Import
- Example 03: Closed 2D Contours and Boolean Operations in 2D
- Example 04: 2.5D Design Using Boolean Operators on Closed Polygon Lists (CPL)
- Example 05: Creation, Relative Positioning and Merging of Solid Geometries (SG)
- Example 06: Relative Positioning and Alignment of Solid Geometries (SG)
- Example 07: Rotation of Closed Polygon Lists for Solid Geometry Design
- Example 08: Slicing, Closing, Cutting and Separation of Solid Geometries
- Example 09: Boolean Operations with Solid Geometries
- Example 10: Packaging of Sets of Solid Geometries (SG)
- Example 11: Attaching Coordinates Frames to Create Kinematik Models
- Example 12: Define Robot Kinematics and Detect Collisions
- Example 13: Mounting Faces and Conversion of Blocks into Leightweight-structures
- Example 14: Manipulation Functions for Closed Polygons and Laser Cutting (SVG)

## 2. Basics on the creation of a solid between two planar contours in different height

The creation of a solid based on two CPL (each containing exactly ONE closed polygon) with a z-difference have to be solved by different method depending on the contour.

In general, the inner angle sum of a closed polygon that has no overlaps is exactly 360 degree, i.e. 2pi.

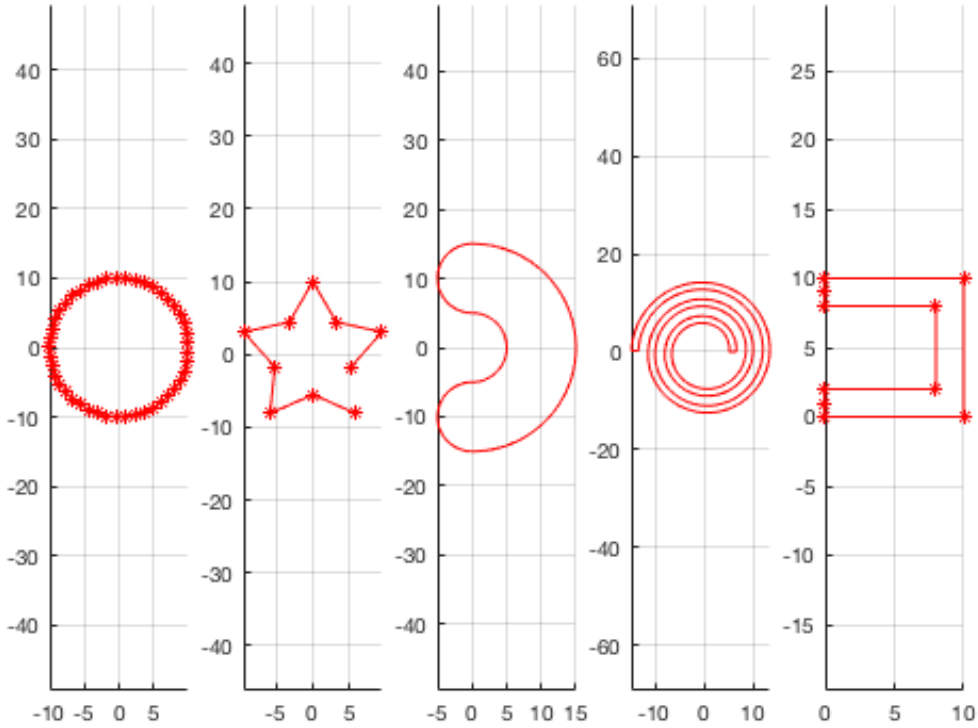
So, for convex polygons there is absolutely no problem by **stepping forward related to the strictly monotonic increasing angle sum**, after **finding a suitable start point** of both polygons (which is a challege itself).

Even **some concave shaped polygons**, such as stars, **have at least monotonic increasing angle sum** and can be connected by this strategy.

Serious challanges exist if we have non monotonic increasing angle sums such as in u-shaped kidneys, or spirals. Here the challange is not only the assignment of points of one contour to a corresponing point on the second, but also how to deal with the starting point if the conturs are rotated.

A challenge is also that the result changes with the order of the input arguments: SGof2CPLz(PLA,PLB) is not the same as (PLB,PLA);

```
SGfigure; view(0,90);
PLU0=[0 0;10 0; 10 10; 0 10; 0 0];
PLU1=[0 0;10 0; 10 10; 0 10; 0 8; 8 8; 8 2; 0 2; 0 0];
PLU2=[0 0;10 0; 10 10; 0 10; 0 9; 0 8; 8 8; 8 2; 0 2; 0 1; 0 0];
subplot(1,5,1); PL=PLcircle(10);PLplot(PL);
view(0,90); grid on; axis equal;
subplot(1,5,2); PL=PLstar(10,10); PLplot(PL);
view(0,90); grid on; axis equal;
subplot(1,5,3); PL=PLkidney(5,15,pi); CPLplot(PL,'r-');
view(0,90); grid on; axis equal;
subplot(1,5,4); PL=CPLspiral(5,15,5*pi); CPLplot(PL,'r-');
view(0,90); grid on; axis equal;
subplot(1,5,5); PL=CPLspiral(5,15,5*pi); CPLplot(PLU2,'r*-');
view(0,90); grid on; axis equal;
```

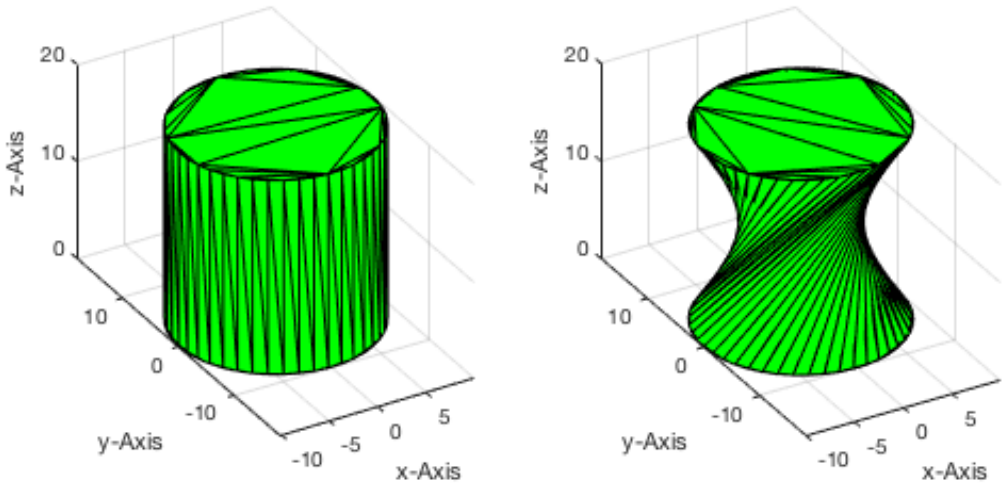


## 3. Solid surface generation and the importance of start point of the contour (turning)

If two identical contours are assigned, the turning angle around the center is of importance. Already a small turning angle, known or unkown, results in a different shape. For solid generation we use the function:

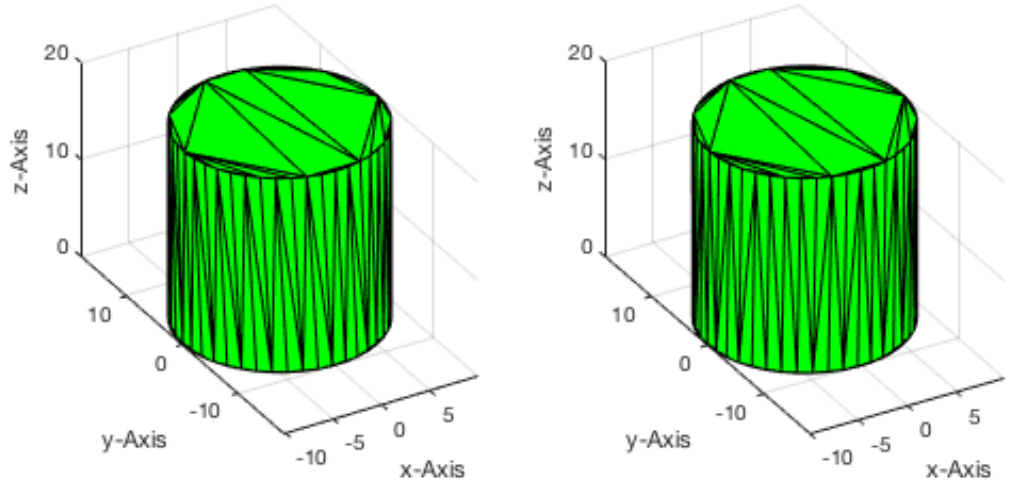
**SGof2CPLz(CPLA,CPLB,z)** - creates a solid between two plane contours in height 0 and z

```
SGfigure; view(-30,30);
subplot(1,2,1);
SG=SGof2CPLz(PLcircle(10),PLcircle(10),20); SGplot(SG,'g'); view(-30,30);
subplot(1,2,2);
SG=SGof2CPLz(PLcircle(10),PLtransR(PLcircle(10),rotdeg(120)),20,[],'none');
SGplot(SG,'g'); view(-30,30);
```




**One input paramter of SGof2CPLz allows to select the turning angle adjustment to 'none', 'rot', or 'miny'.** Please read the documentation of 'czero'=rot (minimal angle near zero of the convex hull) of CPLsortC and PLminyx (Point with the minimal y and minmal x value) to understand 'miny'. In addition it is also possible directly to give the assignment of the first points directly by an 1x2 circshift value [1 1] == 'none'

```
SGfigure; view(-30,30);
subplot(1,2,1);
SG=SGof2CPLz(PLcircle(10),PLtransR(PLcircle(10),rotdeg(90)),20,[],'rot');
SGplot(SG,'g'); view(-30,30);
subplot(1,2,2);
SG=SGof2CPLz(PLcircle(10),PLtransR(PLcircle(10),rotdeg(90)),20,[],'miny');
SGplot(SG,'g'); view(-30,30);
```




**Turning adjustment 'miny' is the default method for turning both contours!** Even if the contour is turned for point assignment, the final order of the points is the same as the original order! This is important for generating tubes based on this function.

## 4. Solid surface generation and the importance of point assignment strategy

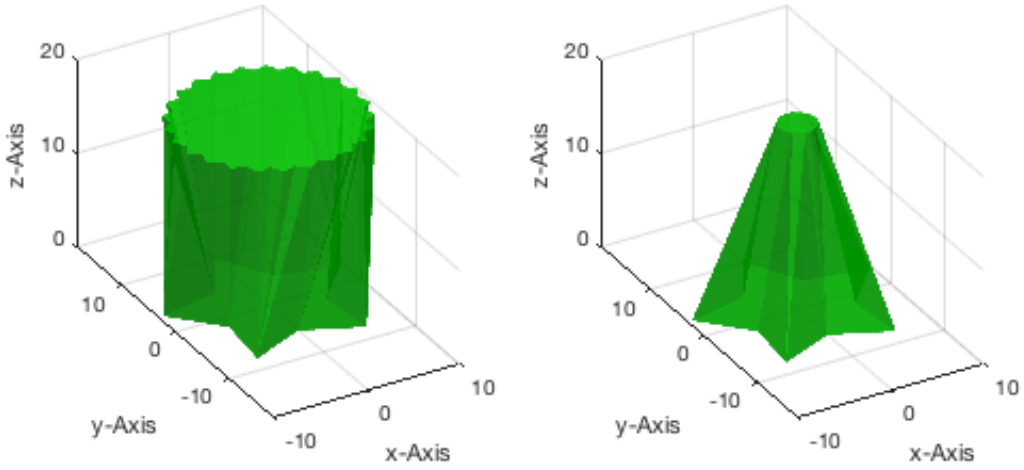
Currently, there are four strategies for the contour point assignment during contour connections:
SGof2CPLZ(PLA,PLB,z,assignment,turning);

- **'number' assignment** based on point index / sum(points) - works well for identical contours with different point numbers. Use in combination with both 'miny' or 'rot'.
- **'length' assignment** based on edge length / sum(edge length) - works well for identical contours (shrinked,grown, different sampling). Use in combination with 'miny' or 'rot', the later especially if the number of points is very large.
- **'angle' assignment** based on abs(edge angle) / sum(abs(edge angle)) - required if the sum(abs(edge angle)) differs remarkable. Use in combination with 'rot' and not with 'miny'.
- **'center' assignment** based on angle between center and point - should not be used anymore.

- **In most cases 'length' and 'miny' works well**, wich are the default values
- **For heavy curved contours use 'angle' and 'rot'**

```
SGfigure; view(-30,30);
subplot(1,2,1);
SG=SGof2CPLz(PLstar(10,10),PLstar(10,50),20); SGplot(SG,'g'); view(-30,30);
VLFLplotlight(1,0.7);
subplot(1,2,2);
SG=SGof2CPLz(PLstar(10,10),PLcircle(2),20); SGplot(SG,'g'); view(-30,30);
```

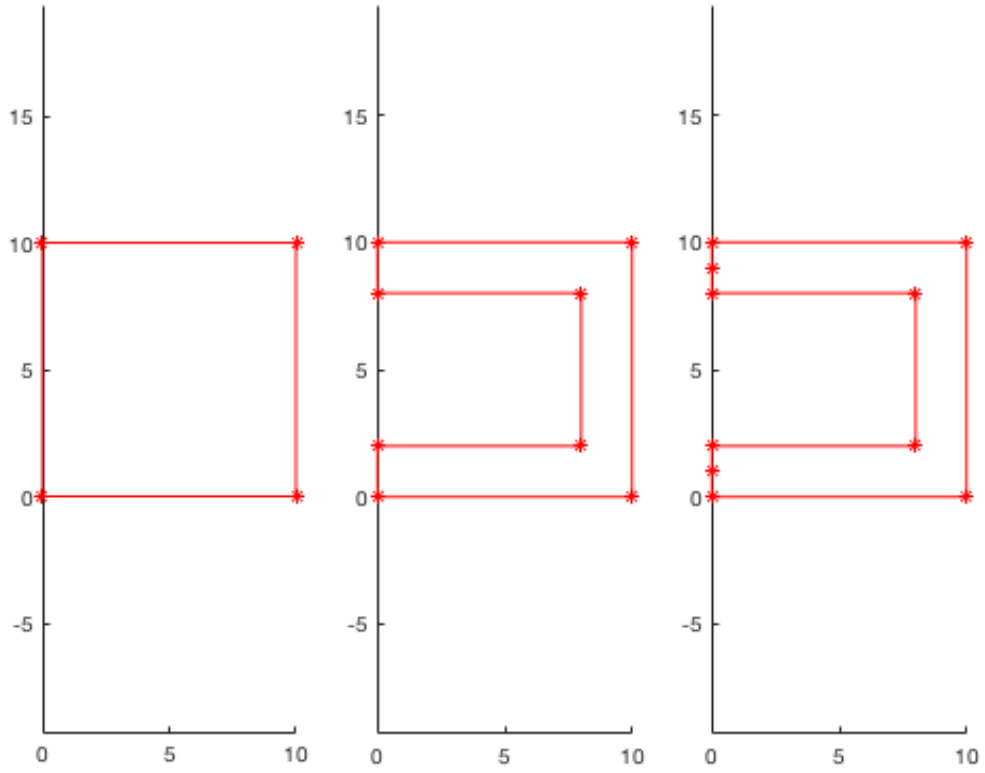
```
VLFLplotlight(1,0.7);
```



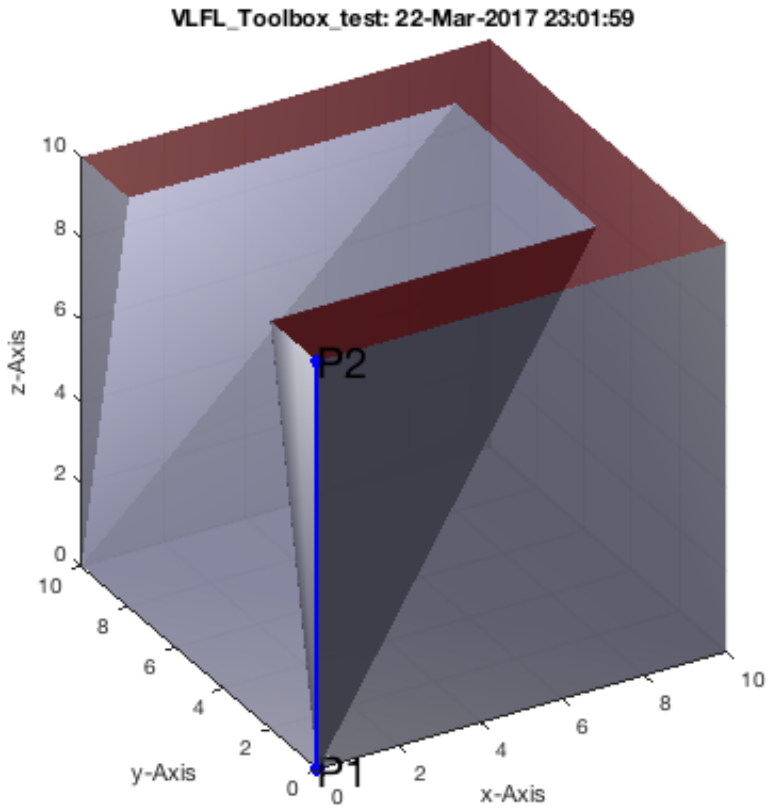

## 5. Solid surface generation for polygons with a small number of points

For polygon of only a few point, typically, the user has clear expectations about the final result of the solid.

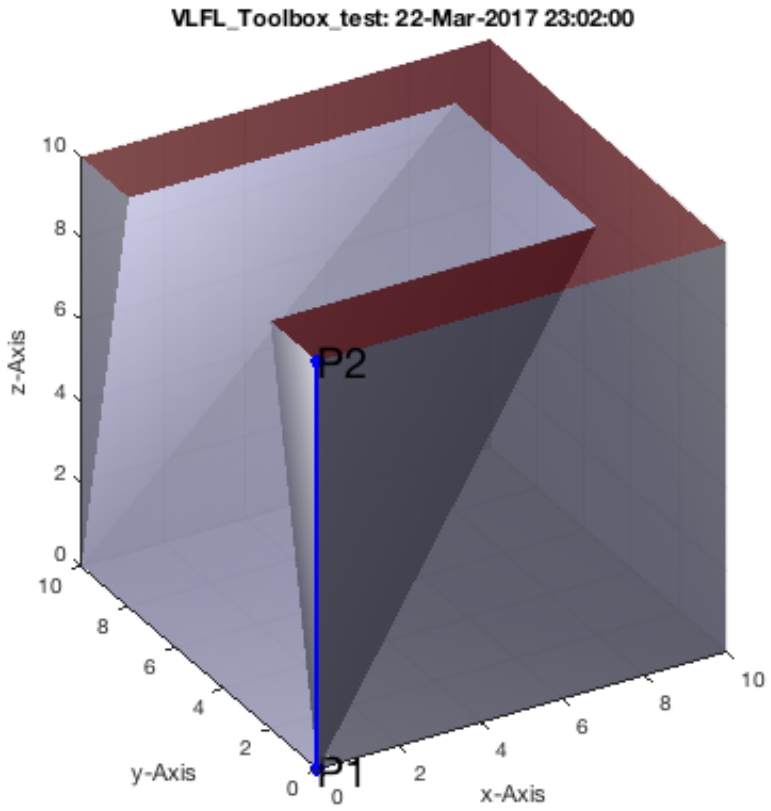
```
SGfigure;
PLU0=[0 0;10 0; 10 10; 0 10; 0 0];
PLU1=[0 0;10 0; 10 10; 0 10; 0 8; 8 8; 8 2; 0 2; 0 0];
PLU2=[0 0;10 0; 10 10; 0 10; 0 9; 0 8; 8 8; 8 2; 0 2; 0 1; 0 0];
subplot(1,3,1); CPLplot(PLU0);  view(0,90); axis equal;
subplot(1,3,2); CPLplot(PLU1);  view(0,90); axis equal
subplot(1,3,3); CPLplot(PLU2);  view(0,90); axis equal
```



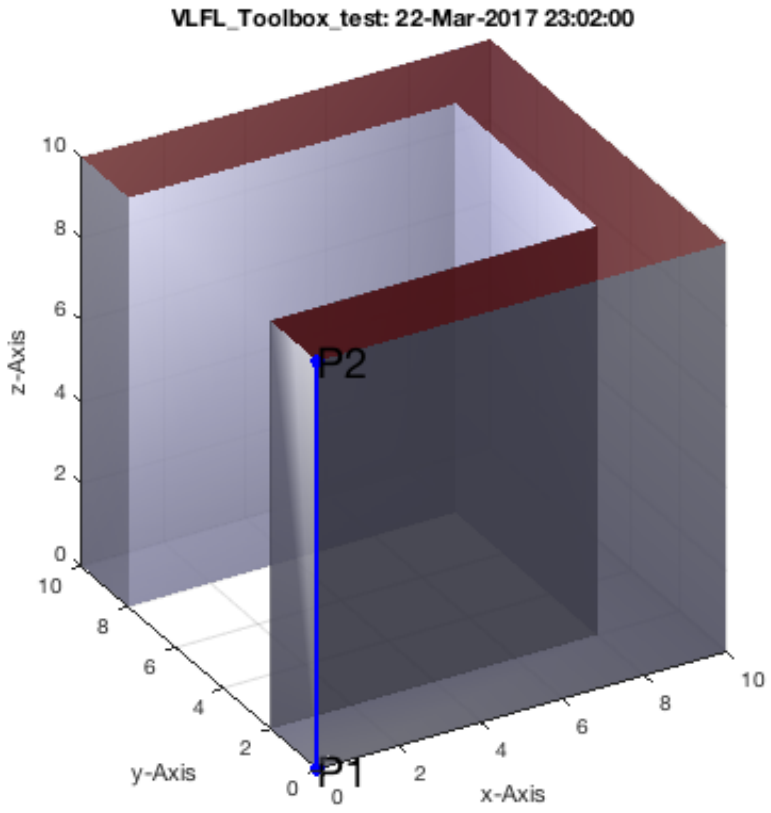

```
SGof2CPLz(PLU0,PLU2,10); VLFLplotlight(1,0.7);
```




```
SGof2CPLz(PLU0,PLU1,10); VLFLplotlight(1,0.7);
```



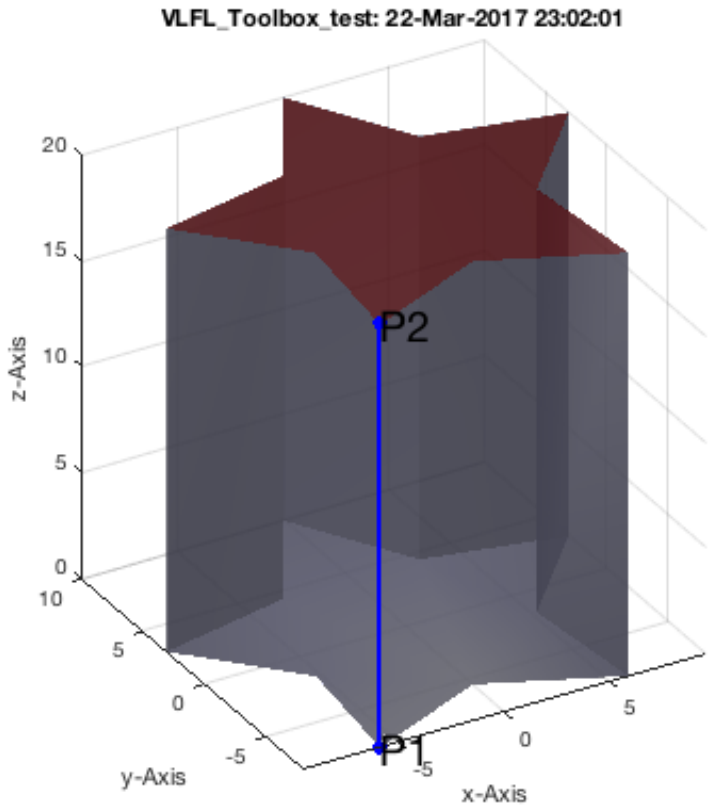

```
SGof2CPLz(PLU1,PLU2,10); VLFLplotlight(1,0.7);
```




## 6. Two identical contours with (strictly) monotonic increasing point-center angle

In case of two identical polygons that are just shifted or rotated, the default values 'length' and 'miny' work almost always perfectly.

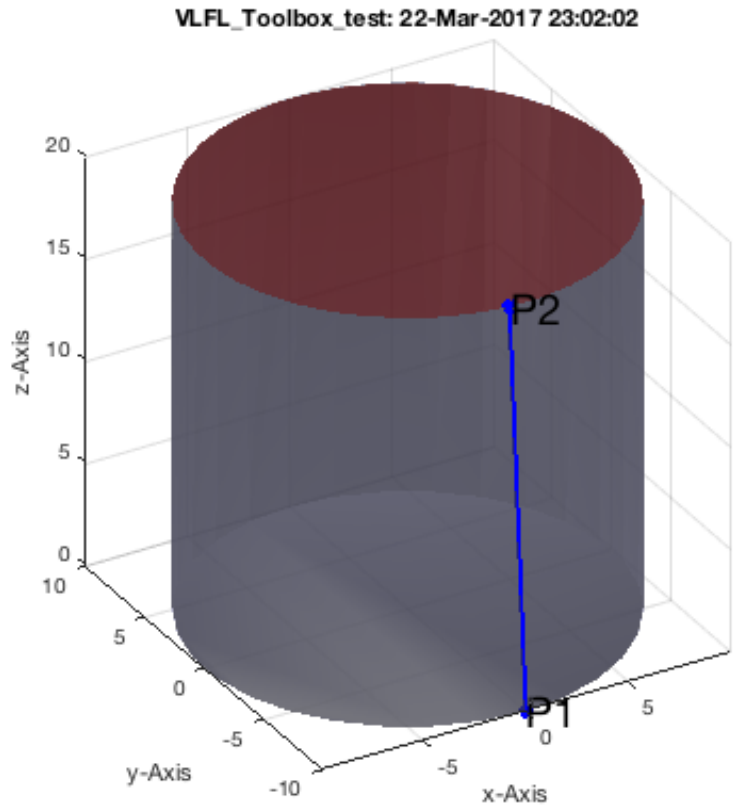
```
SGof2CPLz(PLstar(10,10),PLstar(10,10),20); VLFLplotlight(1,0.7);
```



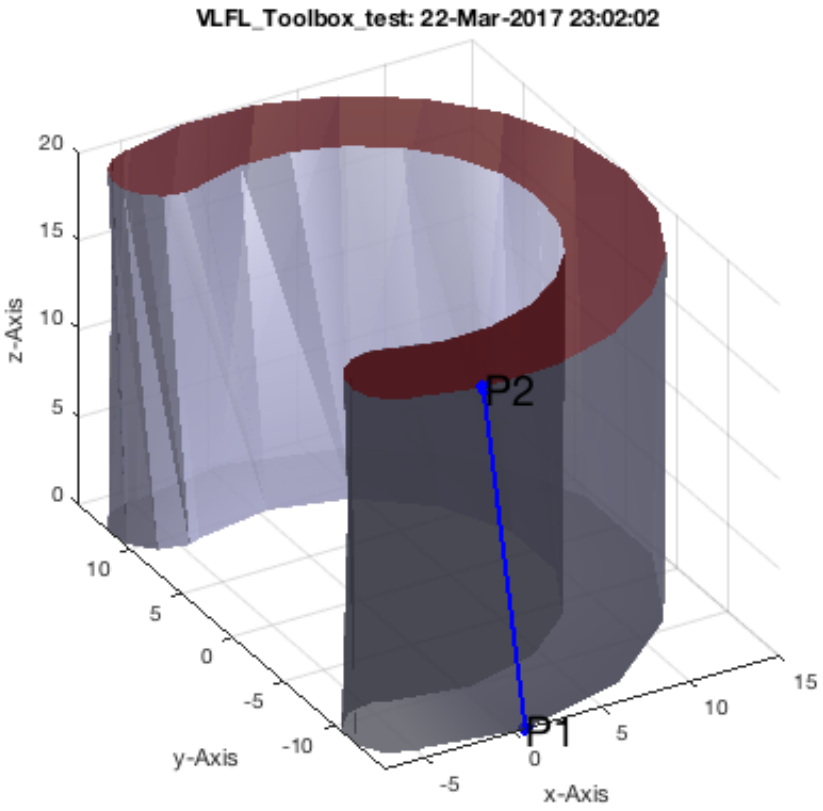

## 7. Two contours of the same shape with different number of points

In case that there are two identically shaped contours but with a different number of points, 'number' would be a solution but the default values 'length' and 'miny' work almost always perfectly. Even in the case of u-shaped contour.

```
SGof2CPLz(PLcircle(10,30),PLcircle(10,40),20);
VLFLplotlight(1,0.7);
```



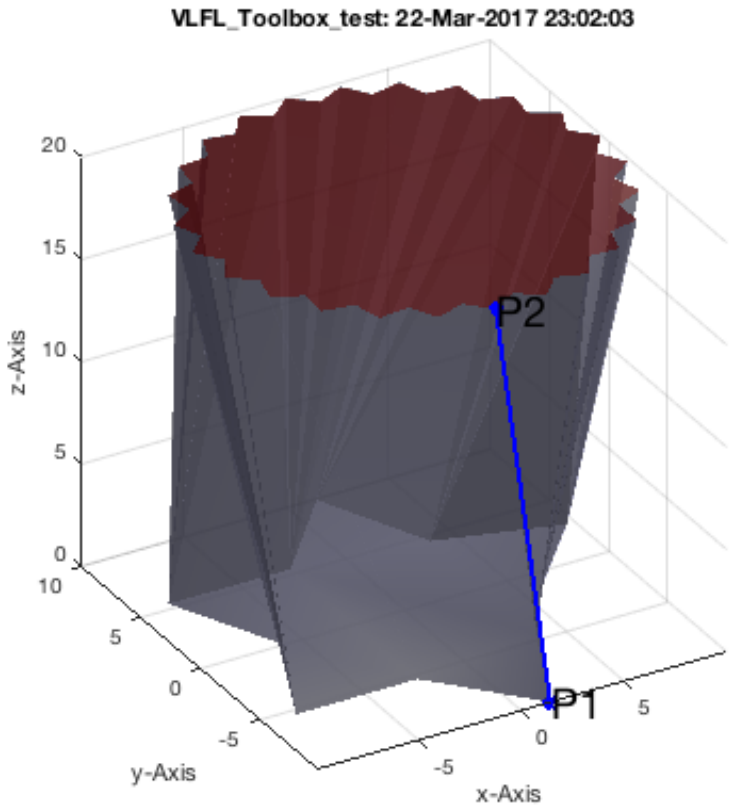

```
SGof2CPLz(PLkidney(10,15,pi/0.8,10),PLkidney(10,15,pi/0.8,15),20);
VLFLplotlight(1,0.7);
```



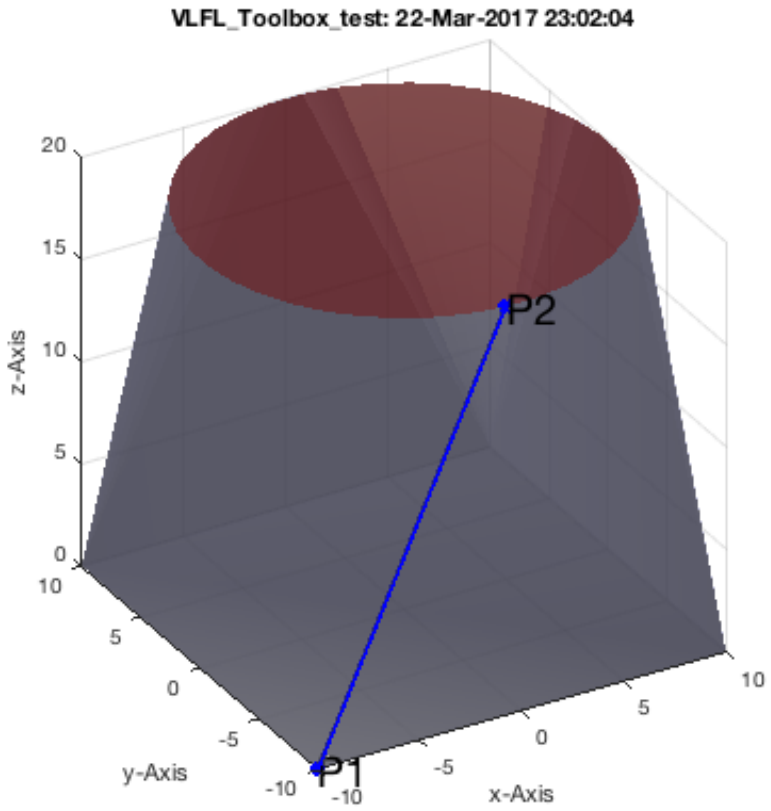

## 8. Two contours of the similar shape with different number of points

In case that there are two similar not indentical contours and with a different number of points, again the default values 'length' and 'miny' work almost always perfectly. Even in the case of u-shaped contour.

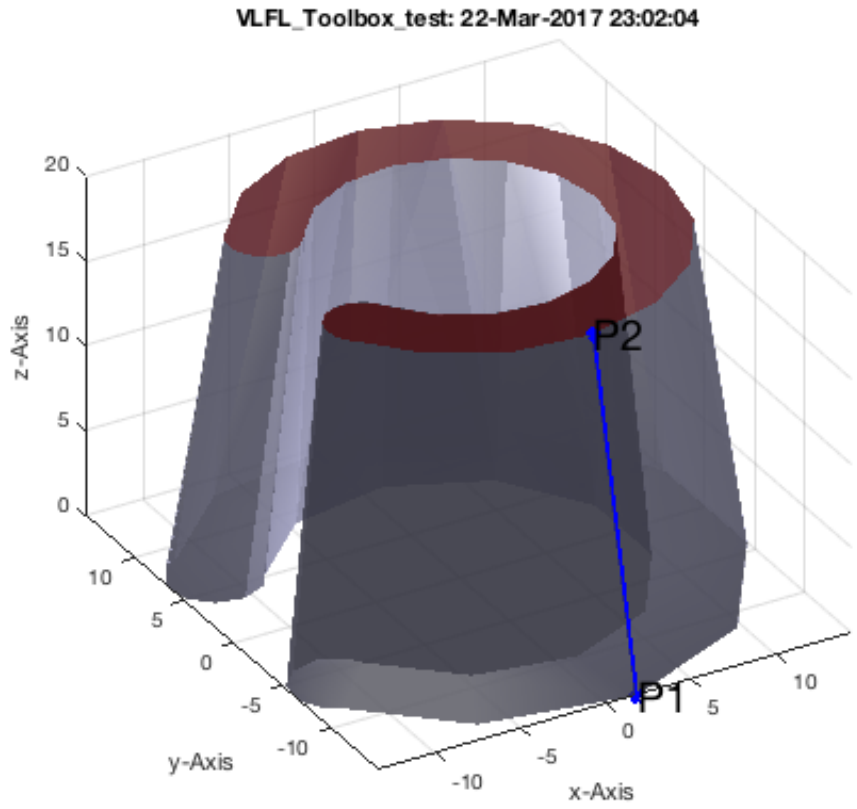
```
SGof2CPLz(PLstar(10,11),PLstar(10,50),20); VLFLplotlight(1,0.7);
```




```
SGof2CPLz(PLcircle(10*sqrt(2),4),PLcircle(10,40),20);
VLFLplotlight(1,0.7);
```




```
SGof2CPLz(PLkidney(10,15,pi/0.6,10),PLkidney(8,12,pi/0.6,15),20);
VLFLplotlight(1,0.7);
```

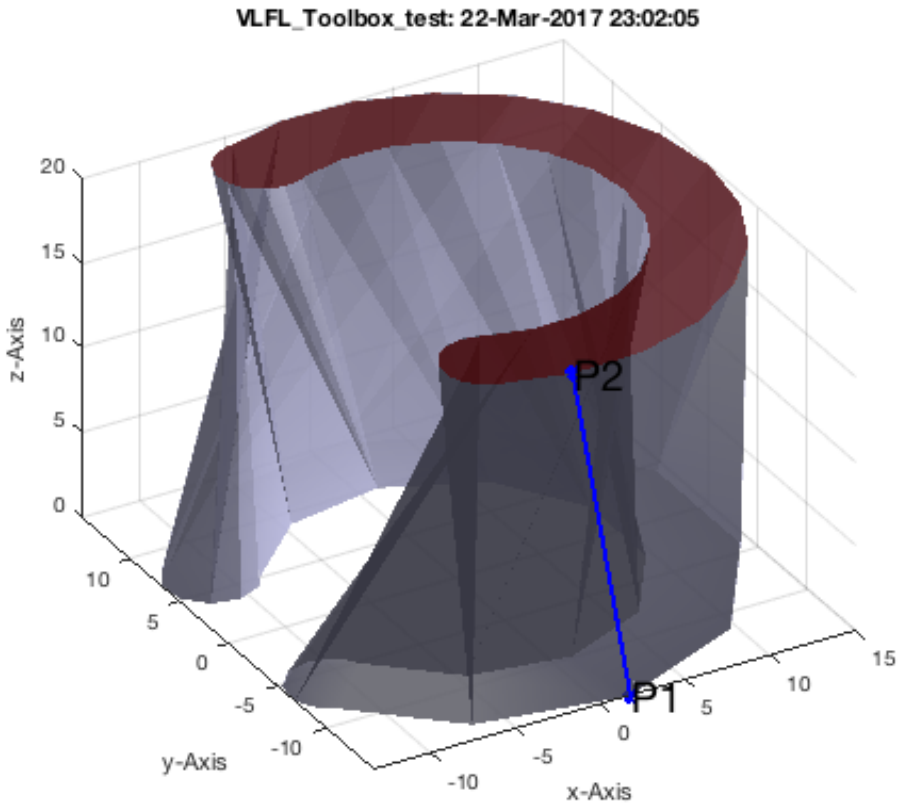



In this case we see that the kidney has different size but the same bending angle of pi/.6.

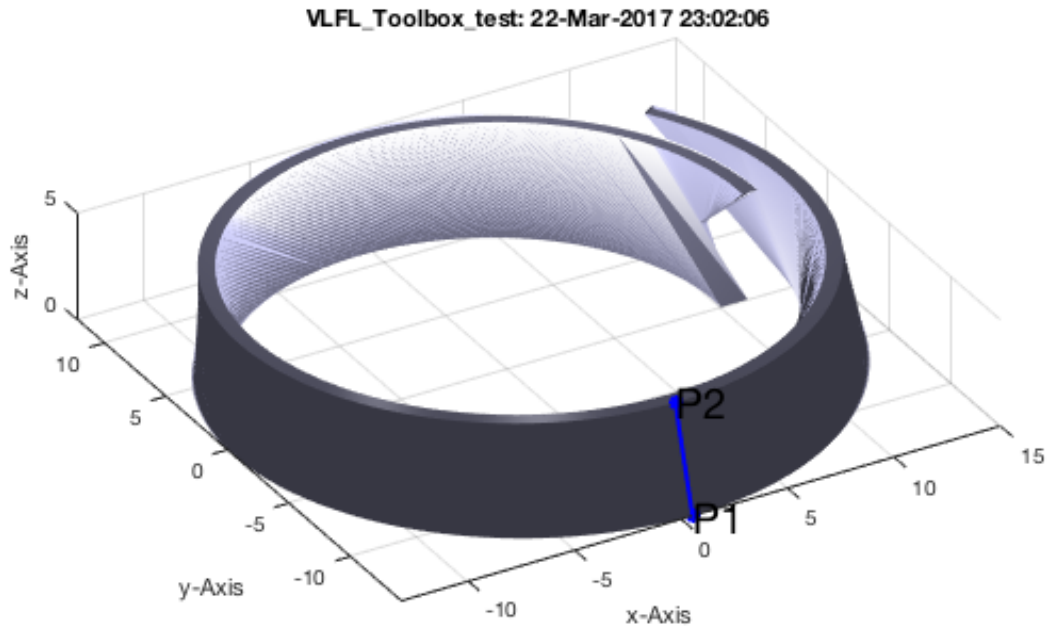
## 9. Two contours of the similar shape with different sum of boundary bending angle sum

In case that the boundary bending angle is greated than 2*pi (360 degree), the assignment by length and the turning strategy of miny is not the best anymore.

```
SGof2CPLz(PLkidney(10,15,pi/0.6,10),PLkidney(10,15,pi/0.8,15),20);
VLFLplotlight(1,0.7);
```

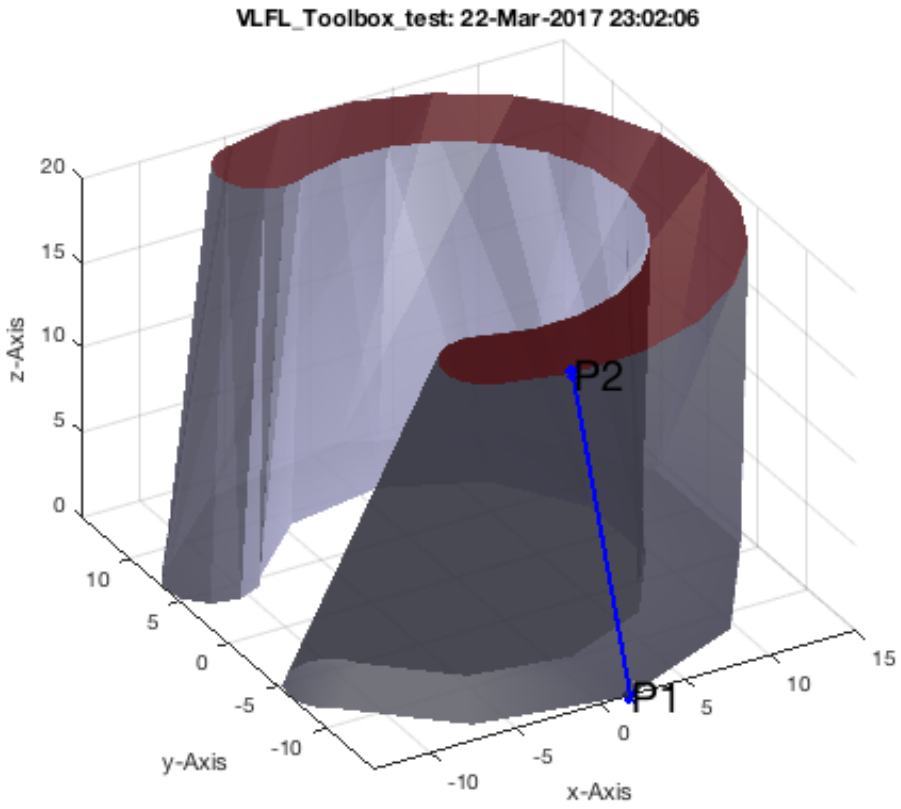



```
SGof2CPLz(CPLspiral(10,15,2*pi),CPLspiral(11,14,2.2*pi),5);
VLFLplotlight (1,1);
```

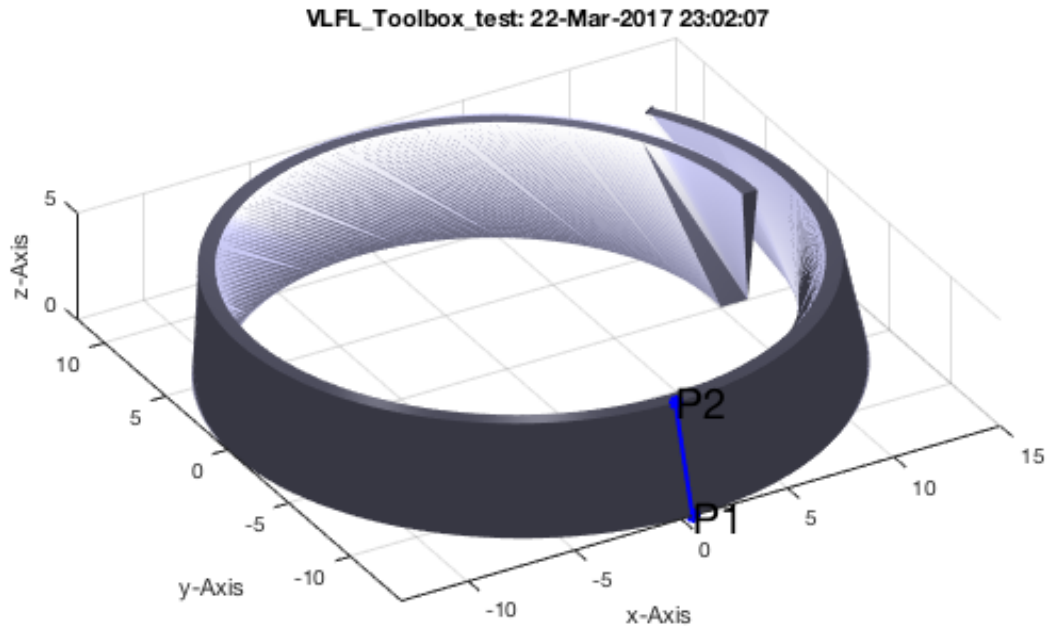



In this case we see malformed solids.

```
SGof2CPLz(PLkidney(10,15,pi/0.6,10),PLkidney(10,15,pi/0.8,15),20,'angle');
VLFLplotlight(1,0.7);
```

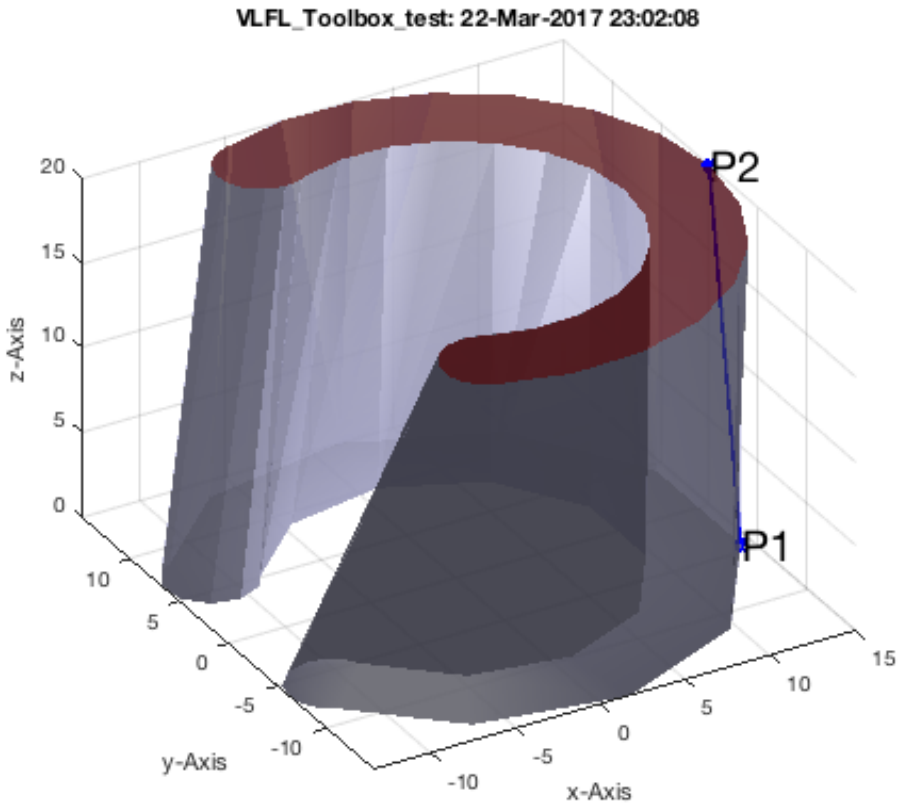



```
SGof2CPLz(CPLspiral(10,15,2*pi),CPLspiral(11,14,2.2*pi),5,'angle');
VLFLplotlight (1,1);
```

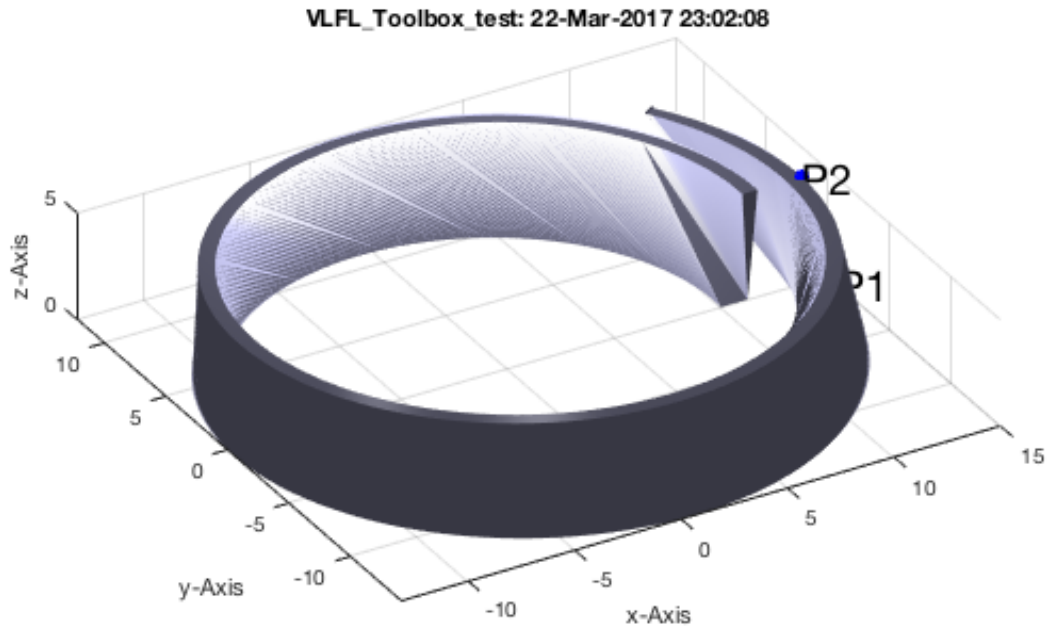



By using 'angle' instead of 'length', the solids are more what we expected.

```
SGof2CPLz(PLkidney(10,15,pi/0.6,10),PLkidney(10,15,pi/0.8,15),20,'angle','rot');
VLFLplotlight(1,0.7);
```

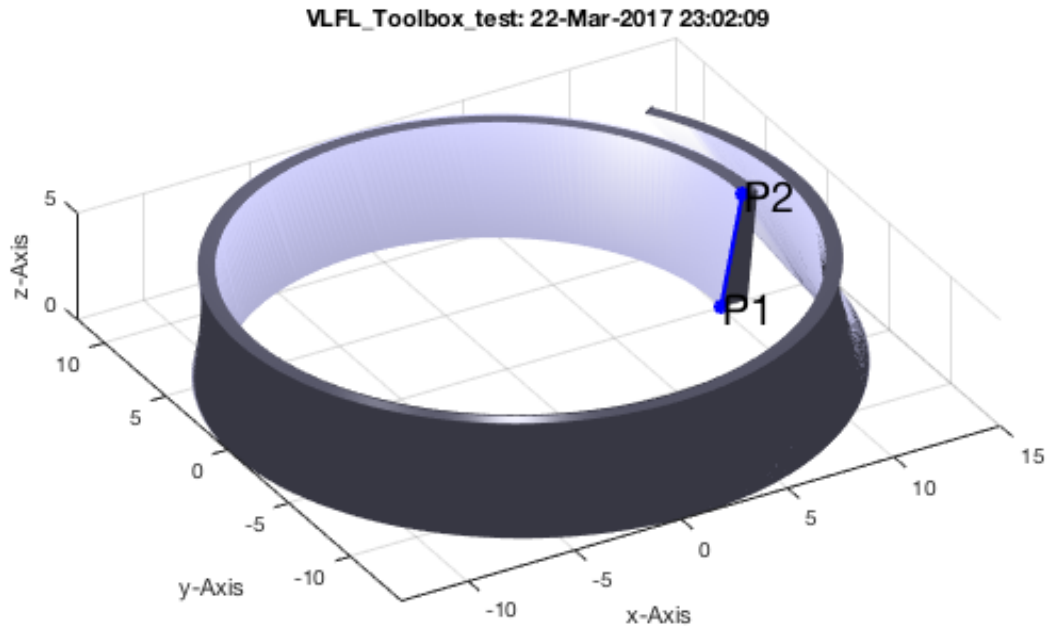



```
SGof2CPLz(CPLspiral(10,15,2*pi),CPLspiral(11,14,2.2*pi),5,'angle','rot');
VLFLplotlight (1,1);
```




By using 'angle' instead of 'length' AND an explicitly given 1st point assignment, the best result can be achieved.

```
PLA=CPLspiral(10,15,2*pi); PLB=CPLspiral(11,14,2.2*pi);
SGof2CPLz(PLA,PLB,5,'angle',[size(PLA,1) size(PLB,1)]); VLFLplotlight (1,1);
```




By using 'angle' instead of 'length' AND 'rot' instead of 'miny', the solids are almost what we expected.

## 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 23:02:12 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 2015-10-03*
- _____, executed and published on 64 Bit PC using Windows with Matlab 2015a on 2015-xx-xx_

*Published with MATLAB® R2016b*