Tutorial 01: First Steps Using the VLFL-Toolbox for Solid Object Design

2014-11-17: 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)

Motivation for this tutorial: (Originally SolidGeometry 1.1 required)

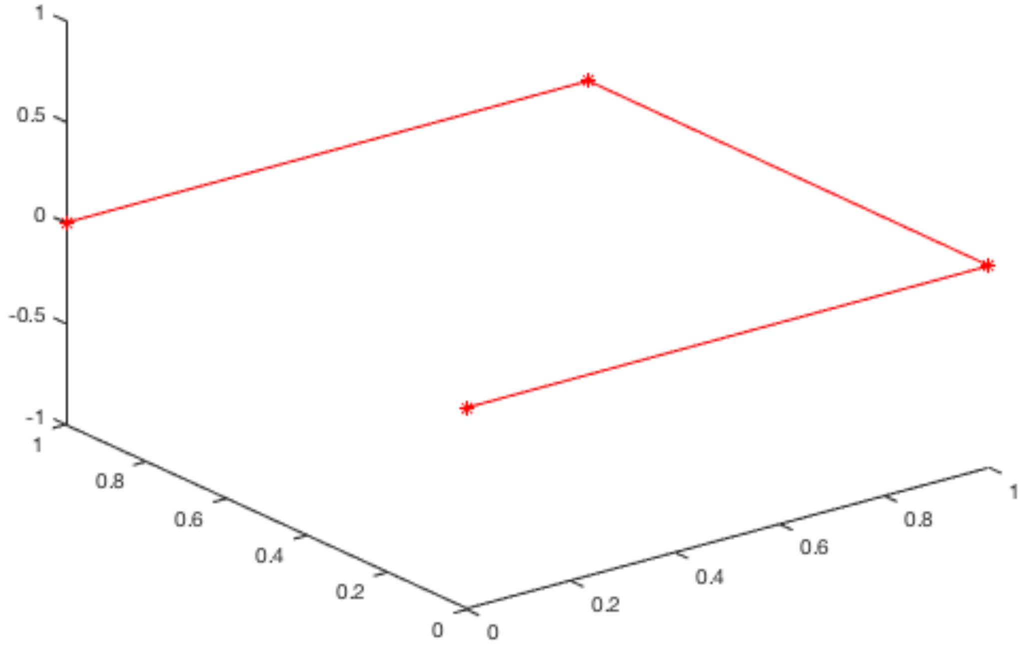
1. Creating and visualizing a simple base-contour by four 2D-points

A point list is a nx2 array. The number n is the number of 2D coordinate points [x y]. In general, such a point list can be the basis for designing a boundary surface model. We start with some simple functions to display polygons:

- **PLplot** to plot in 3D a nx2 point list (PL).

```
PL=[ 0 0; 1 0; 1 1; 0 1]
PLplot(PL);
```

```
PL =

     0     0
     1     0
     1     1
     0     1
```



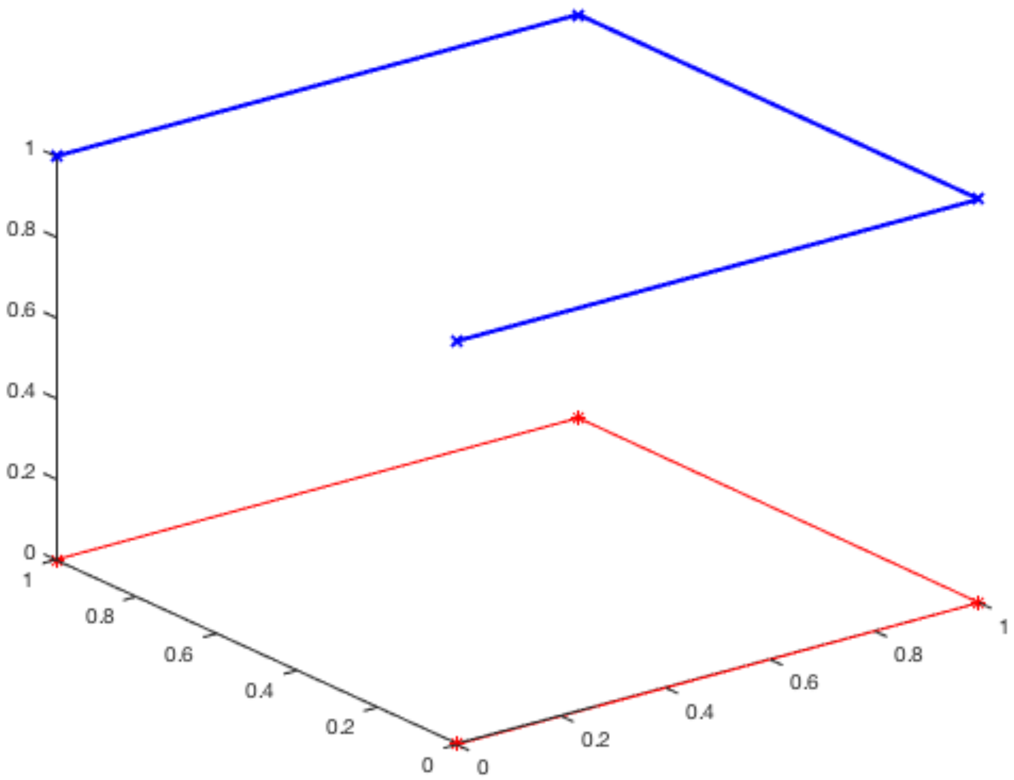
2. Append a 3rd coordinate column to get a vertex list

A vertex list is a nx3 array. The number n is the number of 3D coordinate points [x y z]. In fact, the point list can be transfered into a vertex list by appending a third column containing the z-coordinate such as zero or another z-coordinate.

- **VLaddz** for converting a point list (PL) into a vertex list (VL) by adding a 3rd column for the z-coordinate.
- **VLplot** for displaying in 3D a nx3 vertex list (VL).

```
VL=VLaddz(PL,1)
VLplot (VL,'bx-',2);
```

```
VL =

     0     0     1
     1     0     1
     1     1     1
     0     1     1
```

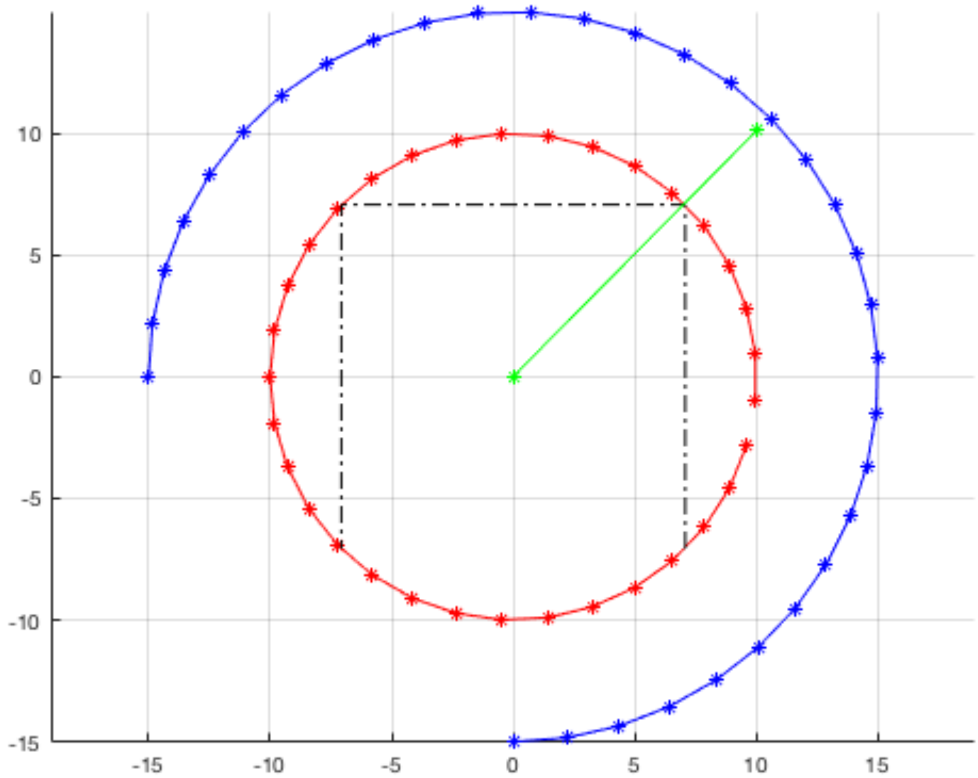


3. Predefined functions for the generation of often used planar polygons (PL)

For some often used contours, there are predefined functions that generate a nx2 coordinate point list (PL).

- **PLcircle** for a polygon with n points.
- **PLcircseg** for a circle segment with n points.
- **PLevolvente** for an evolvente as contour.

```
close all;
PL=PLcircle (10,33);                          % Radius 10, 33 points
PLplot(PL); view (0,90); axis equal; grid on;
PL=PLcircle (10,4);                          % Radius 10, 4 points
PLplot(PL,'k-.'); view (0,90); axis equal; grid on;
PL=PLcircseg (15,33,-pi/2,+pi);               % Radius 15, 33 points, 270 degree
PLplot(PL,'b-*'); view (0,90); axis equal; grid on;
PL=PLevolvente (10,pi/180*270,33)';           % Radius 10, 33 points, 270 degree
PLplot(PL,'g-*'); view (0,90); axis equal; grid on;
```

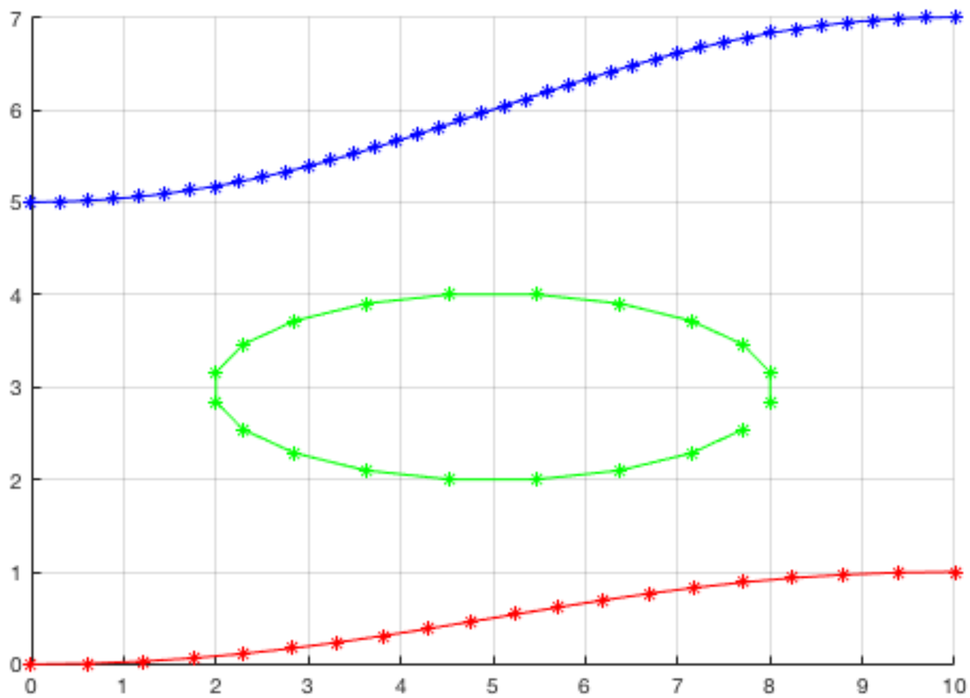


4. More predefined functions for planar polygons in 3D (VL)

Some functions for planar polygons create already 3D points (vertices) and the result of such a function is a vertex list (VL).

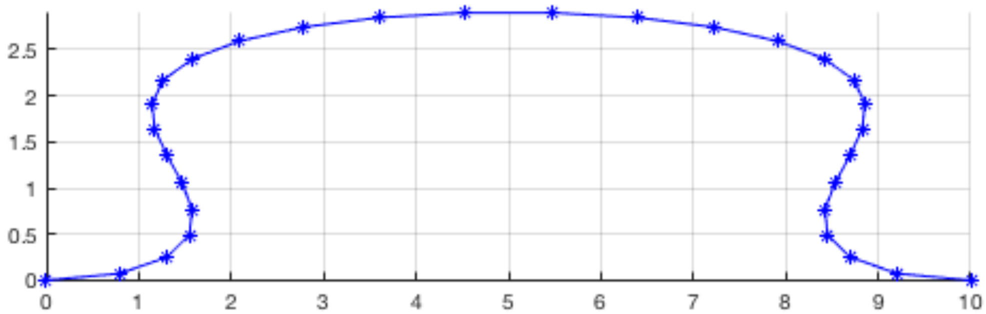
- **VLpolygon** to generate elliptic contours.
- **VLBezier4P** to generate a Bezier-curve using 4 points.
- **VLBezierC** to generate a Bezier-curve using as many points as possible.
- **VLremstraightCVL** to remove obsolete points on straight lines.

```
close all;
VL=VLpolygon(20,3,1,[5 3 0]);
VLplot (VL,'g*-'); show, axis equal, view (0,90); grid on; hold on;
VL=VLBezier4P([0 0 0],[4 0 0],[6 1 0],[10 1 0],20);
VLplot (VL,'r*-'); show, axis equal, view (0,90);
VL=VLBezierC([0 5; 4 5; 6 7; 10 7],40);
VLplot (VL,'b*-'); show, axis equal, view (0,90); grid on
```



- **VLBeziernoose** to generate a Bezier-curve spring-element.

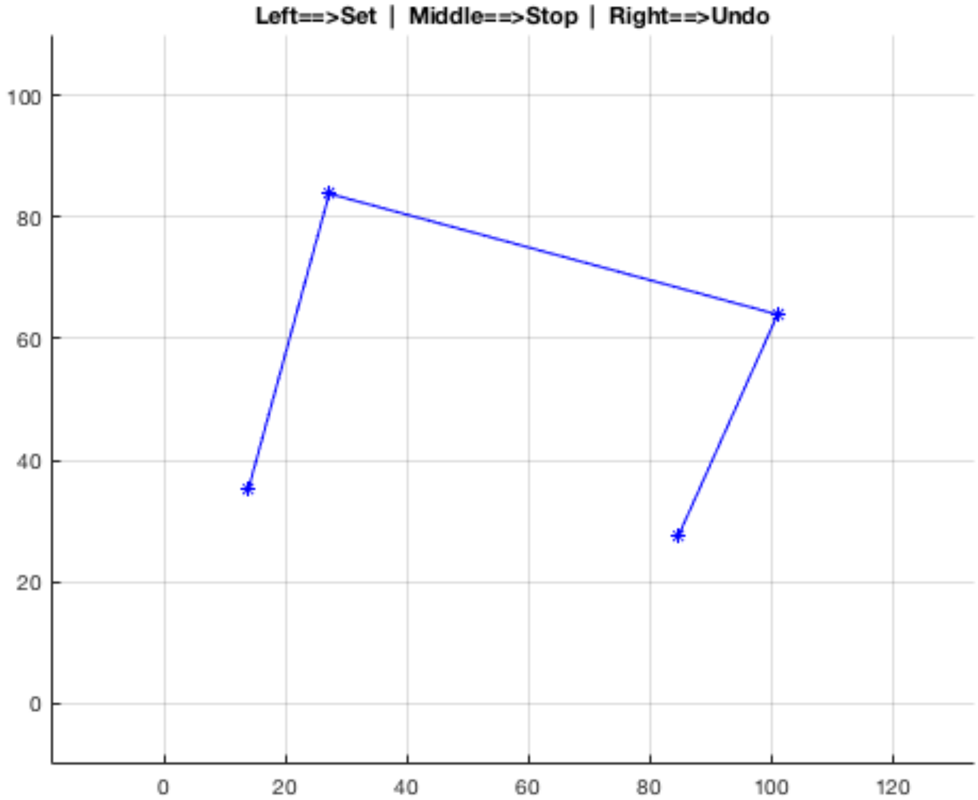
```
close all;
VL=VLBeziernoose(10,2,3,3,30);
VLplot (VL,'b*-'); show, axis equal, view (0,90); grid on
```



- **VLui** as an user interface to enter points by mouse clicks.

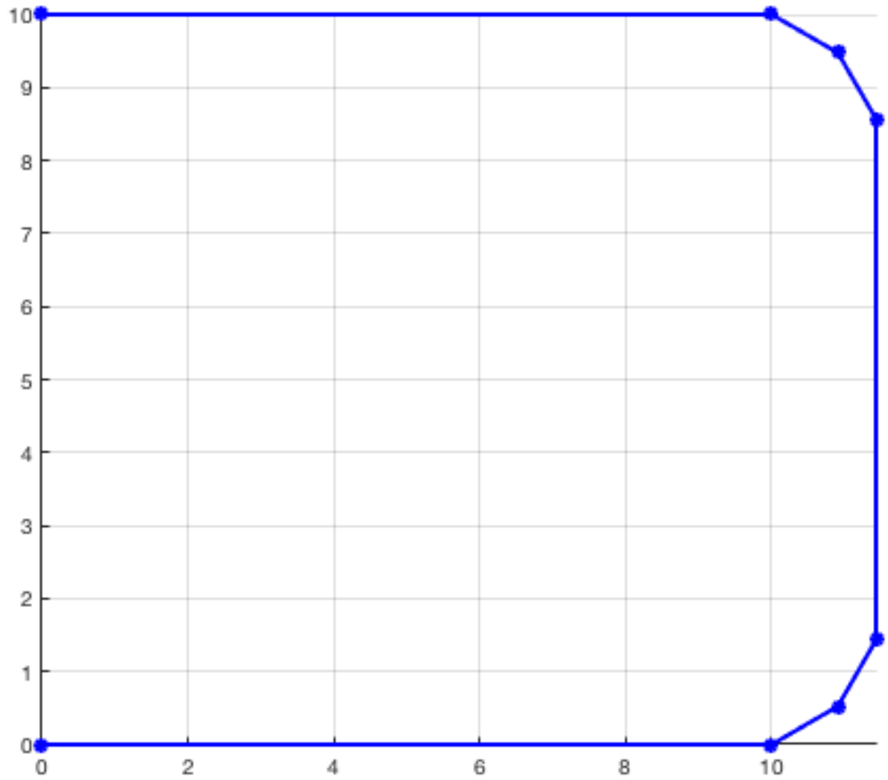
```
close all;
VL=VLui
VLplot (VL,'b*-'); show, axis equal, view (0,90); grid on
```

```
VL =

   13.9000   35.3000         0
   27.2000   83.9000         0
  101.0000   64.0000         0
   84.7000   27.6000         0
```



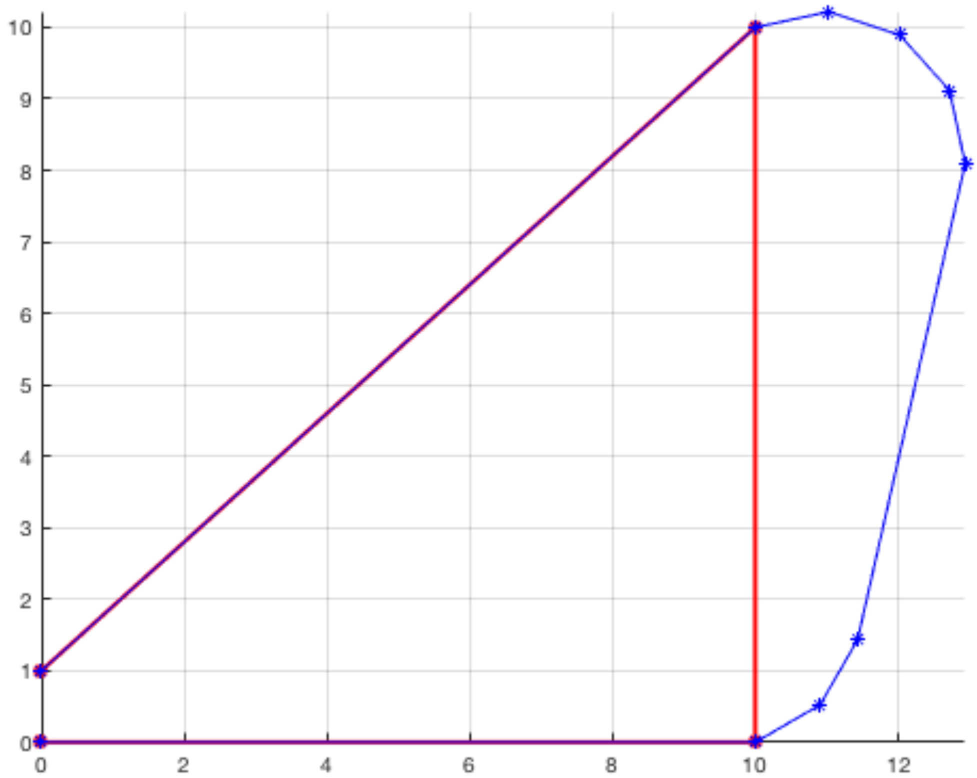
- **VLRadius4P** for inserting points to generate radial curves instead of corners.

```
close all
VL=VLRadius4P([0 0 0],[10 0 0], [10 10 0], [0 10 0], pi/6, 2);
VL=VLremstraightCVL (VL);
VLplot (VL,'b*-',2); show, axis equal, view (0,90); grid on
```



- **VLRadiusC** for inserting points to generate radial curves instead of corners.

```
close all
VLORG=[[0 0 0];[10 0 0];[10 10 0];[0 1 0]];
VLplot (VLORG,'r*-',2); show, axis equal, view (0,90); grid on; hold on
VL=VLRadiusC(VLORG, pi/6, 2);
VL=VLremstraightCVL (VL);
VLplot (VL,'b*-',1); show, axis equal, view (0,90); grid on
```



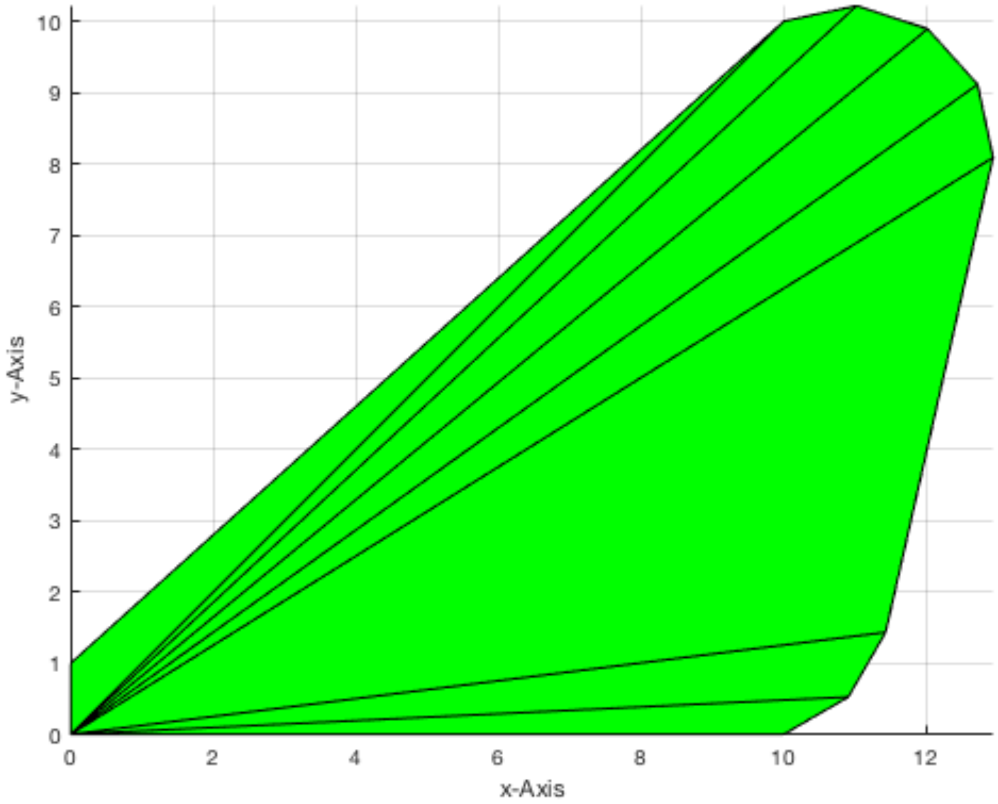
5. Calculation of the surface of a convex polygon

If we have a closed convex polygon, it is possible to generate a surface desciption by a facet list (FL) describing triangle facets. This is called tesselation of the closed polygon/surface. For closed convex polygons, the simplest form are facets from the 1st to the 2nd and 3rd points [1 2 3], then from the 1st to the 3rd and 4th [1 3 4], and so on. The facet list (FL) is therefor a nx3 index list to the point list or vertex list (VL). To use this concept we have some basic functions. For non convex functions we see later some more solutions.

- **FLofVL** to generate the facet list (FL) for a **convex** polygon.
- **VLFLplot** to plot a surface given by a vertex list (VL) and a facet list (FL).

```
close all
FL=FLofVL(VL)
% FL=FLofCVL(VL)
VLFLplot (VL,FL,'g'); axis equal; view (0,90); grid on
% view (-30,30);
```

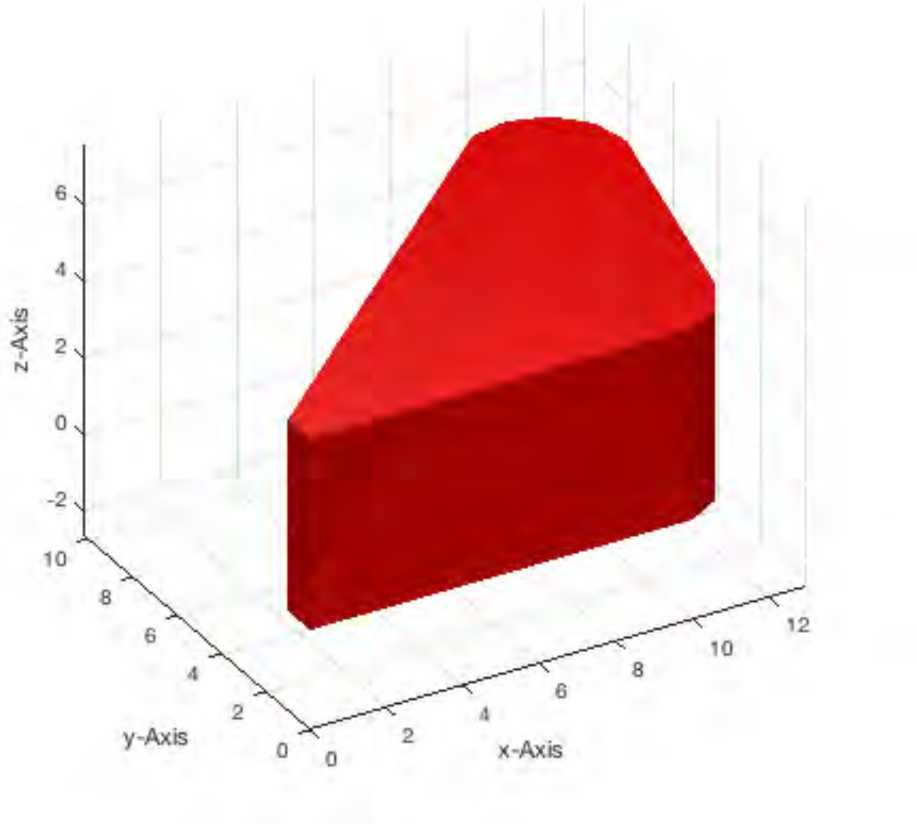
```
FL =

     1     2     3
     1     3     4
     1     4     5
     1     5     6
     1     6     7
     1     7     8
     1     8     9
     1     9    10
```



6. Calculation of all surfaces of convex polygon-based 2.5D-solid-volumes

- **VLFLofPLz** to extrude a convex polygon to a solid volume.
- **VLFLplotlight** to adjust the rendering parameter of the current graphic.

```
close all
[VL,FL]=VLFLofPLz (VL(:,1:2),5);
VLFLplot (VL,FL); axis equal; view (-30,30); grid on
VLFLplotlight(1,0.9); show;
```

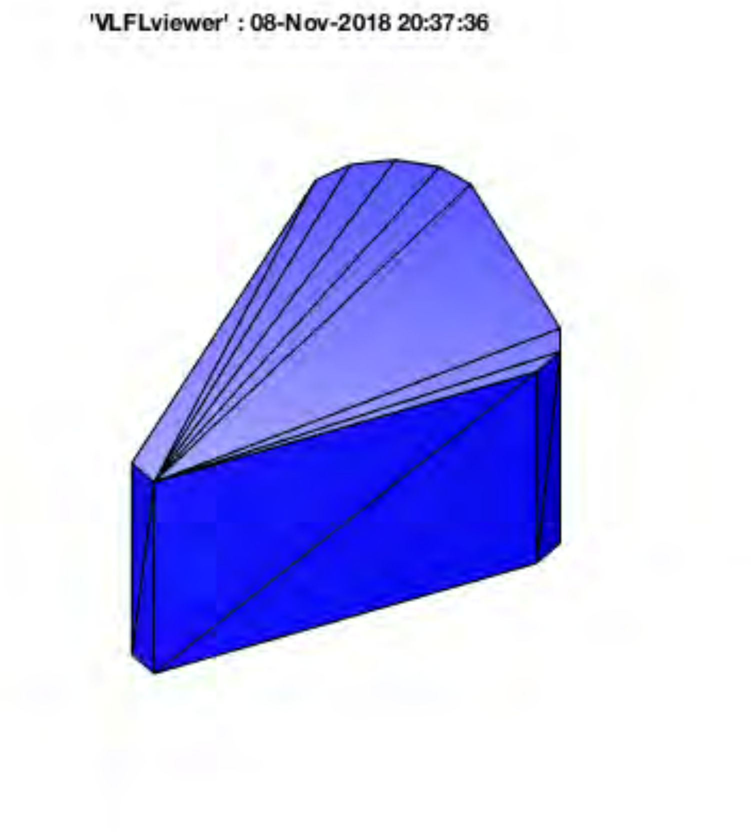


7. Graphical user interface for STL import, export, and viewing

Currently tested only for OSX (Apple Macintosh), there is also a graphical user interface available for displaying the surface objects, to import STL-Files and to export STL-Files. In this example, the tool is just introduced, to explain the capabilities to implement also graphical design tools for solid object modeling.

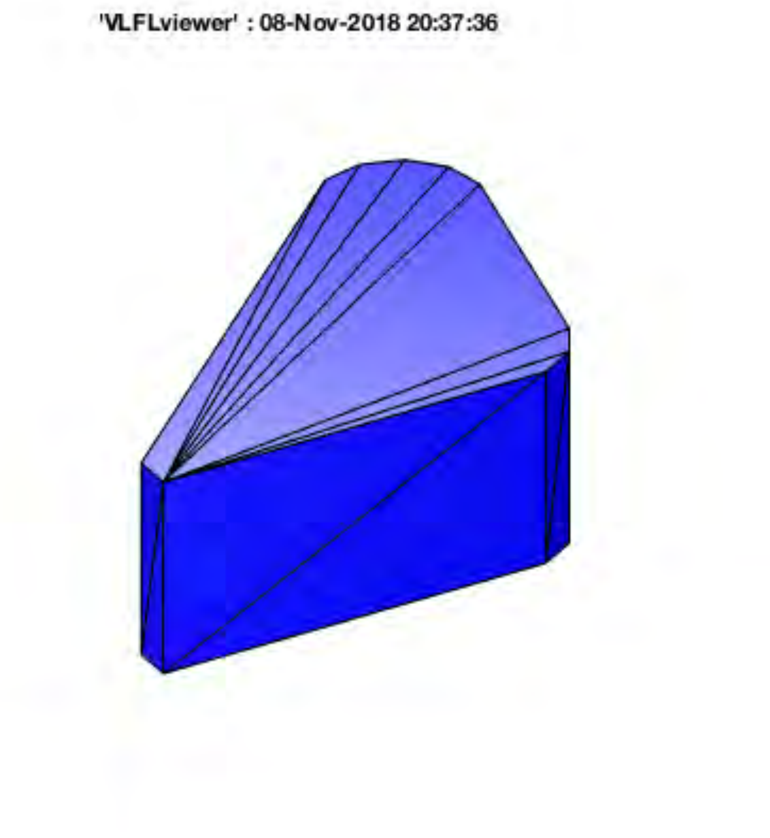
- **VLFLviewer** to show surface models, to import and to export STL-Files.

```
VLFLviewer (VL,FL,'b'); view (-30,30);
```




```
VLFLlicense
% * Tim Lueth, tested and compiled on OSX 10.7.5 with Matlab 2014b on 2014-11-18
% * Tim Lueth, executed and published on 64 Bit PC using Windows with Matlab 2014b on 2014-
11-18
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:37:37!
Executed 08-Nov-2018 20:37:39 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
simulink
video_and_image_blockset
===========================================================================================
==========
```



Published with MATLAB® R2018a

Tutorial 02: Using the VLFL-Toolbox for STL-File Export and Import

2014-11-18: 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

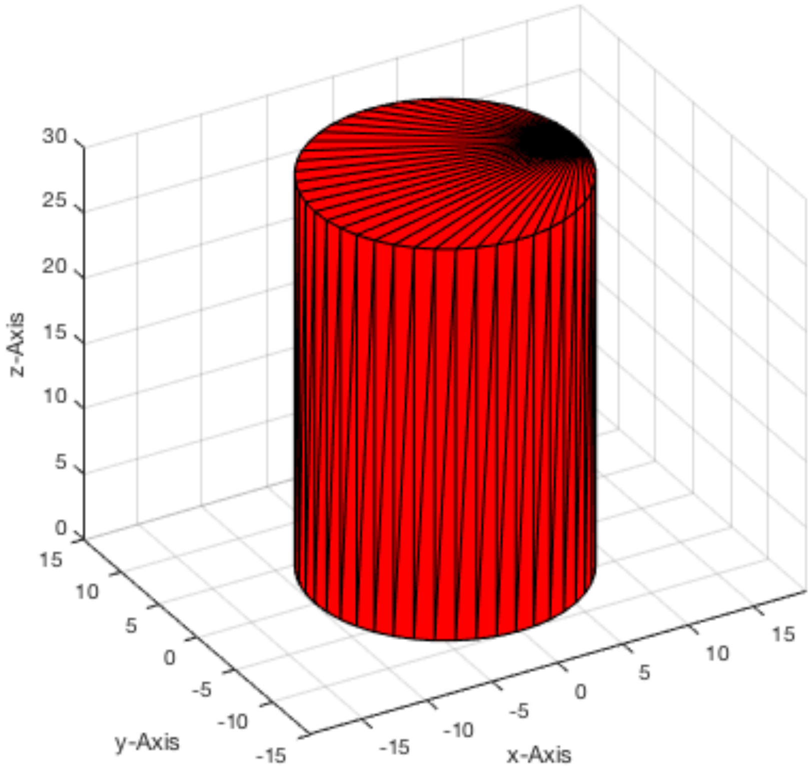
Motivation for this tutorial: (Originally SolidGeometry 1.1 required)

2. Import and export of STL-files in ASCII format and binary format

Often it is useful to import surface data fo solid volumes from STL-Files generated by other programs such as CATIA, ProEngineer, Solidworks etc. On the other hand we want to export our data for documentation, 3D-printing or the exchange with other users. The STL-File format is the most common file format for surface models. It supports ascii-text-format and binary formatted files. For export and import we need a couple of functions:

- **VLFLwriteSTL** for writing STL-files in ascii file format.
- **VLFLwriteSTLb** for writing STL-files in binary file format.

```
close all;
PL=PLcircle(10);
[VL,FL]=VLFLofPLz (PL,30);
VLFLplot(VL,FL); view (-30,30); grid on;
```




```
VLFLwriteSTL(VL,FL,'STL-ASCII','by My Name');
VLFLwriteSTLb(VL,FL,'STL-BINAR','by My Name');
```

```
WRITING STL FILE /Users/lueth/Desktop/Toolbox_test/STL-ASCII.STL in ASCII MODE
completed.
WRITING STL (90 vertices, 176 facets) FILE /Users/lueth/Desktop/Toolbox_test/STL-BINAR.STL
in BINARY MODE
completed.
```

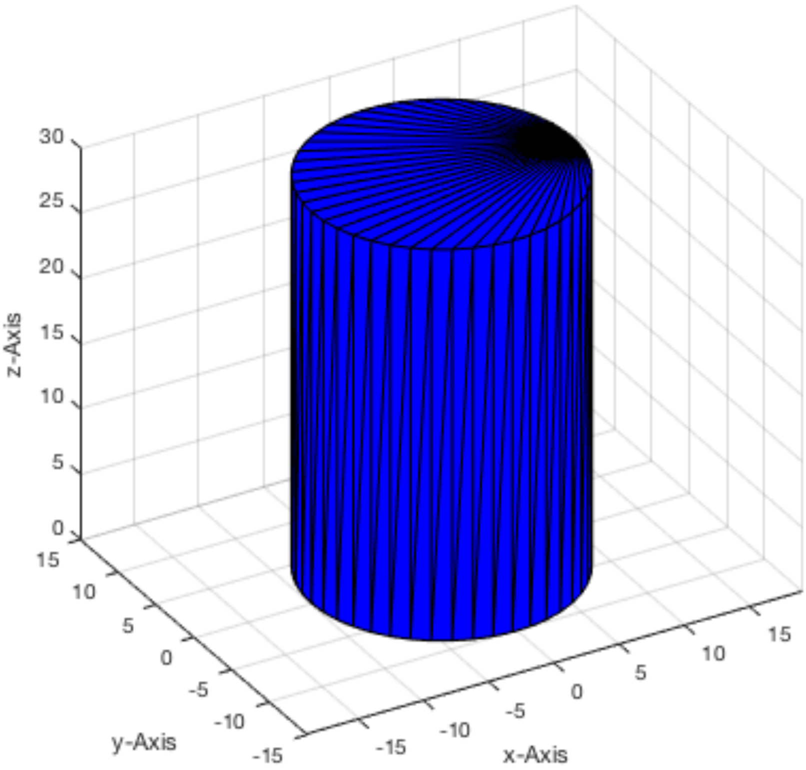
Similar it is possible to read the files in again

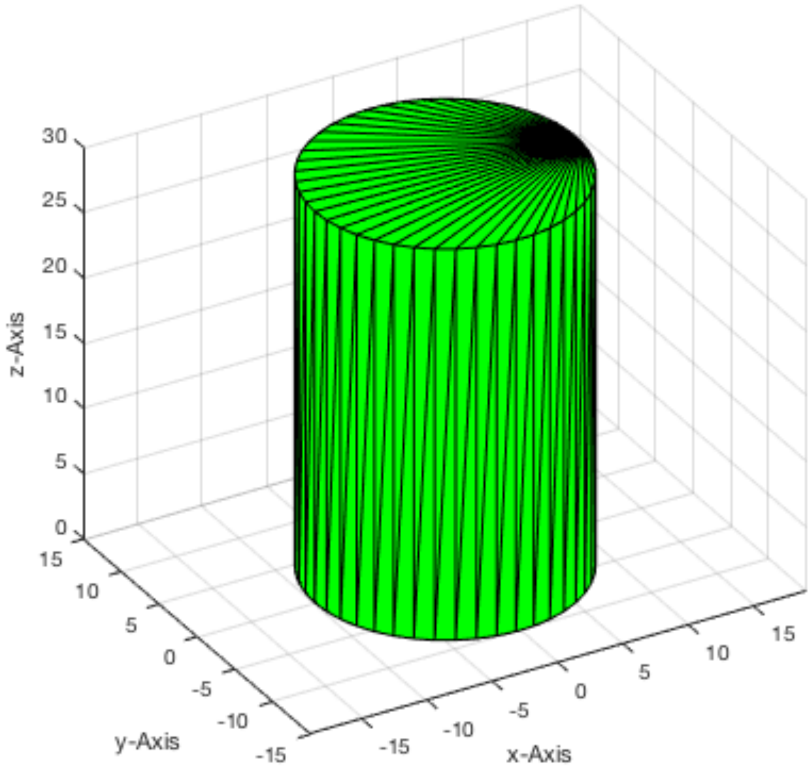
- **VLFLreadSTL** for importing STL-files in ascii file format.
- **VLFLreadSTLb** for importing STL-files in binary file format.

```
close all;
[VL,FL]=VLFLreadSTL ('STL-ASCII');
figure(1); VLFLplot(VL,FL,'b'); view (-30,30); grid on;
[VL,FL]=VLFLreadSTLb ('STL-BINAR');
figure(2); VLFLplot(VL,FL,'g'); view (-30,30); grid on;
```

```
LOADING ASCII STL-File: /Users/lueth/Desktop/Toolbox_test/STL-ASCII.STL scaling factor: 1
Processing 1234 lines:
Finishing solid AOI-LIB:"STL-ASCII by My Name" 08-Nov-2018 20:37:40 08-Nov-2018 20:37:40 LO
```

```
ADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/STL-BINAR.STL
Header: AOI-LIB:"STL-BINAR by My Name" 08-Nov-2018 20:37:40
Number of facets: 176
0..
```





3. Checking surface volume data and STL-files

Especially, when reading in STL-Files that are generated by other programs and libraries it makes sense to check the data quality. For that purpose there is a function that will be explained later in more detail. This function is called at the end of each STL import.

- **VLFLchecker** is used to analyze vertex list (VL) and facet list (FL)

```
VLFLchecker(VL,FL);      % Check the data structure

% There are some more procedures to view and analyze solid volumee data
```

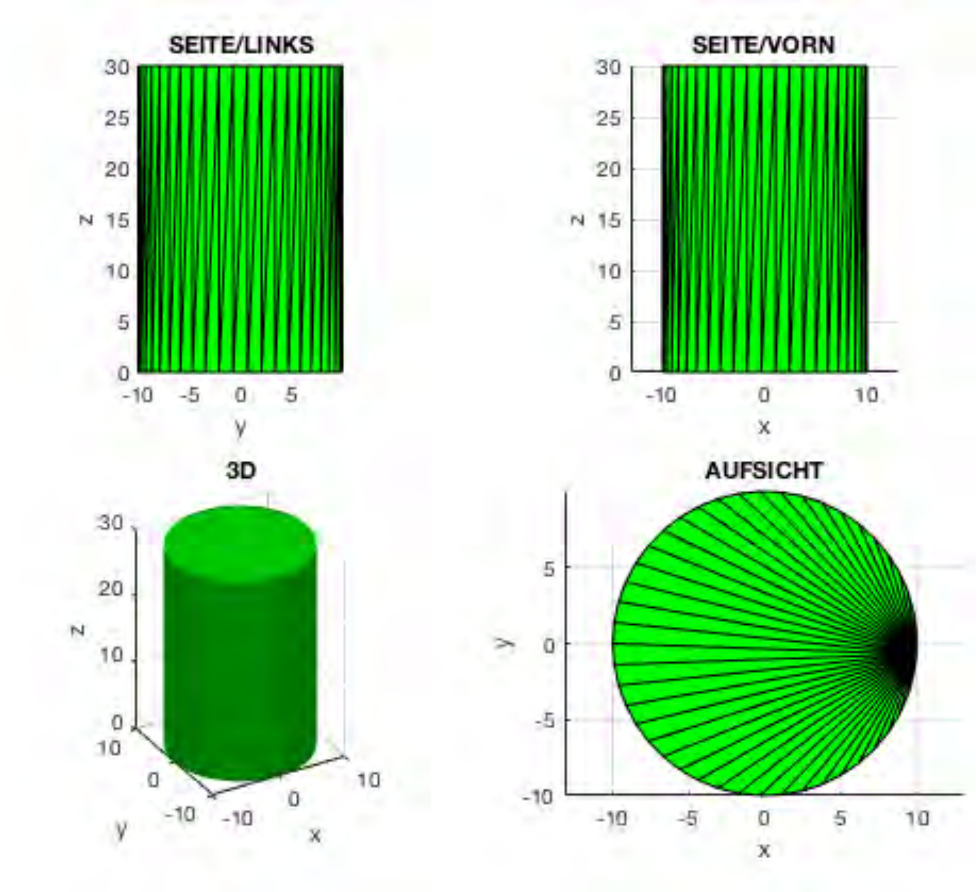
```
VLFLchecker: 90 vertices and 176 facets.
    0 FACET PROBLEMS DETECTED (ERRORS)
    0 VERTEX PROBLEMS DETECTED (OBSOLETE WARNING)
    0 EDGE PROBLEMS DETECTED (NON MANIFOLD WARNING)
    0 SOLID/EDGE PROBLEMS DETECTED (OPEN SOLID WARNING)
```

- **BBofVL** generates the bounding box dimensions of the solid
- **VLFLui** is simple user interface to open an STL file
- **VLFLminimize** eliminates doubles in VL and FL
- **VLFLnormf** calucates norm vector direction and size
- **VLFLplot4** figure with 4 subplots
- **VLFLselect** selected vertex list for a given facet list

- **VLFLseparate** find different independen objects in VL and FL
- **VLFLshort** remove unused vertices from VL
- **VLFLsurface** returns only vertex list and facet list for one surface
- **VLFLvertexfusion** shrinks vertex list by merging extremy near vertices

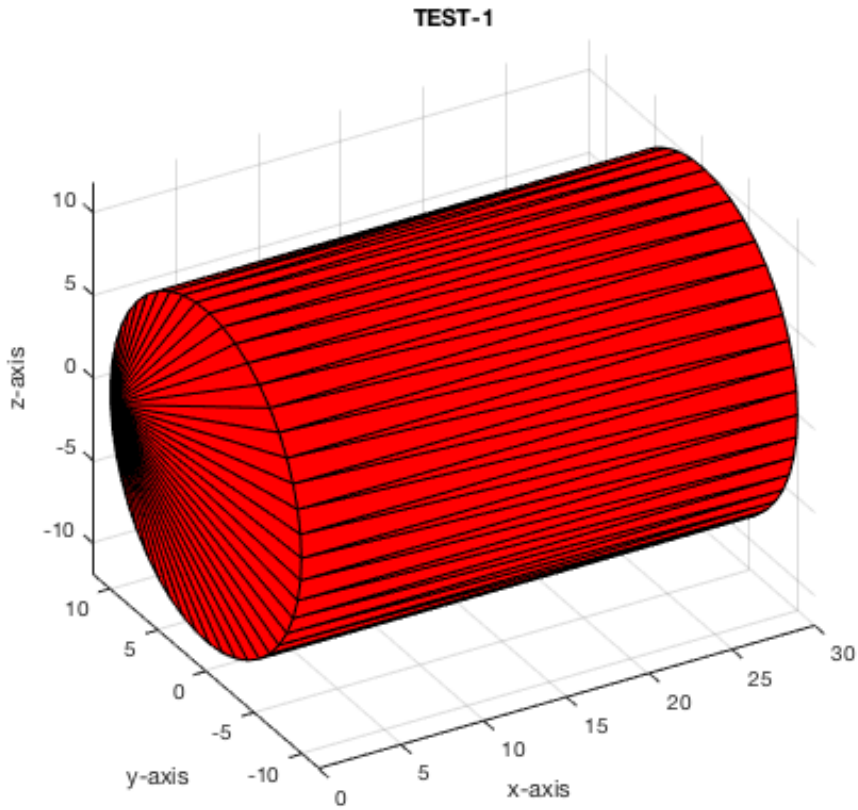
```
BBofVL(VL)
close all; VLFLplots4 (VL,FL,'g');
```

```
ans =

  -10.0000    9.9756   -9.9939    9.9939         0   30.0000
```




```
close all; VLFLseparate(VL,FL);
```

```
Analyzing 90 facets for separation z=[0.0mm|30.0mm]
Object TEST-1 with 176 facets

MVL =

     0   -10     0
```



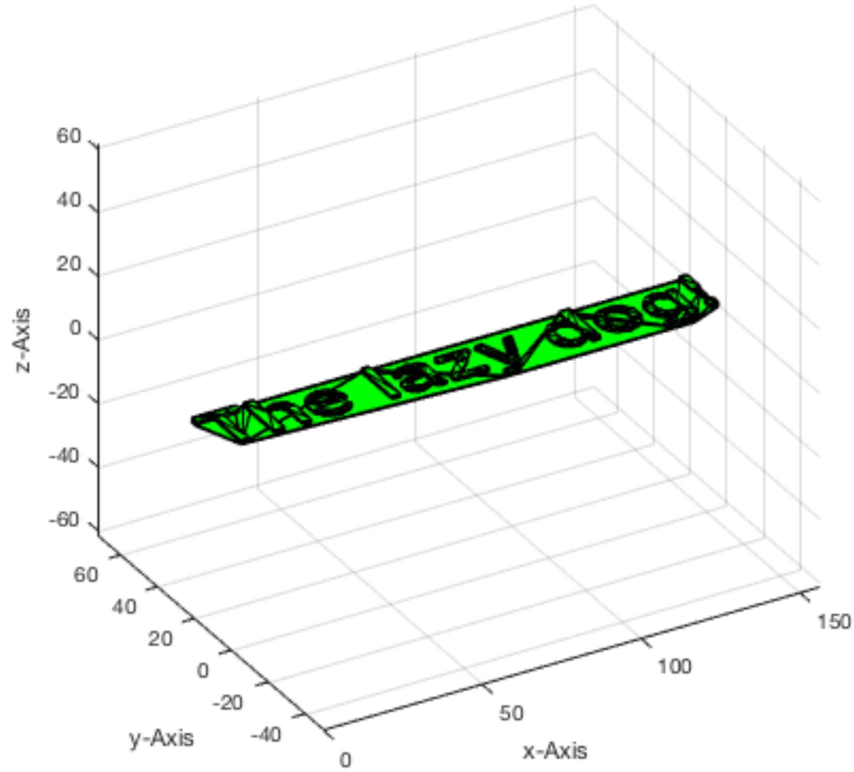
4. Generation of text, numbers, characters and formulas as solid volume

Often you want to write some numbers or code on top of a solid object. For that purpose there is a currently slow function that is able to convert a Matlab-string (even with LaTex-code) into a solid object.

- **VLFLtextimage** writes a line using the text command and converts it into a solid volume
- **VLFLtext** does the same for a very limited number of characters

```
close all;
[VL,FL]=VLFLtextimage('The lazy dog!');
VLFLplot (VL,FL,'g'); view (-30,30);

[VL,FL,d]=VLFLtext('TL-MMXI-XII-XVII');
VLFLwriteSTL (VL,FL,'exp_2011_12_17', 'by Tim C Lueth');
```

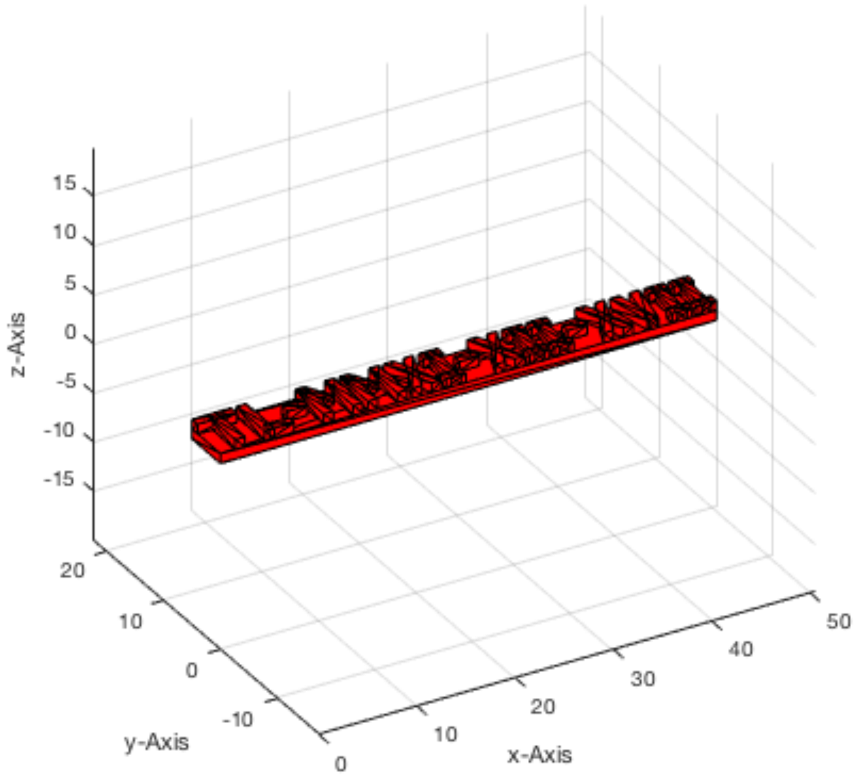
```
WRITING STL FILE /Users/lueth/Desktop/Toolbox_test/exp_2011_12_17.STL in ASCII MODE
completed.
```



5. Turning and mirroring of solids by manipulating the vertex lists (VL)

Turning an object and mirroring is quite simple by exchanging a column of the vertex list to change the sign of a column. To show the use of the functions we generate first a simple roman date string as solid volume.

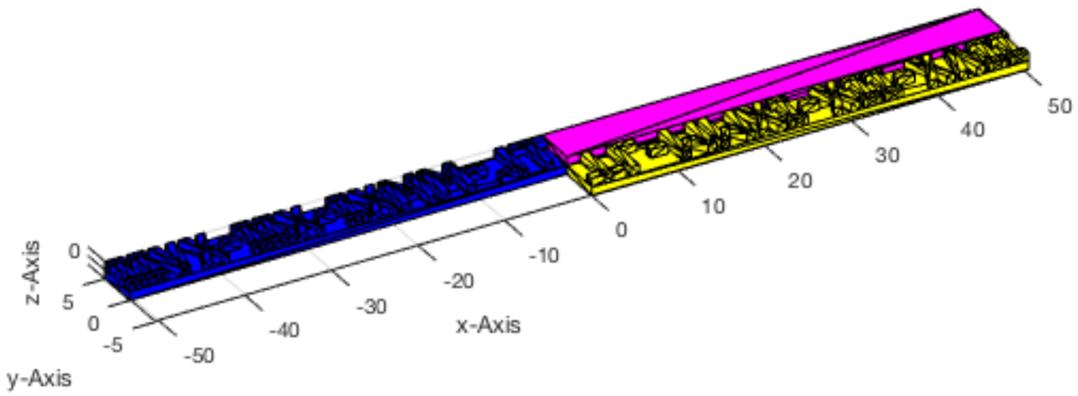
```
close all;
[VL,FL,d]=VLFLtext('TL-MMXI-XII-XVII'); VLFLplot(VL,FL,'r'); view(-30,30);
```



The functions for mirroring solid objects by manipulating the vertex list are the following:

- **VLswapX** mirrors the solid at the x-axis (y/z-plane).
- **VLswapY** mirrors the solid at the y-axis (x/z-plane).
- **VLswapZ** mirrors the solid at the z-axis (x/y-plane).

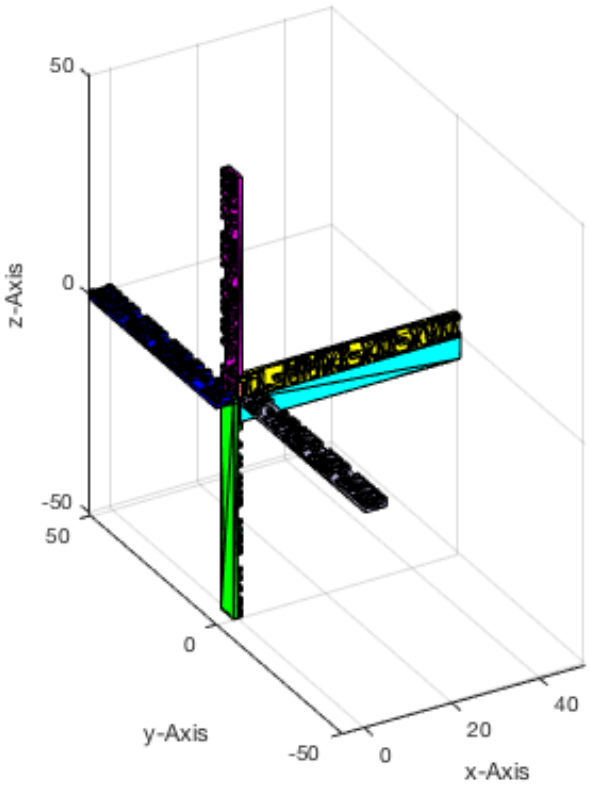
```
close all, view (-30,30); grid on;
VLFLplot(VLswapX(VL),FL,'b');       % mirror at x-axis
VLFLplot(VLswapY(VL),FL,'y');       % mirror at y-axis
VLFLplot(VLswapZ(VL),FL,'m');       % mirror at z-axis
```



The functions for turning solid objects by manipulating the vertex list are the following:

- **VLswapXY** turn the x-axis to the y-axis.
- **VLswapXZ** turn the x-axis to the z-axis.
- **VLswapYX** turn the y-axis to the x-axis.
- **VLswapYZ** turn the y-axis to the z-axis.
- **VLswapZX** turn the z-axis to the x-axis.
- **VLswapZY** turn the z-axis to the y-axis.

```
close all, view (-30,30); grid on
VLFLplot(VL,FL,'r');                % original solid
VLFLplot(VLswapXY(VL),FL,'b');      % turn the x-axis to the y-axis
VLFLplot(VLswapXZ(VL),FL,'m');      % turn the x-axis to the z-axis
VLFLplot(VLswapYZ(VL),FL,'y');      % turn the y-axis to the z-axis
VLFLplot(VLswapZY(VL),FL,'c');      % turn the z-axis to the y-axis
VLFLplot(VLswapZX(VL),FL,'g');      % turn the z-axis to the x-axis
VLFLplot(VLswapYX(VL),FL,'w');      % turn the y-axis to the x-axis
```



6. Spatial transformation of solids by manipulating the vertex lists (VL)

All solid objects consisting of vertices and facets can be moved and rotated by only manipulating the vertex list (VL). Since the facet list is an index list, the facet list (FL) is not affected by a transformation of the vertex list. The following example generates a cylinder and perform different postion and orientation transformations.

```
closeall;
VLFLviewer([]);
PL=PLcircle(10);                        % define a base-contour
[VL,FL]=VLFLofPLz (PL,30);              % extrude to a solid volume
VL=VLswapZX (VL);                       % swap X and X axis
VLFLplot(VL,FL); view (-30,30); grid on;    % plot as red cylinder
```

In detail, there are five basic transformation functions for manipulation a vertex list (VL)

- **VLtrans0** for translating the solid in the coordiante system origin.
- **VLtrans1** for translating the solid into quadrant 1.
- **VLtransP** for translating the solid using a translation vector.
- **VLtransR** for rotation the solid using a rotation matrix.
- **VLtransT** for transformating using an homogenous transformation matrix.

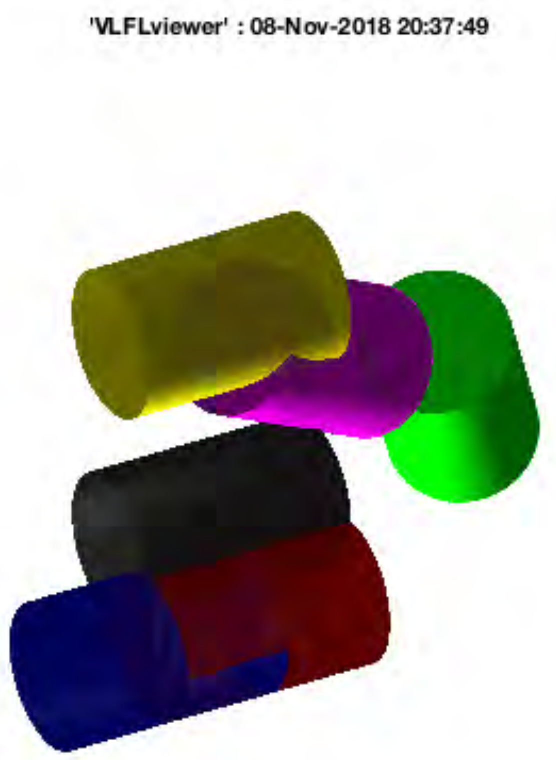
In addition to the already existing matlab functions rotx, roty, and rotz, two new functions are useful.

- **rot** for generating a 3x3 rotation matrix for x y z given in rad.
- **rotdeg** for generating a 3x3 rotation matrix for x y z given in degree.

```
VL=VLtrans0 (VL);                      % Transformation into the origin (blue)
VLFLplot(VL,FL,'b'); view (-30,30);

VL=VLtrans1 (VL);                       % Transformation into quadrant 1 (black)
VLFLplot(VL,FL,'k'); view (-30,30);

VL=VLtransP (VL,[0 ;0; 30]);            % Transformation upwards 30 mm (yellow)
VLFLplot(VL,FL,'y'); view (-30,30);

VL=VLtransR (VL,rotdeg(0,30,15));       % Rotate 30 degree around y and 15 around z (magenta)
VLFLplot(VL,FL,'m'); view (-30,30);

T=[rotdeg(0,30,15), [20;0;0];[0 0 0 1]] % define a homogenous transformation matrix (green)
```

```
VL=VLtransT (VL,T);                   % Transformation using an HT matrix
VLFLplot(VL,FL,'g'); view (-30,30); grid on;
VLFLplotlight (1,0.9); grid on;
```

```
T =

    0.8365   -0.2241    0.5000   20.0000
    0.2588    0.9659         0         0
   -0.4830    0.1294    0.8660         0
         0         0         0    1.0000
```



Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:37:50!
Executed 08-Nov-2018 20:37:52 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
```

```
map_toolbox
matlab
simulink
video_and_image_blockset
===========================================================================================
==========
```

- Tim Lueth, tested and compiled on OSX 10.7.5 with Matlab 2014b on 2014-11-19
- Tim Lueth, executed and published on 64 Bit PC using Windows with Matlab 2014b on 2014-11-19

Published with MATLAB® R2018a

Tutorial 03: Closed 2D Contours and Boolean Operations in 2D

2014-11-19: 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

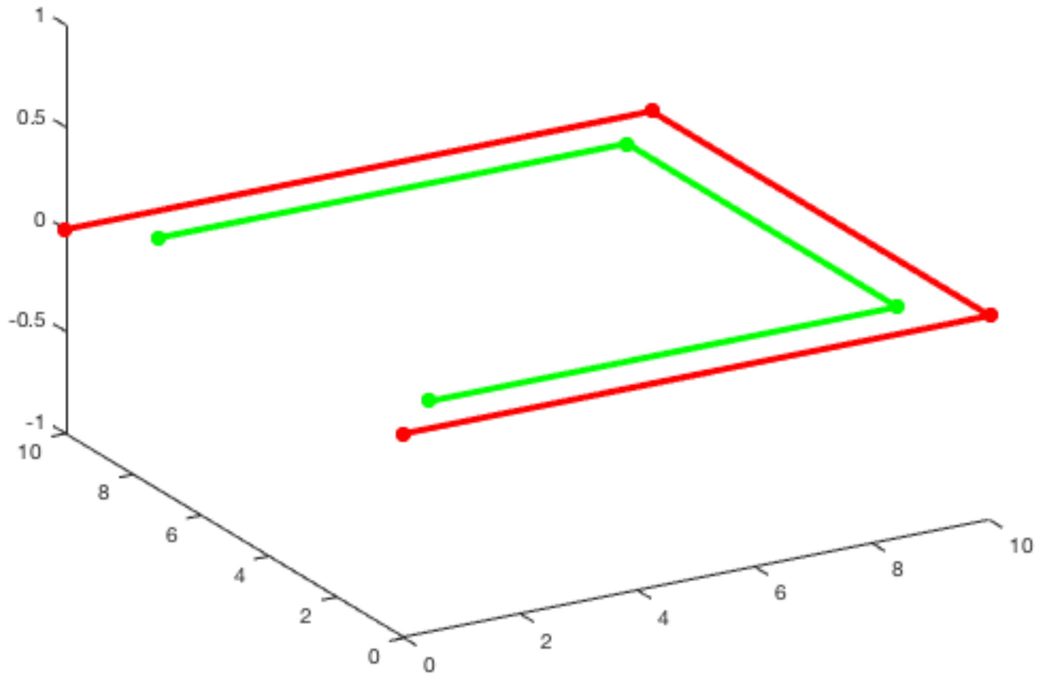
Motivation for this tutorial: (Originally SolidGeometry 1.3 required)

2. The contour polybool list (mapping toolbox)

This 3rd example deals with a different data structure for the description of 2D closed contour polygons: Contour Polybool List (CPL). The CPL is a nx2 x/y-coordinate point list [x y] similar to a point list (PL). Always, the first and last point of the list are considered as closed. In addition, it is possible to concatenate two point list after another. To separate the individual contours, a separator-point [NaN NaN] is inserted between them.

```
close all;
PLA=[0 0; 10 0; 10 10; 0 10], PLB=[1 1; 9 1; 9 9; 1 9]
PLplot (PLA,'r-*',3);PLplot (PLB,'g-*',3); view(-30,30);
```

```
PLA =

     0     0
    10     0
    10    10
     0    10


PLB =

     1     1
     9     1
     9     9
     1     9
```

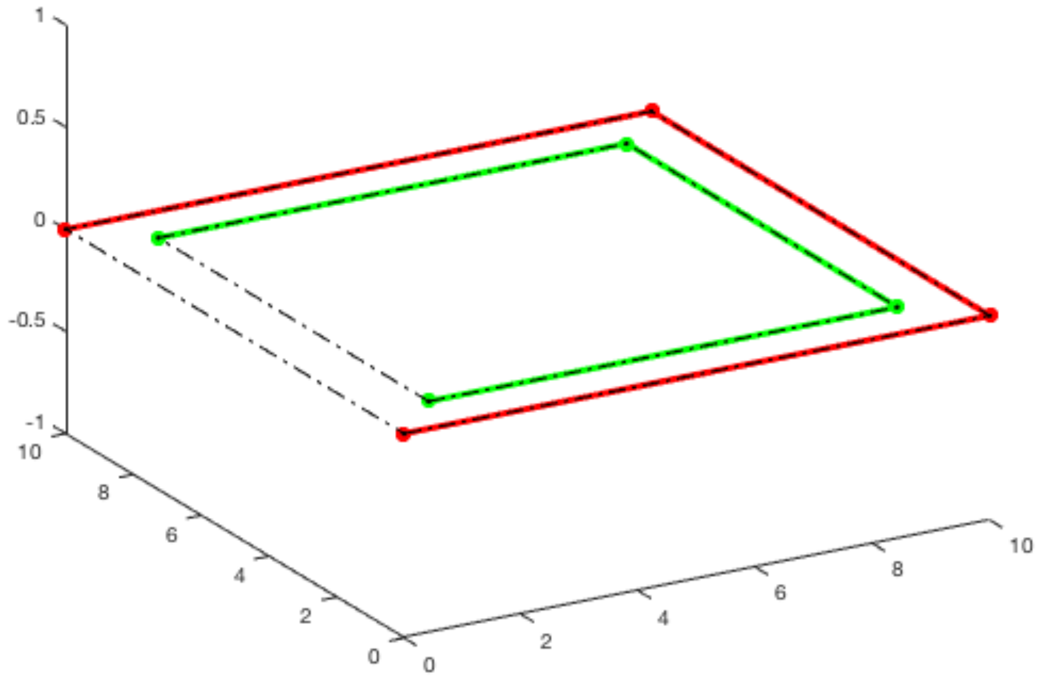


The concatenation generates then the closed polybool list (CPL). Two functions are helpful to draw a CPL and also to show the start point (black cube), the endpoint (black ring) and the direction of each edge of the CPL:

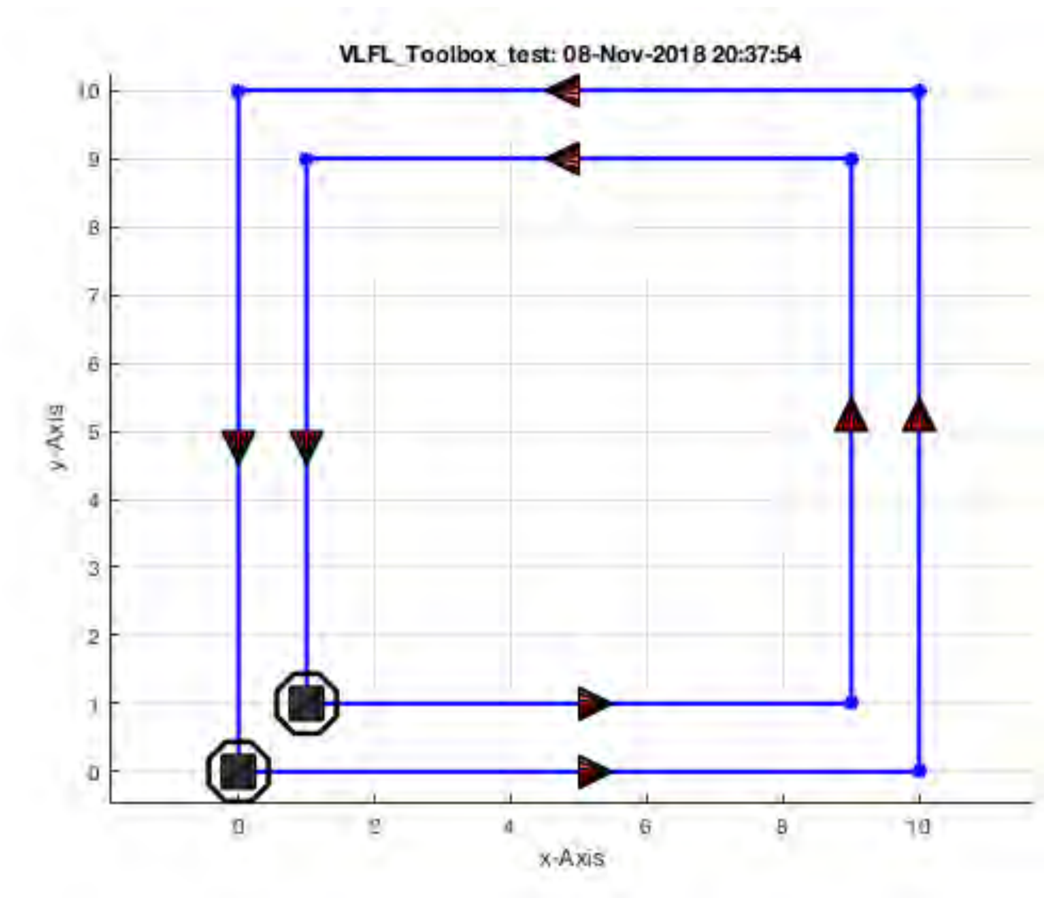
- **CPLplot** draws a closed contour polybool list (CPL).
- **PLELofCPL** draws a start point, end point and direction-arrows, when called without any output variable.

```
CPL=[PLA;NaN NaN;PLB],
CPLplot (CPL,'k.-.',1);
```

```
CPL =

     0     0
    10     0
    10    10
     0    10
   NaN   NaN
     1     1
     9     1
     9     9
     1     9
```




```
close all;
PLELofCPL (CPL);
```



3. Surface tesselation for contour polybool list (CPL)

A closed polygon list can be considered as bounding contour for a surface. In general, there exist different strategies, to tesselate a bounding contour, to get a triangle surface description. There is no optimal one. We can distinguish **simple strategies** or more advanced strategies such as **Delaunay-Triangulation** or **Row-Scanning-Triangulation**. In example 1 we used a simple strategy for closing convex polygons.

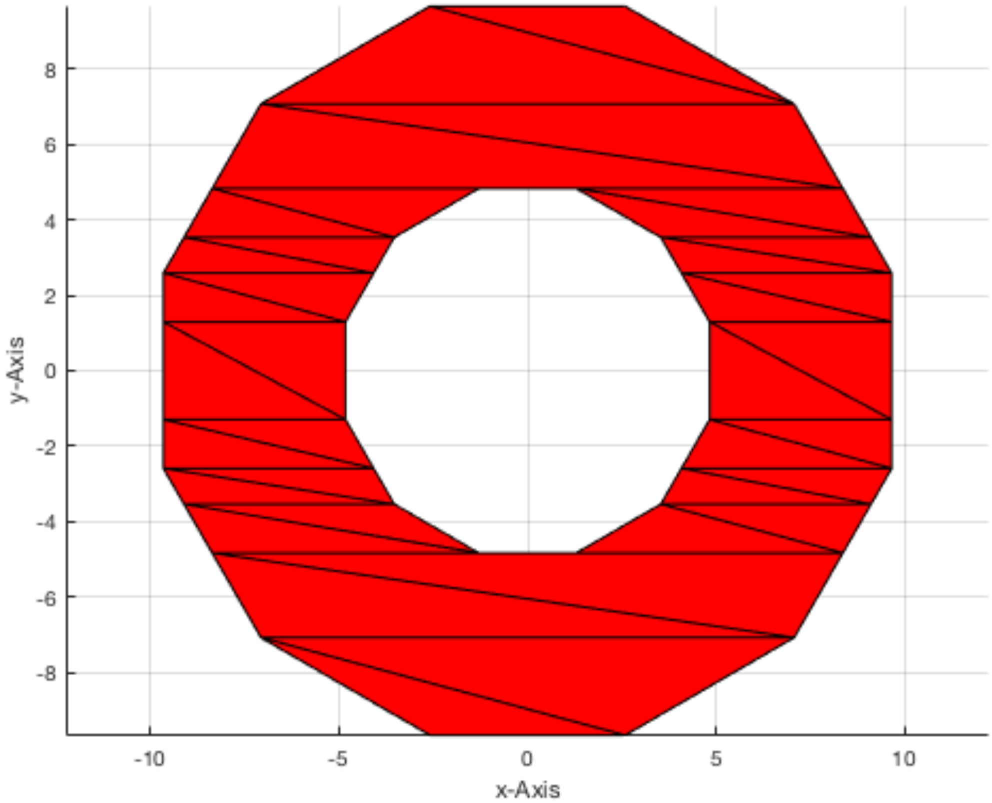
- **Row-Scanning-Triangulation** is able to handle all kinds of polygons (even enclosed), but the triangle facets are sometimes very small. Furthermore, the point contains redundant information.
- **Delaunay-Triangulation** is able to handle all kinds of polygons (even enclosed), but has problems with polygons that cross each other or share one point or more points, i.e. overlapping edges.

It is not unique to tesselate polygons, if they are enclosed or cross each other. There exist no general purpose solution. Nevertheless, in most cases, the Delaunay-Triangulation is preferrable, since this concept does work also in 3D. To generate a facet list (FL) for a CPL, there exist two functions. In case of crossing polygons or overlapping polygons, additional points have to be caluclated automatically, and therefore, the points in the point list can change. In this case, you will get a warning, but only in case of the Delaunay-triangulation. Conventionally, additional split/crossing points are added at the end of the point list, in case of the Delaunay-triangulation.

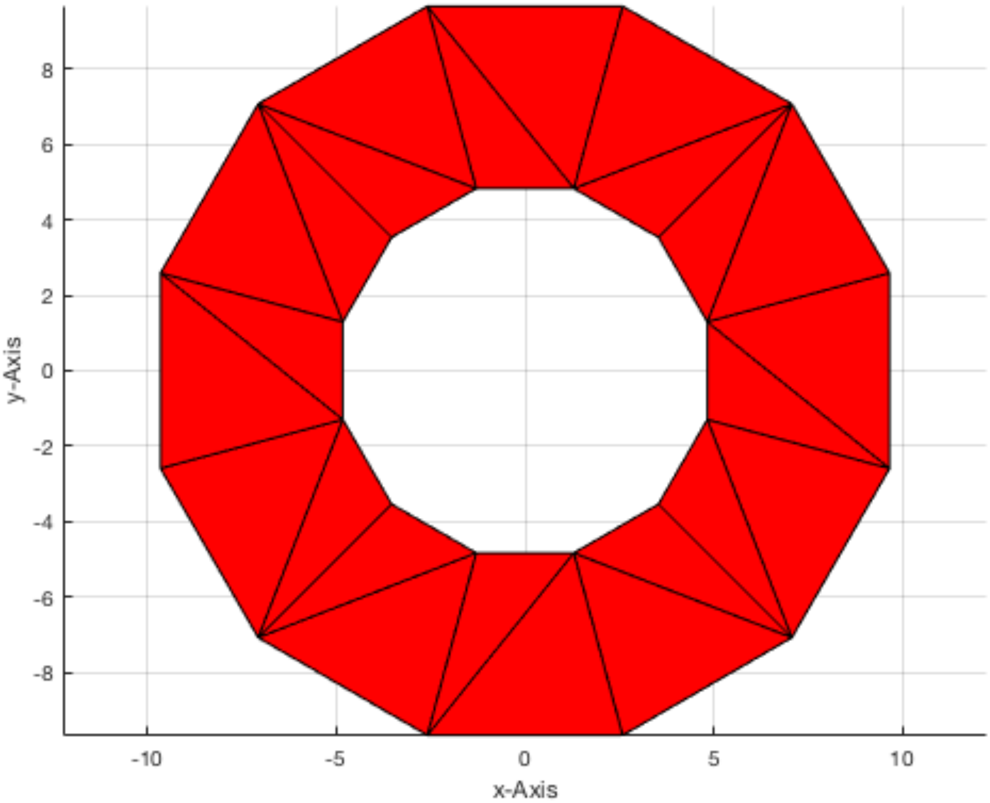
- **PLFLofCPLpoly** returns a facet tesselation by a simple y-coordinate row scanning (the points are ordered by increasing y, contour by contour, do not mix, but are redundant. Not as efficient as Delaunay, and not useful for 3D).
- **PLFLofCPLdelaunay** returns a facet tesselation by a Delaunay-triangulation (no crossings or joint points or joint edges are allowed, i.e create additional split points).

```
close all;
CPL=[PLcircle(10,12);NaN NaN;PLcircle(5,12)]
[PL,FL]=PLFLofCPLpoly(CPL); VLFLplot(PL,FL);
```

```
CPL =

    9.6593   -2.5882
    9.6593    2.5882
    7.0711    7.0711
    2.5882    9.6593
   -2.5882    9.6593
   -7.0711    7.0711
   -9.6593    2.5882
   -9.6593   -2.5882
   -7.0711   -7.0711
   -2.5882   -9.6593
    2.5882   -9.6593
    7.0711   -7.0711
       NaN       NaN
    4.8296   -1.2941
    4.8296    1.2941
    3.5355    3.5355
    1.2941    4.8296
   -1.2941    4.8296
   -3.5355    3.5355
   -4.8296    1.2941
   -4.8296   -1.2941
   -3.5355   -3.5355
   -1.2941   -4.8296
    1.2941   -4.8296
    3.5355   -3.5355
```



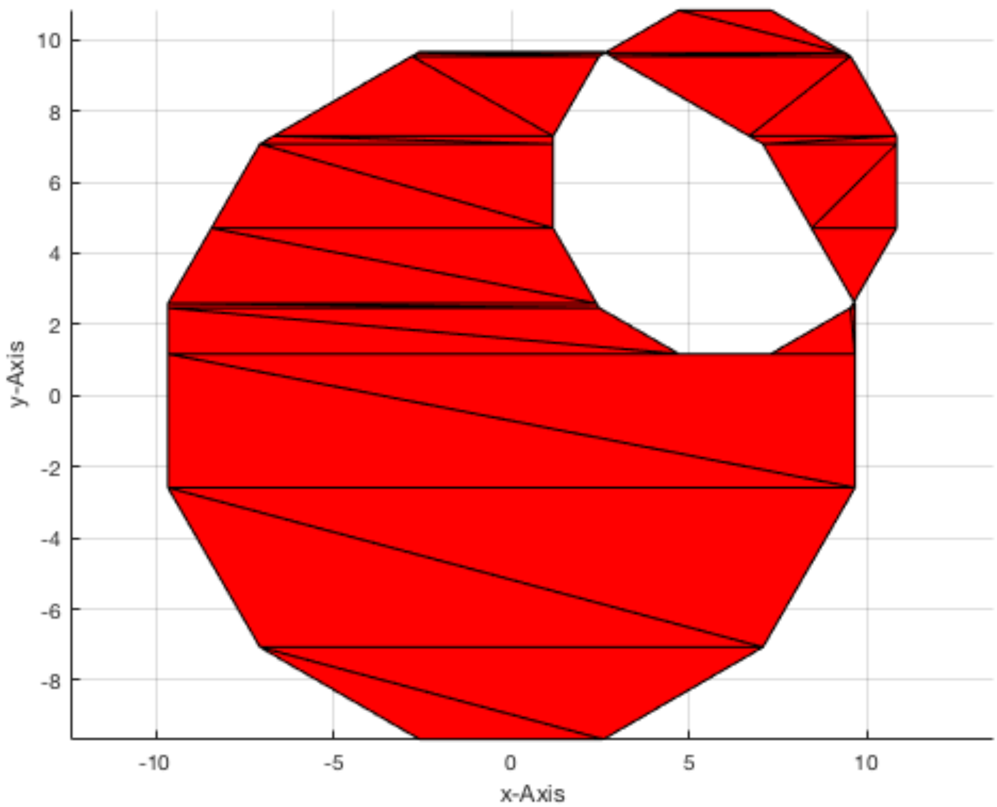

```
close all;
[PL,FL]=PLFLofCPLdelaunay(CPL); VLFLplot(PL,FL);
```

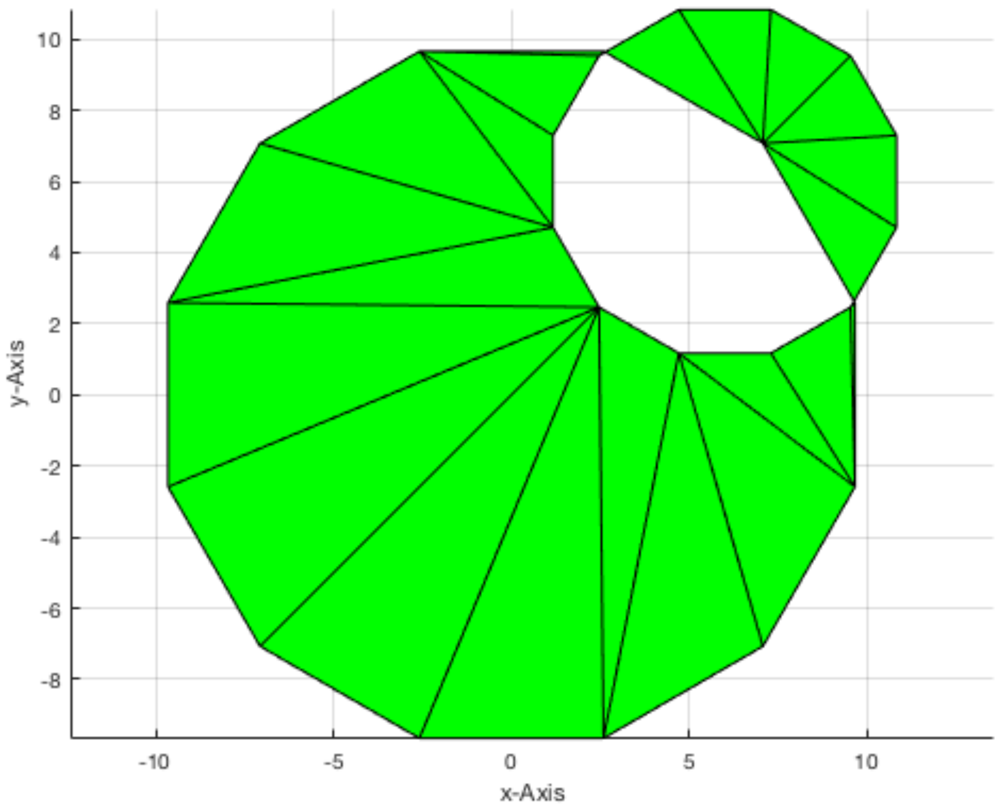


The next example shows the need for splitting contours:

```
close all
CPL=[PLcircle(10,12);NaN NaN;PLcircle(5,12)+6];
figure(1); [PL,FL]=PLFLofCPLpoly(CPL); VLFLplot(PL,FL,'r');
figure(2); [PL,FL]=PLFLofCPLdelaunay(CPL); VLFLplot(PL,FL,'g');
```

```
Warning: Intersecting edge constraints have been split, this may have added new
points into the triangulation.
```





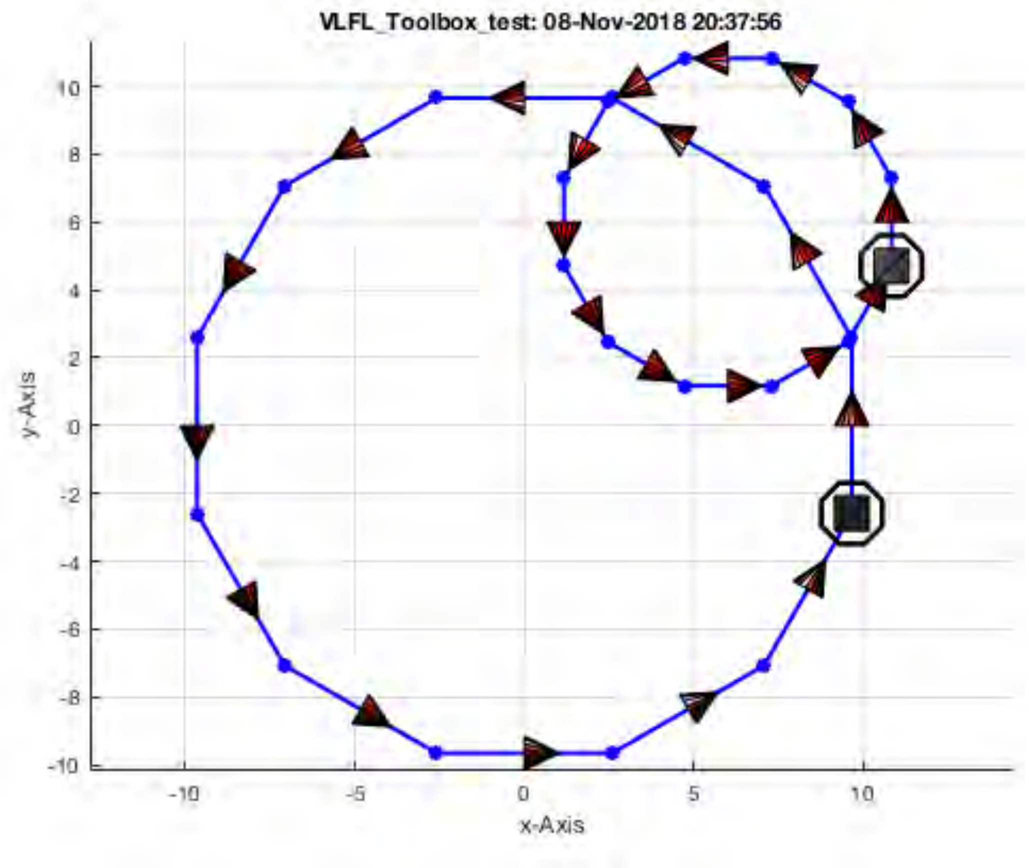
4. Orientation of outer and inner polygons of a CPL

In the mapping tool box, that supports the boolean operation of contours, there is a rule to use outer contours in clockwise (cw) directions and embedded contours always in the opposite direction, which means counter-clockwise (ccw) for the first level of embedding. Unfortunatly, this is exactly the other way around to the rules that are used for Delaunay representation and 3D surface description. So we have to be careful later when switching from CPL to surface description for 3D modelling. In 3D modelling, to distinguish outer contours and inner contours of a CPL, we use counter clockwise (ccw) polygons for outside and clockwise (cw) polygons for inside contours. At a later stage we want to generate walls extruded upwards on the contours. If the contour direction is defined correctly for 3D modelling, the facet orientation can be calcuated automatically from the contour direction.

- **PLELofCPL** shows the direction of the used contours.
- **flip(PL)** changes the direction of a point list.

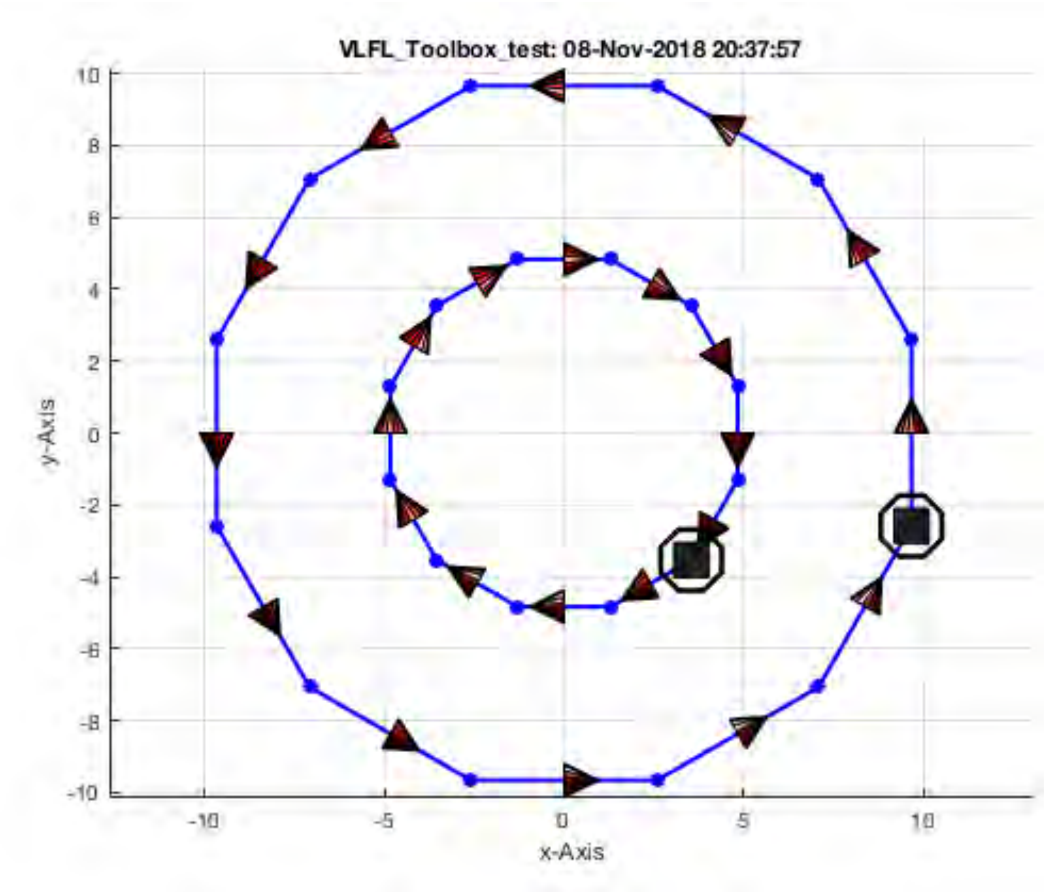
In the next figure, we see both polygons counter-clockwise:

```
PLELofCPL(CPL);
```



Now, we see the outer polygons counter-clockwise and the inner polygons clockwise

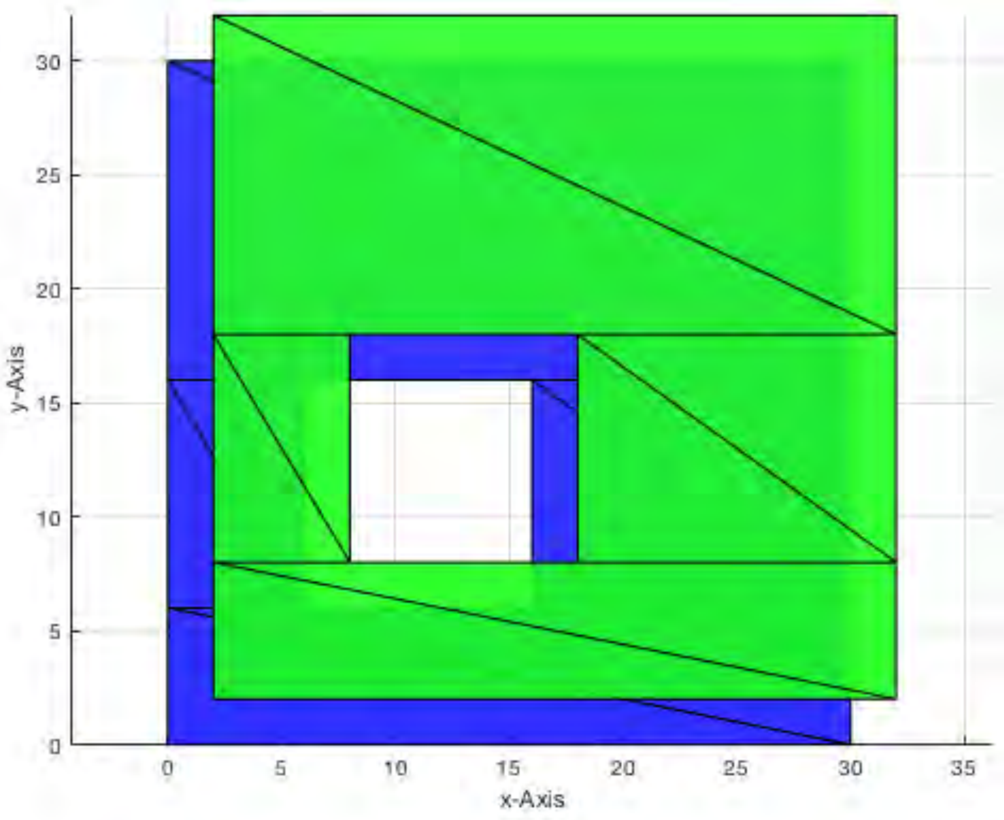
```
close all;
CPL=[PLcircle(10,12);NaN NaN;flip(PLcircle(5,12))];
PLELofCPL(CPL);
```



5. Boolean operations of contour polybool lists (CPL)

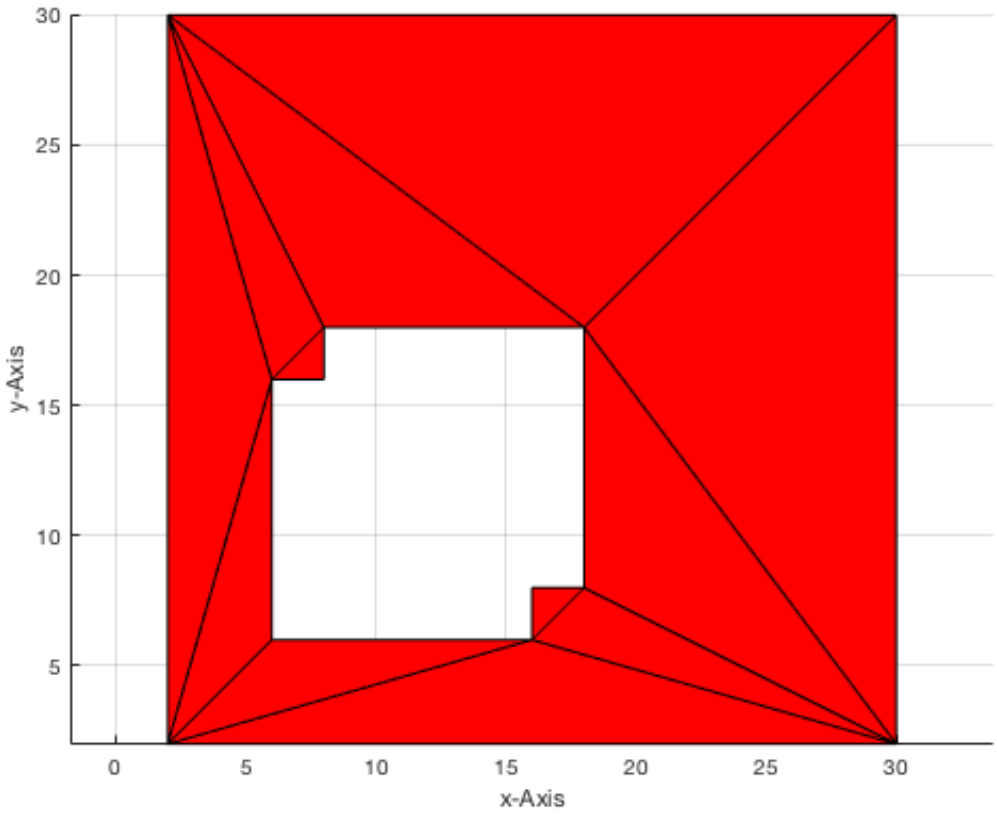
The main advantage of the CPL representation is currently the possibility to use the polybool functions of the mapping toolbox. Here, we show the use in embedded functions that help later to make the step forward to 3D modeling. We start with boolean operations of contour polybool lists (CPL).

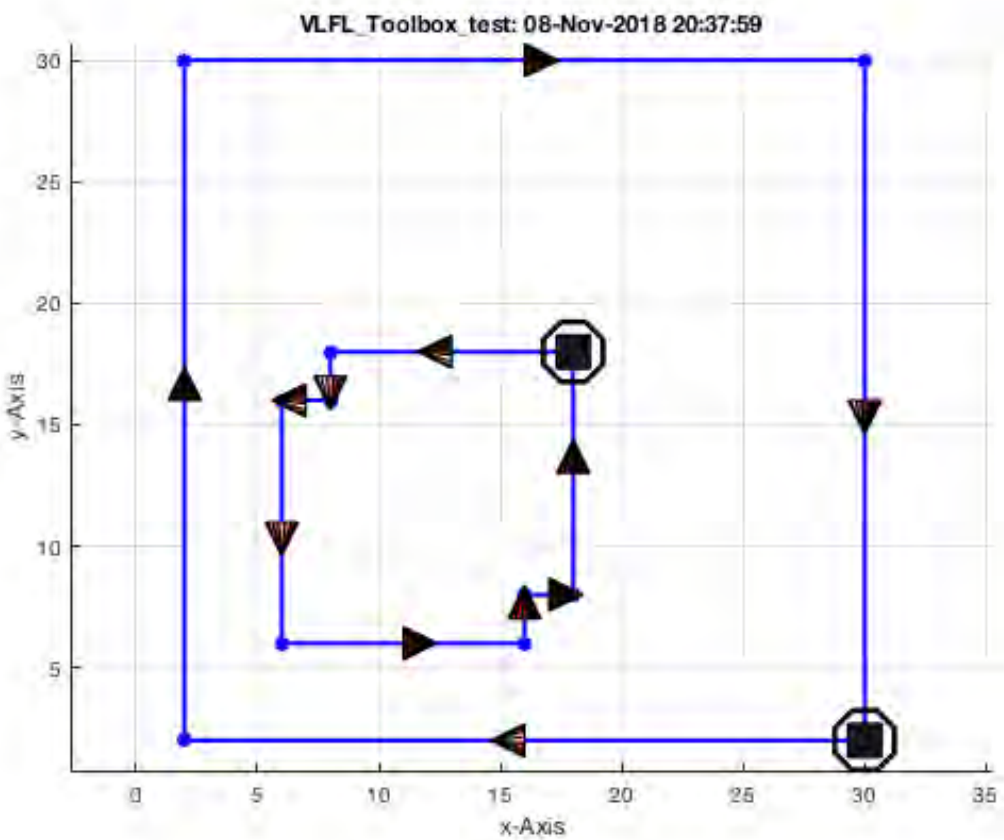
```
close all; figure;
CPL=[PLA*3;NaN NaN;(PLA)+6];
CPLA=CPL;    [PL,FL]=PLFLofCPLpoly(CPLA); VLFLplot(PL,FL,'b');
CPLB=CPL+2;  [PL,FL]=PLFLofCPLpoly(CPLB); VLFLplot(PL,FL,'g');
VLFLplotlight (0,0.9)
```



- **CPLpolybool('and',CPLA,CPLB)** delivers CPLA intersecting CPLB.

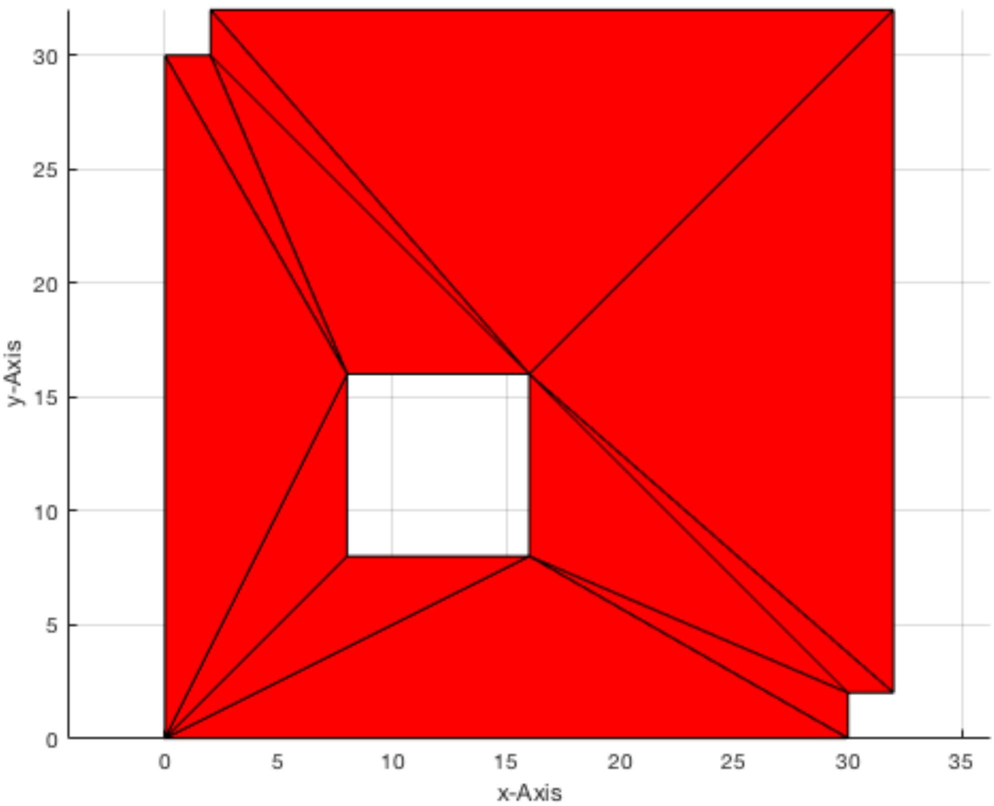
```
close all;
CPLN=CPLpolybool('and',CPLA,CPLB);
[PL,FL]=PLFLofCPLdelaunay(CPLN); VLFLplot(PL,FL); PLELofCPL(CPLN);
```

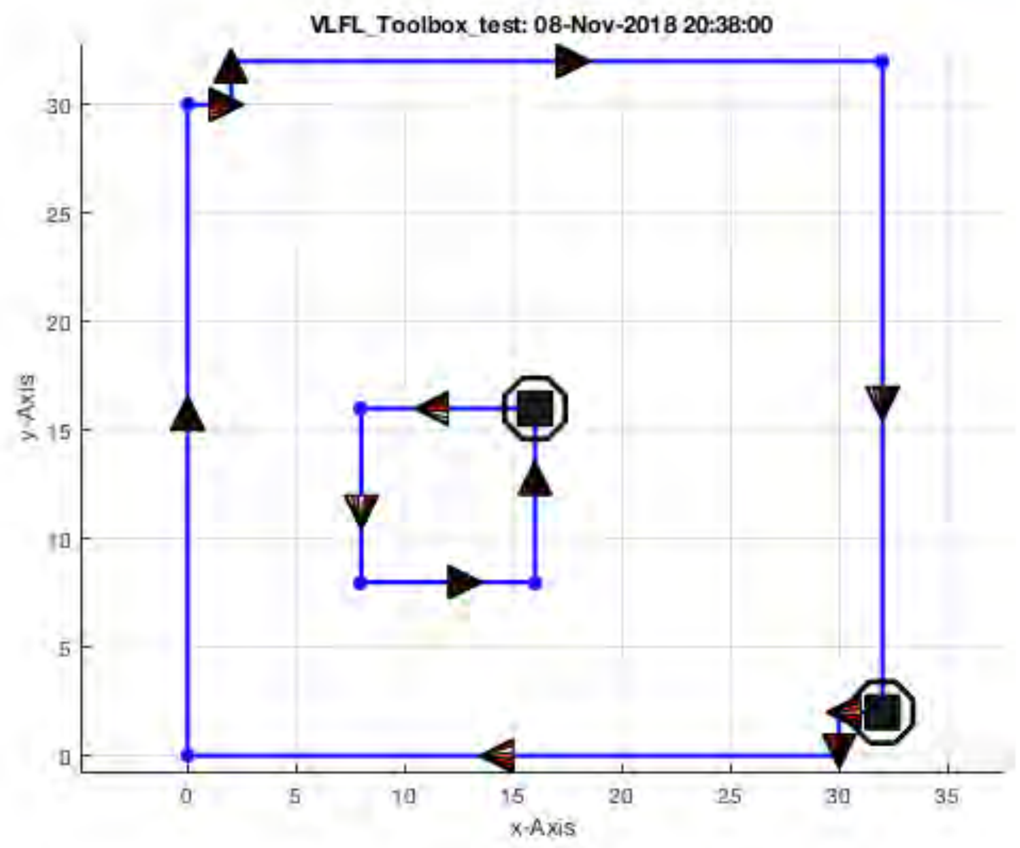




- **CPLpolybool('or',CPLA,CPLB)** delivers CPLA united with CPLB.

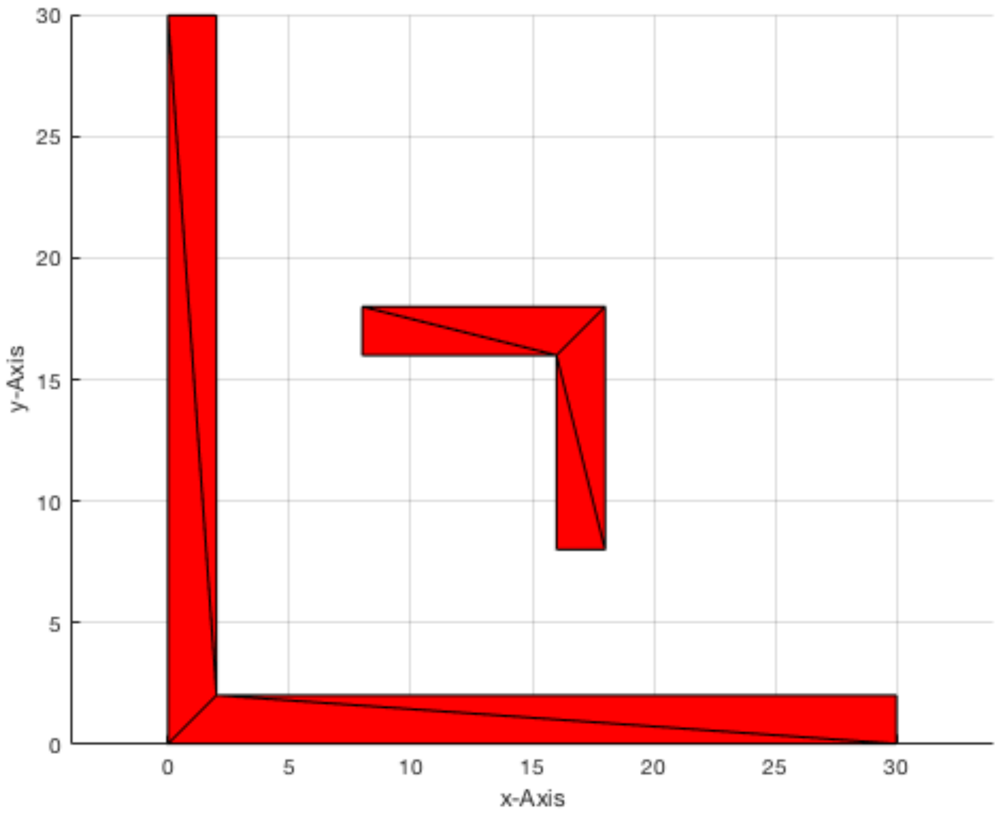
```
close all;
CPLN=CPLpolybool('or',CPLA,CPLB);
[PL,FL]=PLFLofCPLdelaunay(CPLN); VLFLplot(PL,FL); PLELofCPL(CPLN);
```

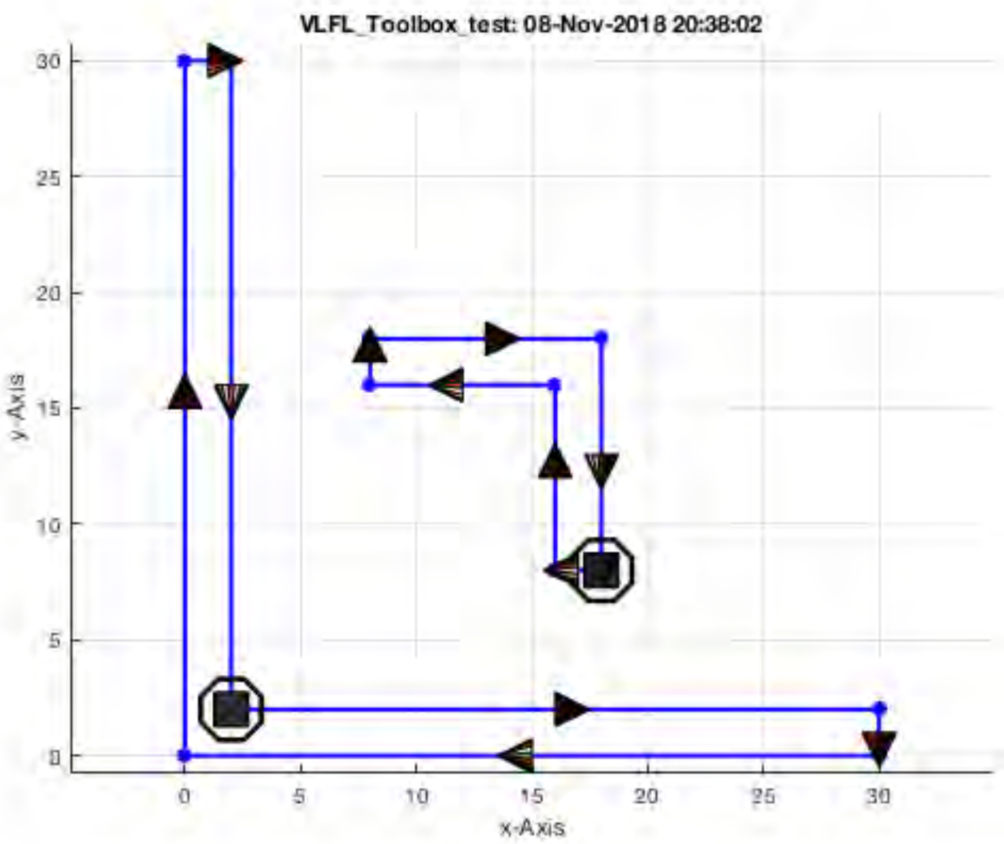




- **CPLpolybool('minus',CPLA,CPLB)** delivers CPLA minus CPLB.

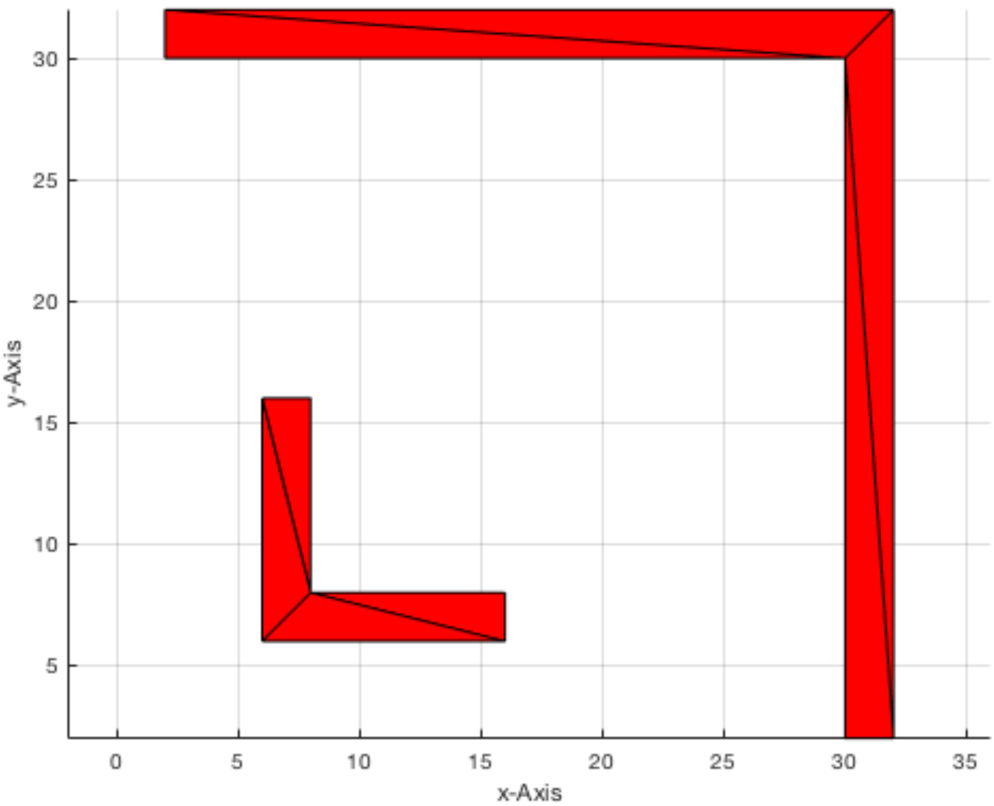
```
close all;
CPLN=CPLpolybool('minus',CPLA,CPLB);
[PL,FL]=PLFLofCPLdelaunay(CPLN); VLFLplot(PL,FL); PLELofCPL(CPLN);
```

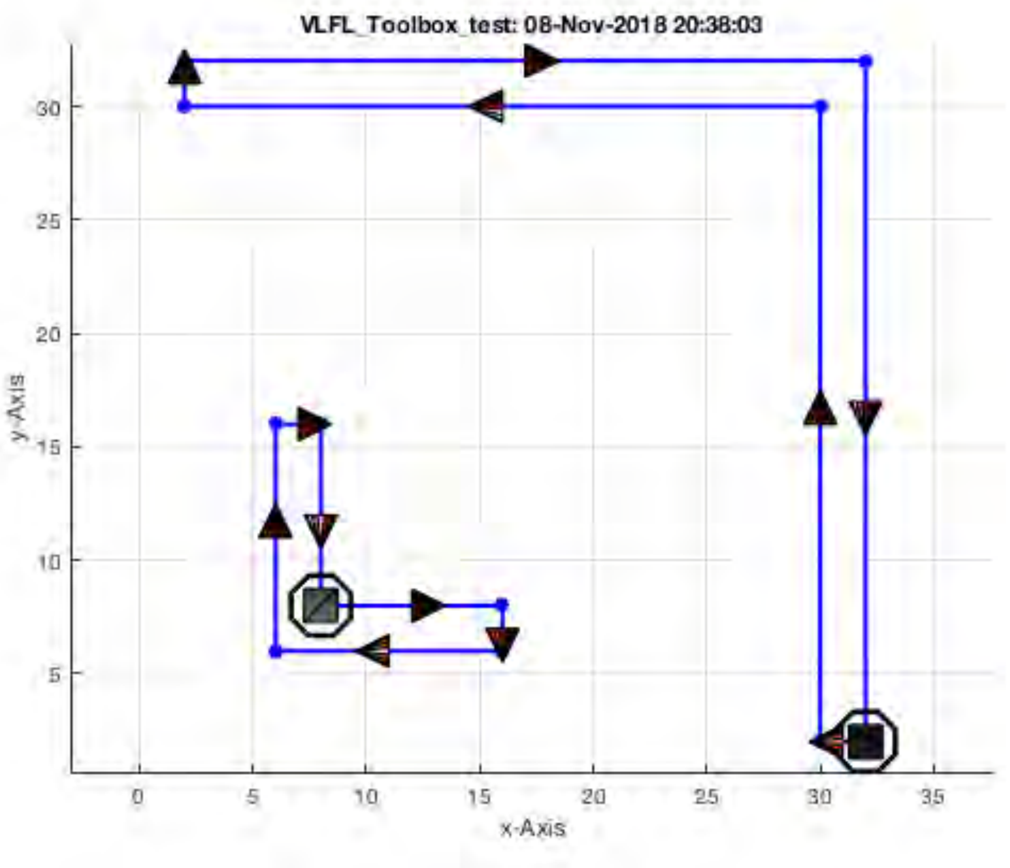




- **CPLpolybool('minus',CPLB,CPLA)** delivers CPLB minus CPLA.

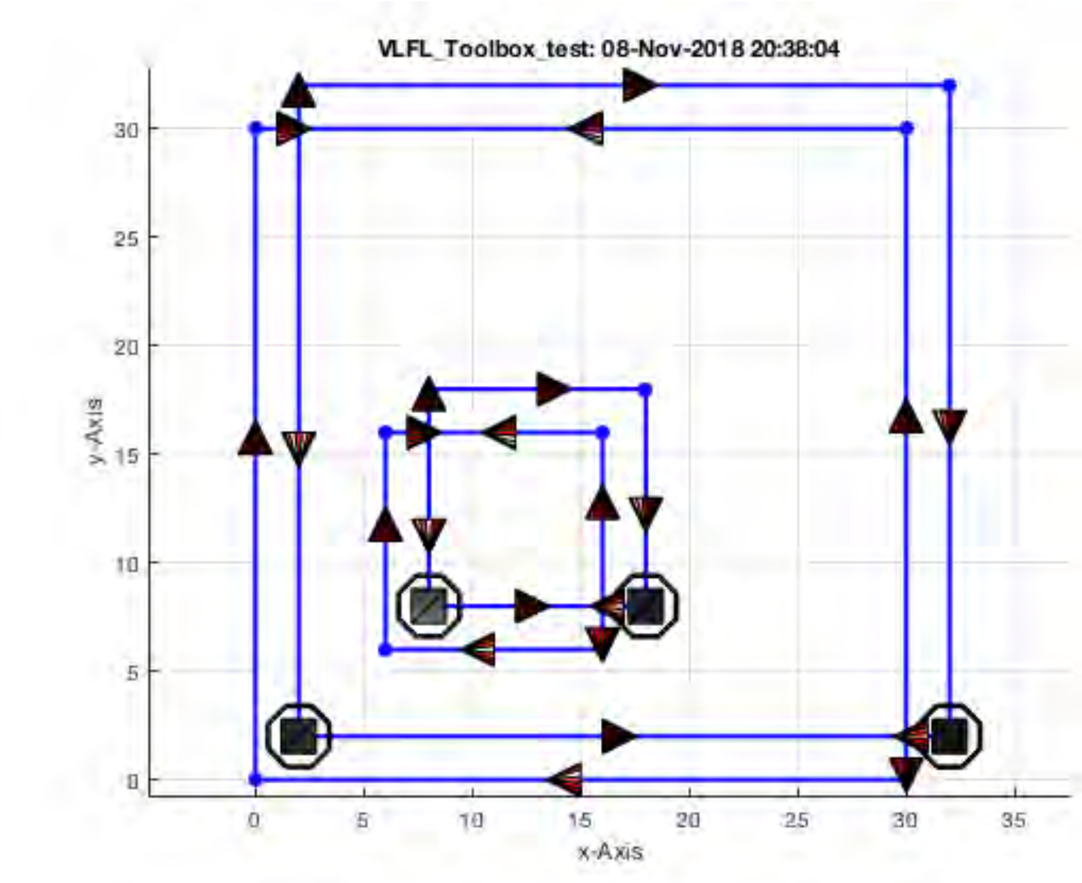
```
close all;
CPLN=CPLpolybool('minus',CPLB,CPLA);
[PL,FL]=PLFLofCPLdelaunay(CPLN); VLFLplot(PL,FL); PLELofCPL(CPLN);
```





- **CPLpolybool('xor',CPLA,CPLB)** delivers CPLA exclusiveor CPLB.

```
close all;
CPLN=CPLpolybool('xor',CPLA,CPLB);
[PL,FL]=PLFLofCPLpoly   (CPLN); VLFLplot(PL,FL); PLELofCPL(CPLN);
```



6. Converting a closed polybool list into a point list (PL) and an edge list (EL)

To generate an extruded 2½D solid from a CPL, it makes sense first to convert a CPL to a point list (PL) with an explicit description of the edges of the polygons as edge list (EL). Since the EL, as a result of CPLpolybool, has an inverted direction, ELflip is used to change the direction of the edges.

- **PLELofCPL** transforms the CPL into a point list (PL) and an edge list (EL).
- **ELflip** corrects the edge direction after CPLpolybool.

```
CPLN=CPLpolybool('minus',CPLA,CPLB)
[PL,EL]=PLELofCPL(CPLN), EL=ELflip(EL),
```

```
CPLN =

    18     8
    16     8
    16    16
     8    16
     8    18
    18    18
    18     8
   NaN   NaN
     2     2
    30     2
    30     0
     0     0
     0    30
     2    30
     2     2


PL =

    18     8
    16     8
    16    16
     8    16
     8    18
    18    18
     2     2
    30     2
    30     0
     0     0
     0    30
     2    30


EL =

     1     2
     2     3
     3     4
     4     5
     5     6
     6     1
     7     8
```

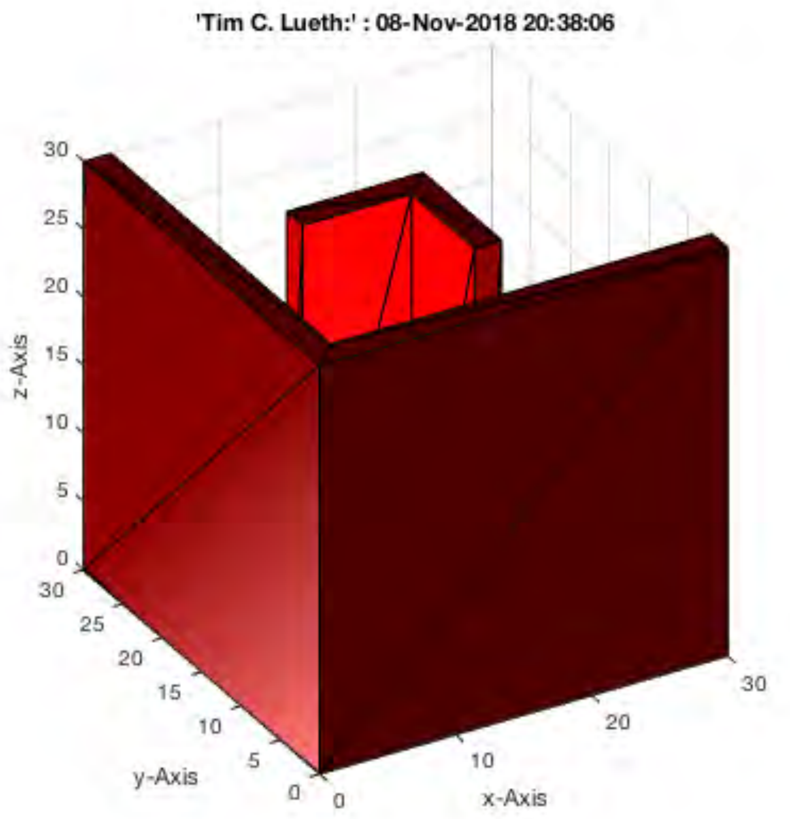
```
     8     9
     9    10
    10    11
    11    12
    12     7


EL =

     7    12
    12    11
    11    10
    10     9
     9     8
     8     7
     1     6
     6     5
     5     4
     4     3
     3     2
     2     1
```

7. Extruding point list (PL) and edge list (EL) to a solid volume

After a boolean operation there is often the wish to extrude the resulting base contour into a 3D solid volume. The function is explained later in more detail. Anyway,it is helpful to see in 3D a model that is the result of a 2D boolean operation.

- **VLFLofPLELz** extruding a point list (PL) and edge list (EL) into 3D.

```
close all; VLFLfigure; view(-30,30);
[VL,FL]=VLFLofPLELz(PL,EL,30); VLFLplots(VL,FL);
```



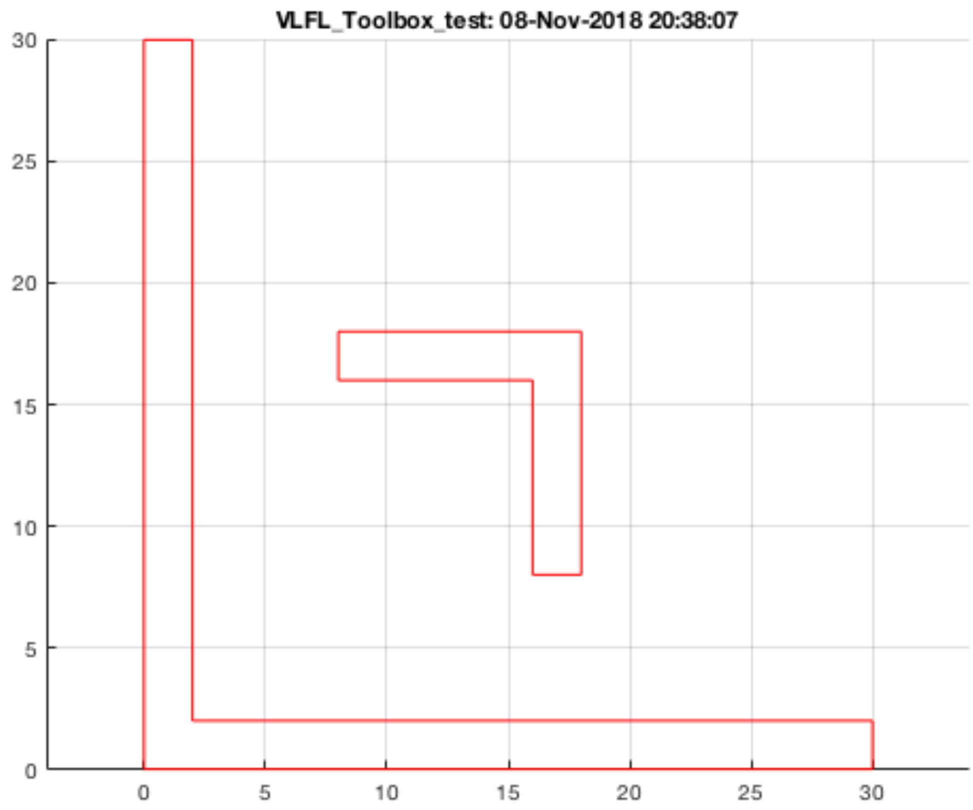
8. Converting a point list and edge list into a closed polybool list

Finally, as it was possible to convert a CPL into and point list and an edge list, there is a function for the opposite.

- **CPLofPLEL** converting a point list (PL) and an edge list (EL) into a closed polybool list.

```
CPLofPLEL(PL,EL)
```

```
ans =

    18     8     0
    18    18     0
     8    18     0
     8    16     0
    16    16     0
    16     8     0
    18     8     0
   NaN   NaN   NaN
     2     2     0
     2    30     0
     0    30     0
     0     0     0
    30     0     0
    30     2     0
     2     2     0
```



Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:38:07!
Executed 08-Nov-2018 20:38:09 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
simulink
video_and_image_blockset
===========================================================================================
==========
```

- *Tim Lueth, tested and compiled on OSX 10.7.5 with Matlab 2014b on 2014-11-19*
- *Mattias Traeger, executed and published on 64 Bit PC using Windows with Matlab 2014b on 2014-11-20*

Published with MATLAB® R2018a

Tutorial 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) - 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)

Motivation for this tutorial: (Originally SolidGeometry 1.6 required)

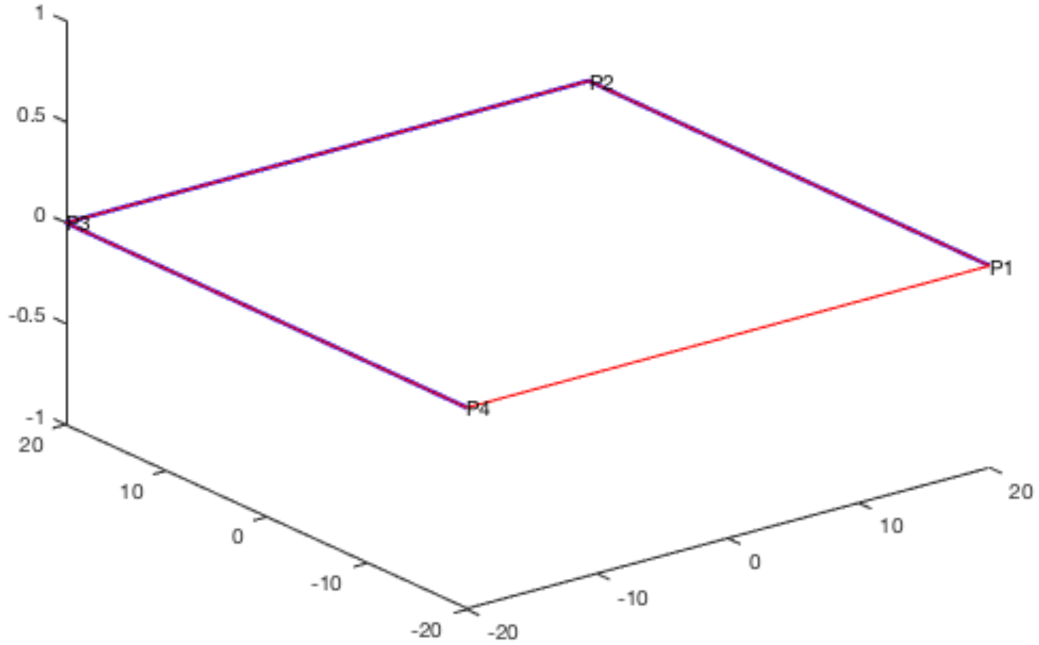
2. Moving and rotating point lists (PL) and closed polygon lists (CPL)

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

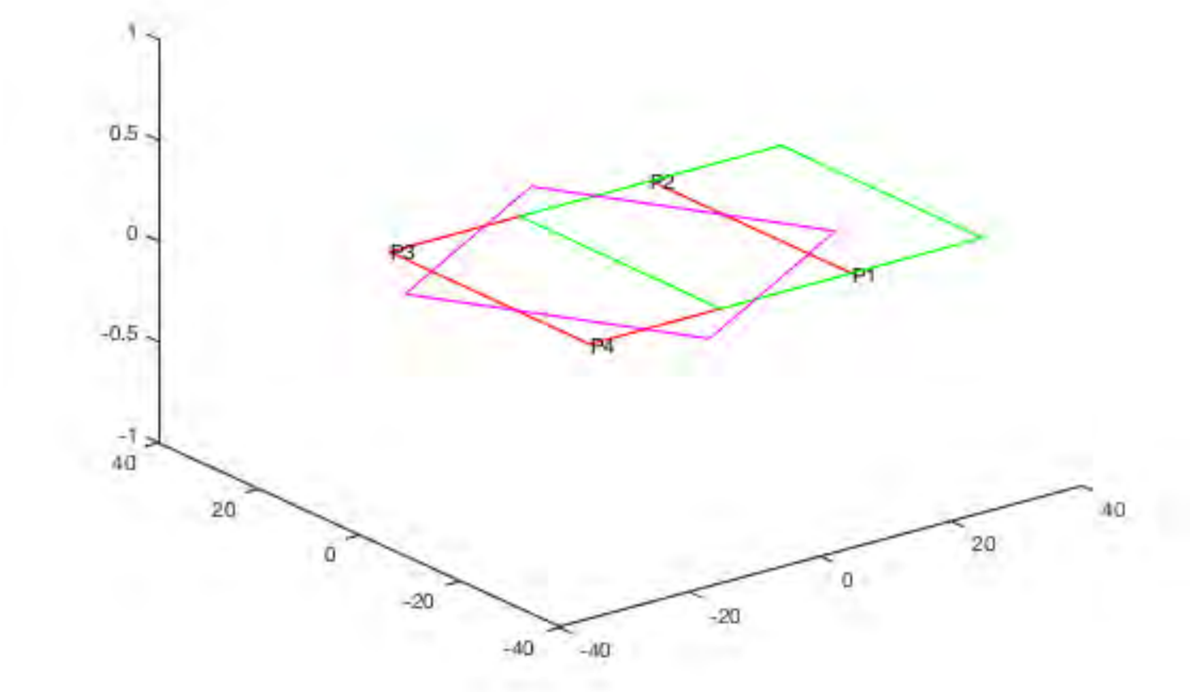
```
close all;
PLA=PLcircle(20*sqrt(2),4);              % Generate a circle with 4 point, i.e. square
PLplot(PLA,'b',2);                       % Plots the points in blue
CPLplot(PLA,'r');                        % Plots the closed polygon in red
textVL (PLA);                            % Plots point descriptors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



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
```

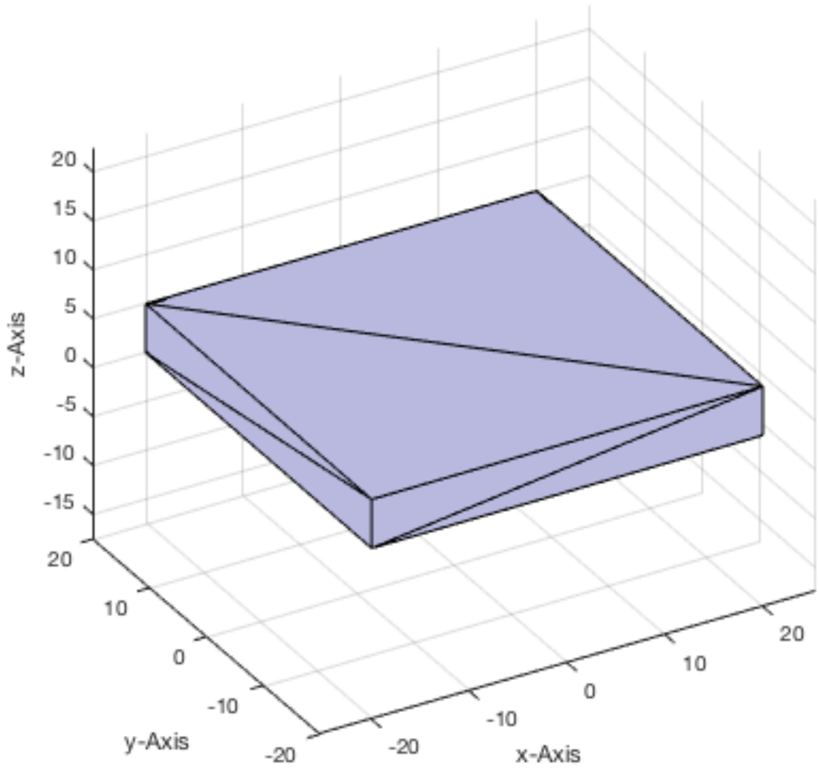


3. Simple extrusion of point lists (PL/CPL) to design 2½D solids

Next we extrude the square in 3D

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

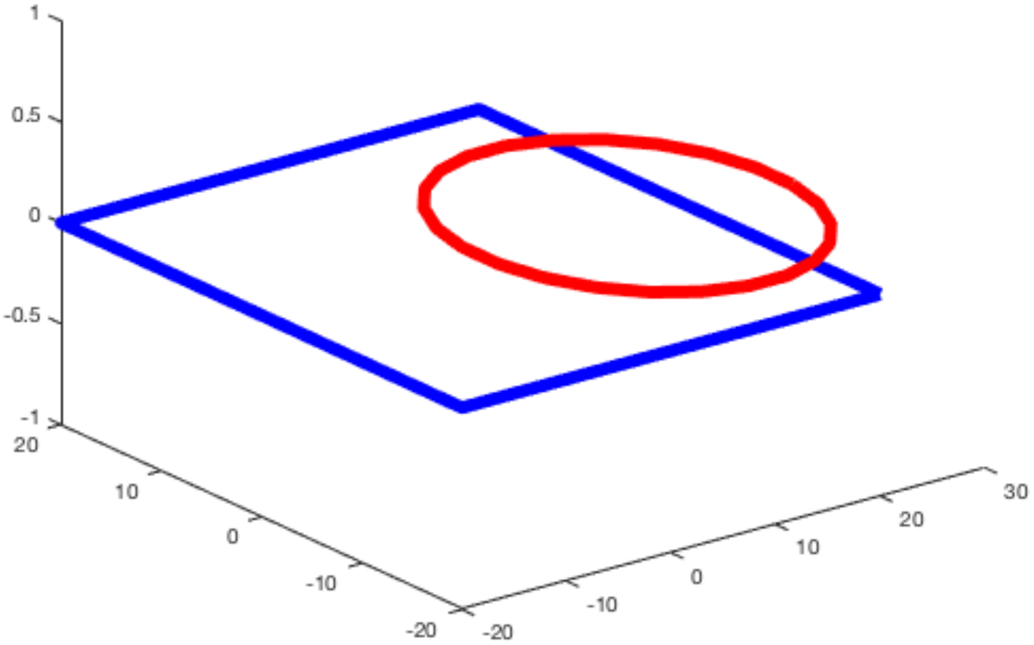
```
close all
[VL,FL]=VLFLofCPLz (PLA,5);
VLFLplot (VL,FL,'w'), axis equal, view(-30,30);
```



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

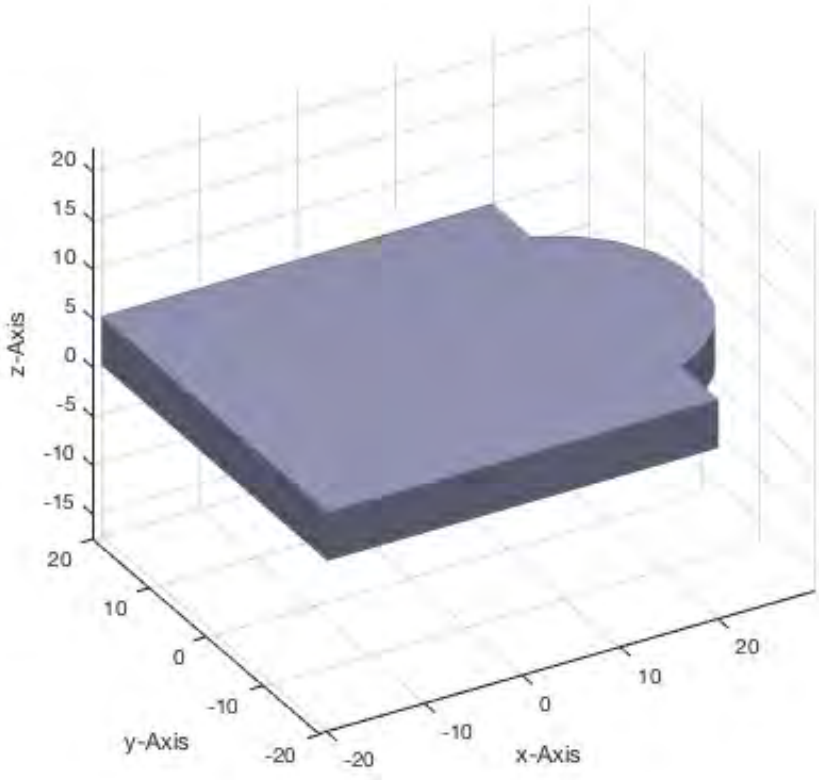
```
close all;
PLA=PLcircle(20*sqrt(2),4);
PLB=PLcircle(10*sqrt(2),24);  PLB=PLtransP(PLB,[15 0]);
CPLplot(PLA,'b',6); CPLplot(PLB,'r',6);
```



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')
```

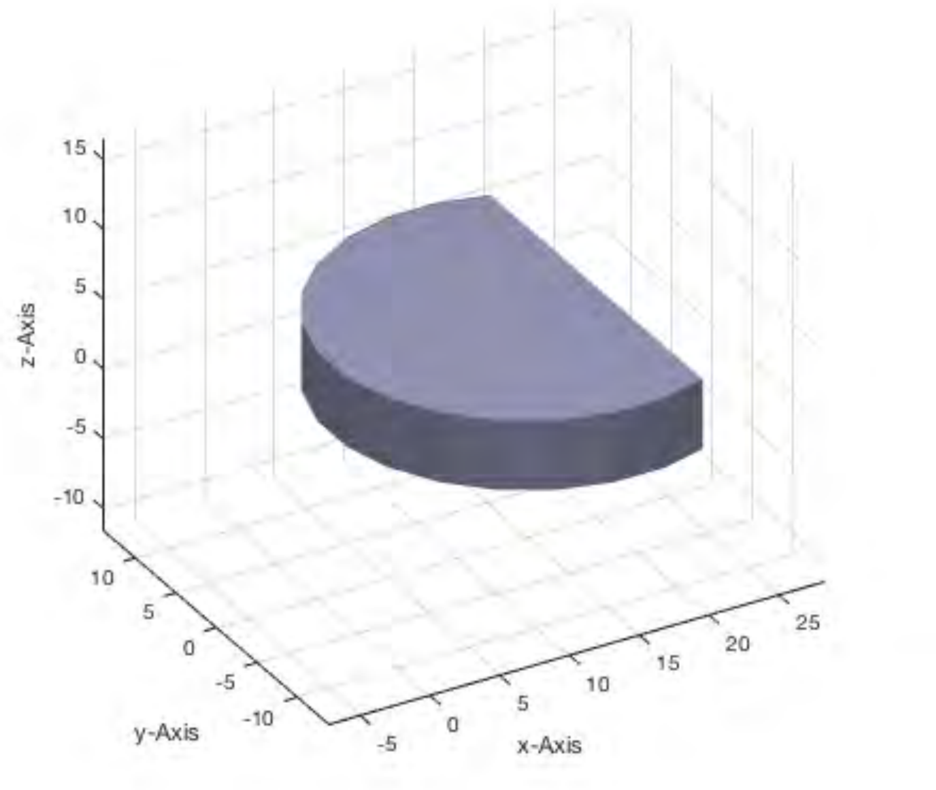
```
WRITING STL FILE /Users/lueth/Desktop/Toolbox_test/A.STL in ASCII MODE
completed.
```



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')
```

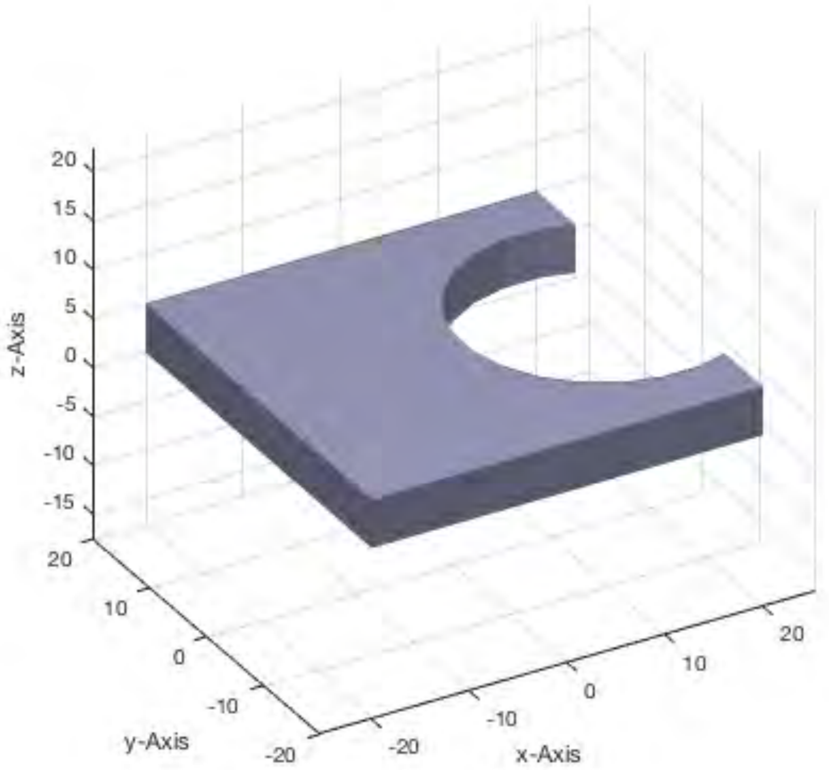
```
WRITING STL FILE /Users/lueth/Desktop/Toolbox_test/A.STL in ASCII MODE
completed.
```



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')
```

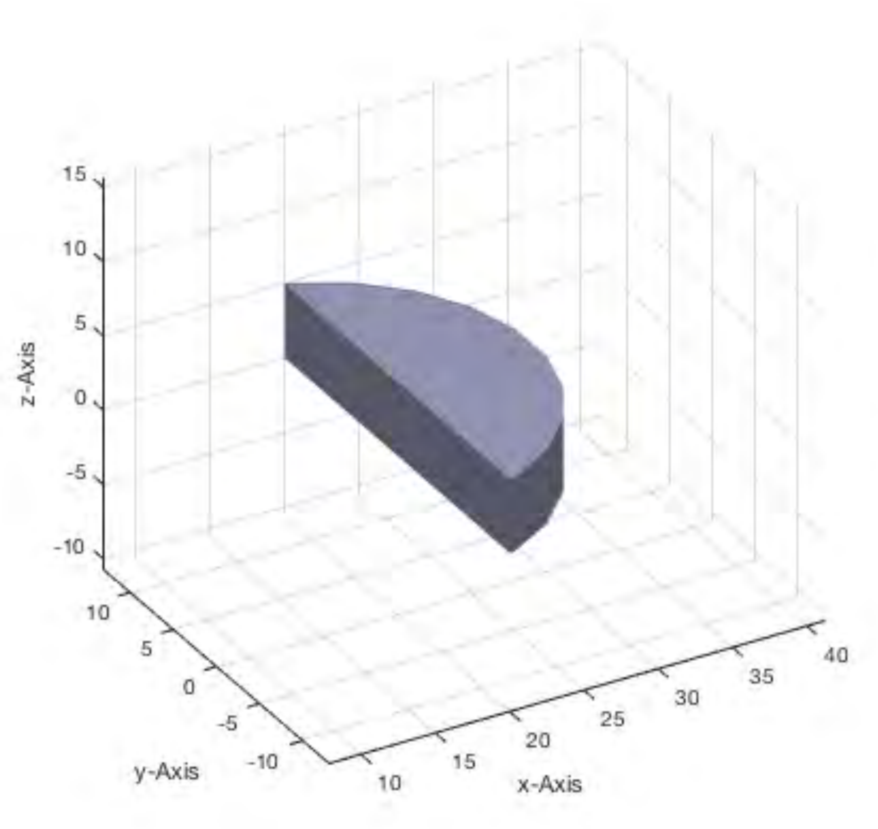
```
WRITING STL FILE /Users/lueth/Desktop/Toolbox_test/A.STL in ASCII MODE
completed.
```



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')
```

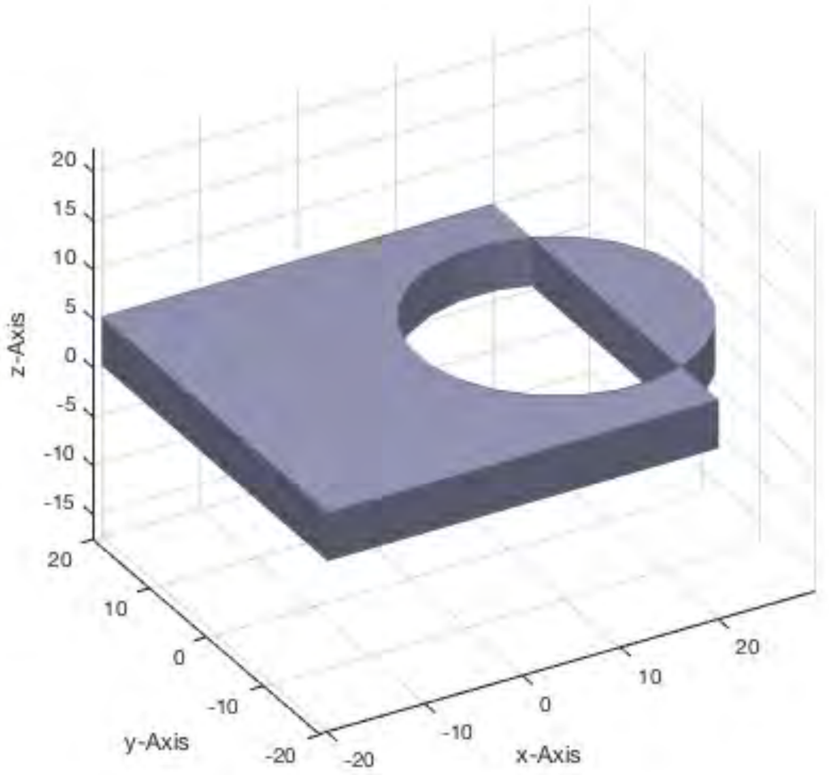
```
WRITING STL FILE /Users/lueth/Desktop/Toolbox_test/EXP04-AminusB.STL in ASCII MODE
completed.
```



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 delaunayTriangulation.
WRITING STL FILE /Users/lueth/Desktop/Toolbox_test/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.  (2018-Nov-08), 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 11-Aug-2073 20:38:15!
```

```
Executed 08-Nov-2018 20:38:17 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
simulink
video_and_image_blockset
===========================================================================================
==========
```

- *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® R2018a

Tutorial 05: Creation, Relative Positioning and Merging of Solid Geometries (SG)

2014-11-22: 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

Motivation for this tutorial: (Originally SolidGeometry 1.7 required)

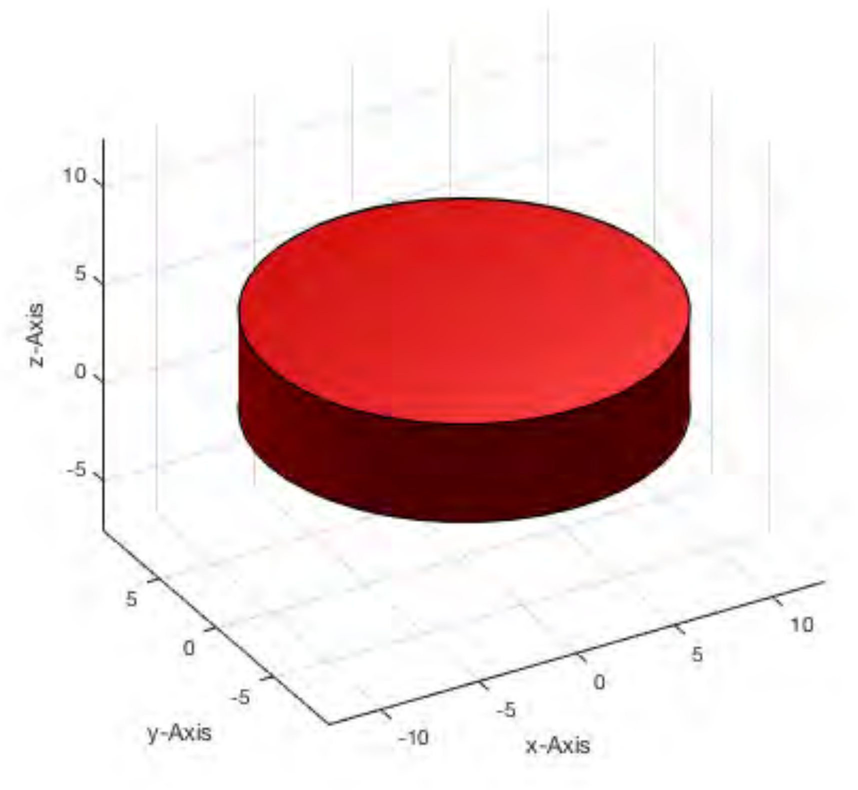
2. Creating, plotting, writing of the struct *Solid Geometry* (SG)

Even if it is useful to know that a vertex list (VL), and a facet list (FL) is required for 3D modeling, it is more convenient to use matlab structs for solid geometries (SG). Instead of writing VL or FL, we use SG.VL, SG.FL as variables. The advantage is, that each solid object is described by one struct that contains vertex list and facet list, but can contain other defined information (such as the underlying CPL, PL or EL) or it is open for your own defined information.

- **SGofCPLz** for extruding a solid object from a CPL similar to VLFLofCPLz.
- **SGplot** for plotting one or more solid objects similar VLFLplots.
- **SGchecker** for checking a solid object similar to VLFLchecker.
- **SGwriteSTL** for writing a solid object similar to VLFLwriteSTLb.
- **SGsize** for generating the bounding box of a solid geometry (SG).

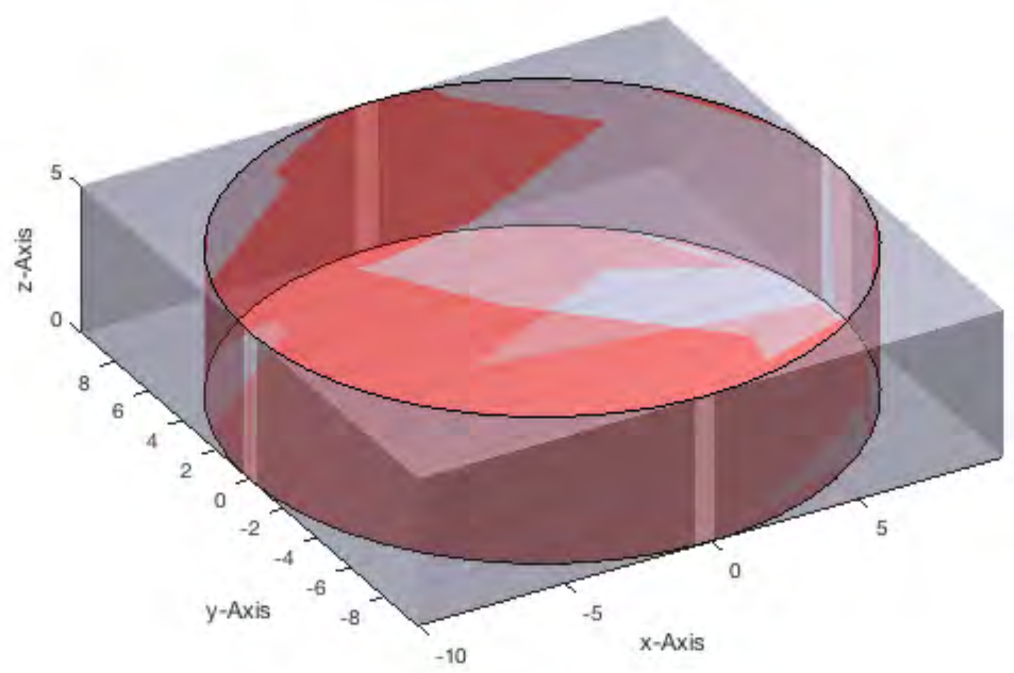
```
close all;
nSG=SGofCPLz(PLcircle(10),5)
SGplot(nSG,'r',1); VLFLplotlight(1); view (-30,30);
SGchecker(nSG);
SGwriteSTL (nSG,'EXP05-1');
```

```
nSG =

  struct with fields:

    CPL: [45×2 double]
     VL: [90×3 double]
     FL: [176×3 double]
     PL: [45×2 double]
     EL: [45×2 double]
```



Often it is useful to know the size of the bounding box of an object and to plot it.

```
bs=SGsize(nSG); [BB.VL,BB.FL]=VLFLofBB(bs); SGplot (BB,'w'); VLFLplotlight(1,0.4);
```

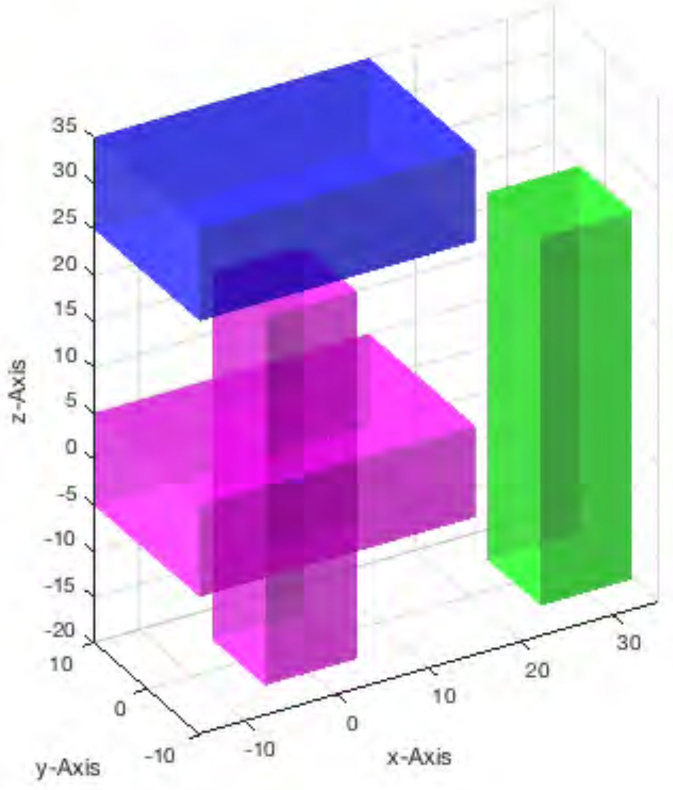


3. Spatial transformations of solid geometries and *sets of solid geometries*

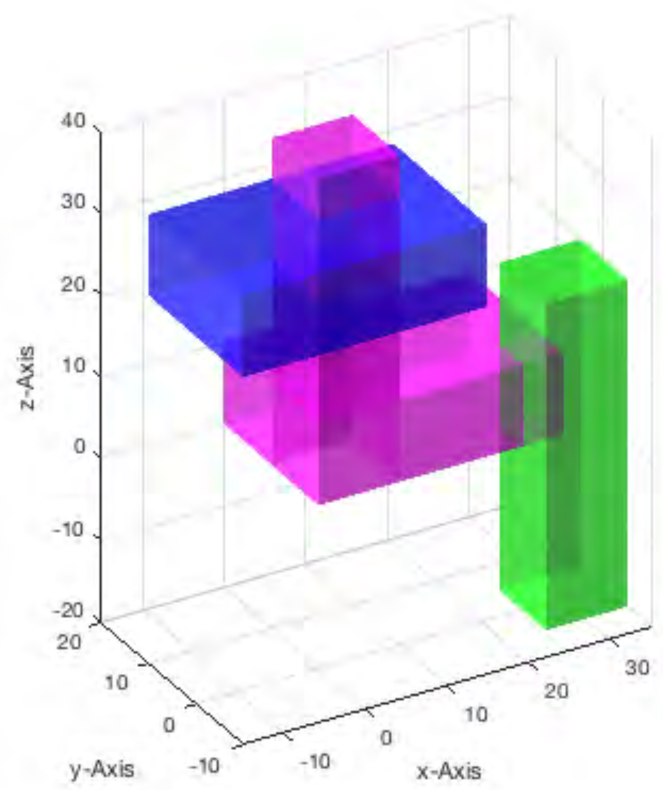
In contrast to manipulate an individual solid geometry as struct, it is often useful to manipulate or handle a set of solid geometries. For this purpose, we use the cell concept of Matlab. A=SGbox([30,20,10]); {A, A, A} is a set of three solid geometries that can be given as arguments of a function and can also be the output argument of a function.

- **SGbox** creates a simple box at the origin indimensions [x y z].
- **SGtransP** moves a solid geometry (SG) or a set of SG by a translation vector.
- **SGtransR** rotate a solid geometry (SG) or a set of SG by a rotation matrix .
- **SGtransT** transform a solid geometry (SG) or a set by a homogenous transformation matrix.
- **SGtrans0** moves a solid geometry (SG) or a set of SG into the coordinate systems origin.
- **SGtrans1** moves a solid geometry (SG) or a set of SG into quadrant 1.

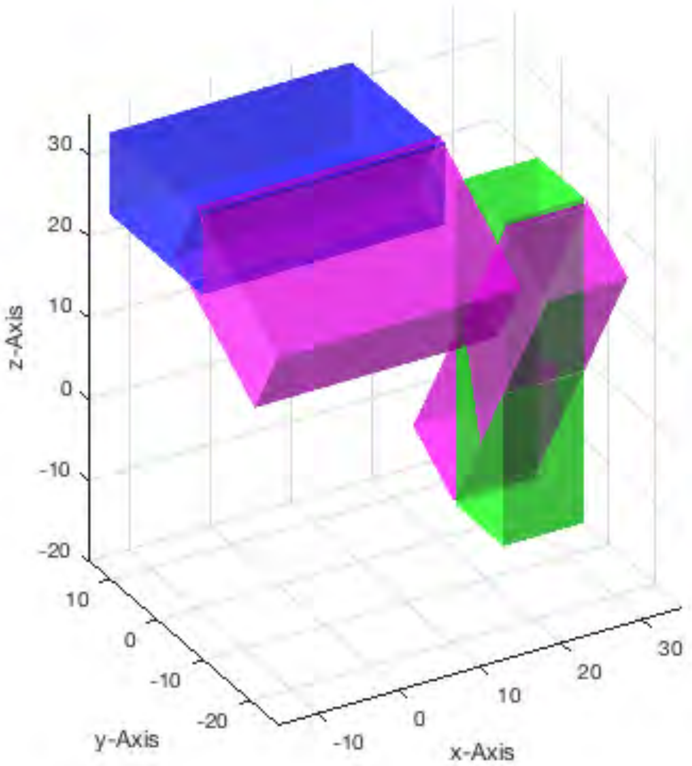
```
close all;
A=SGtransP(SGbox([30,20,10]),[0;0;30]); SGplot (A,'b'); show;
B=SGtransP(SGbox([10,10,40]),[30;0;0]); SGplot (B,'g'); show
view (-30,30);
SGplot(SGtrans0({A,B}),'m'); VLFLplotlight (1,0.5)
```



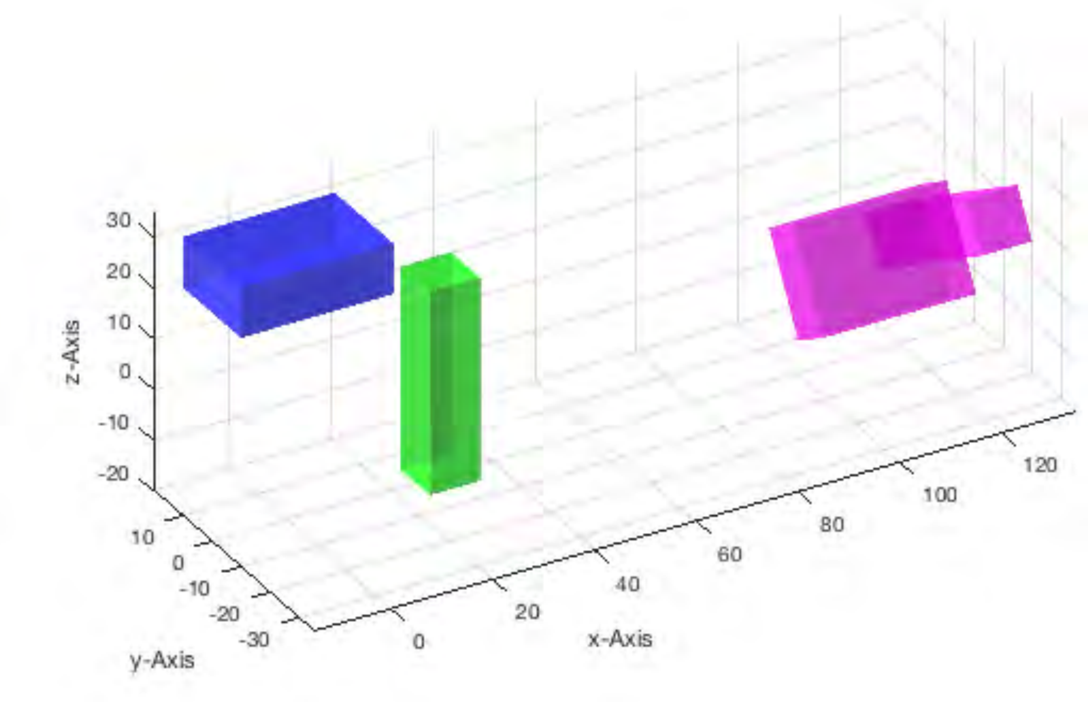

```
close all;
A=SGtransP(SGbox([30,20,10]),[0;0;30]); SGplot (A,'b'); show;
B=SGtransP(SGbox([10,10,40]),[30;0;0]); SGplot (B,'g'); show
view (-30,30);
SGplot(SGtrans1({A,B}),'m'); VLFLplotlight (1,0.5)
```




```
close all;
A=SGtransP(SGbox([30,20,10]),[0;0;30]); SGplot (A,'b'); show;
B=SGtransP(SGbox([10,10,40]),[30;0;0]); SGplot (B,'g'); show
view (-30,30);
SGplot(SGtransR({A,B},rot(pi/6,0,0)),'m'); VLFLplotlight (1,0.5)
```




```
close all;
A=SGtransP(SGbox([30,20,10]),[0;0;30]); SGplot (A,'b'); show;
B=SGtransP(SGbox([10,10,40]),[30;0;0]); SGplot (B,'g'); show
view (-30,30);
SGplot(SGtransT({A,B},[rot(pi/3,0,0),[100;0;0]]),'m'); VLFLplotlight (1,0.5)
```

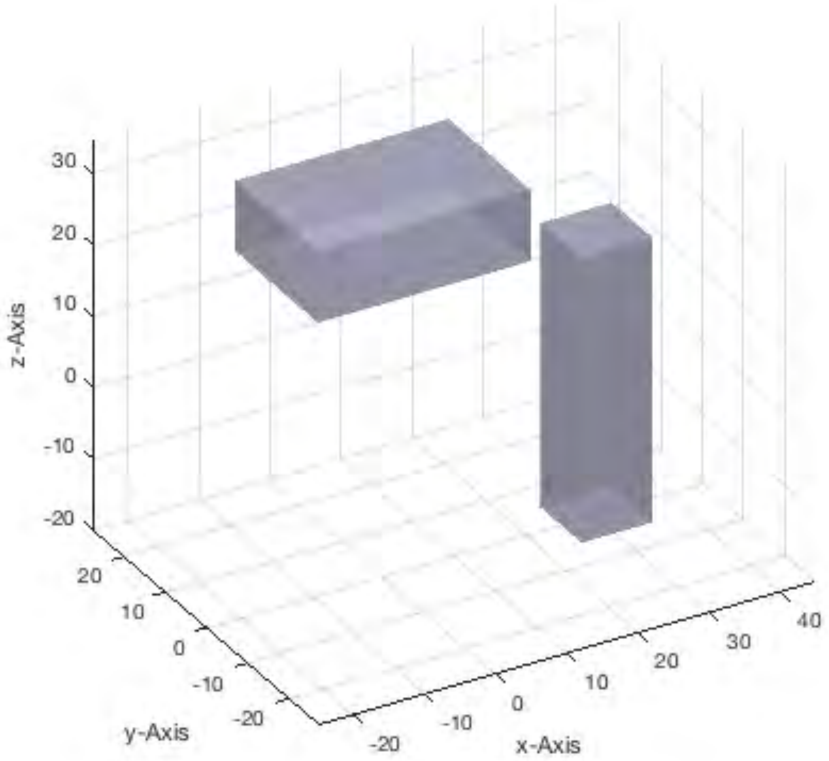


4. Merging of solid geometries (SG) and sets of solid geometries

In the previous example we often created and manipulated two solids. As explained, it is possible to handle several objects at the same time by using sets of elements. Nevertheless, in mosts cases, after some operations we want to merge several solid geometries into one single object. SGcat concatenates the vertex list and the facet list of a set og given solids into one list. Furthermore like in VLFLcat, doubled vertices are detected and removed. It is not possible anymore to separate the objects in general afterwards.

- **VLFLcat** merges two VL/FL into one VL/FL.
- **VLFLcat2** simply concatenates two VL/FL into one VL/FL.
- **SGcat** merges single solids or a set of solids into one solid object.

```
close all;
nSG=SGcat ({A,B}); SGplot (nSG,'w'); view (-30,30); VLFLplotlight (1,0.5)
SGwriteSTL (nSG,'EXP05-2');
```

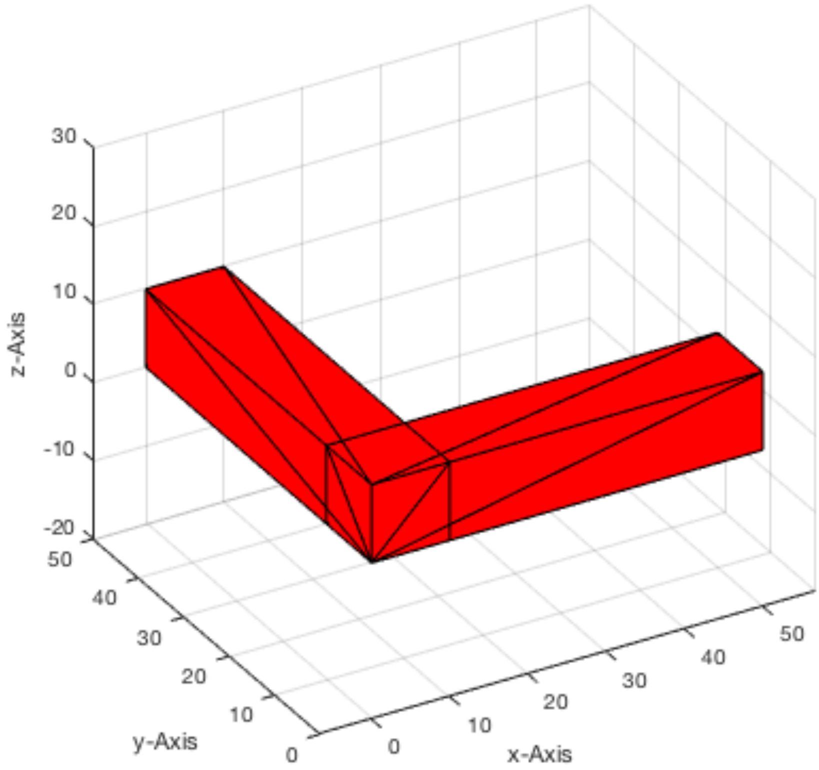


5. Non-manifold points, edges, and facets of solid geometries (SG)

By using functions such as SGcat or VLFLcat/VLcat2, it is very easy and efficient to create solid models and STL-files by simply attaching or penetrating individual solid objects. It is some kind of *additive design of solid objects* in 3D. For a 3D printing process, those additive designed objects are not a real problem, i.e. several independent parts are simply attached or penetrate each other. 3D contour printing of penetrating objects is automatically handled by the slicer, a piece of software that we get to know later.

Nevertheless, as soon as a vertex is used by two independent solids, an edge is used by two independent solids. In this case a slicer software will not be able to solve the manifold problem.

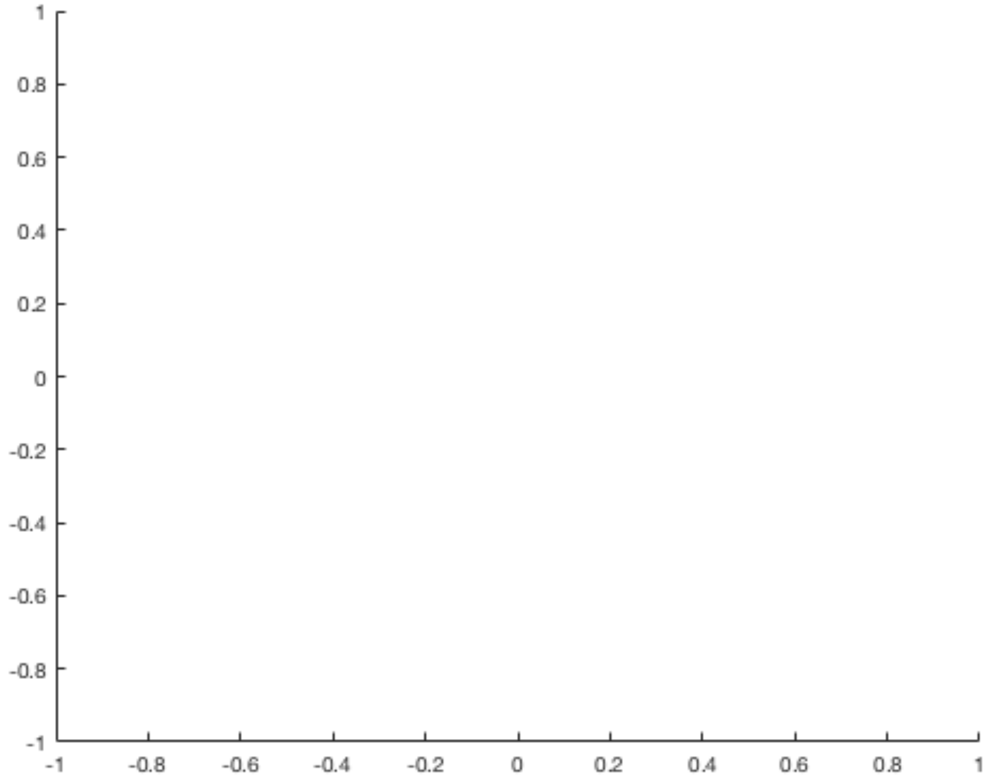
```
close all;
A=SGtrans1(SGbox([10,50,10])); B=SGtrans1(SGbox([50,10,10]));
nSG=SGcat({A,B});  SGplot (nSG); view (-30,30);
[VL,EL,PEL]=SGchecker (nSG);
if ~isempty(VL); VLELplots (VL(:,2:4),PEL,'m*-',4); VLFLplotlight (1,0.7); end
```



If we call SGchecker with a second argument ('plot'), we get a figure showing the non manifold objects that generate the conflict

```
close all; SGchecker (nSG,'plot');
```

```
2 edges [blue] are doubled, not removed
```



6. Additive Design: Separate or penetrate solid geometries (SG)

During additive solid geometry design, we will always get problems with non-manifold points or edges, as long as we try always to align objects point-to-point, edge-to-edge or face-to-face. As a basic rule, it is better to shorten or increase the length of a object slighly by a micrometer and do not align it with another face, edge, point. Even if the number of points of a solid geometry is increased by this strategy, the number of facets of this solid is decreased. Additive solid geoemtry design is therefore, not an inefficient but an efficient design methodology.

```
close all;
slot=1e-3;
A=SGtransl(SGbox([10,50,10]));
B=SGtransl(SGbox([50,10,10])); B=SGtransP(B,[slot;0;0]);
nSG=SGcat({A,B});
SGchecker (nSG,'plot');
```

The value for shifting the object about 1 micrometer is much lower than the manufacturing accuracy of the 3D printer. Anyway, if this would not be the case, then simply change it to 1 nanometer (1e-6) or one picometer (1e-9) if we consider a millimeter as default integer unit. It is also clear that we can automate the correction by simply splitting the objects and adding a random submicrometer value to the coordinates.

Another possibility is separating the objects instead of penetrating them. This will lead to the same solution to avoid non manifold edges. Nevertheless, some manufacturing preprocessors analyze STL-Files and detect objetcs that are separated and do not penetrate each other. These objects are then separated and repositioned in the 3D printing working volume to optimize the use of the print job's working volumen and material use.

The later presented function for relative spatial alignment of solid geometries will support a parameter for a gap between objects. Negative gap sizes correspond to a slightly penetration of the solid geometries.

Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:38:25!
Executed 08-Nov-2018 20:38:27 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
simulink
video_and_image_blockset
===========================================================================================
==========
```

- *Tim Lueth, tested and compiled on OSX 10.7.5 with Matlab 2014b on 2014-11-23*
- *Tim Lueth, executed and published on 64 Bit PC using Windows with Matlab 2014b on YYYY-MMM-DD*

Published with MATLAB® R2018a

Tutorial 06: Relative Positioning and Alignment of Solid Geometries (SG)

2014-11-24: 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

Motivation for this tutorial: (Originally SolidGeometry 1.8 required)

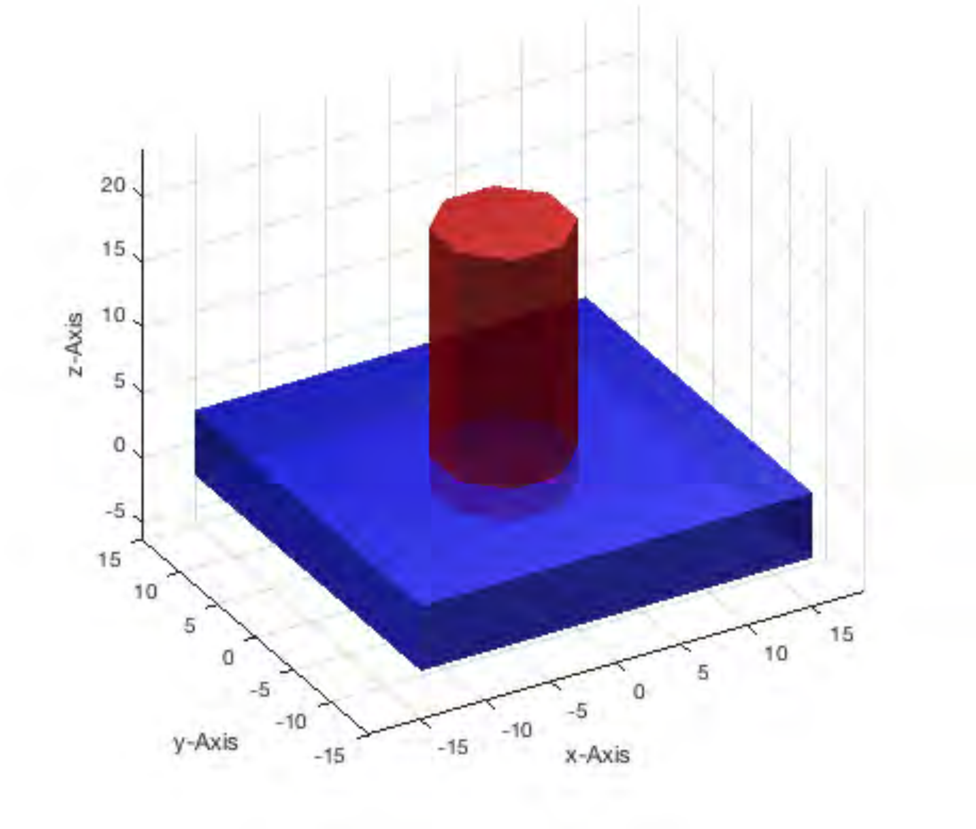
2. Relative spatial positioning of solid geometries (SG) using bounding boxes

Since it is quite convenient to use solid geometries (SG), there is a need for relative spatial positioning of these objects. For 'on top', 'under' (modifies the z-coordinates), 'in front', 'behind' (modifies the y-coordinates), 'left' and 'right' (modifies the x-coordinates), we have six different positioning functions that generate copies of the SG with just a changed vertex list. A third parameter of those functions is a gap, that can be defined. A positive gap value means a separation of thoses solids, a negative gap value means a penetration of those solids.

- **SGontop** positions a solid geometry 'A' on top of solid geometry 'B'
- **SGunder** positions a solid geometry 'A' under of solid geometry 'B'
- **SGinfront** positions a solid geometry 'A' in front of solid geometry 'B'
- **SGbehind** positions a solid geometry 'A' behind of solid geometry 'B'
- **SGleft** positions a solid geometry 'A' left of solid geometry 'B'
- **SGright** positions a solid geometry 'A' right of solid geometry 'B'

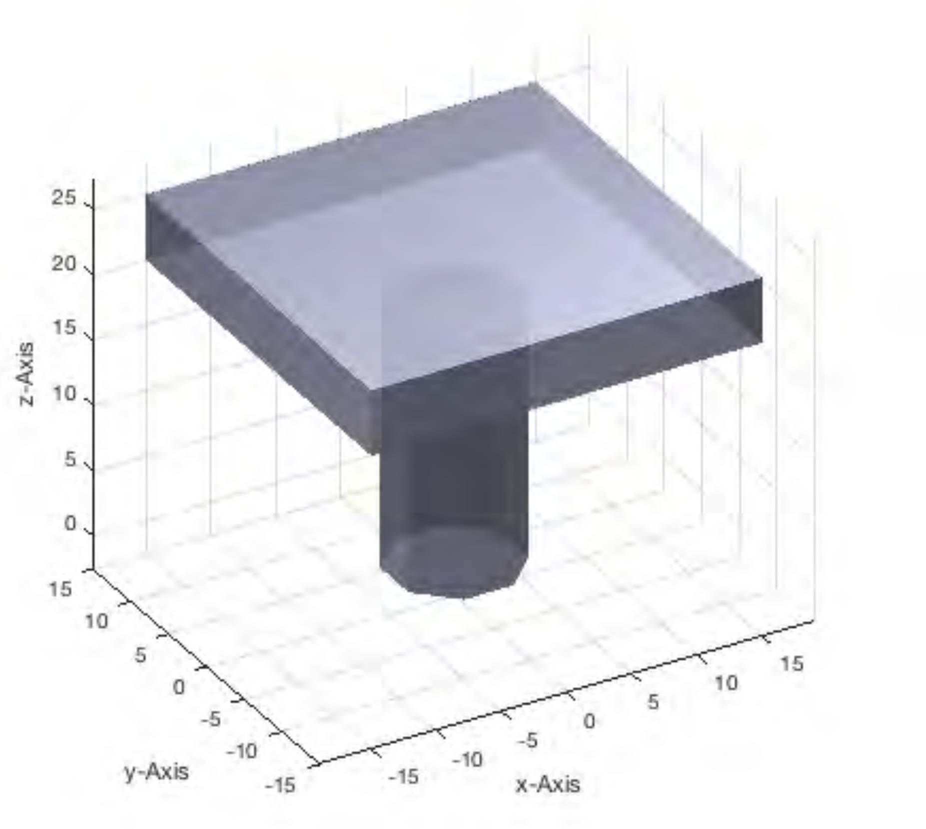
Define two solid geometrys 'A' and 'B'

```
close;
A=SGbox([30,30,5]); B=SGofCPLz(PLcircle(5,8),20);
SGplot (A,'b'); SGplot (B,'r'); VLFLplotlight(1,0.7); view (-30,30);
```



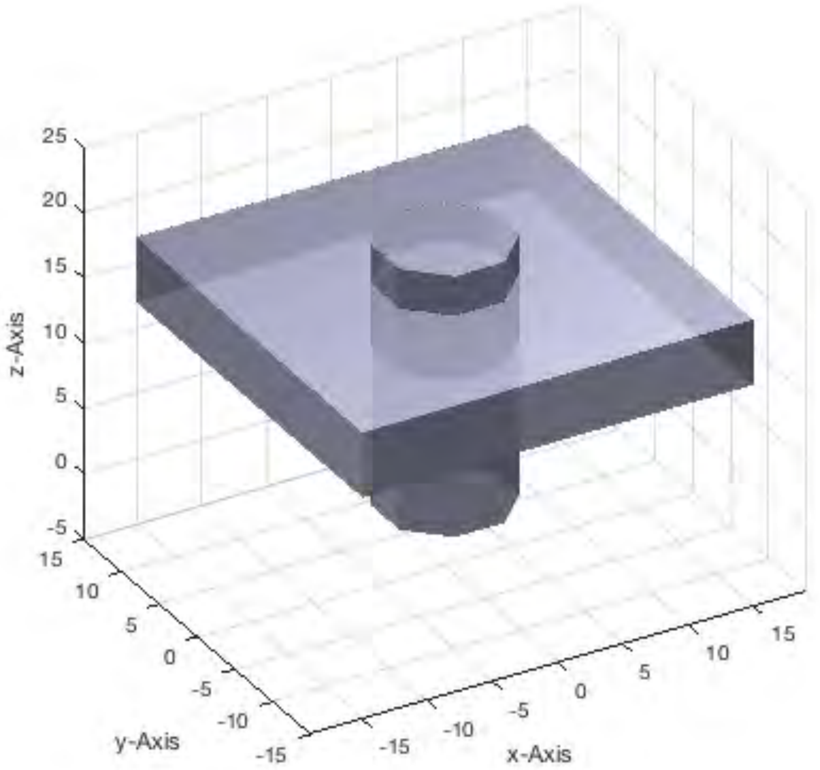
SGontop positions a solid geometry 'A' on top of solid geometry 'B'

```
close;
SG=SGcat(SGontop(A,B),B);
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



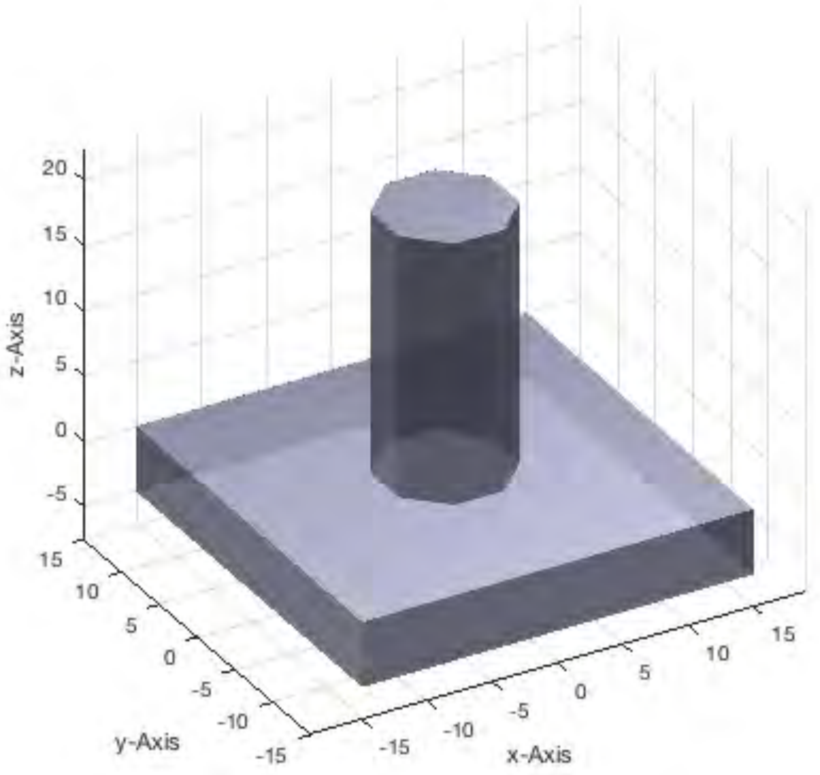
SGontop positions a solid geometry 'A' on top of solid geometry 'B' with a gap of -8

```
close all;
SG=SGcat(SGontop(A,B,-8),B);
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



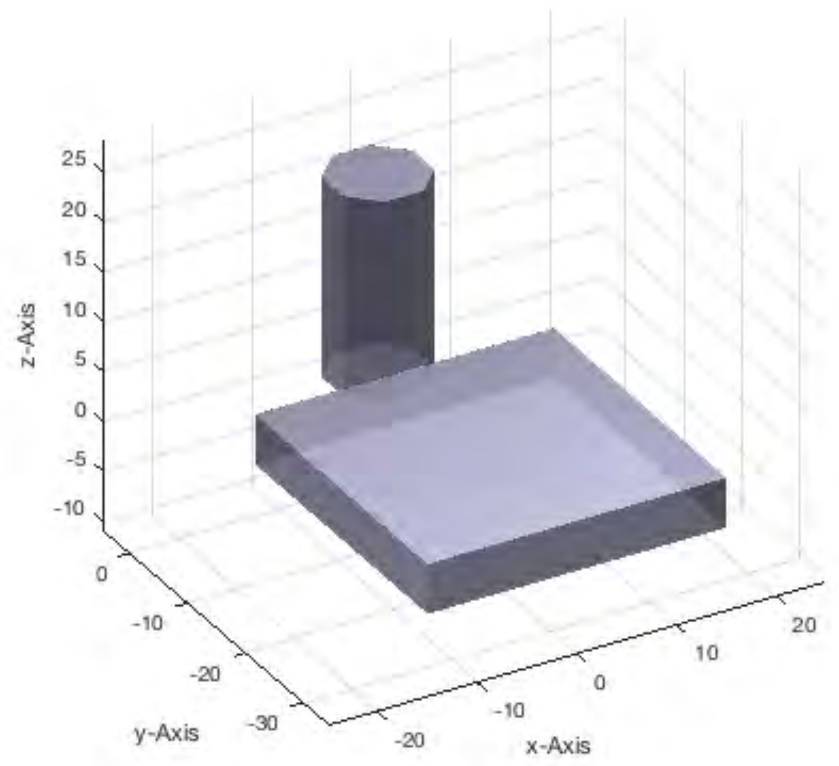
SGunder positions a solid geometry 'A' under of solid geometry 'B'

```
close all;
SG=SGcat(SGunder(A,B),B);
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



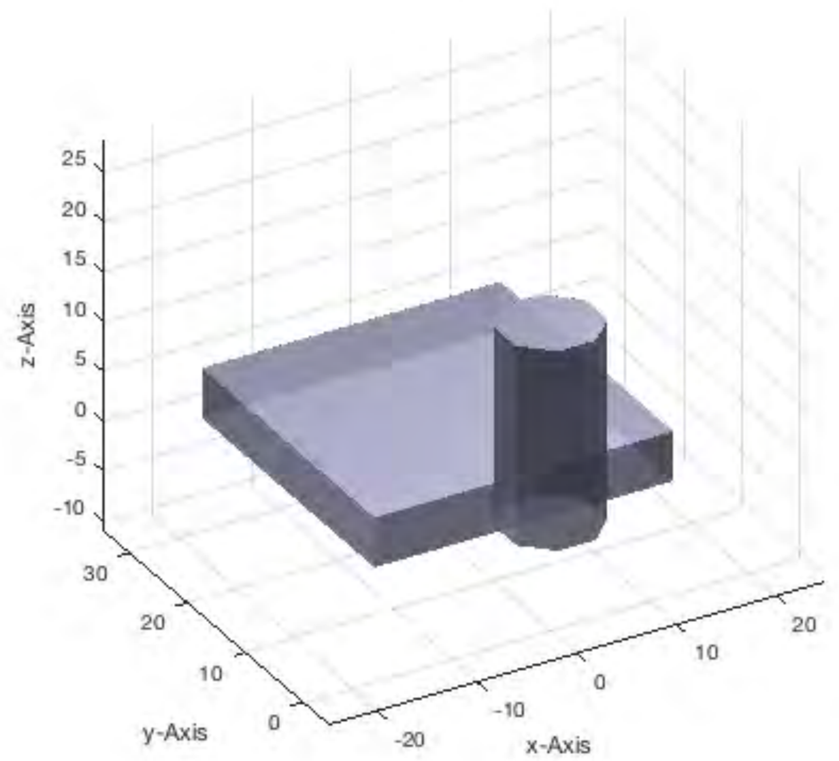
SGinfront positions a solid geometry 'A' in front of solid geometry 'B'

```
close all;
SG=SGcat(SGinfront (A,B),B);
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



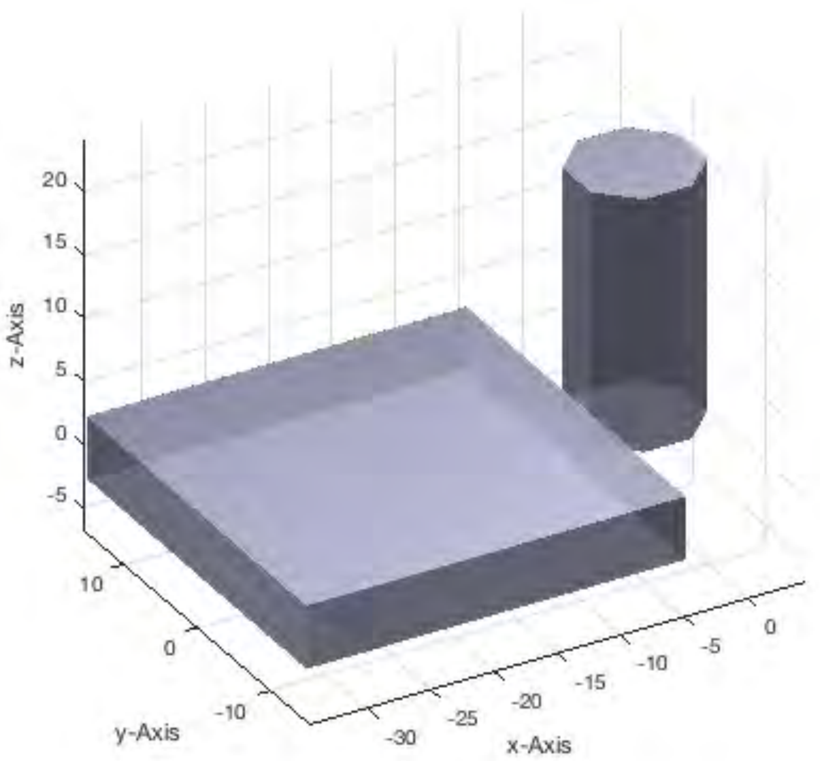
SGbehind positions a solid geometry 'A' behind of solid geometry 'B'

```
close all;
SG=SGcat(SGbehind (A,B),B);
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



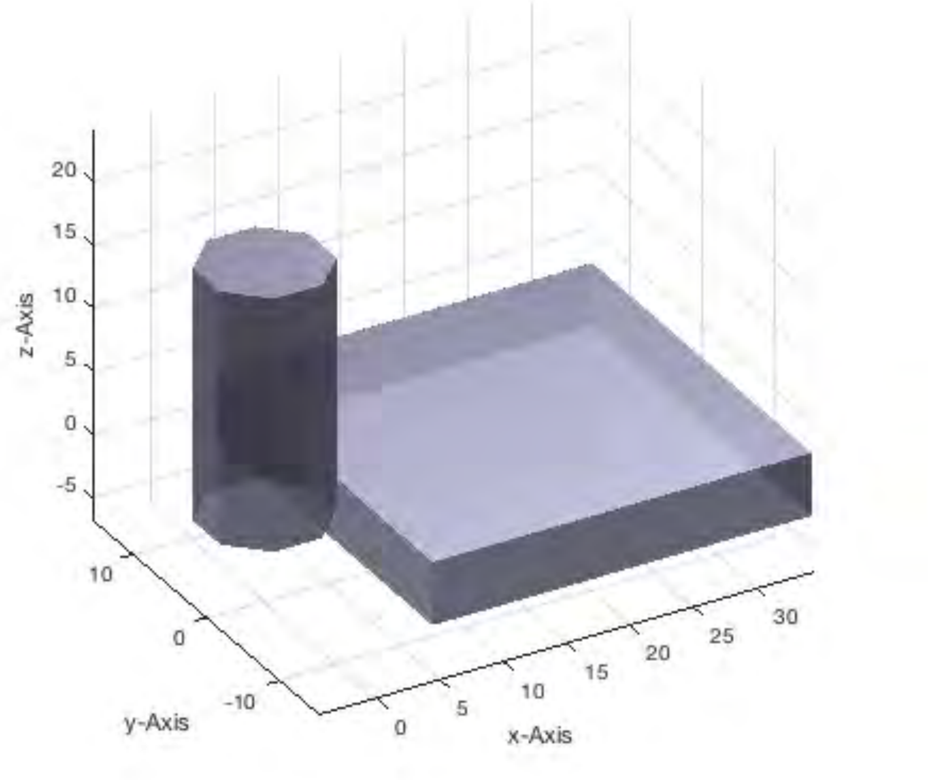
SGleft positions a solid geometry 'A' left of solid geometry 'B'

```
close all;
SG=SGcat(SGleft (A,B),B);
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



SGright positions a solid geometry 'A' right of solid geometry 'B'

```
close all;
SG=SGcat(SGright (A,B),B);
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



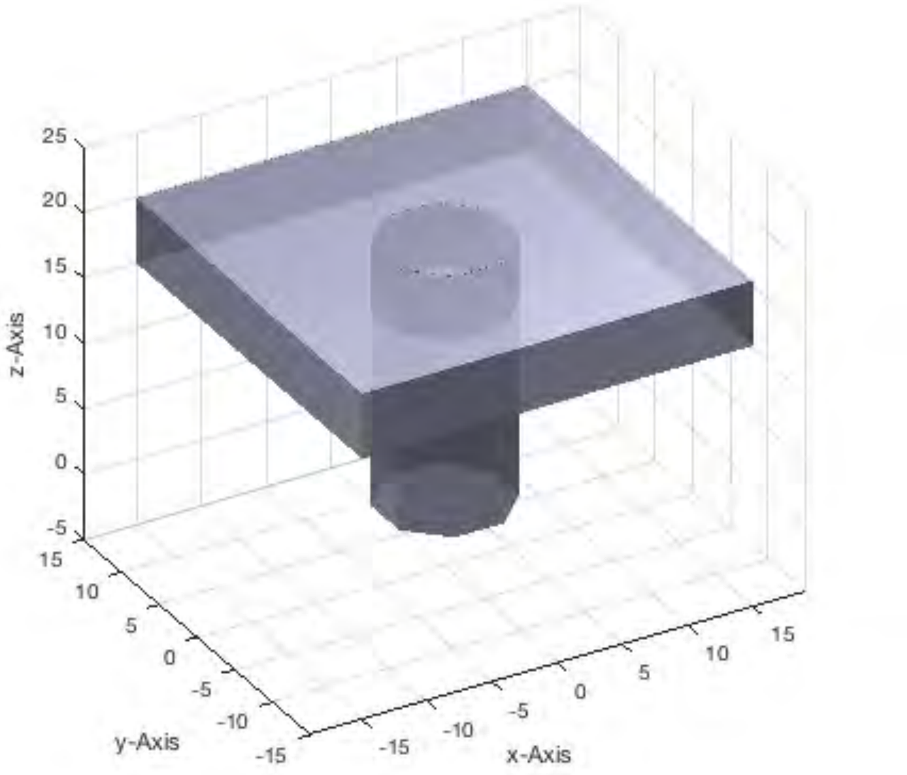
3. Relative spatial alignment of solid geometries (SG) using bounding boxes

Similar to the relative positioning, also the spatial alignment is helpful. For example solid A is aligned with solid B to achive the same 'top', 'bottom' (modifies the z-coordinates), 'front', 'back' (modifies the y-coordinates), 'left side' or 'right side' (modifies the x-coordinates).

- **SGaligntop** aligns the top of solid A with the top of solid B
- **SGalignbottom** aligns the bottom of solid A with the bottom of solid B
- **SGalignfront** aligns the front of solid A with the front of solid B
- **SGalignback** aligns the back of solid A with the back of solid B
- **SGalignleft** aligns the left side of solid A with the left side of solid B
- **SGalignright** aligns the right side of solid A with the right side of solid B

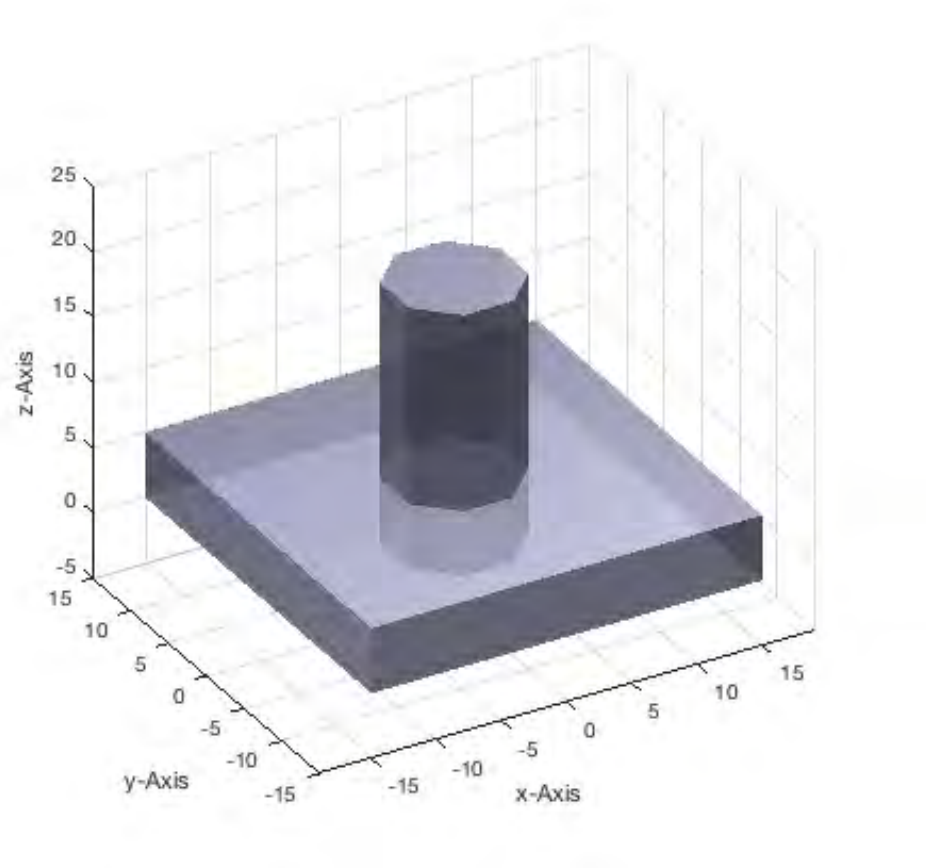
SGaligntop aligns the top of solid A with the top of solid B

```
close all;
SG=SGcat({SGaligntop(A,B),B});
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



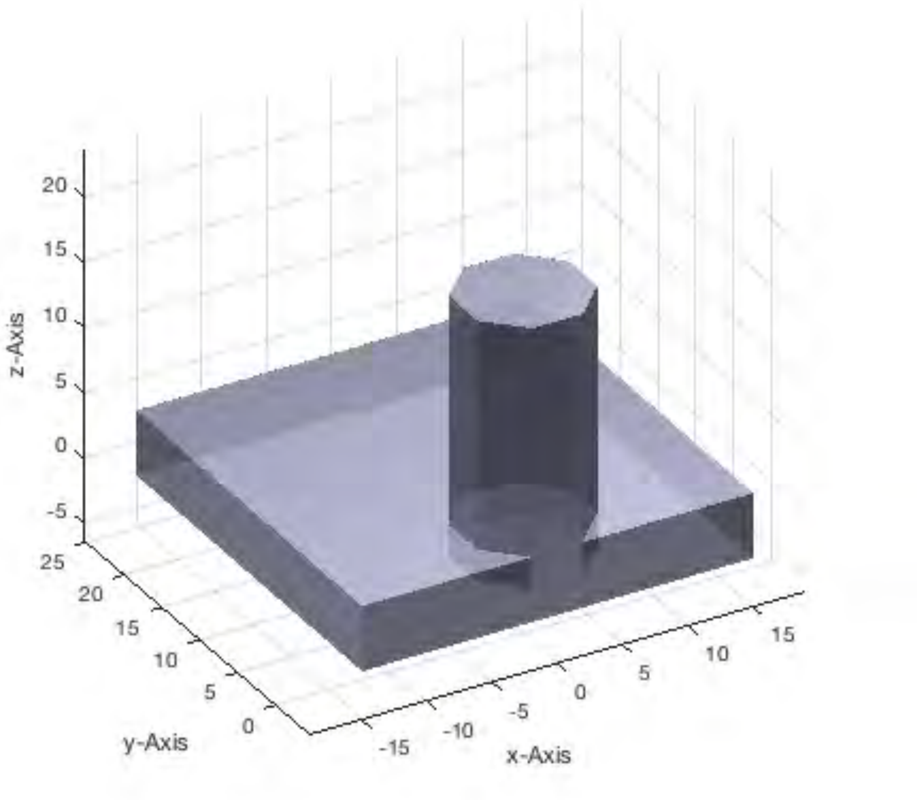
SGalignbottom aligns the bottom of solid A with the bottom of solid B

```
close all;
SG=SGcat({SGalignbottom(A,B),B});
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



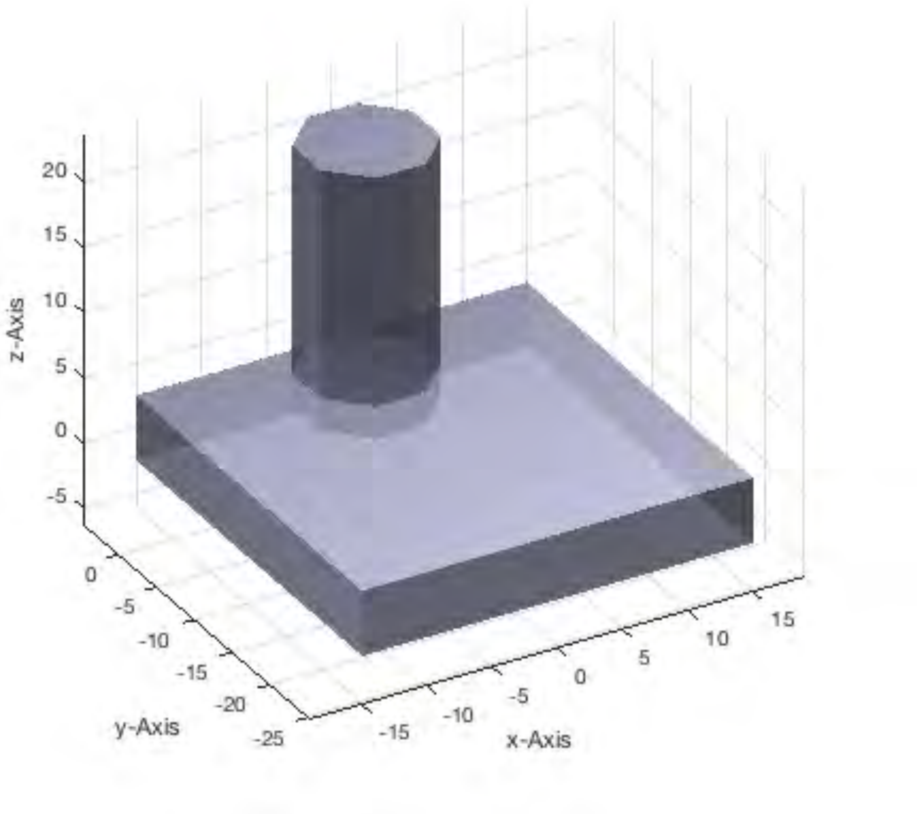
SGalignfront aligns the front of solid A with the front of solid B

```
close all;
SG=SGcat({SGalignfront(A,B),B});
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



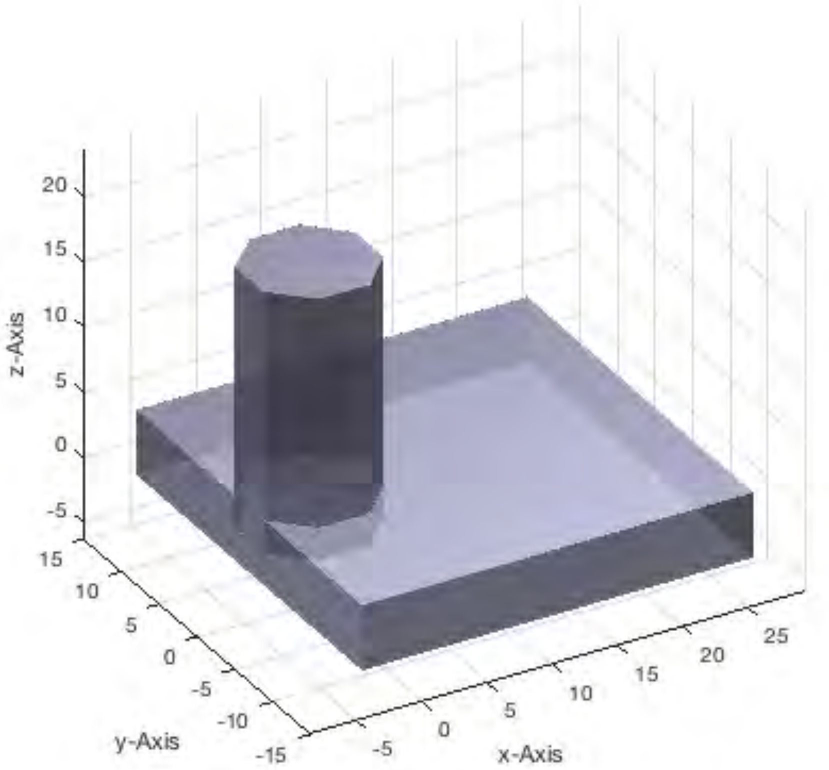
SGalignback aligns the back of solid A with the back of solid B

```
close all;
SG=SGcat({SGalignback(A,B),B});
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



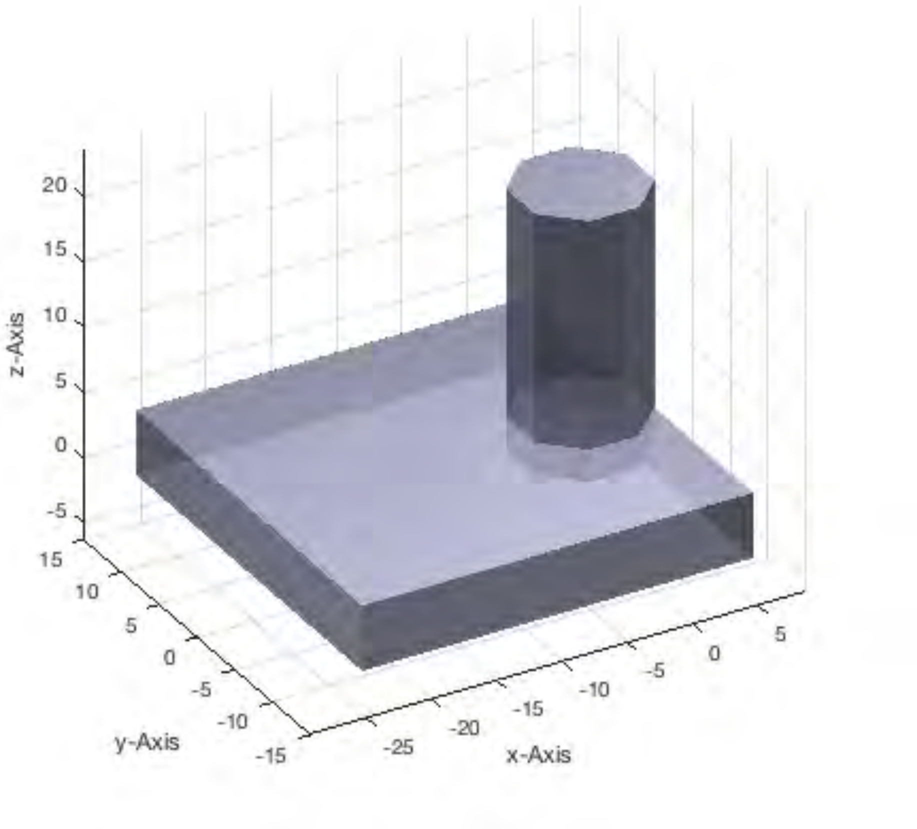
SGalignleft aligns the left side of solid A with the left side of solid B

```
close all;
SG=SGcat({SGalignleft(A,B),B});
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



SGalignright aligns the right side of solid A with the right side of solid B

```
close all;
SG=SGcat({SGalignright(A,B),B});
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:38:35!
Executed 08-Nov-2018 20:38:37 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
simulink
video_and_image_blockset
===========================================================================================
==========
```

- *Tim Lueth, tested and compiled on OSX 10.7.5 with Matlab 2014b on 2014-11-25*
- *Mattias Traeger, executed and published on 64 Bit PC using Windows with Matlab 2014b on YYYY-MMM-DD*

Published with MATLAB® R2018a

Tutorial 07: Rotation of Closed Polygon Lists for Solid Geometry Design

2014-11-26: 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)

Motivation for this tutorial: (Originally SolidGeometry 2.0 required)

2. List of functions used in this example

As we learned in example 2 and 4, it is possible to extrude a planar point list (PL) or closed polygon (CPL) list into a 2.5D solid geometry. Now, we will rotate a CPL around th z-axis. In this case, we consider the CPL or the PL always as a x/z-list. Using closed polygon lists, we have to remember that before extruding them or rotating them it is necessary to guarantee that the outer contour has a counter-clockwise order (ccw).

In this example, some new functions are introduced:

- CPLplot to draw the closed polygon list in the x/y plane.
- PLELofCPL to draw the direction, starting point and end point.
- CPLuniteCPL to unite several CPL into one and adapt their original directions.
- SGofCPLrot to rotate a contour around the z-axis

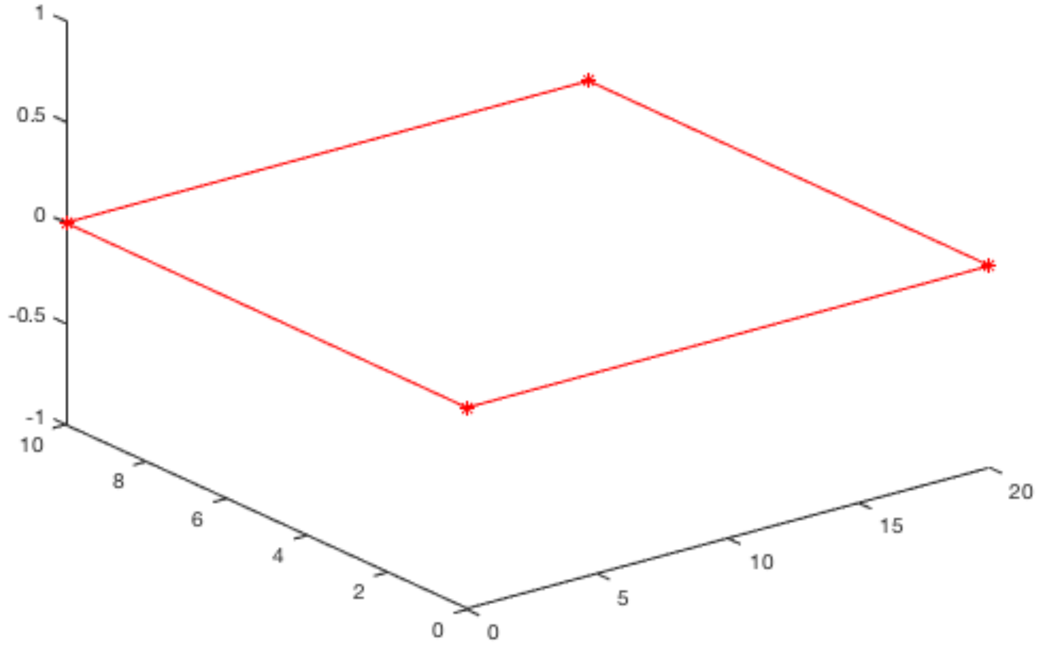
3. Rotation of closed polygon lists (CPL)

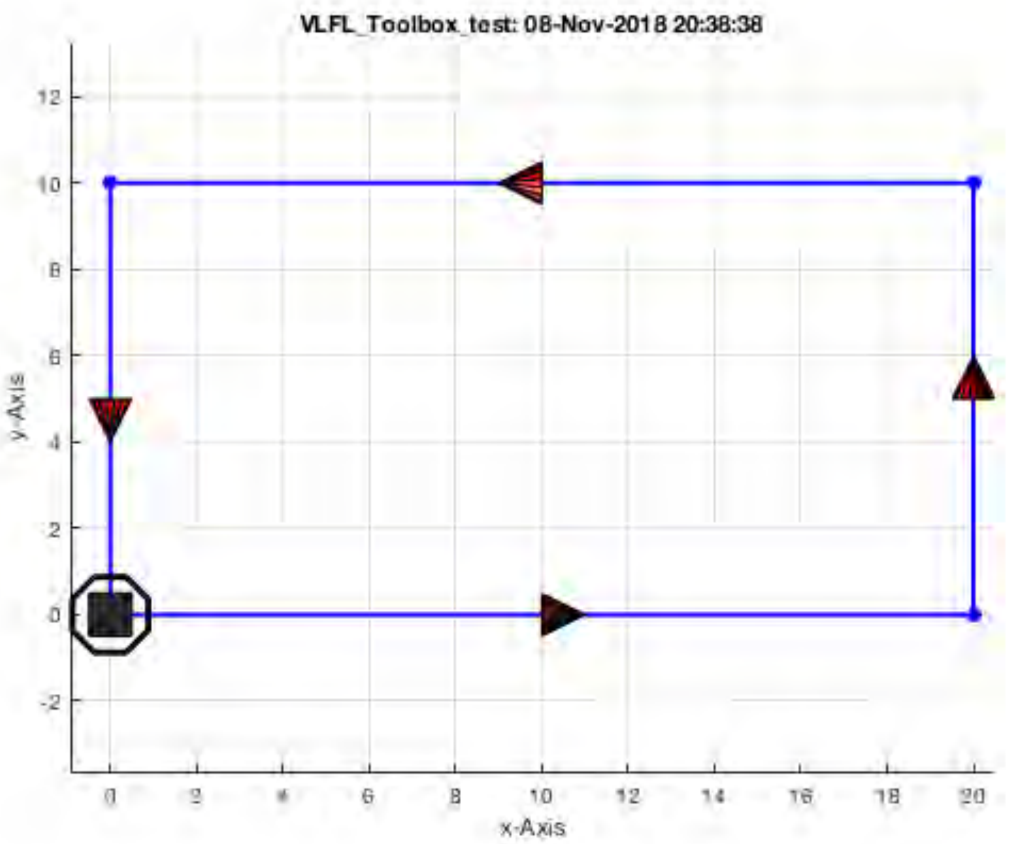
For the rotation of a simple contour we use the following functions

- **CPLplot** to draw the closed polygon list in the x/y plane.
- **PLELofCPL** to draw the direction, starting point and end point.
- **SGofCPLrot** to rotate a contour around the z-axis

Exercise: Create a simple point list that touches the y-axis

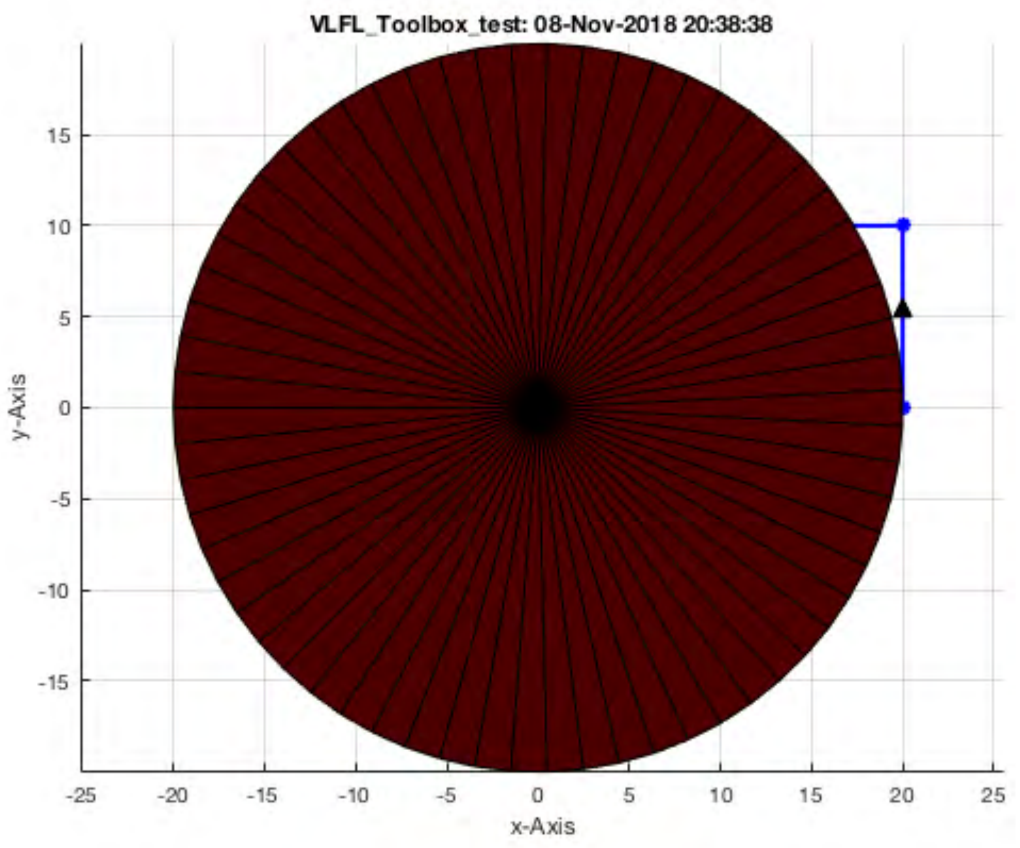
```
close all;
CPL=[0 0; 20 0; 20 10; 0 10];        % Create a simple rectangle (ccw)
CPLplot(CPL);                        % plot the rectangle
PLELofCPL(CPL);                      % show edges and directions
```





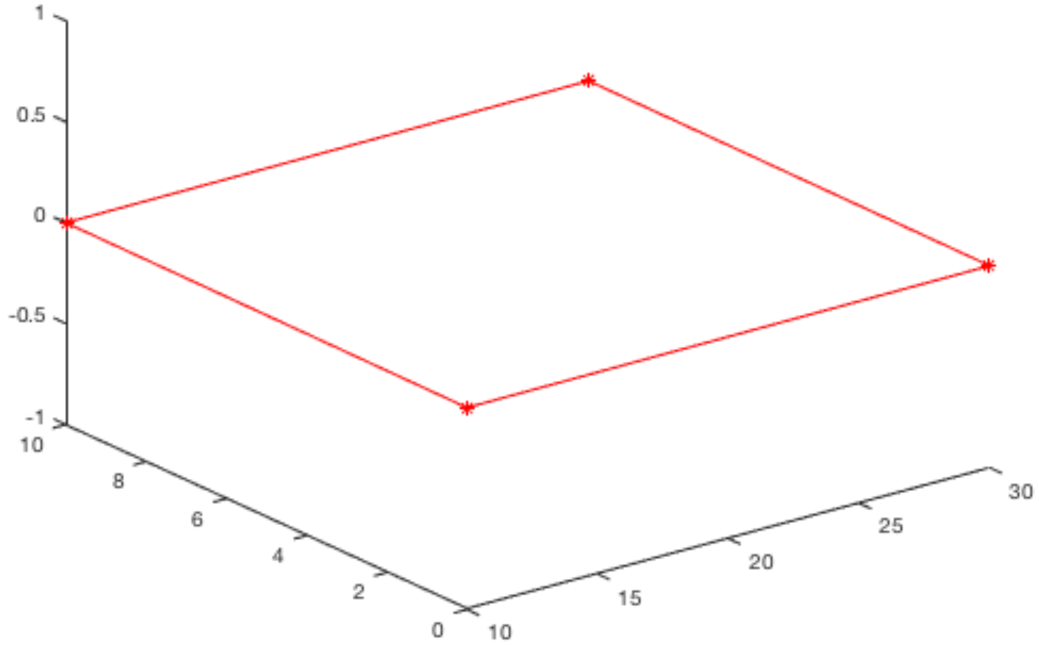
Exercise: Rotate the point list around the z-axis to create a cylinder

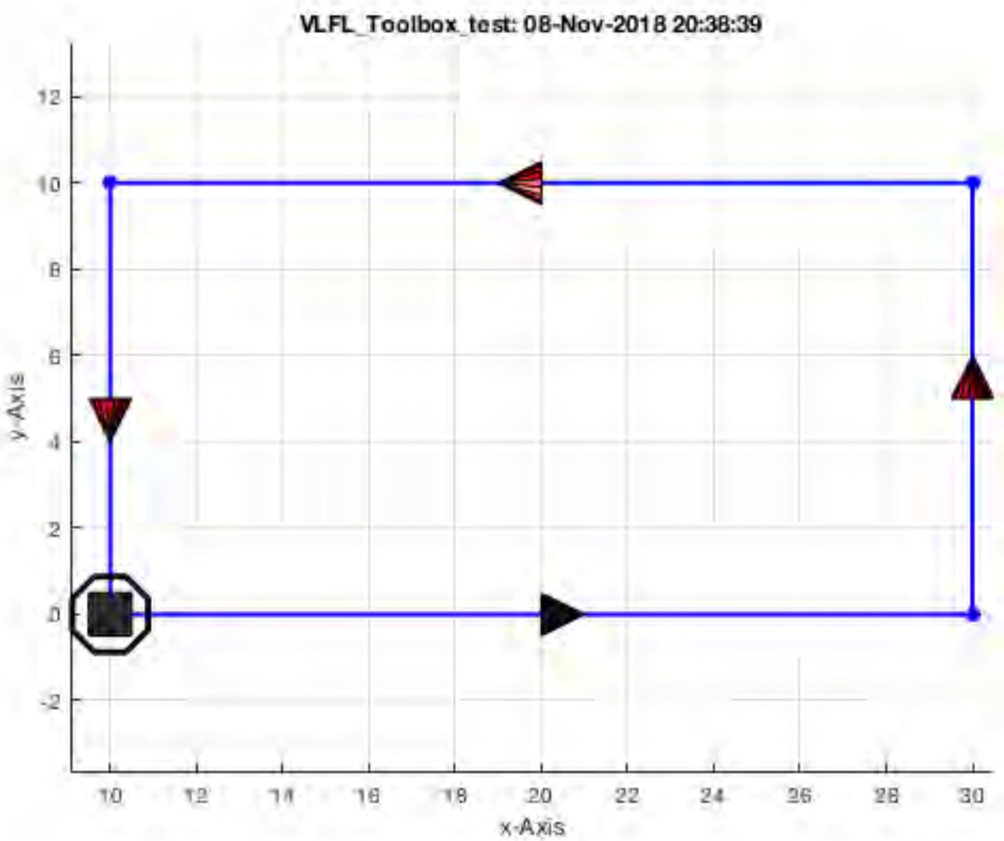
```
SG=SGofCPLrot(CPL);                 % Solid contour rotation
SGplot(SG);                         % show the solid
```



Exercise: Create a simple point list with distance to the y-axis

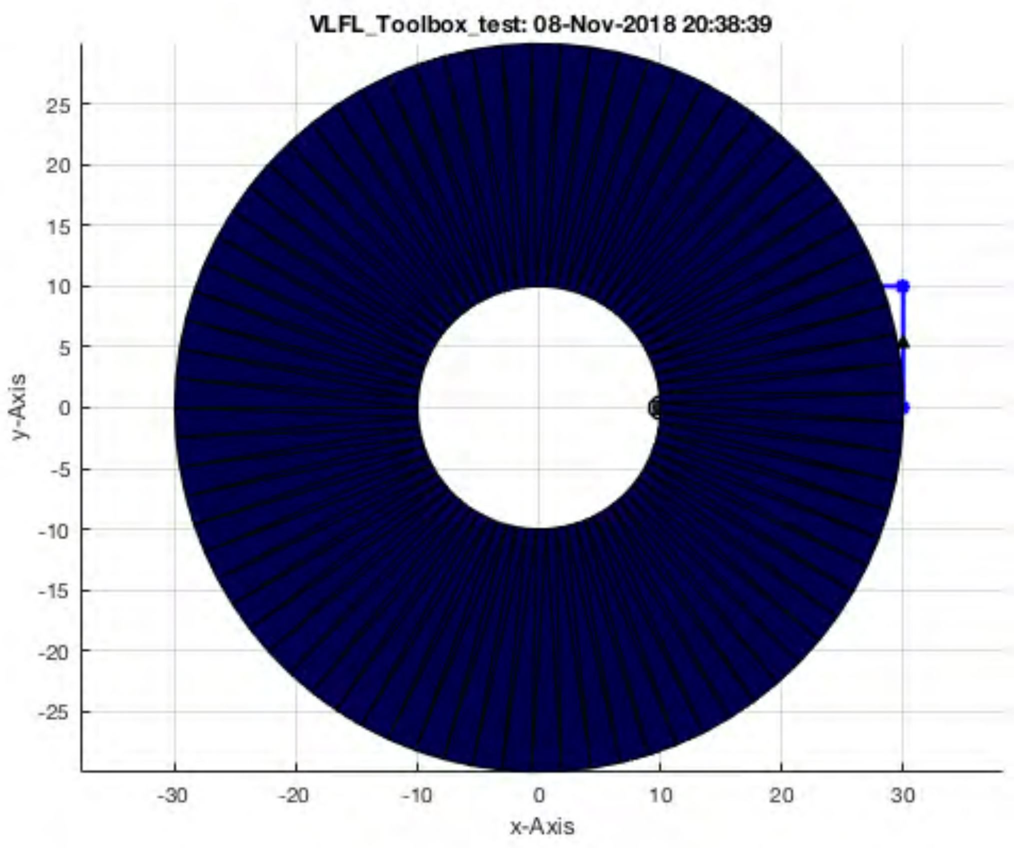
```
close all;
CPL=[0 0; 20 0; 20 10; 0 10];       % Create a simple rectangle (ccw)
CPL(:,1)=CPL(:,1)+10;               % shift by 1 on the x-axis
CPLplot(CPL);                       % plot the rectangle
PLELofCPL(CPL);                     % show edges and directions
```





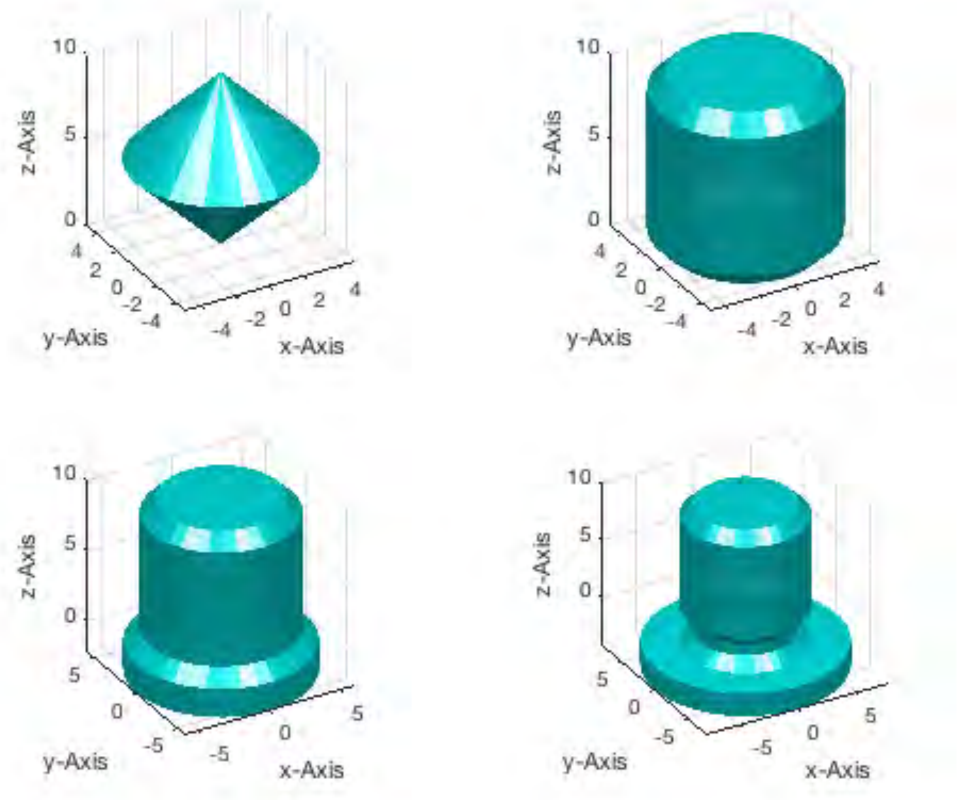
Exercise: Rotate the point list around the z-axis to create a hollow cylinder

```
SG=SGofCPLrot(CPL);                 % Solid contour rotation
SGplot(SG,'b');
```



Exercise: Some other examples for massiv rotational symetric solids

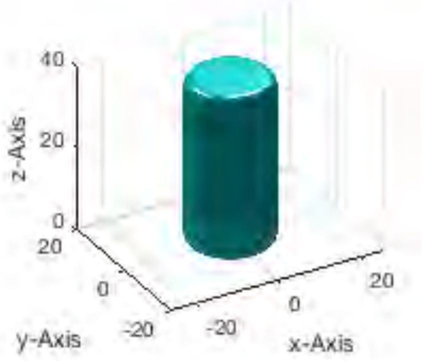
```
SG=SGofCPLrot([0 0; 5 5; 0 10]);                     % Solid contour rotation
subplot(2,2,1); view (-30,30); SGplot(SG,'c'); VLFLplotlight (1,0.9);
SG=SGofCPLrot([0 0; 4 0;  5 1; 5 9; 4 10; 0 10;]); % Solid contour rotation
subplot(2,2,2); view (-30,30); SGplot(SG,'c'); VLFLplotlight (1,0.9);
SG=SGofCPLrot([0 -2; 6 -2; 6 0;  5 1; 5 9; 4 10; 0 10;]); % Solid contour rotation
subplot(2,2,3); view (-30,30); SGplot(SG,'c'); VLFLplotlight (1,0.9);
SG=SGofCPLrot([0 -4; 8 -4; 8 -2; 5 -2; 4 -1; 4 0;  5 1; 5 9; 4 10; 0 10;]); % Solid contour
 rotation
subplot(2,2,4); view (-30,30); SGplot(SG,'c'); VLFLplotlight (1,0.9);
```

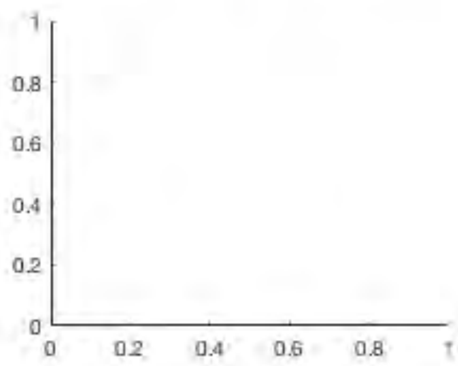


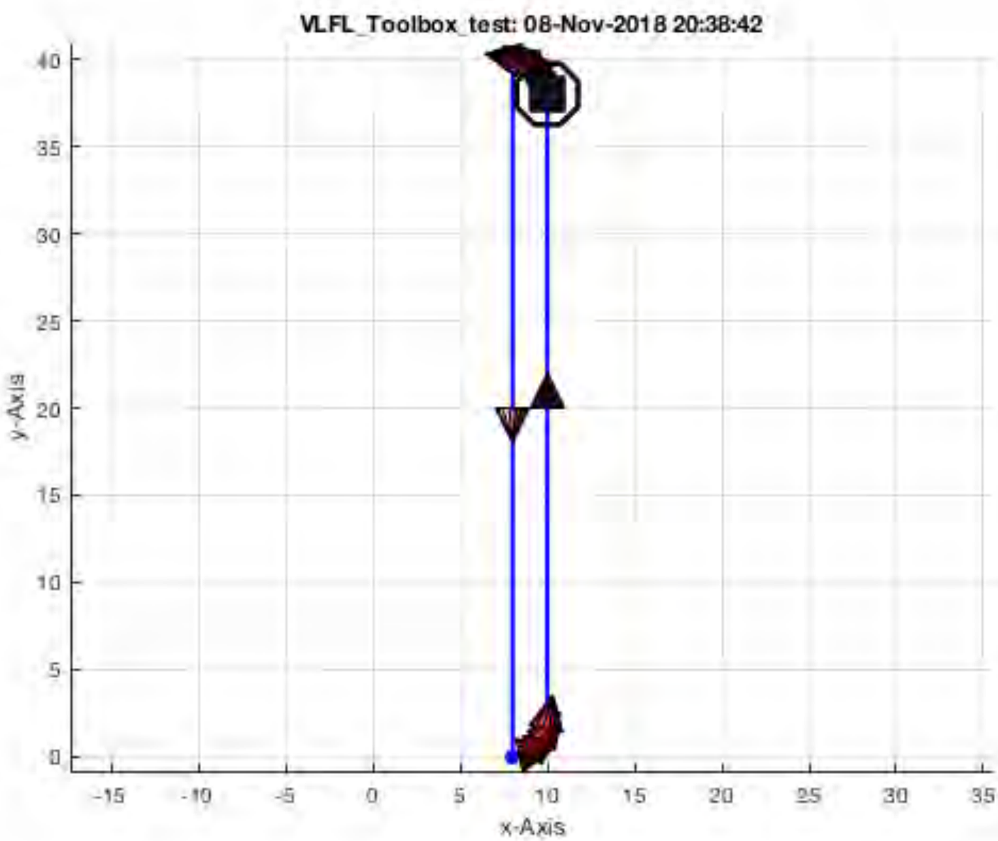
The warnings 'Removed n(m) facets' can be ignored. These warning appear if a part of the contour touches or crosses the x=0 line (y-axis).

Exercise: Creating a bold and a sleeve

```
closeall;
r=2; H=40; R=10;

PL=PLcircseg (r,[],0,pi/2);  CPL=PLtransP(PL,[R-r,H-r]);
PL=PLcircseg (r,[],-pi/2,0); CPL=[CPL;0 H; 0 0;PLtransP(PL,[R-r,r])];
SG=SGofCPLrot(CPL);                  % Solid contour rotation

subplot(2,2,1); SGplot(SG,'c'); VLFLplotlight (1,0.9); view (-30,30);
subplot(2,2,3); PLELofCPL (CPL);

PL=PLcircseg (r,[],0,pi/2);  CPL=PLtransP(PL,[R-r,H-r]);
PL=PLcircseg (r,[],-pi/2,0); CPL=[CPL;PLtransP(PL,[R-r,r])];
SG=SGofCPLrot(CPL);
subplot(2,2,2); SGplot(SG,'c'); VLFLplotlight (1,0.9); view (-30,30);
subplot(2,2,4); PLELofCPL (CPL);
```



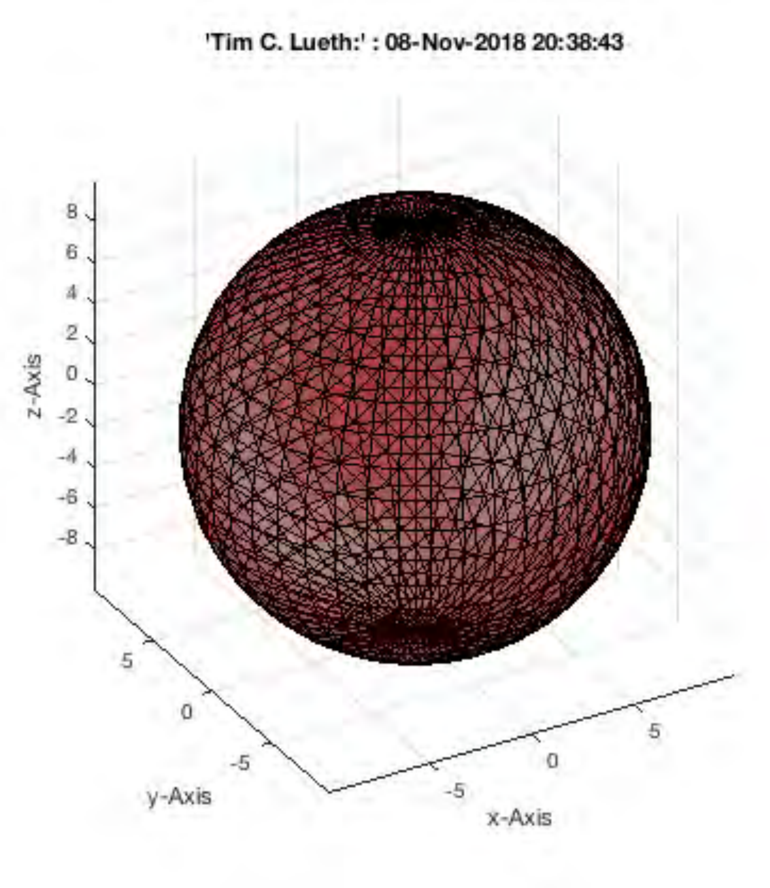




4. Creating spheres by rotating half-circles

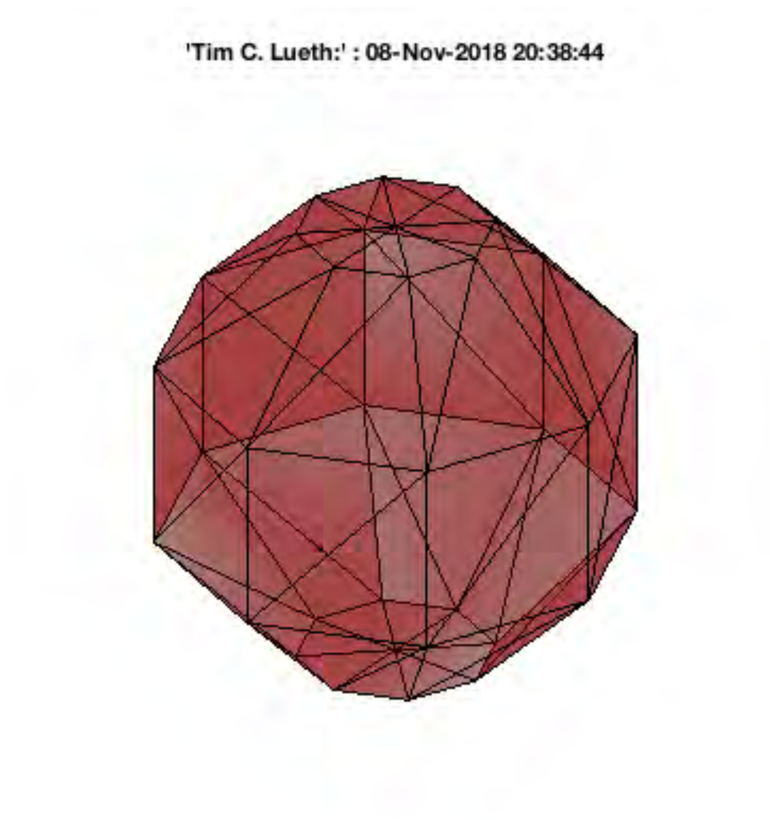
Exercise: Creating a full sphere

```
close all;
PL=PLcircle(10);

VLFLfigure; view(-30,30); grid on;
SG=SGofCPLrot(PL);
SGplot(SG); VLFLplotlight (0,0.5);
```



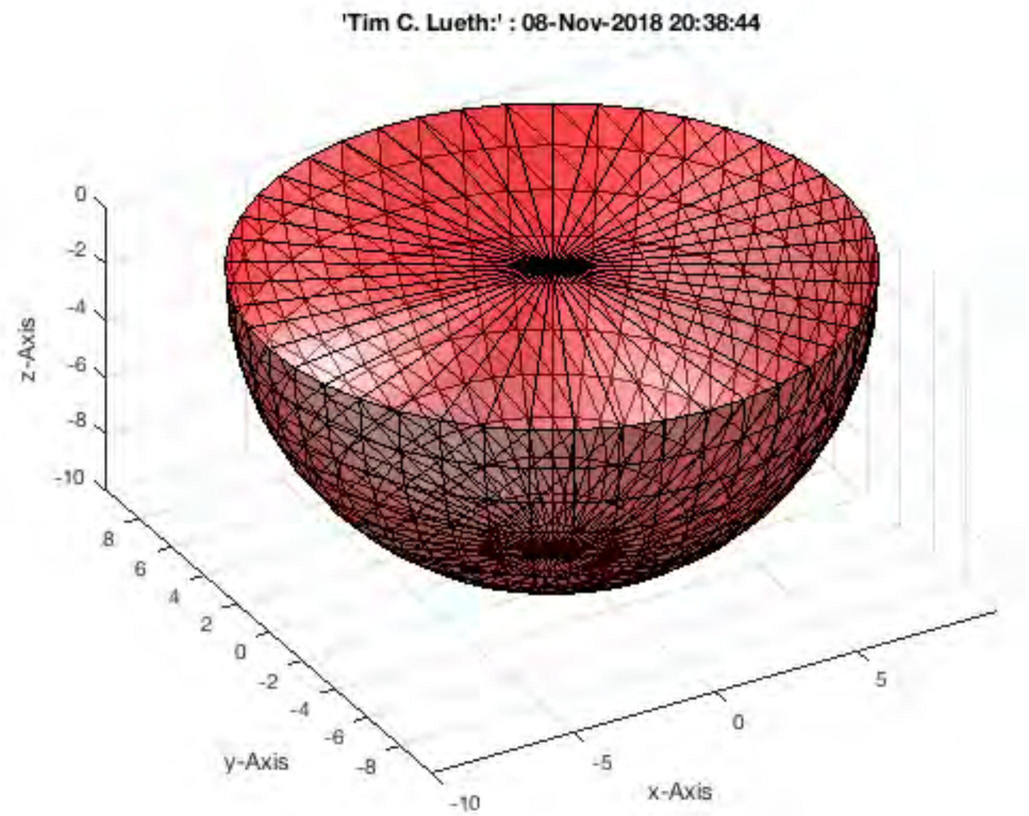
Exercise: Creating a 8 by 8 sphere

```
close all;
PL=PLcircle(10,8);

VLFLfigure; view(-30,30); grid on;
SG=SGofCPLrot(PL,8);
SGplot(SG); VLFLplotlight (0,0.5);
```



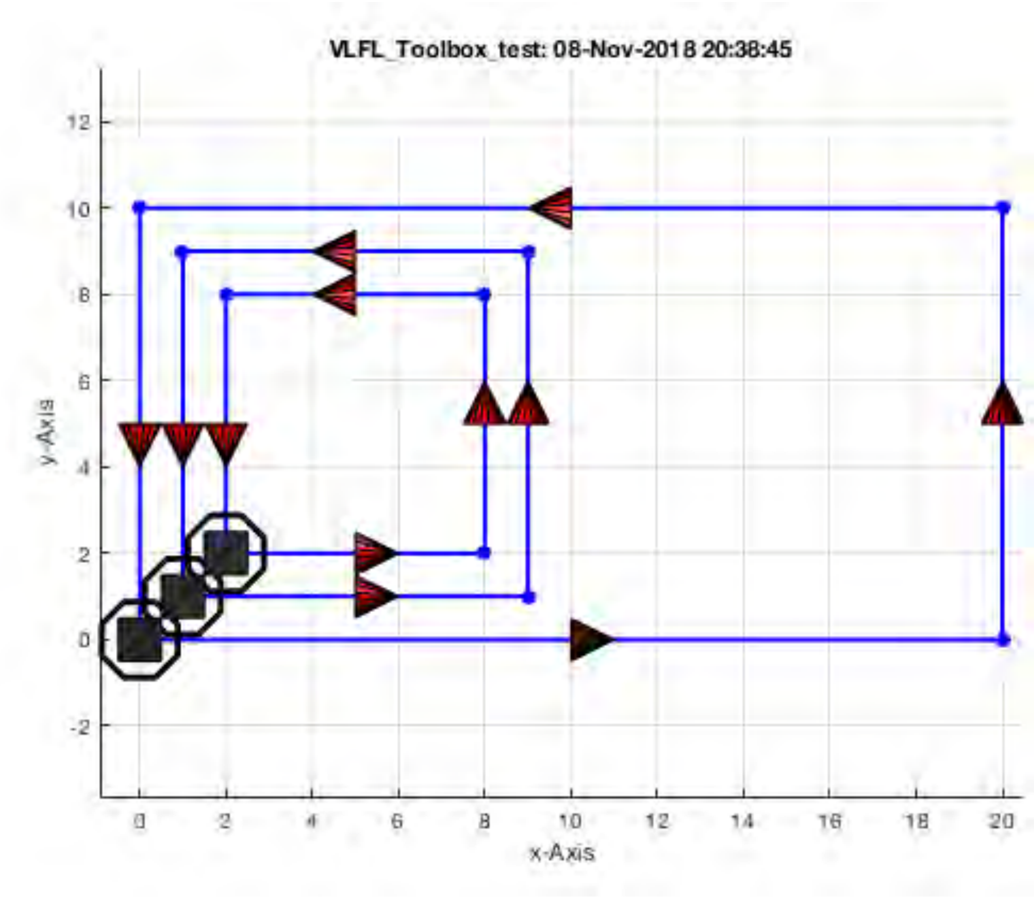
Exercise: Creating a half sphere

```
close all;
PL=[PLcircseg(10,[],-pi/2,0); 0 0];

VLFLfigure; view(-30,30); grid on;
SG=SGofCPLrot(PL);
SGplot(SG); VLFLplotlight (0,0.5);
```

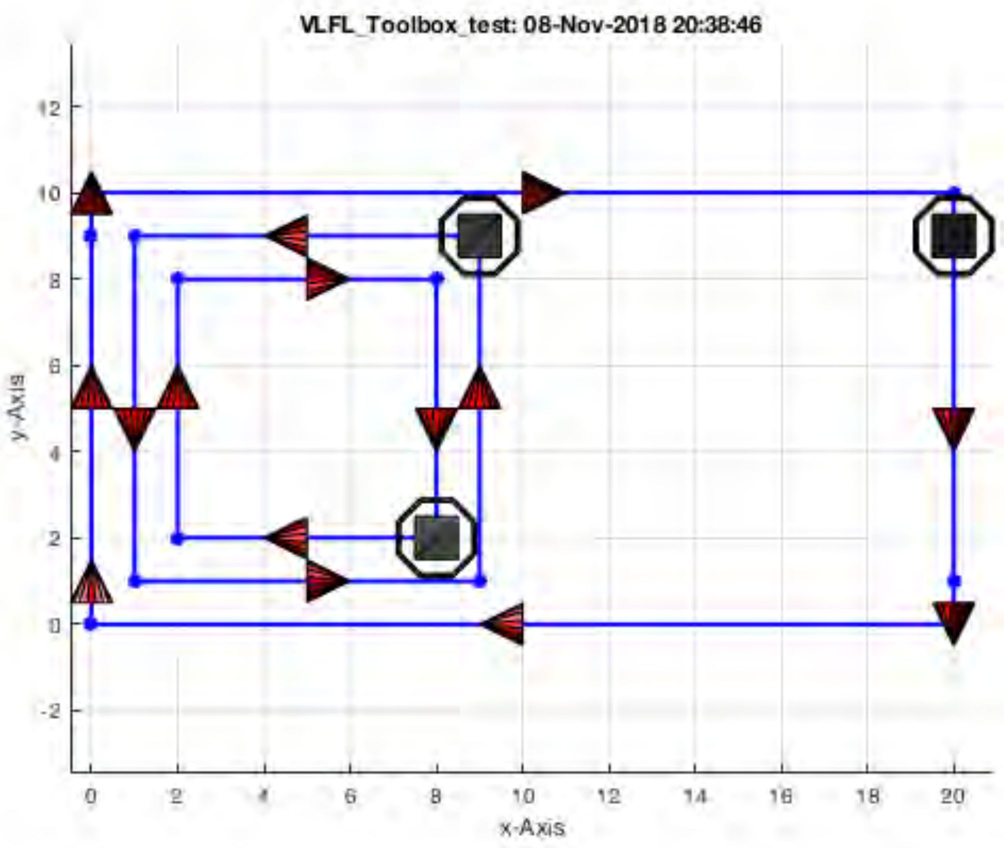


5. Creating embedded contours

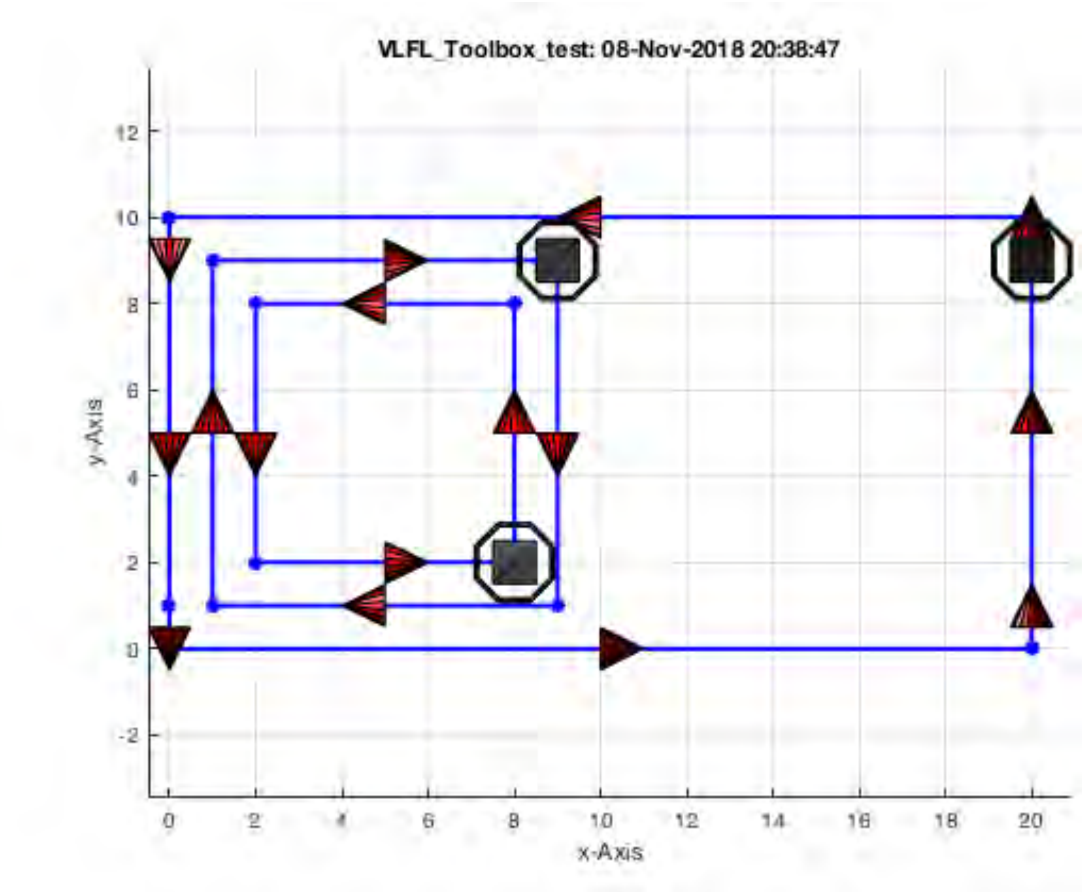
```
CPL=[0 0; 20 0; 20 10; 0 10; NaN NaN; 1 1; 9 1; 9 9; 1 9; NaN NaN; 2 2; 8 2; 8 8; 2 8 ];

close all;PLELofCPL(CPL);
```




```
CPL=CPLuniteCPL(CPL);
close all; PLELofCPL(CPL);
```

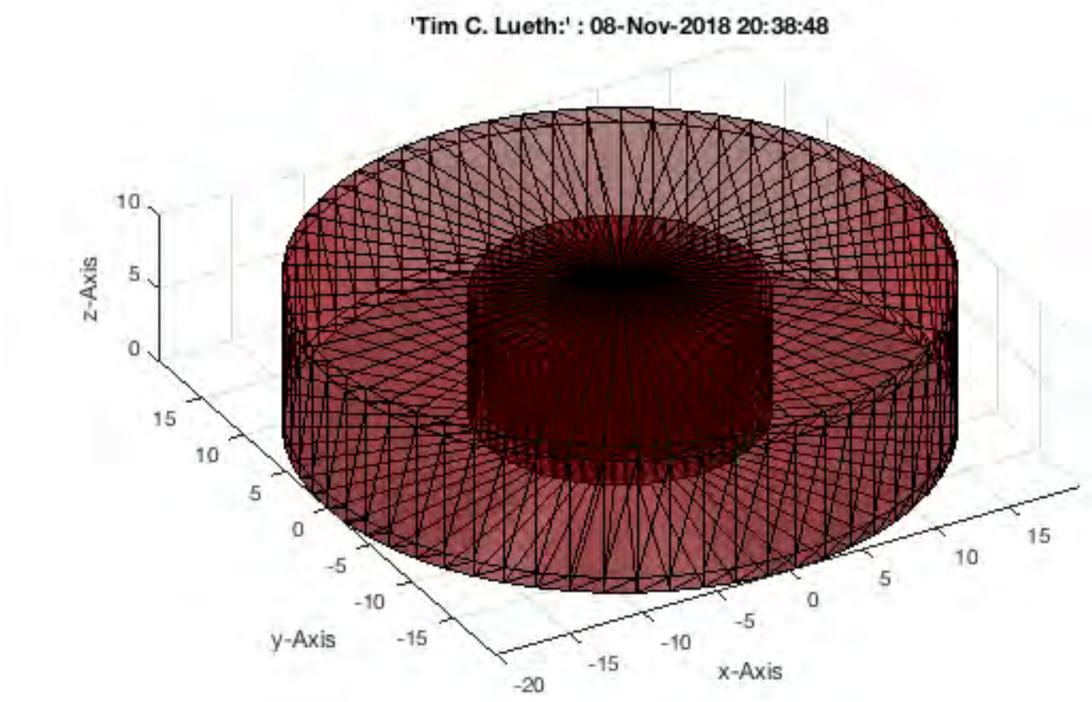



```
CPL=flip(CPL);
close all; PLELofCPL(CPL);
```

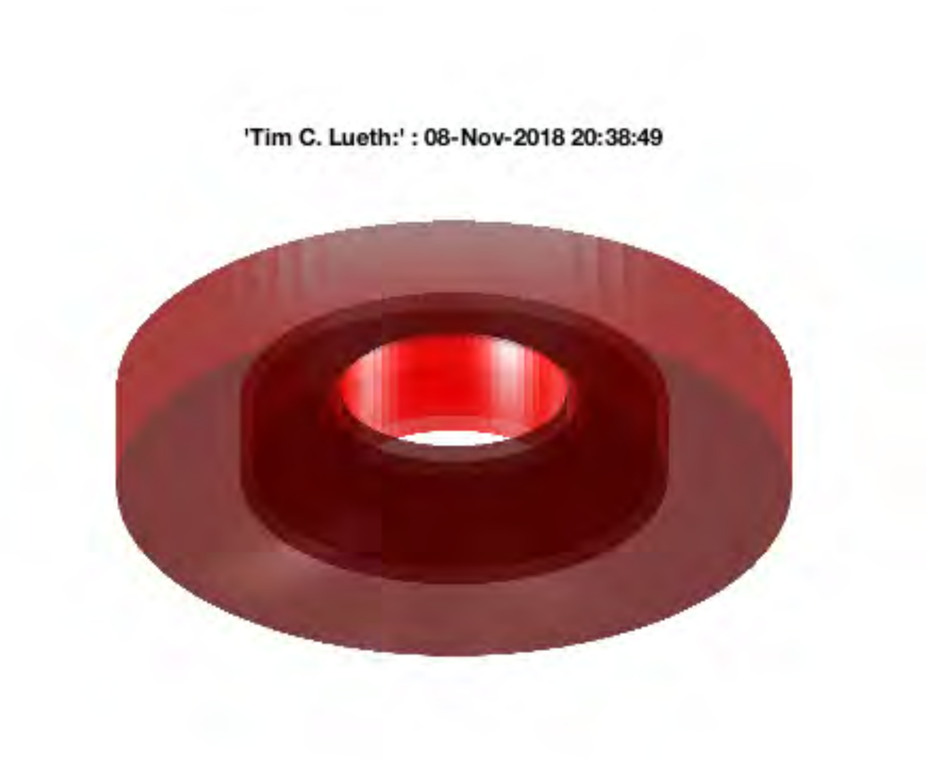


6. Rotate Contours around the z-axis

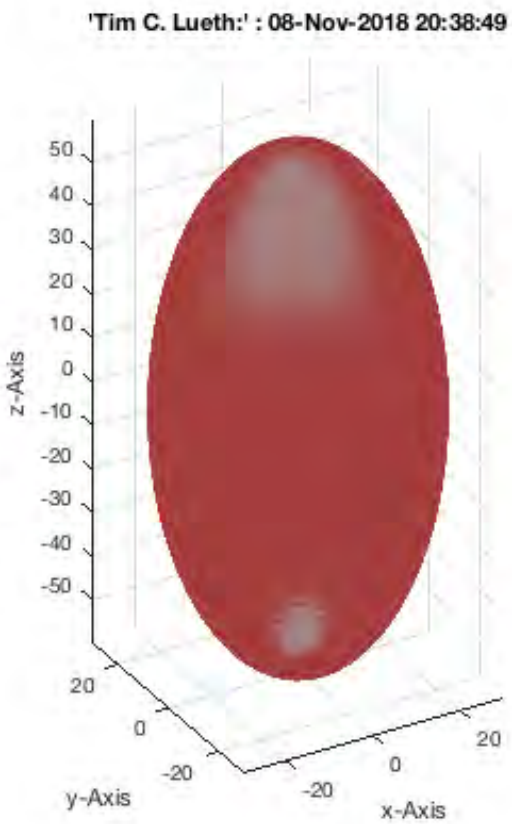
```
VLFLfigure; view(-30,30); grid on;
SG=SGofCPLrot(CPL);
SGplot(SG); VLFLplotlight (0,0.5);
```



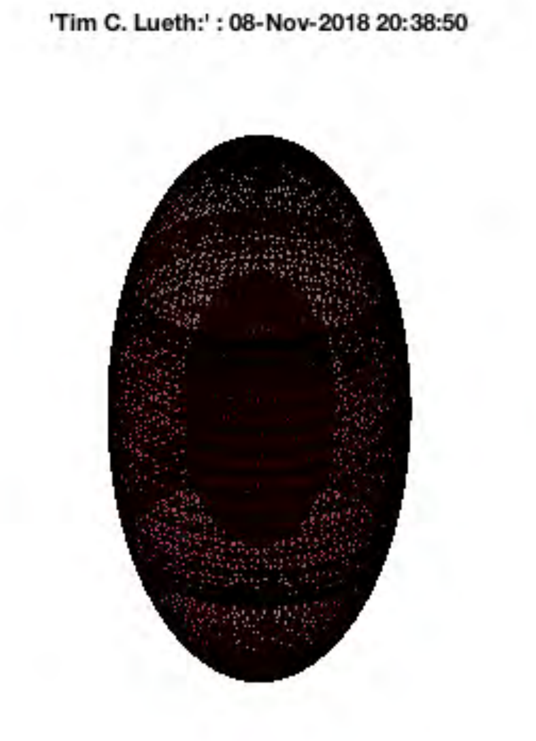

```
VLFLfigure; view(-30,30); grid on;
CPL(:,1)=CPL(:,1)+10;
SG=SGofCPLrot(CPL);
SGplot(SG); VLFLplotlight (1,0.5);
```



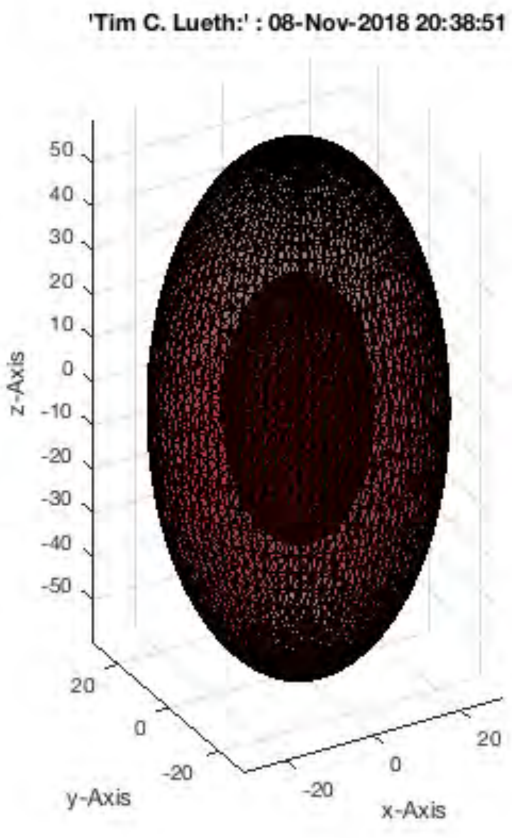

```
VLFLfigure; view(-30,30); grid on;
CPL=PLcircle(30); CPL(:,2)=CPL(:,2)*2;
SG=SGofCPLrot(CPL);
SGplot(SG); VLFLplotlight (1,0.5);
```




```
VLFLfigure; view(-30,30); grid on;
CPL=PLcircle(30); CPL(:,2)=CPL(:,2)*2;
CPL=[CPL;NaN NaN;CPL*0.5];
SG=SGofCPLrot(CPL);
SGplot(SG); VLFLplotlight (0,0.5);
```



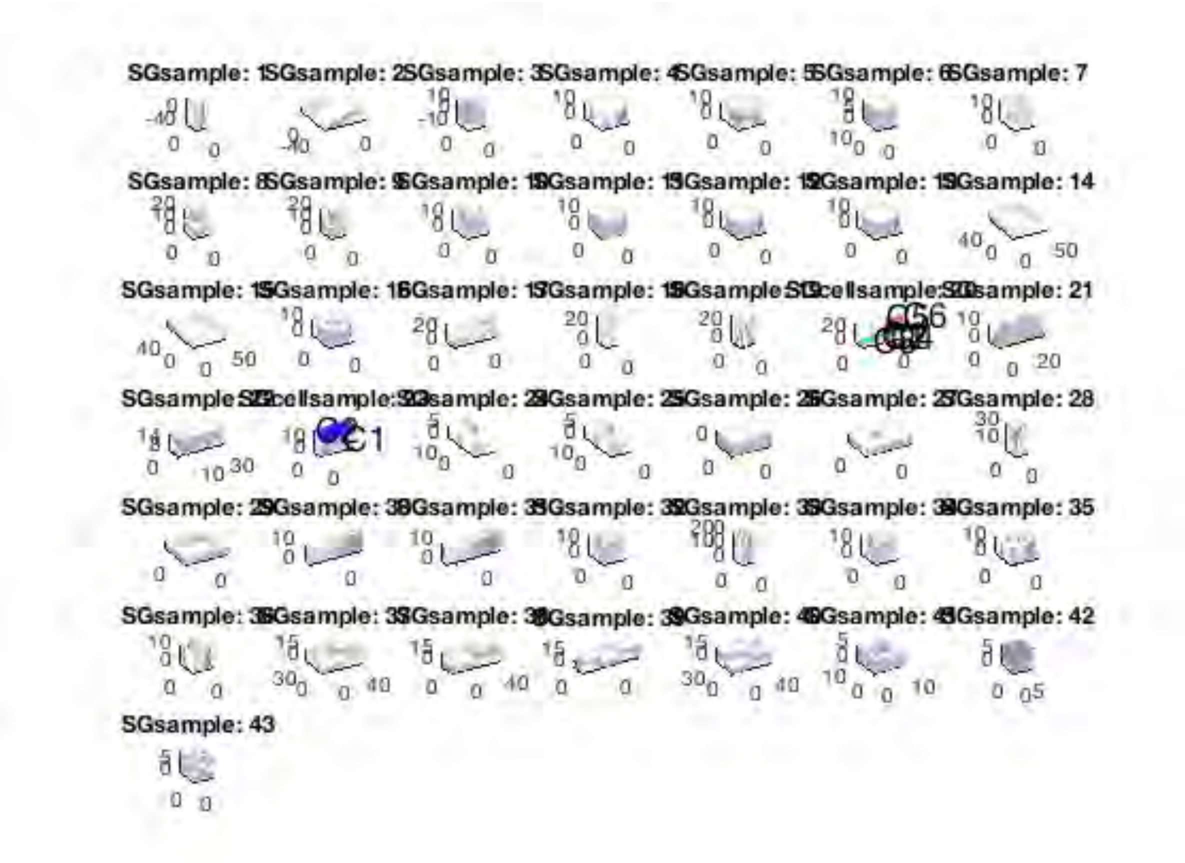

```
VLFLfigure; view(-30,30); grid on;
CPL=PLcircle(30); CPL(:,2)=CPL(:,2)*2;

SG=SGcat(SGofCPLrot(CPL),SGswap(SGofCPLrot(CPL*0.5)));
SGplot(SG); VLFLplotlight (0,0.5);
```



7. Samples of 3D Design

```
SGsample;
```

```
SGchecker "AXB":
SGchecker "A-B":
Warning: Intersecting edge constraints have been split, this may have added new
points into the triangulation.
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 delaunayTriangulation.
```



Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:39:00!
Executed 08-Nov-2018 20:39:02 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================= Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
simulink
video_and_image_blockset
==========================================================================================
==========
```

- *Tim Lueth, tested and compiled on OSX 10.7.5 with Matlab 2014b on 2014-11-30*
- *Mattias Traeger, executed and published on 64 Bit PC using Windows with Matlab 2014b on YYYY-MMM-DD*

Published with MATLAB® R2018a

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:

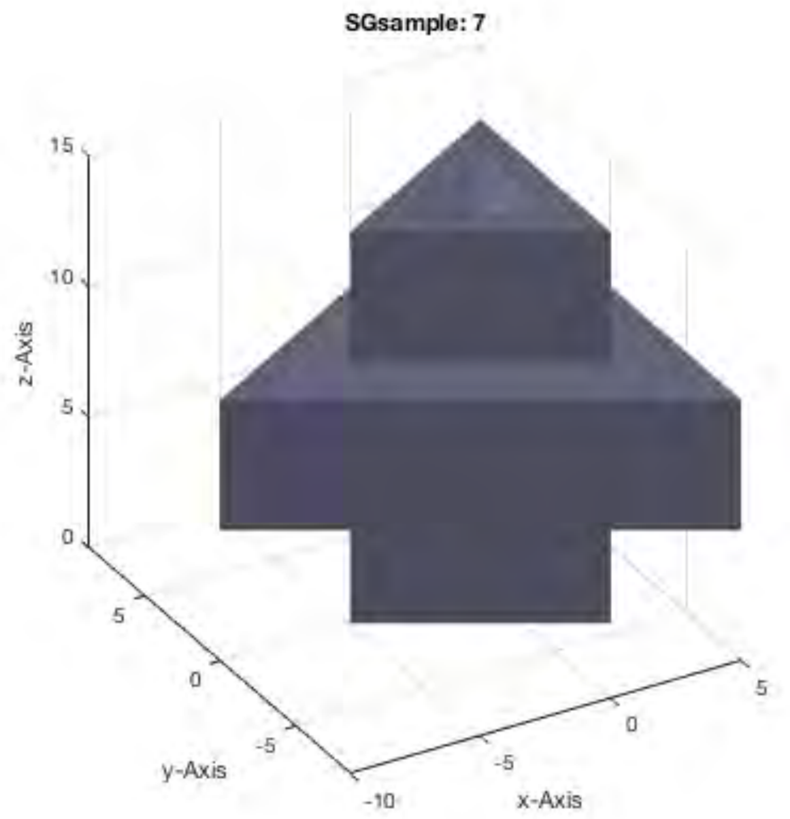
- Tutorial 01: First Steps Using the VLFL-Toolbox for Solid Object Design
- Tutorial 02: Using the VLFL-Toolbox for STL-File Export and Import
- Tutorial 03: Closed 2D Contours and Boolean Operations in 2D
- Tutorial 04: 2½D Design Using Boolean Operators on Closed Polygon Lists (CPL)
- Tutorial 05: 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

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.

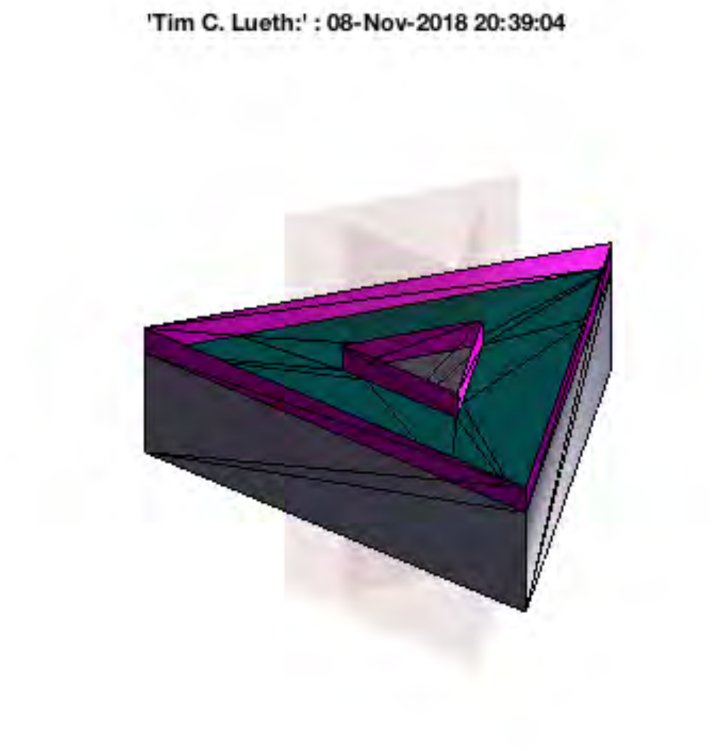
```
close all
SGsample(7);
A=SGsample(7);
```



3. Analyze a slice plane through a solid geoemtry

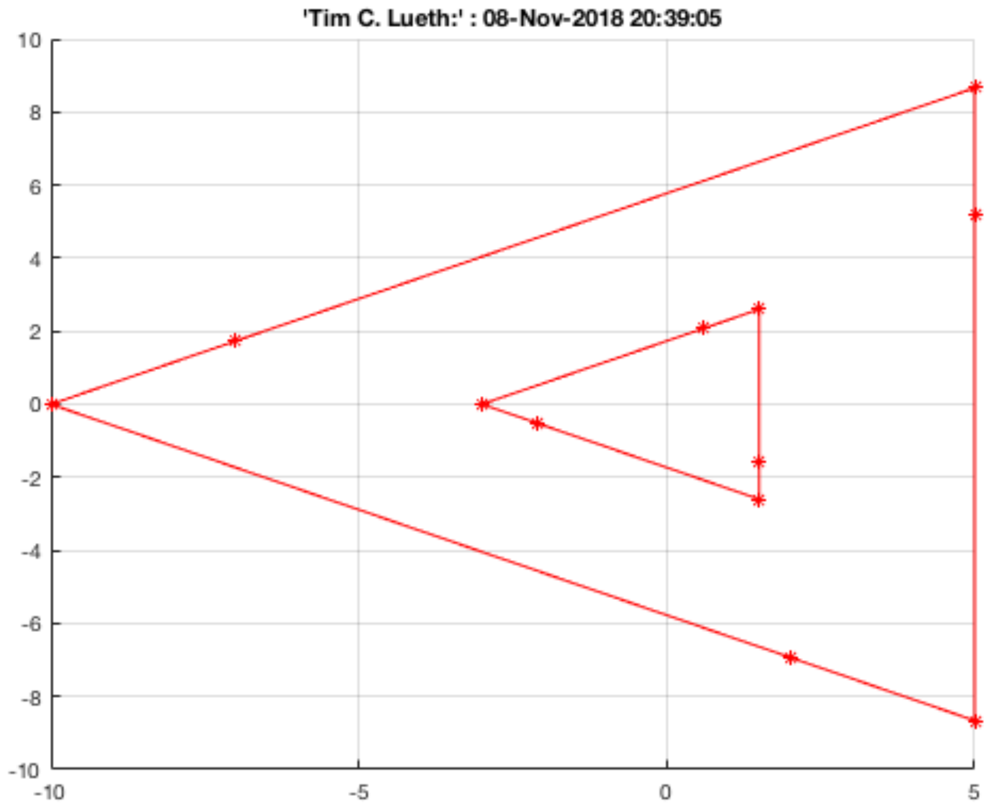
Slicing at a specified z-coordinates is a more complex procedure than expected if several solids are processed that can penetrate each other. By slicing a single solid, the crossed facets/triangles are separated into 2 upper and lower parts that will lead to 2 lower and 1 upper facets or 1 lower and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



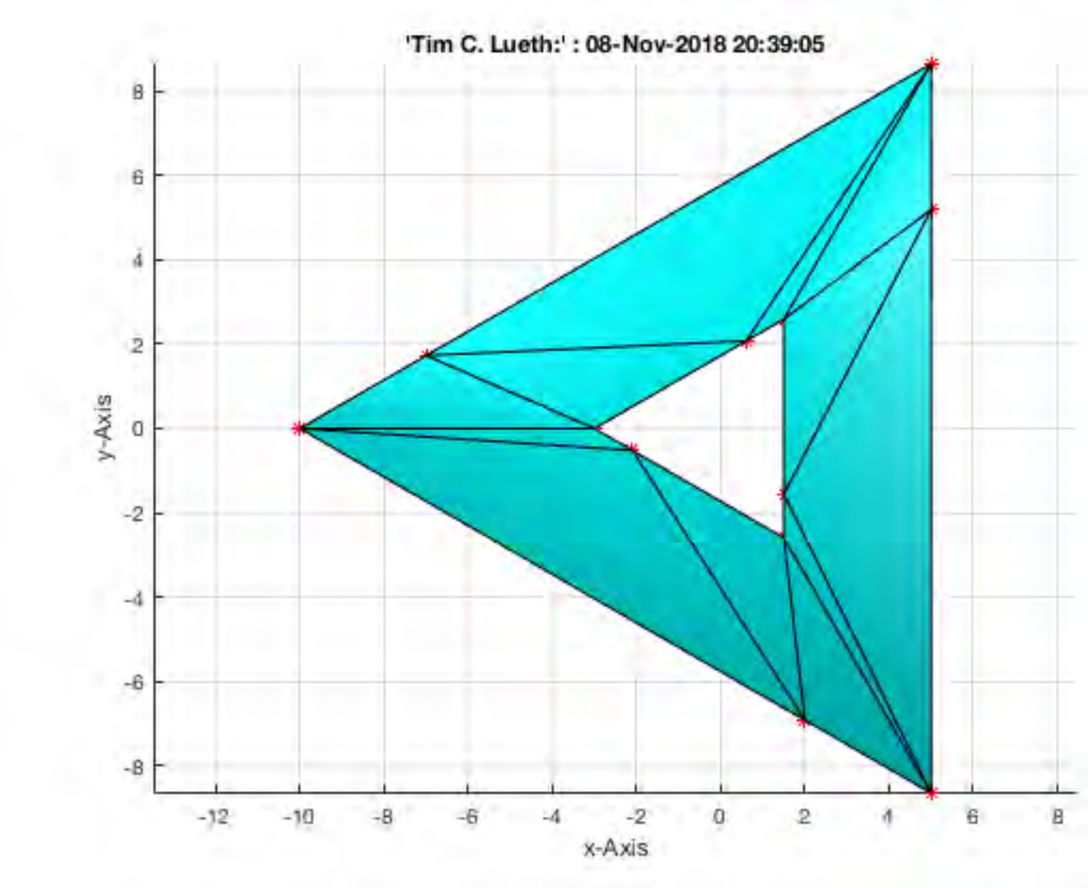
It is also possible just to show the cutting edges of the cutting contour

```
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.

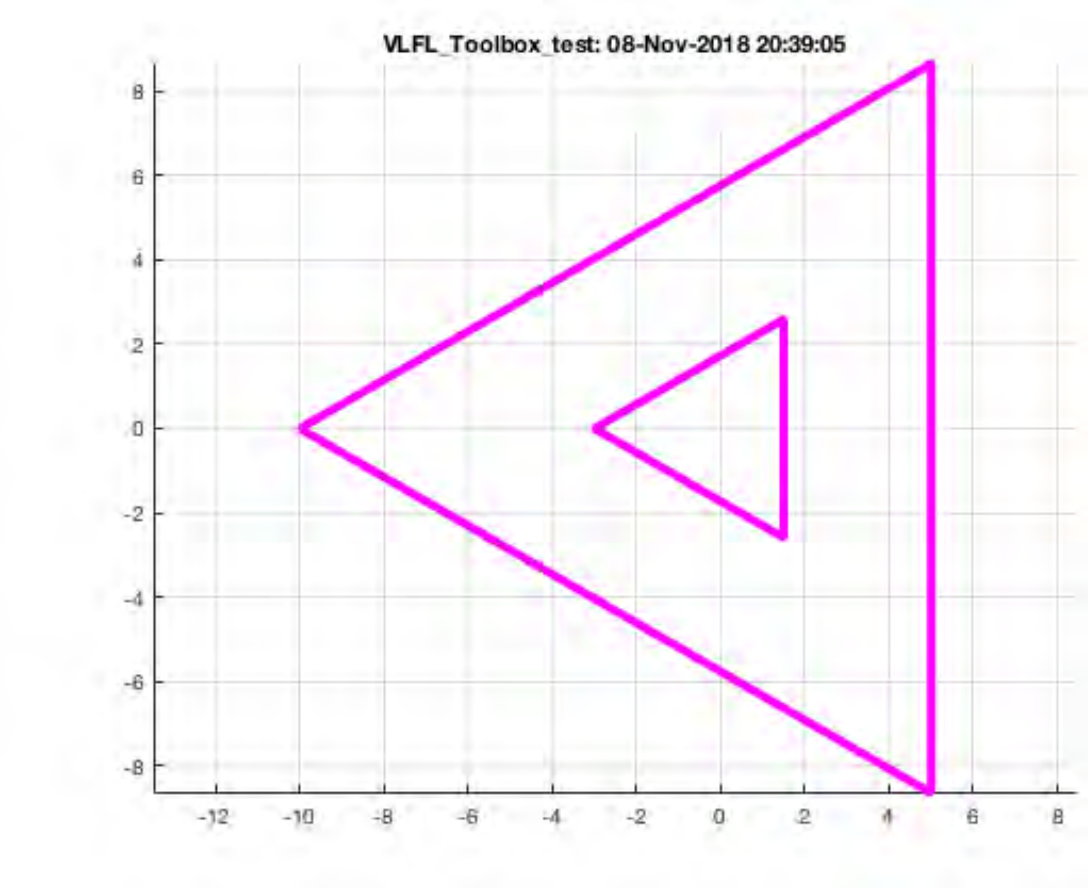
```
in=isInterior(TR2);
VLFLplots(TR2.Points, TR2.ConnectivityList(in,:),'c');
```



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

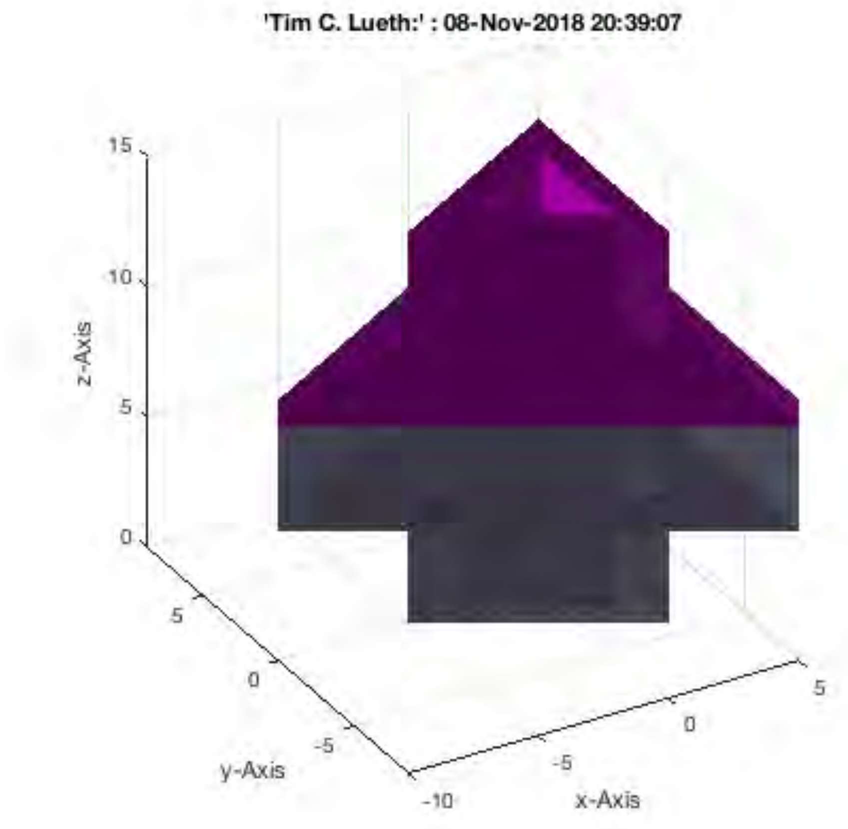
```
CPLofSGslice(A,10); [CPL,warn]=CPLofSGslice(A,10); warn
```

```
Warning: Crossing plane cannot be calculated error-free
Warning: CPLofPLEL: EL contains open and ignored edges!
Warning: Crossing plane cannot be calculated error-free
Warning: CPLofPLEL: EL contains open and ignored edges!

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**.

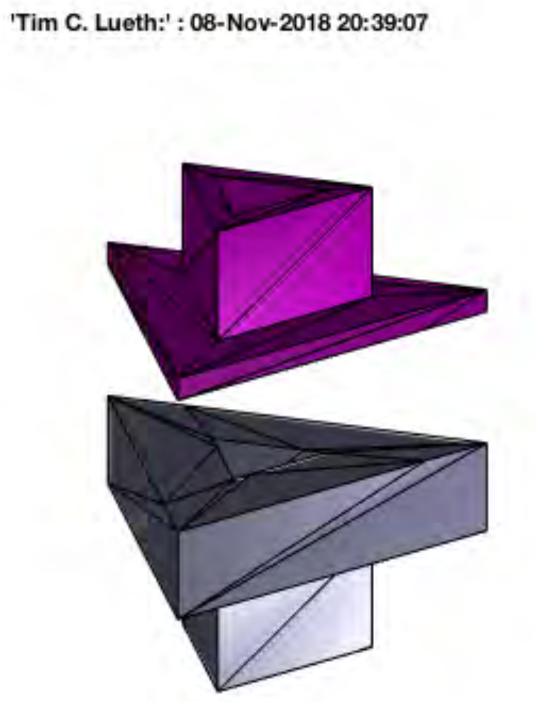
```
VLFLfigure;
SGcut(A,9);
```



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);
```

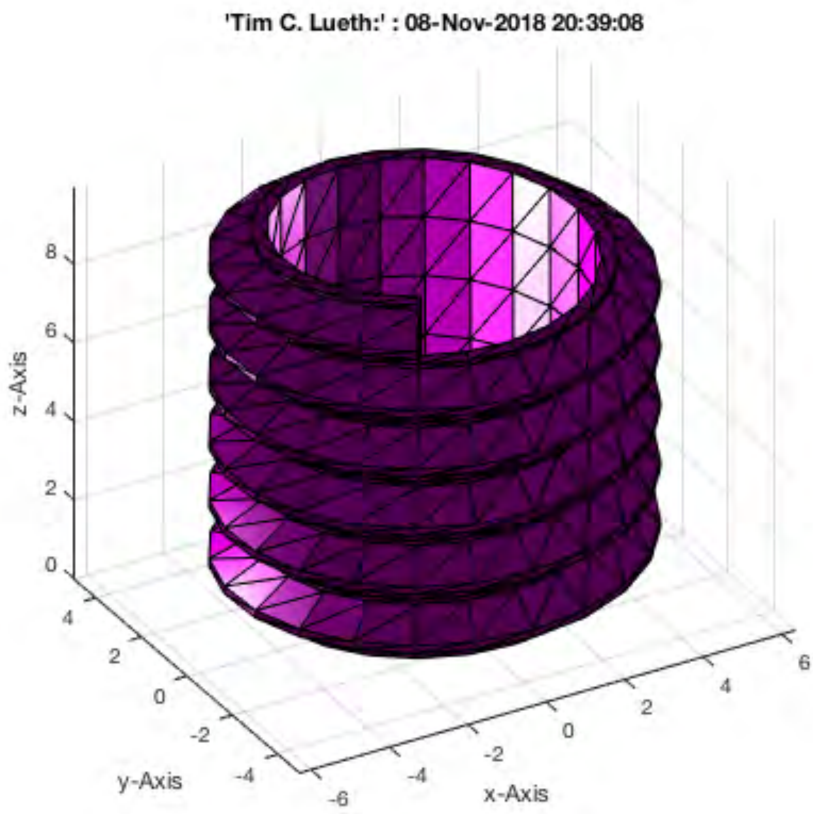
```
L =

  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

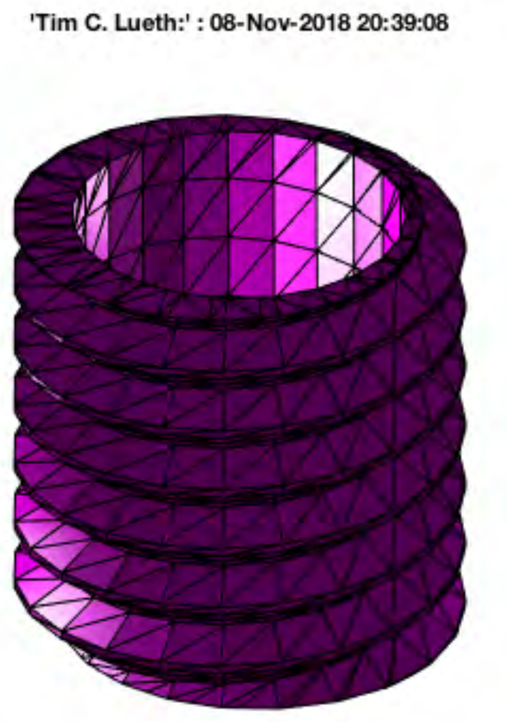
Some geometries such as screwnuts have specific geometries that have their origin in the manufacturing process of the threads. To create also similar shapes it is necessary to create a longer thread and to cut out the required length later:

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



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.  (2018-Nov-08), 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 11-Aug-2073 20:39:09!
Executed 08-Nov-2018 20:39:11 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
simulink
video_and_image_blockset
==========================================================================================
==========
```

- *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® R2018a

Tutorial 09: Boolean Operations with Solid Geometries

2014-11-30: 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)

Motivation for this tutorial: (Originally SolidGeometry 2.0 required)

The implementation of the boolean operations based on STL geometries took the author several years. The reason was not only the complexity of the numerous special cases but also the numerical accuracy or resolution of the required geometrical calucations. So even if a normal position is cacluated with 12 digits accuracy, the cross product often has a just an accuracy of 6 digits. Unfortunatly crossing triangles with position errors of 6 digits can easily lead to phantom triangles, phantom edges wich either do not really exist or are just doubles of already existing lines. Since normally all edges must have a second edge with the opposite direction, doubled lines/egdes with the same direction make trouble. So to be successful with the boolean operations you should make sure that

1. No facet should be in the same plane as or overlap another facet or cross with almost parallel edges to the plane of another facet. This is always valid for one solid, but in case of a second solid for boolean operations, it is quite difficult to guarantee this.
2. It is fact that, the more boolean operations took place, to create a new solid, the more vertices and facets were created. The removement of dispensable vertices and facets is possible but is a boring non productive pieve of source code. So the motivation to programm such a procedure is not high.
3. No edge of a triangle should be in the sample plane or crossing but almost parallel to a plane of a facet.
4. It is fact that a normal user just want to use the boolean operator without thinking about those problems. The normal user will just be disappointed if the way to design a physical solid object finally fails because of the limitations of the crossing
5. Make definitly sure that after all boolean operations you use SGchecker to analyze the solid geoemtry do detect errors immedeatly.
6. May be the only solution is to use a fixed coordiate grid during all calculations to make shure that two vertices are either definitly separated or definitly the same. #

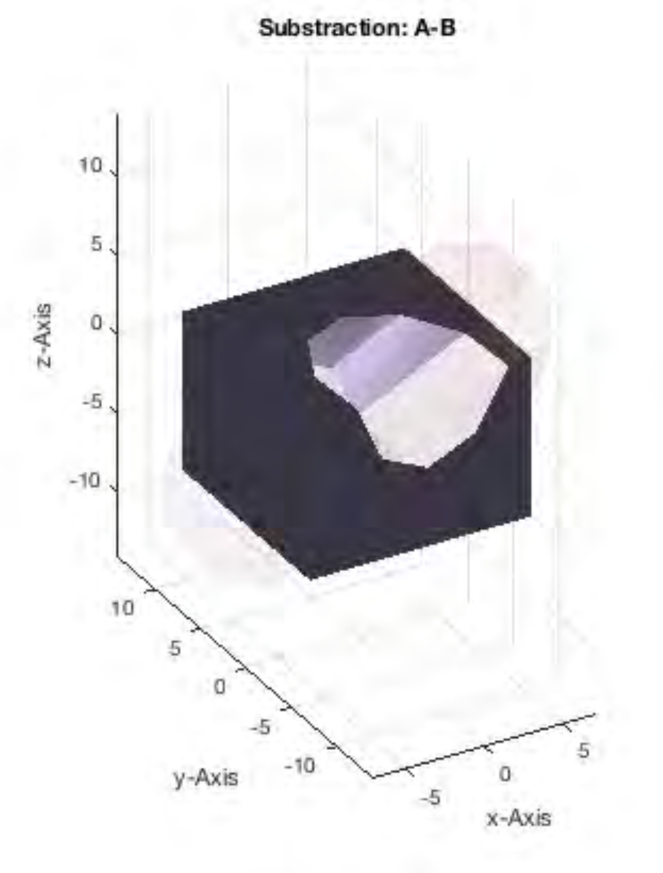
3. Creating two solids for showing the boolean operations

```
function VLFL_EXP09
```

```
VLFLfigure; view(-30,30); grid on;
```

```
A=SGofCPLz(PLcircle(10,4),10);  A=SGtrans0(A);
B=SGofCPLz(PLcircle(5,10),30);  B=SGtrans0(B); B=SGtransR(B,rotdeg(45,5,0));

SGplot(A,'b');
SGplot(B,'r');
VLFLplotlight (0,0.9);
A=SGstripfields(A)
B=SGstripfields(B)

SGbool ('-',A,B);
```

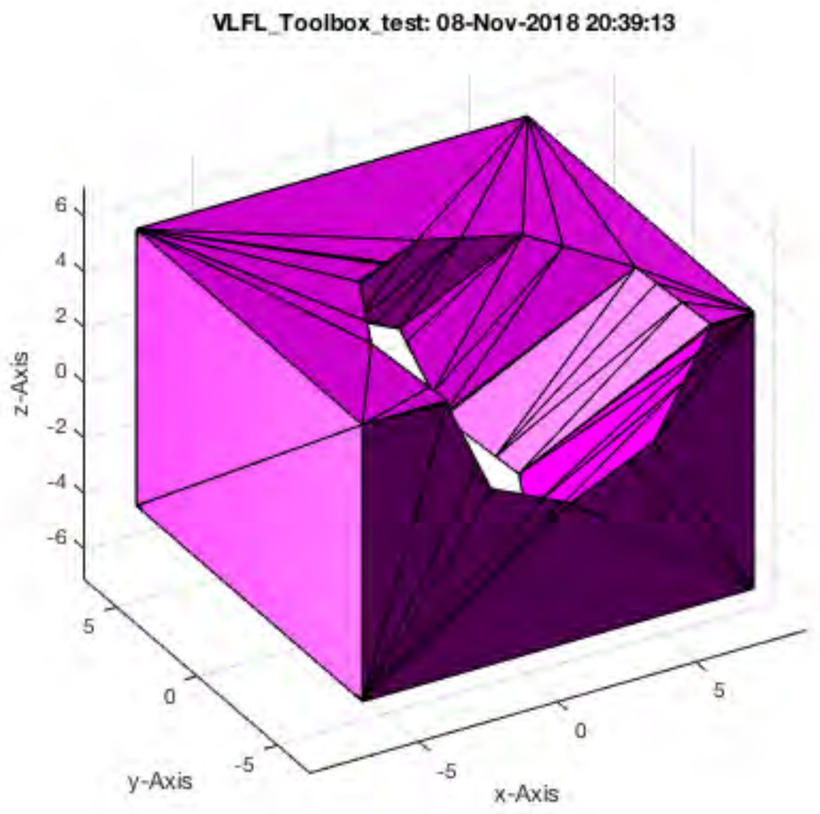
```
A =

  struct with fields:

    VL: [8×3 double]
    FL: [12×3 double]


B =

  struct with fields:

    VL: [20×3 double]
    FL: [36×3 double]

SGchecker "A-B":
```



4. Boolean operator: Substraction A-B or A\B

```
X=SGbool ('A',A,B);
SGfigure(X); view(-30,30);
```

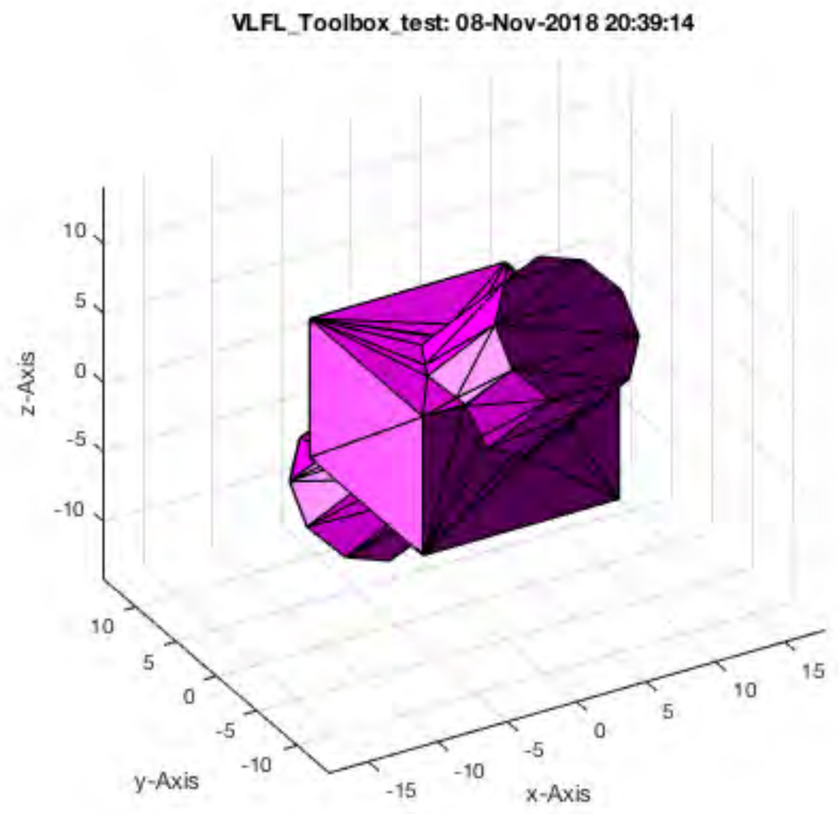
```
SGchecker "AAB":
```



5. Boolean operator: Substraction A+B

```
X=SGbool ('+',A,B);
SGfigure(X); view(-30,30);
```

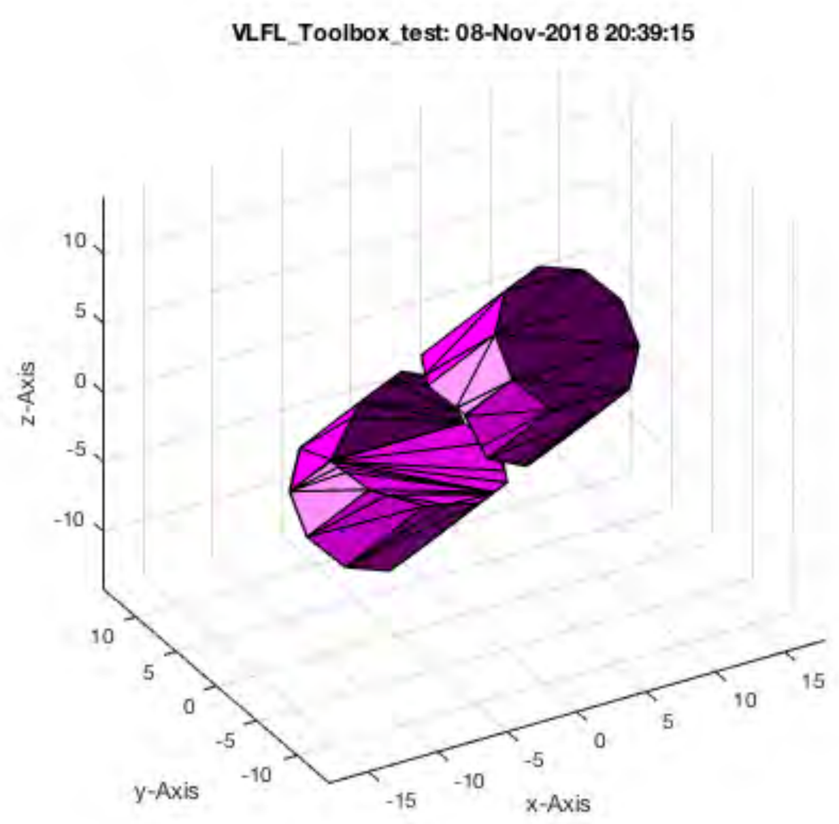
```
SGchecker "A+B":
```



6. Boolean operator: Substraction B\A

```
X=SGbool ('B',A,B);
SGfigure(X); view(-30,30);
```

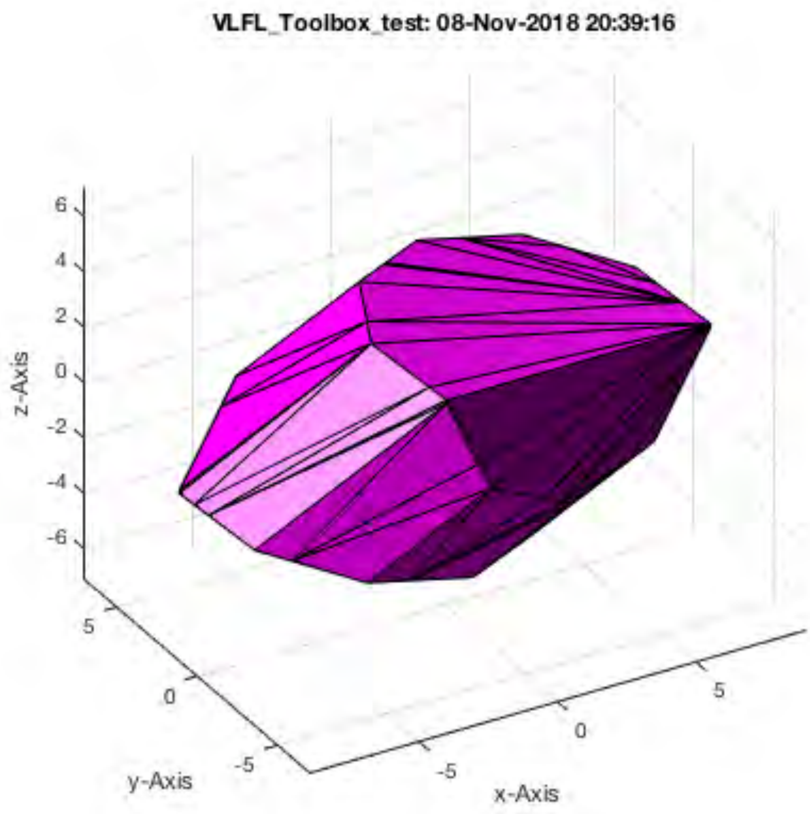
```
SGchecker "ABB":
```



7. Boolean operator: A xor B

```
X=SGbool ('x',A,B);
SGfigure(X); view(-30,30);
```

```
SGchecker "AxB":
```



8. Analyzing the results and comparision with additive design.

Analyzing the number of vertices and facets of the results of a boolean operation shows clearly that there are much more vertices and facets than the sum of the vertices and facets. In general it makes for STL more sense to add simple solids to a more complex by attaching them together by simply pushing them into another. In this case the final number of vertices and facets is the sum of the individual facets and vertices.

Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:39:16!
Executed 08-Nov-2018 20:39:18 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
simulink
video_and_image_blockset
==========================================================================================
==========
```

- *Tim Lueth, tested and compiled on OSX 10.7.5 with Matlab 2014b on 2015-06-07*
- *Christina Friedrich, executed and published on 64 Bit PC using Windows with Matlab 2015a on 2015-06-17*

Published with MATLAB® R2018a

Tutorial 10: Packaging of Sets of Solid Geometries (SG)

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

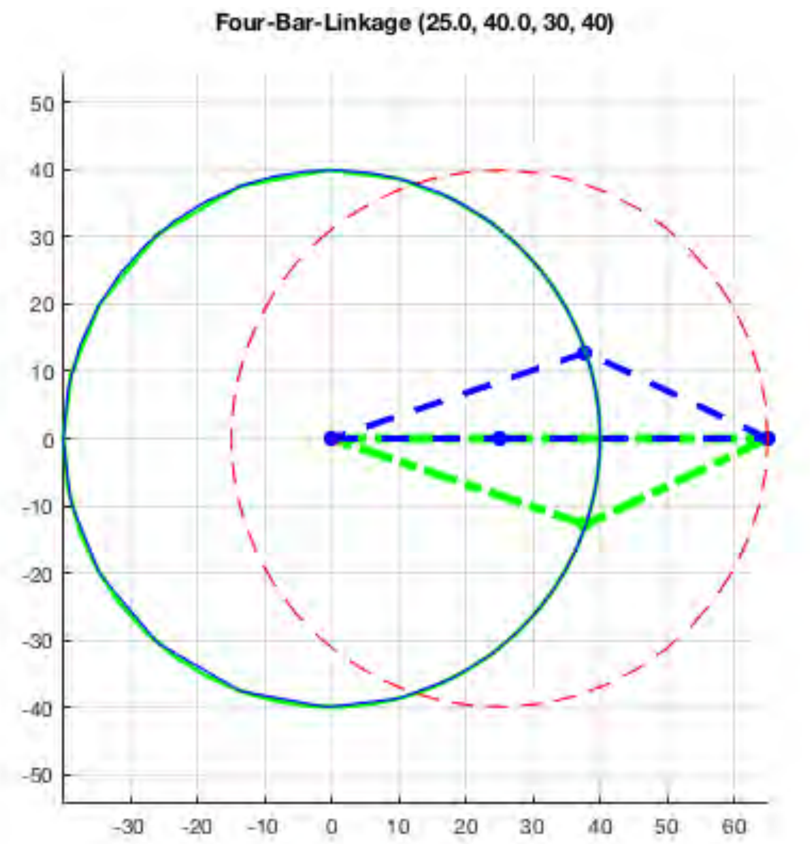
Motivation for this tutorial: (Originally SolidGeometry 2.4 required)

2. The four-bar-linkage kit as example for a set of multiple solids

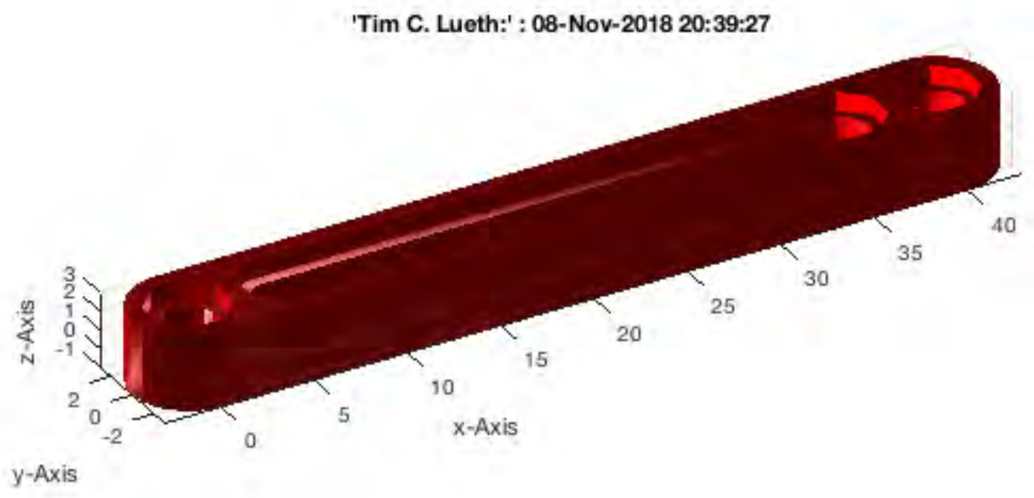
A very interesting mechanism in mechanics is the four-bar-linkage. It consists of four bars that are linked together by 4 rotatorial joints. Such a mechanism can be built by 4 different elememts

1. Bar: The basic mechanic link
2. Bolt: A simple bolt that allows rotation
3. Shaft: A simple shaft that transfer torque
4. Spacer: A simple element that is required to achieve parallel bars

```
close all;
fourBarLinkage (25,40,30,40);
```

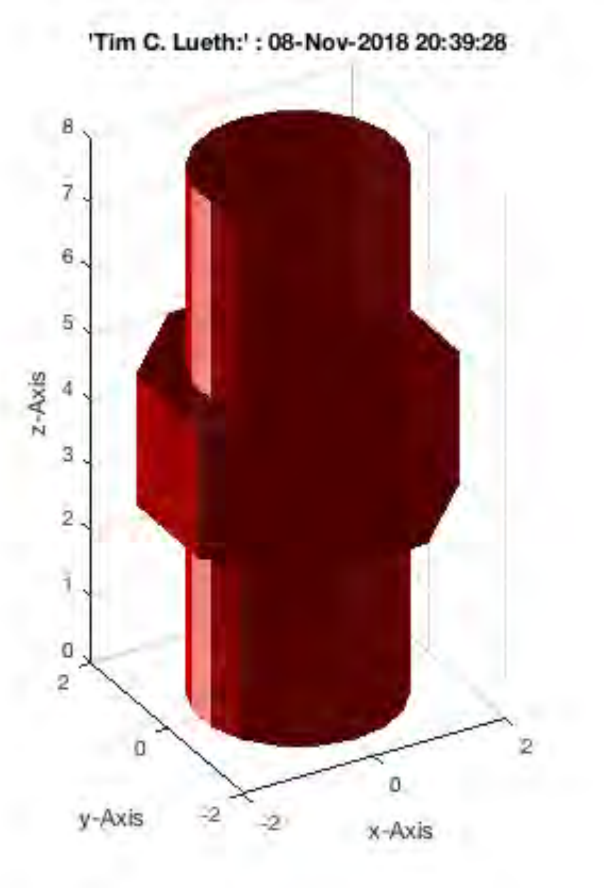



```
fourBarLinkageKit ('Bar',40);
```




```
fourBarLinkageKit ('Bolt');
```


```
fourBarLinkageKit ('Shaft');
```




```
fourBarLinkageKit ('Spacer');
```

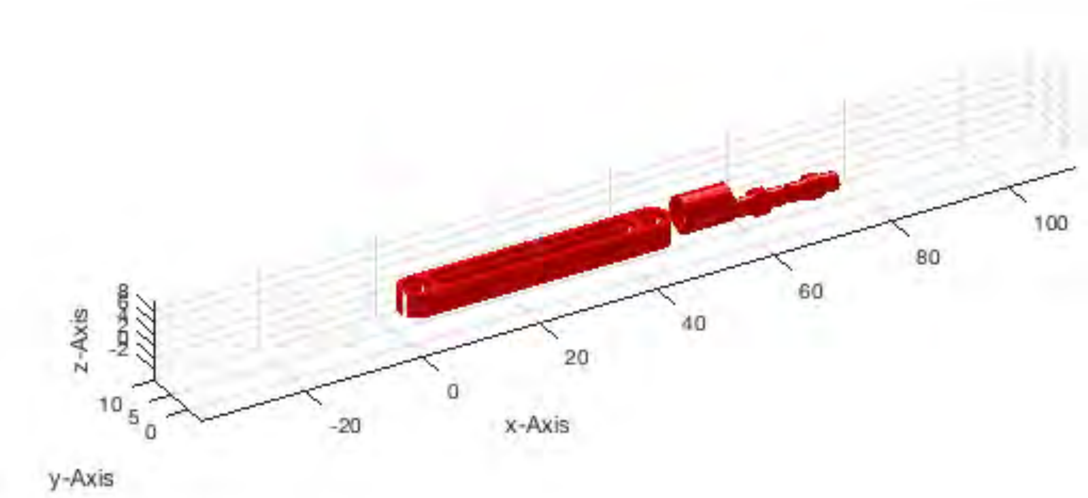
3. Packaging a set of solid geometries in a volume

For a four-bar-linkage we need 4 bars and 4 bolts and may be 2 spacer and 2 shafts. For this purpose there is one function

- **SGpacking** arranges several solid geometries side by side in a volume

```
close all;
A=fourBarLinkageKit ('Bar',40);
B=fourBarLinkageKit ('Bolt');
C=fourBarLinkageKit ('Shaft');
D=fourBarLinkageKit ('Spacer');
SG=SGpacking({A,B,C,D});
SGplot(SG); view (-30,30); VLFLplotlight (1,0.9); zoompatch;
```

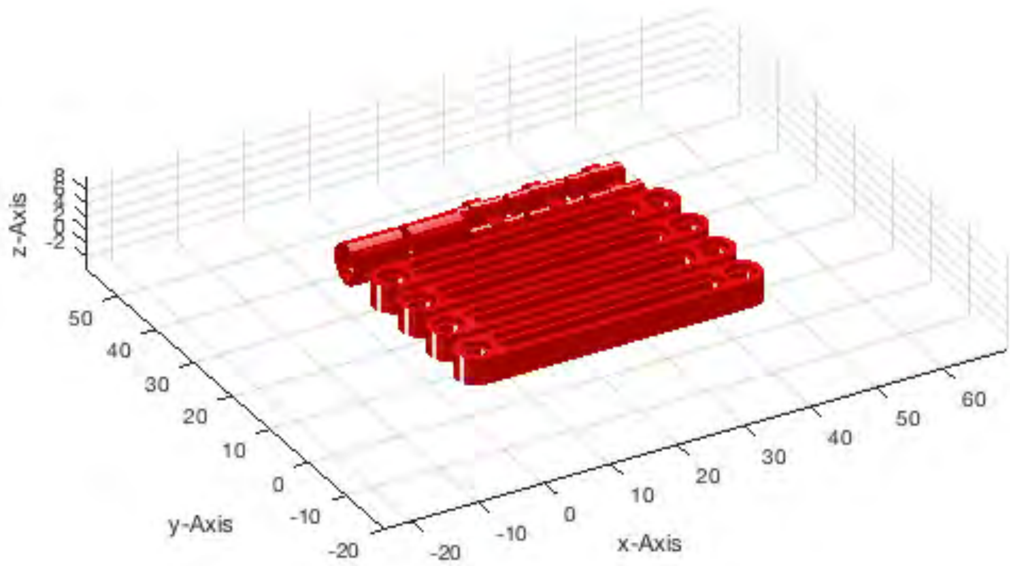
```
Packing 4 objects (h=24):
```



Similar it is possible to pack several objects of the same kind into the volume and also to define the dimensions of the packing volume. Typically the z-coordiante of the volume specification is unlimited or much bigger than the xy-coordinats.

```
close all;
SG=SGpacking({A,A,A,A,B,B,B,B,C,C,D,D},[50,50,1000]);
SGplot(SG); view (-30,30); VLFLplotlight (1,0.9); zoompatch;
```

```
Packing 12 objects (h=45):
```



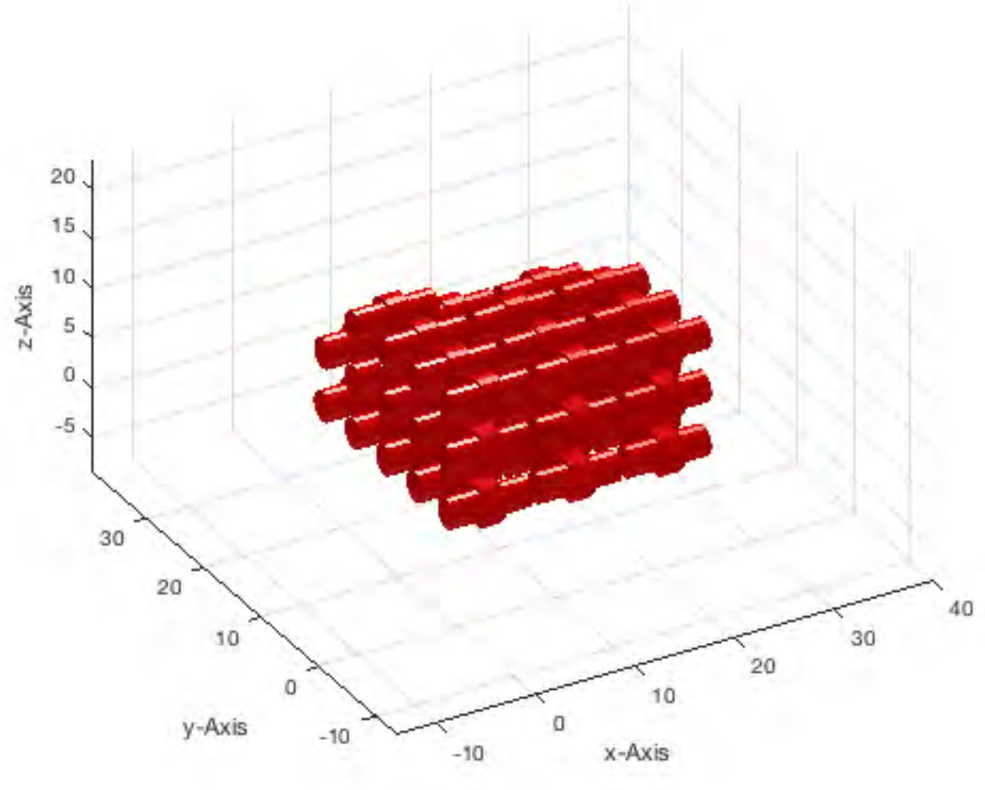
4. Using container/collections insted of itemizing the solids

In many cases we are not interested to list the items in the source code but to create a structure containing all objects we want to pack later Therefor, we need a data structure that allows to collect several solids into something like a container. This can be done by the following functions:

- **SGCaddSG** Add a single solid geometry to a collection
- **SGCaddSGn** Add multiple copies of a single solid geometry to a collection

```
close all
SGC=[];                                               % Create a Solid Geometry Collection
SGC=SGCaddSGn(SGC,fourBarLinkageKit ('Bolt'),20);    % Add 20 bolt to the container
SGC=SGCaddSGn(SGC,fourBarLinkageKit ('Shaft'),20);   % Add 20 shafts to the container

SG=SGpacking(SGC,[30, 30 ,1000]);                     % SGpacking accepts also SGC structs
SGplot(SG); view (-30,30); VLFLplotlight (1,0.9); zoompatch;
```

```
Packing 40 objects (h=48): ....................
```

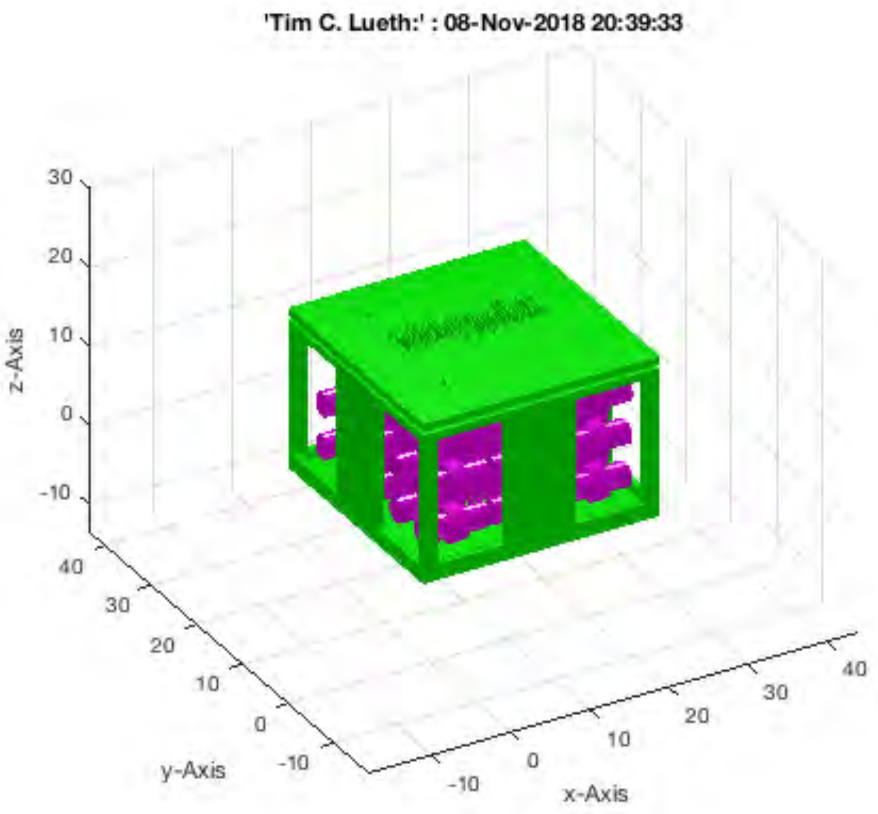


5. Create boxes around the packed solids for the final 3D print job

To handle the print job in a convinient way, it makes sense to create a box around the parts and also to write on top of the cover the content or the intended use of the box plus may by a date.

```
close all;
SGboxing(SG,[],[],'.\nTest for Packaging and Boxing\n.');
view (-30,30); VLFLplotlight (1,0.9); zoompatch;
```

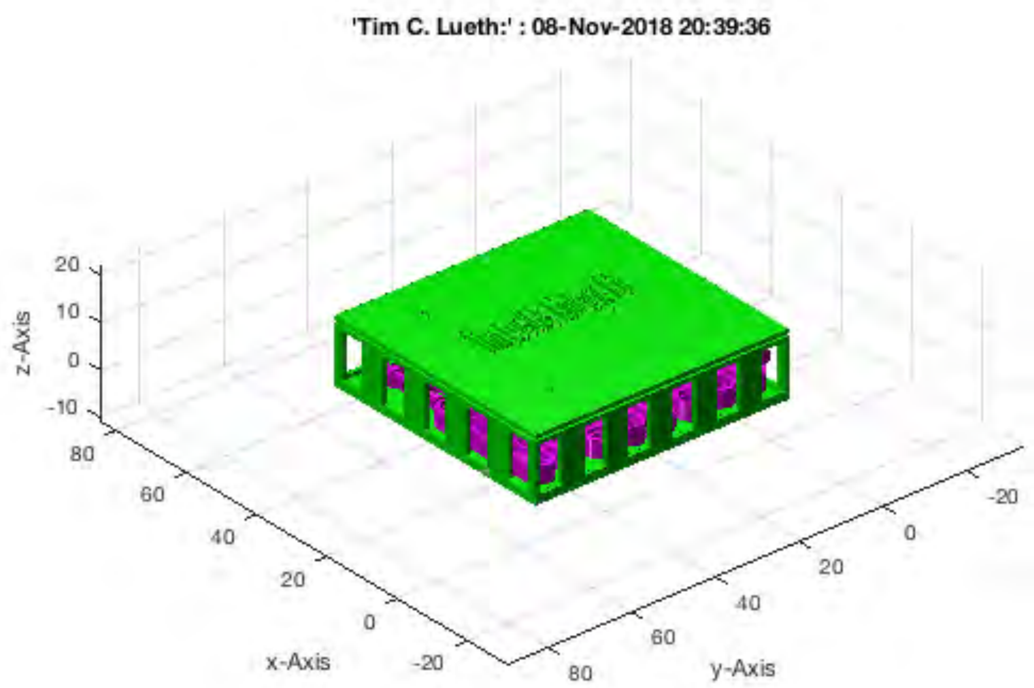
```
==>TEXT GENERATION..Analyze union areas for 60 facets:
Major union vectors: 6 found with maximum size of 1982.
Finally 2 union areas found with size > 100
Text attached to union Nr: 2
..finished!
```



6. Create the four-bar-linkage kit as print job

```
close all;
SGC=[];
SGC=SGCaddSGn(SGC,fourBarLinkageKit ('Bar',25),2);
SGC=SGCaddSGn(SGC,fourBarLinkageKit ('Bar',35),2);
SGC=SGCaddSGn(SGC,fourBarLinkageKit ('Bar',40),4);
SGC=SGCaddSGn(SGC,fourBarLinkageKit ('Bolt'),4);
SGC=SGCaddSGn(SGC,fourBarLinkageKit ('Shaft'),4);
SGC=SGCaddSGn(SGC,fourBarLinkageKit ('Spacer'),4);
SGA=SGpacking(SGC,[55, 60 ,100]);
SGB=SGboxing(SGA,[],[],'.\nTim Lueth''s Linkage Kit\n.');
VLFLfigure(SGA); SGplot(SGB,'g');
SG=SGcat(SGA,SGB); view (-130,30); VLFLplotlight (1,0.9); zoompatch;
SGwriteSTL(SG,'EXP10: Four-Bar-Linkage-Kit');
```

```
Packing 20 objects (h=58): ....................
==>TEXT GENERATION..Analyze union areas for 60 facets:
4 Dimension warnings
Major union vectors: 6 found with maximum size of 7323.
Finally 22 union areas found with size > 100
Text attached to union Nr: 2
..finished!
1000..2000..3000..4000..5000..6000..7000..8000..9000..10000..11000..12000..
```



Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:39:36!
Executed 08-Nov-2018 20:39:38 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
simulink
video_and_image_blockset
===========================================================================================
==========
```

- *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® R2018a

Tutorial 11: Attaching Coordinates Frames to Create Kinematik Models

2015-06-08: 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 2.3 required)

```
function VLFL_EXP11
```

```
close all;
```

2. Loading the 5 components of a 4DoF robot solid model

Before explaining how to create the parts of a robot kinematik we just load such components in. The command line load AIM_robot

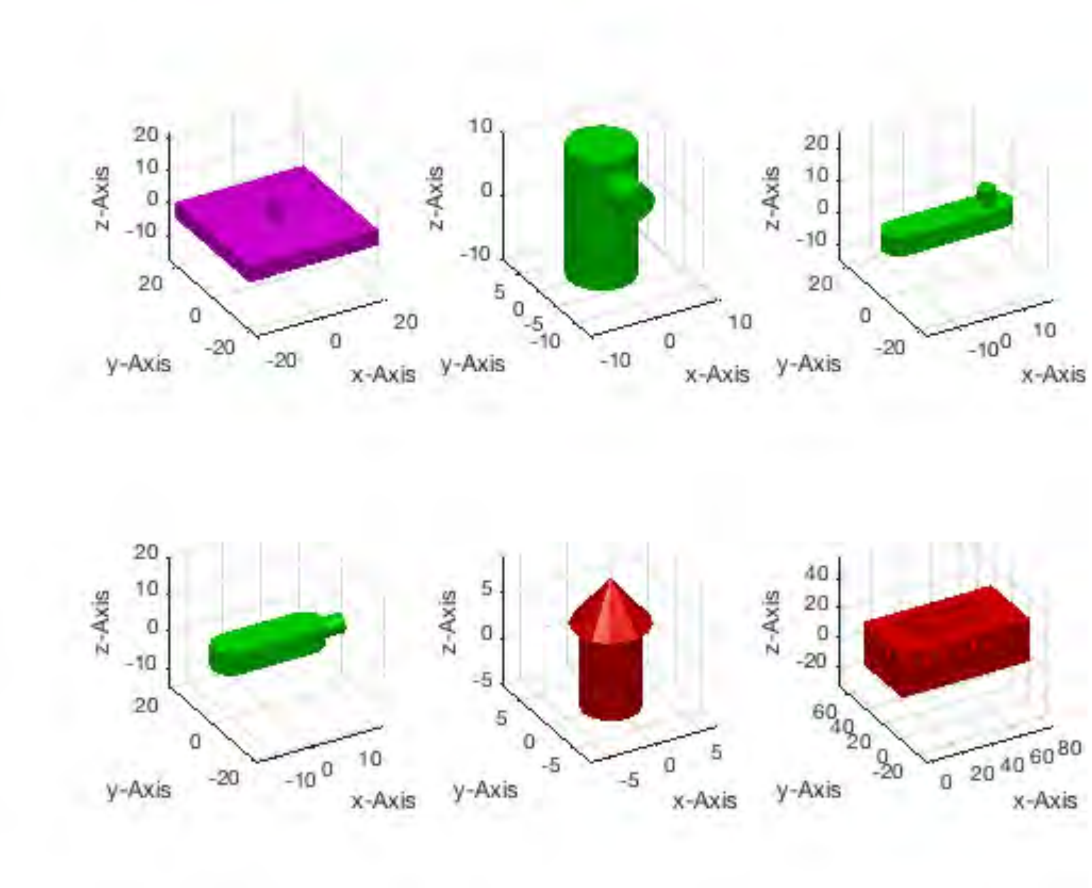
```
load ('AIM_SGrobot.mat');
% SG0=SGfixerrors(SG0,1e-3); SGchecker(SG0);
% SG1=SGfixerrors(SG1,1e-3); SGchecker(SG1);
% SG2=SGfixerrors(SG2,1e-3); SGchecker(SG2);
% SG3=SGfixerrors(SG3,1e-3); SGchecker(SG3);
% SG4=SGfixerrors(SG4,1e-3); SGchecker(SG4);
% save ('AIM_SGrobot','SG0','SG1','SG2','SG3','SG4','SGrobot');
```

- returns a solid geometry SG0 which is a base plate with a rotatorial joint
- returns a solid geometry SG1 which is a link with a rotatorial joint
- returns a solid geometry SG2 which is a link with a rotatorial joint
- returns a solid geometry SG3 which is a link with a rotatorial joint
- returns a solid geometry SG4 which is a hand with a pointing tip
- returns a solid geometry SGrobot which can be written as STL-File an printed using a 3D printer.

```
SGfigure;
```

```
subplot(2,3,1); SGplot(SG0); view (-30,30); VLFLplotlight(1,0.9);
subplot(2,3,2); SGplot(SG1); view (-30,30); VLFLplotlight(1,0.9);
subplot(2,3,3); SGplot(SG2); view (-30,30); VLFLplotlight(1,0.9);
subplot(2,3,4); SGplot(SG3); view (-30,30); VLFLplotlight(1,0.9);
subplot(2,3,5); SGplot(SG4); view (-30,30); VLFLplotlight(1,0.9);
subplot(2,3,6); SGplot(SGrobot); view (-30,30); VLFLplotlight(1,0.9);
SGwriteSTL (SGrobot,'4-DOF Robot Set');
```

```
1000..2000..3000..4000..5000..6000..7000..8000..9000..10000..11000..12000..13000..14000..15
000..16000..17000..18000..19000..20000..21000..22000..23000..
```



3. The concept of attaching coordinate frames as 4x4 honogenous transformation matrix

If we analyze the structure of one of the components of the robot we see that we have now more than just the surface of the geometry.

```
SG0

% We see that beside vertices and facets (VL, FL) we have a color and a
% alpha value for transparency when plotting.
```

```
SG0 =

  struct with fields:
```

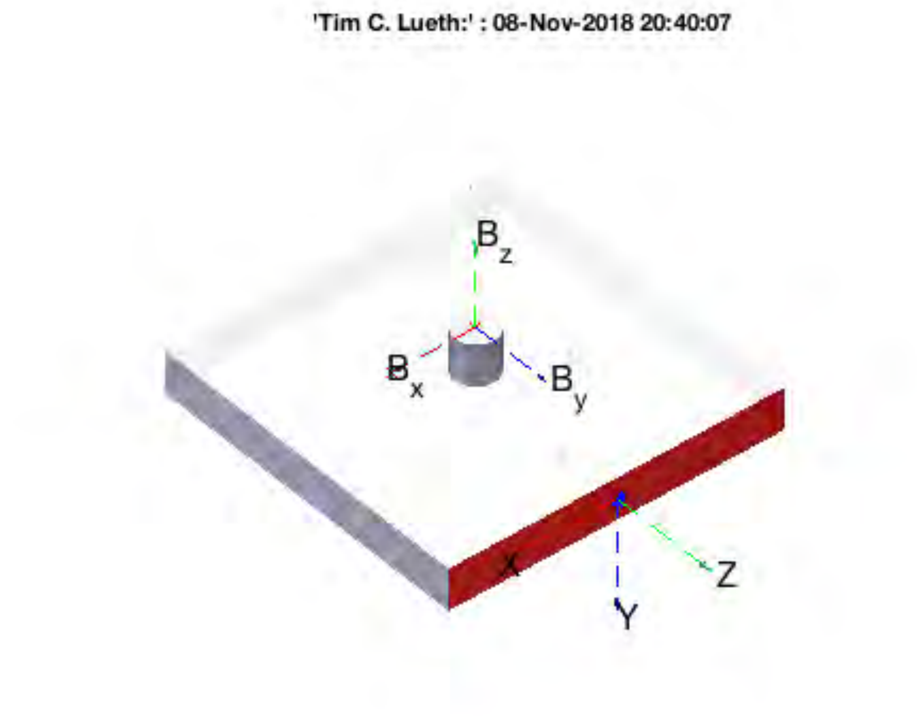
```
        VL: [79×3 double]
        FL: [154×3 double]
     alpha: 0.8000
     color: 'm'
     Tname: {'A'  'B'}
         T: {[4×4 double]  [4×4 double]}
      TFiL: {[2×1 double]  [21×1 double]}
```

- *Tname* is a cell list contain the ascii-string of the names of the coordinate frames
- *T* is the 4x4 homogenous transformation matrix related to the indexed name
- *TFiL* is an optional index of the facets that belong to the surface that defines the coordinate system To the homogenous transformation matrix out of the struct, the most convinient way is to use the function:

- **SGT** returns for a given solid and a given frame name the 4x4 matrix
- **SGT** draws the part and the frames and the defining facets if there is no output parameter

```
A=SGTui(SG0,'A')
B=SGTui(SG0,'B')

SGTplot(SG0);
view(-40,40);
```

```
A =

  struct with fields:

        VL: [79×3 double]
        FL: [154×3 double]
     alpha: 0.8000
     color: 'm'
     Tname: {'A'  'B'}
         T: {[4×4 double]  [4×4 double]}
      TFiL: {[2×1 double]  [21×1 double]}


B =

  struct with fields:

        VL: [79×3 double]
        FL: [154×3 double]
     alpha: 0.8000
     color: 'm'
     Tname: {'A'  'B'}
         T: {[4×4 double]  [4×4 double]}
      TFiL: {[2×1 double]  [2×1 double]}
```



4. Attaching manually coordinate frames as 4x4 honogenous transformation matrix

Since there is no requirement to use the facets TFiL, T matrices and their name can easily added by a programm during the design phase. Nevertheless, there is also a need to add frames interactively. For that purpose there are two other functions to add or to remove frames.

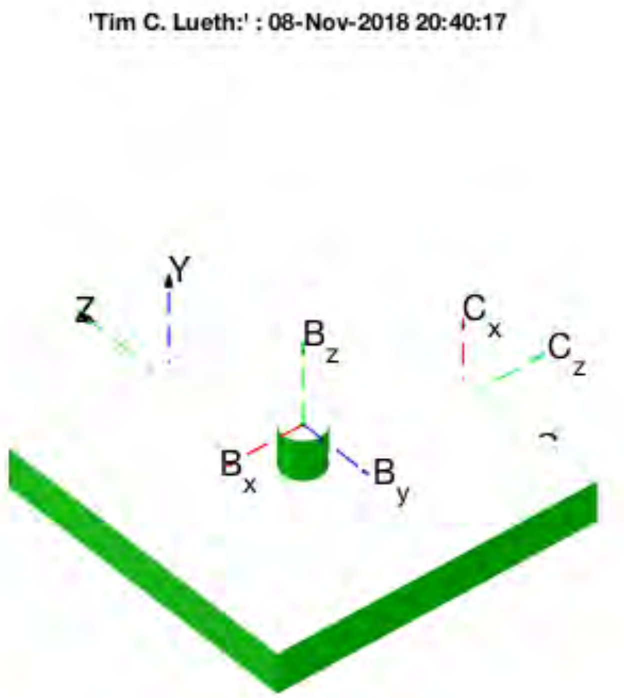
- **SGTremove** removes a named frame from the structure
- **SGTui** opens a figure and allows to generate a frame by touching a surface or point

To use SGTui you should a) first rotate the part on the screen until you see the surface where you like to set a frame, b) press enter and c) klick on the plane to set the frame. Now set two Frames 'C' and 'D'

```
A=SGTui(SG0,'C')
A=SGTui(A,'D')
view(-40,40);
```

```
A =

  struct with fields:

       VL: [79×3 double]
       FL: [154×3 double]
    alpha: 0.8000
    color: 'm'
    Tname: {'A'  'B'  'C'}
        T: {[4×4 double]  [4×4 double]  [4×4 double]}
```

```
     TFiL: {[2×1 double]  [21×1 double]  [2×1 double]}


A =

  struct with fields:

       VL: [79×3 double]
       FL: [154×3 double]
    alpha: 0.8000
    color: 'm'
    Tname: {'A'  'B'  'C'  'D'}
        T: {[4×4 double]  [4×4 double]  [4×4 double]  [4×4 double]}
     TFiL: {[2×1 double]  [21×1 double]  [2×1 double]  [2×1 double]}
```

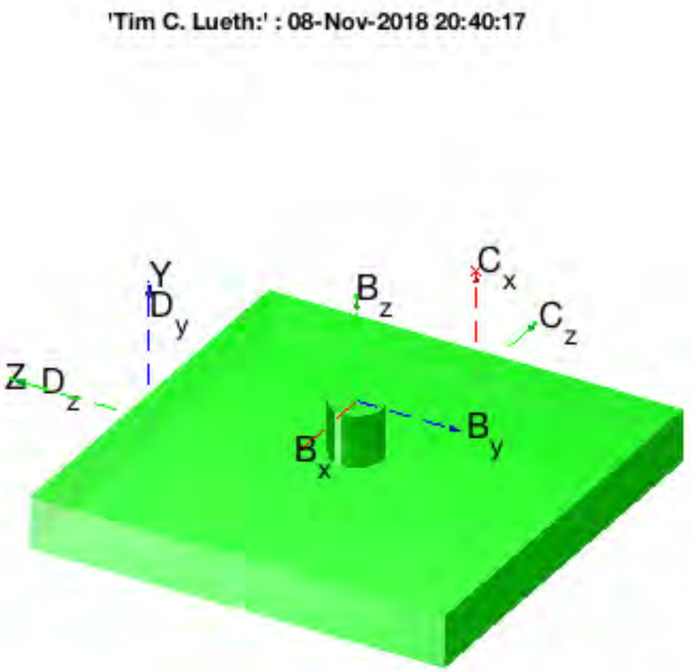


No we remove the first two frames 'A' and 'B'

```
A=SGTremove(A,'A')
A=SGTremove(A,'B')
SGT(A); view(-60,30);
```

```
A =

  struct with fields:
```

```
        VL: [79×3 double]
        FL: [154×3 double]
     alpha: 0.8000
     color: 'm'
     Tname: {'B'  'C'  'D'}
         T: {[4×4 double]  [4×4 double]  [4×4 double]}
      TFiL: {[21×1 double]  [2×1 double]  [2×1 double]}


A =

  struct with fields:

        VL: [79×3 double]
        FL: [154×3 double]
     alpha: 0.8000
     color: 'm'
     Tname: {'C'  'D'}
         T: {[4×4 double]  [4×4 double]}
      TFiL: {[2×1 double]  [2×1 double]}
```



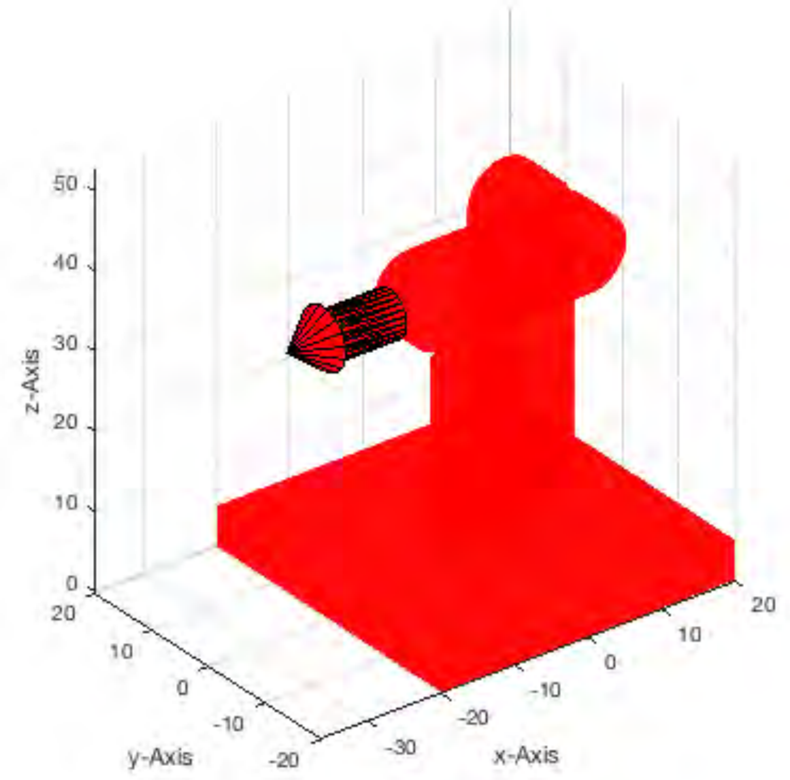
5. Creating kinematic models consisting of named solids

After being able to attach coodinate systems by frames to a solid, we can chain these solids by a string that describes which frames of the indiviudal objects are linked together. For this purpose we define a structure **KM (kinematik model)** that is a list of solids, followed by an ascii identifier and a transformation matrix for the origin of the solid. If the solids are chained, a function **KMchain** calculates those 3rd column transformation matrix to move and rotate the solid so that is fits to the given description of linked frames. **KMplot** shows the position of the individual solids in space.

```
% KM={SG0,'A',eye(4);SG1,'B',eye(4);SG2,'C',eye(4);SG3,'D',eye(4);SG4,'E',eye(4)}

KM.SG={SG0,SG1,SG2,SG3,SG4};
KM.Sname={'A','B','C','D','E'};
KM.BT={eye(4),eye(4),eye(4),eye(4),eye(4)};


KMchain(KM,'A.A-A.B-B.A-B.B-C.A-C.B-D.A-D.B-E.A-E.B-');
KM=KMchain(KM,'A.A-A.B-B.A-B.B-C.A-C.B-D.A-D.B-E.A-E.B-')
KMplot(KM);

% Now let us see how the 3rd column matrices describe the position of the
% solids in 3D space to create the robot model
KM.BT{:}
```

```
KM =

  struct with fields:

       SG: {1×5 cell}
    Sname: {'A'   'B'   'C'   'D'   'E'}
       BT: {1×5 cell}


ans =

   -0.0000    1.0000   -0.0000   -0.0000
   -1.0000   -0.0000         0         0
   -0.0000    0.0000    1.0000    2.5000
         0         0         0    1.0000


ans =

    1.0000    0.0000   -0.0000   -0.0120
   -0.0000    1.0000   -0.0000   -0.0120
         0    0.0000    1.0000   15.0000
         0         0         0    1.0000


ans =

   -0.0000   -1.0000    0.0000   -0.0000
    0.0000   -0.0000   -1.0000   -4.9880
    1.0000    0.0000   -0.0000   32.9590
         0         0         0    1.0000


ans =

   -1.0000         0    0.0000   -7.4530
    0.0000   -0.0000   -1.0000  -10.9880
    0.0000   -1.0000   -0.0000   45.4350
         0         0         0    1.0000
```

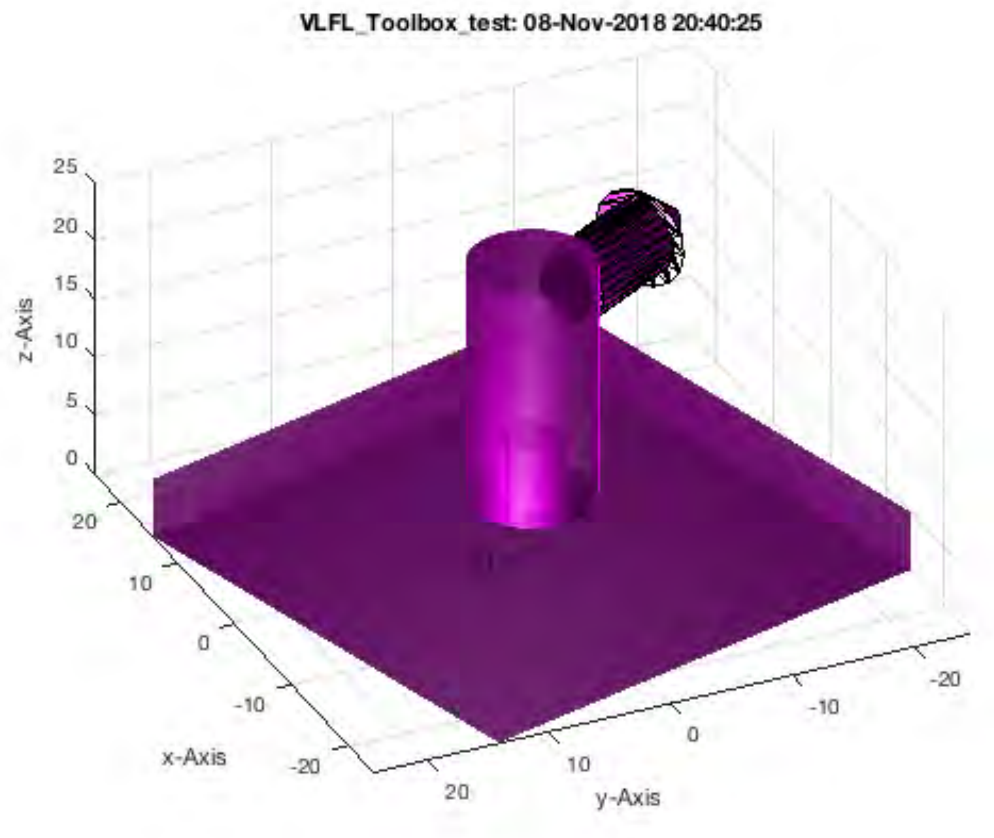
```
ans =

    0.0000   -0.0000   -1.0000  -27.4290
   -0.0000   -1.0000    0.0000  -13.9760
   -1.0000   -0.0000    0.0000   45.4470
         0         0         0    1.0000
```



6. Automatic creation of a chain

```
KMofSGs({SG0,SG1,SG4})
```

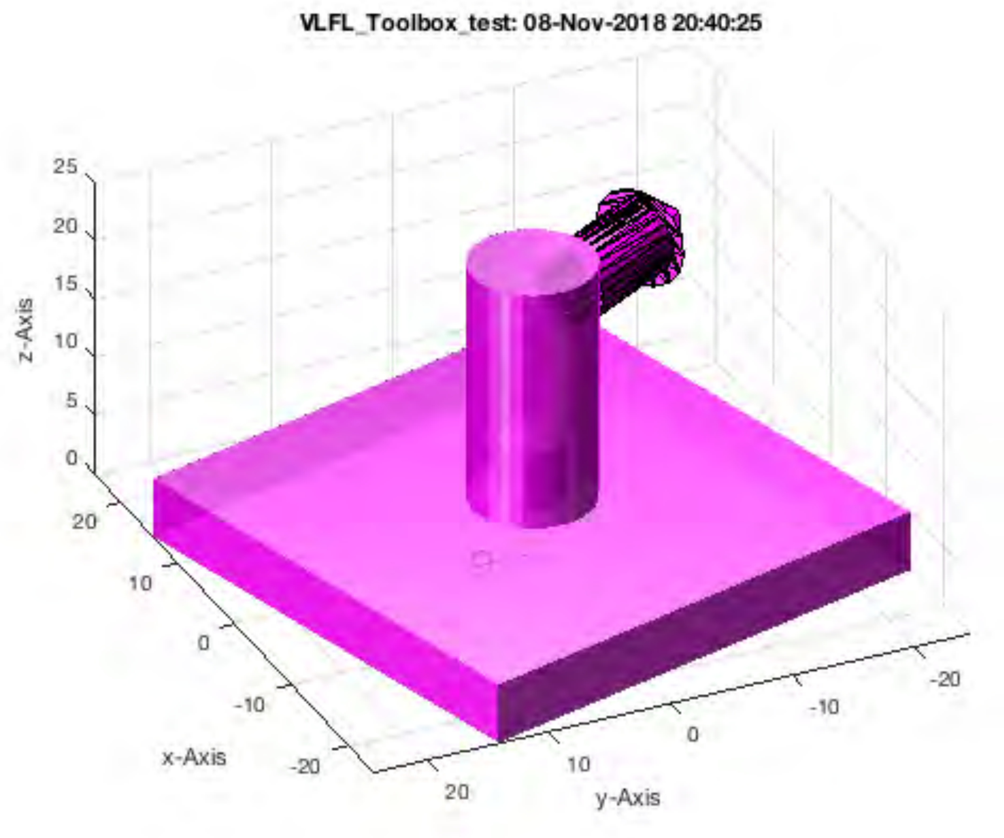
```
KMofSGs: No collisions found for tolerance: 0.10

ans =

  struct with fields:

       SG: {[1×1 struct]  [1×1 struct]  [1×1 struct]}
    Sname: {3×1 cell}
       BT: {3×1 cell}
       KC: {'A.A-A.B-B.A-B.B-C.A-C.B-'}
```



Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:40:26!
Executed 08-Nov-2018 20:40:28 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
simulink
video_and_image_blockset
===========================================================================================
==========
```



- *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® R2018a

Tutorial 12: Define Robot Kinematics and Detect Collisions

2015-08-09: 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

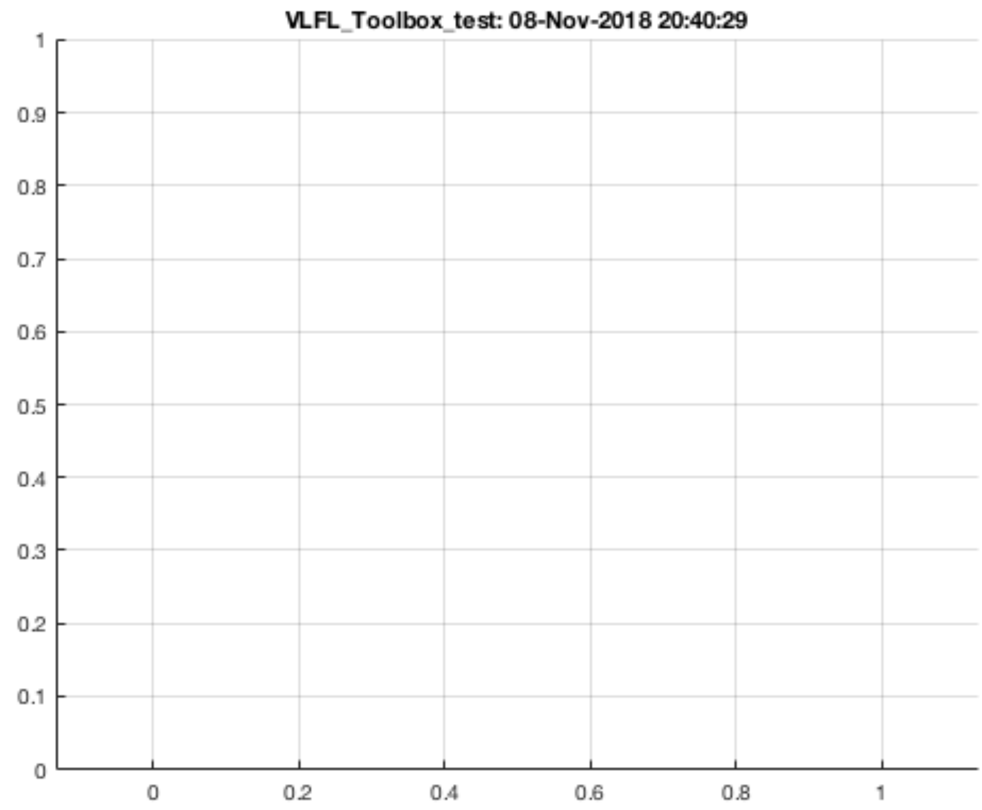
Motivation for this tutorial: (Originally SolidGeometry 2.4 required)

2. Loading the 5 components of a 4DoF robot solid model as last time

Before explaining how to create the parts of a robot kinematik we just load such components in. The command line load AIM_robot

```
function VLFL_EXP12
```

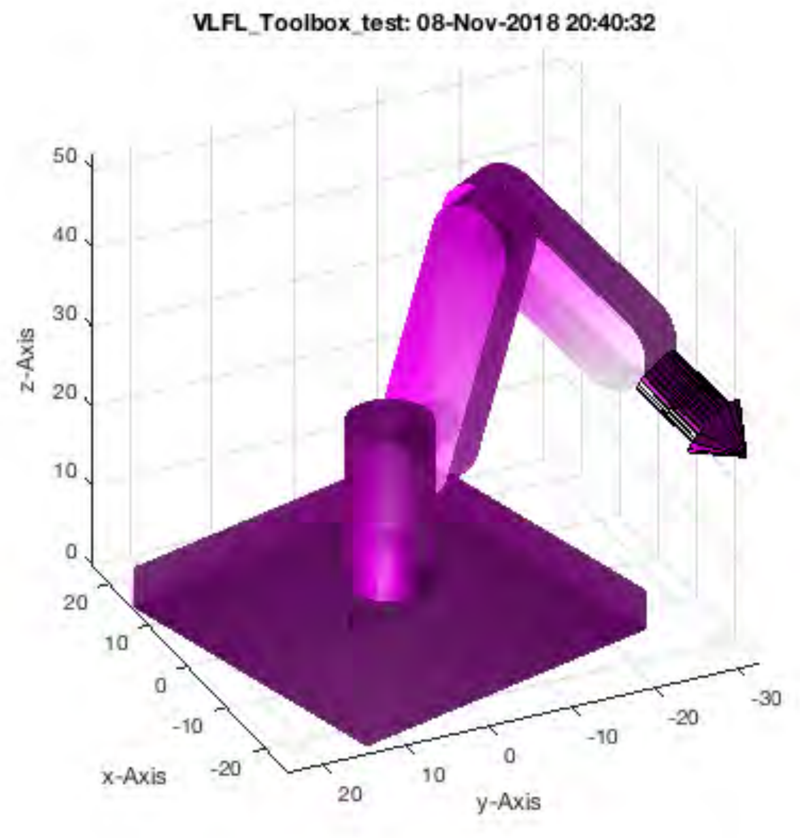
```
close all; SGfigure;
load ('AIM_SGrobot')
SG0=SGfixerrors(SG0,1e-3); SGchecker(SG0);
SG1=SGfixerrors(SG1,1e-3); SGchecker(SG1);
SG2=SGfixerrors(SG2,1e-3); SGchecker(SG2);
SG3=SGfixerrors(SG3,1e-3); SGchecker(SG3);
SG4=SGfixerrors(SG4,1e-3); SGchecker(SG4);
% save ('AIM_SGrobot','SG0','SG1','SG2','SG3','SG4','SGrobot');
VLFLplotlight(1,0.8);
```



3. Automatic creation of a the robot

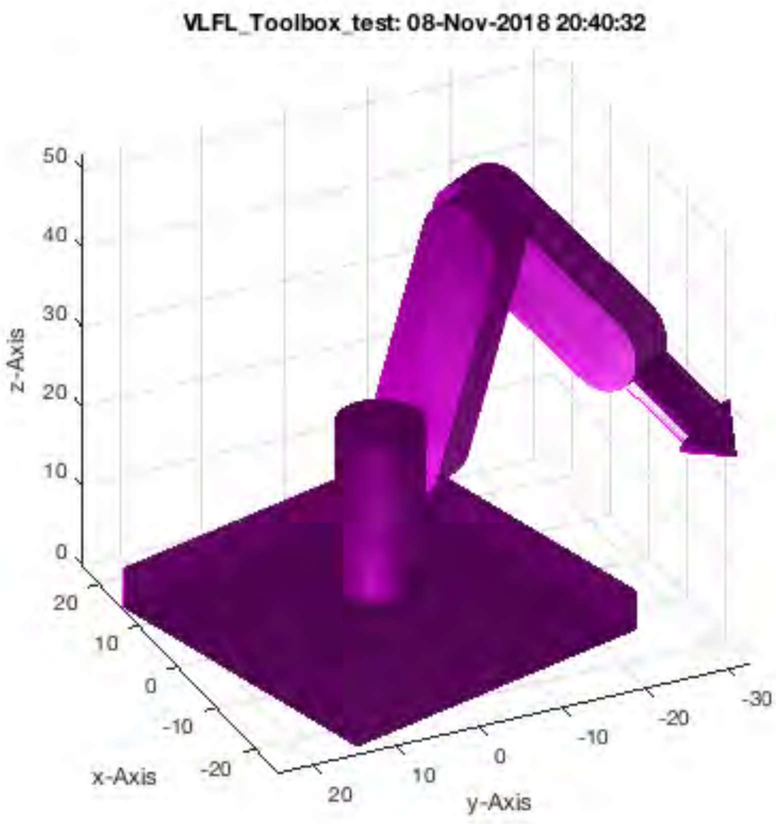
```
KMofSGs({SG0,SG1,SG2,SG3,SG4});
```

```
KMofSGs: No collisions found for tolerance: 0.10
```



3. Collision in the joint by the resolution of the surfaces

```
[KM,XVL]=KMofSGs({SG0,SG1,SG2,SG3,SG4},[],0.05);
KMplot(KM,'m'); VLFLplotlight (1,0.9);
```

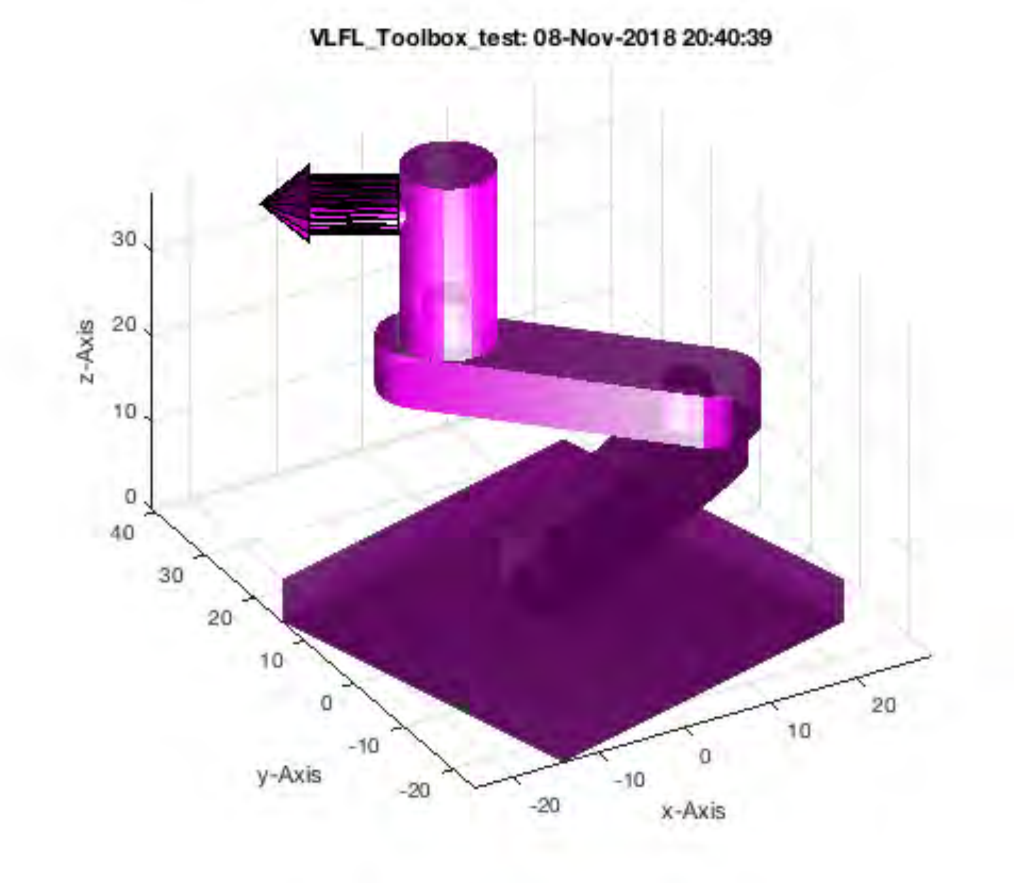



```
if ~isempty(XVL); zoompatch(XVL); VLplot(XVL,'k*',10);  end;
```

4. Showing a different robot

```
KMofSGs({SG0,SG2,SG2,SG1,SG4});
view(-30,30);
```

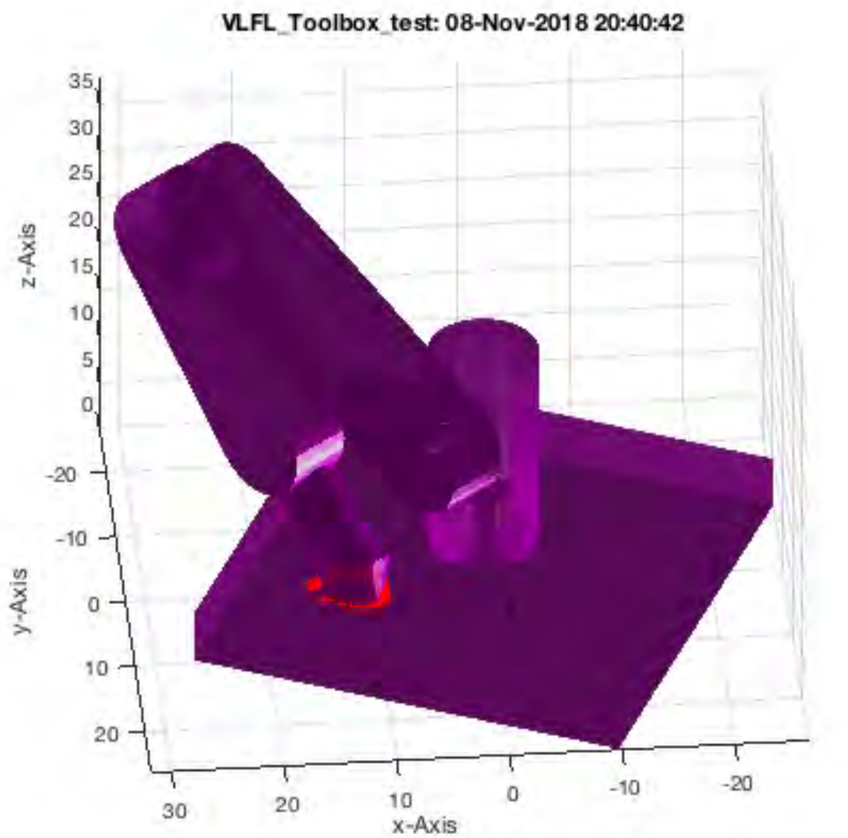
```
KMofSGs: No collisions found for tolerance: 0.10
```



5. Showing a self collision of a robot

```
KMofSGs({SG0,SG1,SG2,SG3,SG4},155);
view(-185,35); VLFLplotlight(1,0.8);
```

```
Warning in KMofSGs: 110 collisions found for tolerance: 0.10
```



Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:40:43!
Executed 08-Nov-2018 20:40:45 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
simulink
video_and_image_blockset
===========================================================================================
==========
```

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

Published with MATLAB® R2018a

Tutorial 13: Mounting Faces and Conversion of Blocks into Leightweight-structures

2015-09-11: 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

Motivation for this tutorial: (Originally SolidGeometry 2.5.1 required)

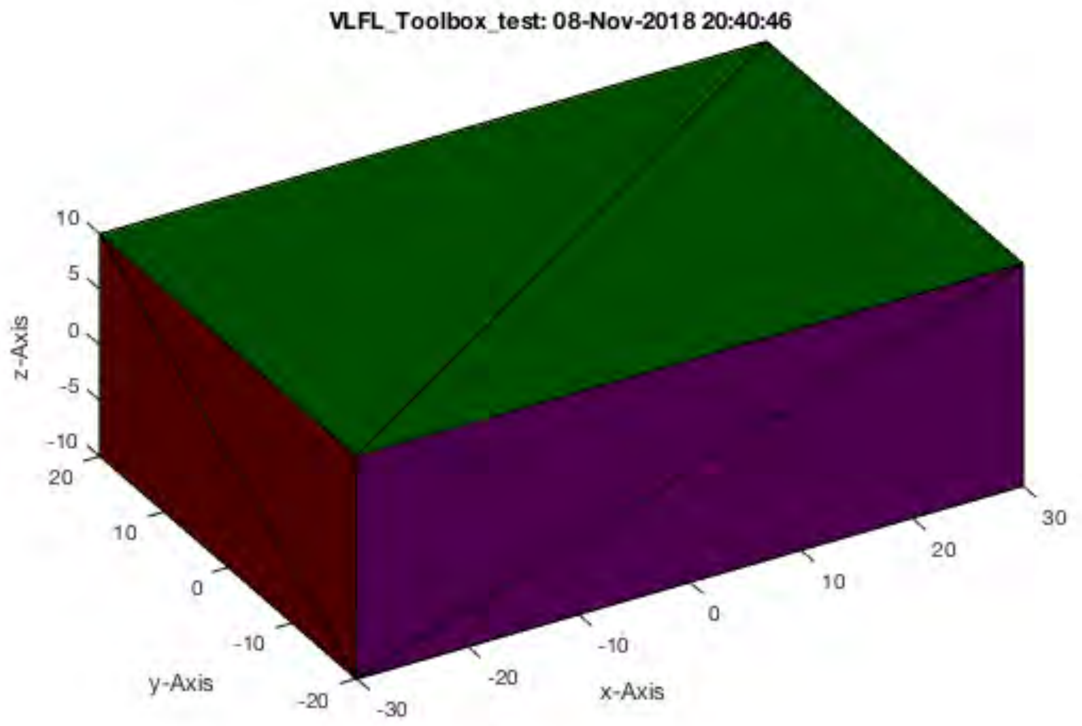
2. Analyzing mouting faces of flat surfaces

All planar faces of a solid can be considered as mounting faces for different design purpses. It is useful to calculate or to handle them using the following functions:

- MLofSG - creates the mounting faces and calculates normal vectors and sizes
- MLplot - plots the mounting faces in different colors

The following example shows the separation of a solid into a set of mouting faces wich are represented by a number and a correlation list between triangle faces and mounting faces.

```
close all; SGfigure; view (-30,30);
[ML,MA,SG]=MLofSG(SGbox([60,40,20]));
MLplot(SG);
```



- ML defines for all entries of FL the corresponding mounting face.
- In this example, 12 faces are ordered to 6 mounting faces

```
ML
```

```
ML =

     1
     1
     2
     2
     3
     3
     4
     4
     5
     5
     6
     6
```

- MA describes for each mounting face, the number, the size, and the normal vector.
- In this example, we see 6 faces with different normal vectors and sizes

```
MA
```

```
MA =

     1        4800           0           0          -1
     2        4800           0           0           1
     3        1600           1           0           0
     4        1600          -1           0           0
     5        2400           0          -1           0
     6        2400           0           1           0
```

- SG is a struct of VL and FL extended by ML and MA

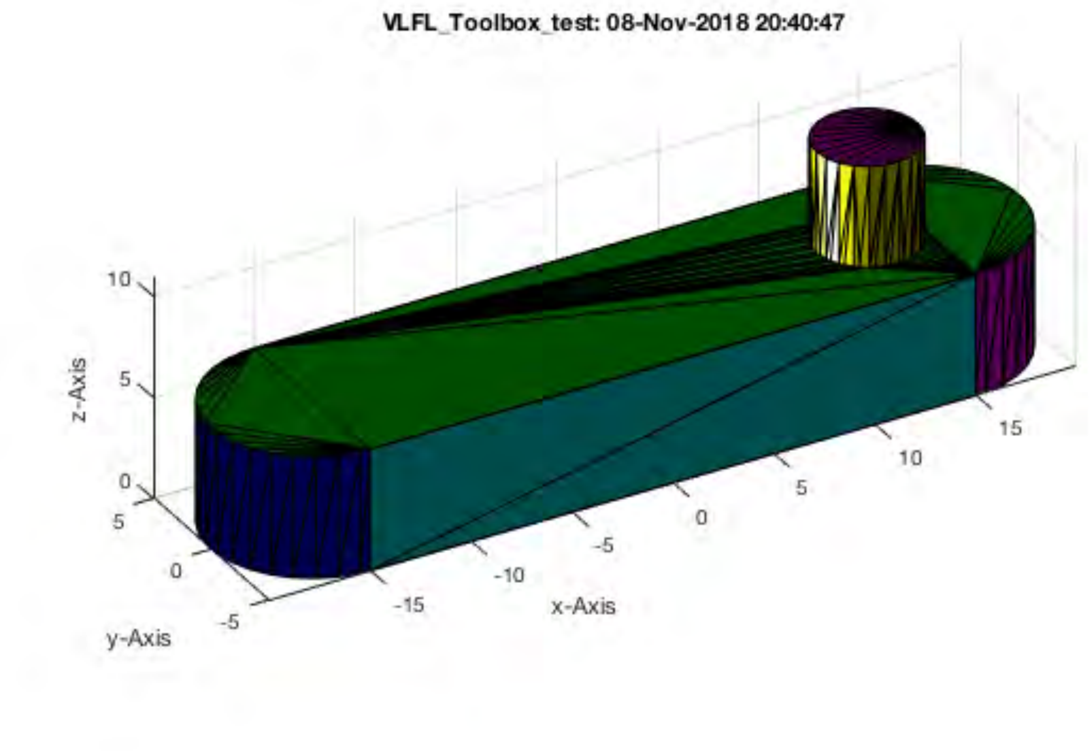
```
SG
```

```
SG =

  struct with fields:

    VL: [8×3 double]
    FL: [12×3 double]
    ML: [12×1 double]
    MA: [6×5 double]
```

3. Analyzing mouting faces of spherical/freeform surfaces

The concepts of mounting faces supports also spherical mounting faces. In case of spherical mounting faces the length of the normal vector is shorter than 1!. This information can be used to distinguish between planar and spherical or freeformed mounting faces

```
close all; SGfigure; view (-30,30);
load AIM_SGrobot
[ML,MA,SG]=MLofSG(SG2);
MLplot(SG);
```

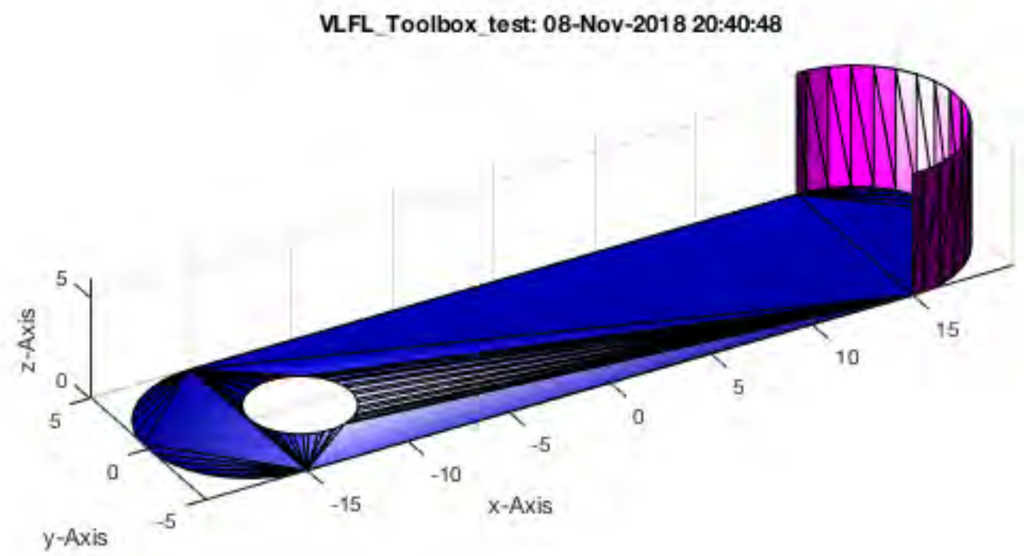


Now we plot only the two half-spherical mounting faces 5 and 1 of the robot link

```
close all; SGfigure; view (-30,30);
MA
MLplot(SG,5); % bended surface, since length of normal vector is less than 1
MLplot(SG,1); % planar surface since length of normal vector is 1
z1=MA(12,3:5) % normal vector of the cylindric surface 1
z2=MA(14,3:5) % normal vector of the cylindric surface 2
```

```
MA =

    1.0000  717.2978         0         0   -1.0000
    2.0000  717.2974         0         0    1.0000
    3.0000  360.0000         0   -1.0000         0
    4.0000    0.2880    1.0000         0         0
    5.0000  188.1982    0.5972         0         0
    6.0000    0.2880    1.0000         0         0
    7.0000  360.0000         0    1.0000         0
    8.0000    0.2880   -1.0000         0         0
    9.0000  188.1982   -0.5972         0         0
   10.0000    0.2880   -1.0000         0         0
   11.0000   38.7854         0         0   -1.0000
   12.0000  156.5960   -0.0001    0.0000    0.0000
   13.0000   38.7829         0         0    1.0000
   14.0000  156.5910    0.0000    0.0000    0.0000
```

```
z1 =

   1.0e-04 *

   -0.6543    0.2818    0.1521


z2 =

   1.0e-04 *

    0.1239    0.3370    0.0978
```

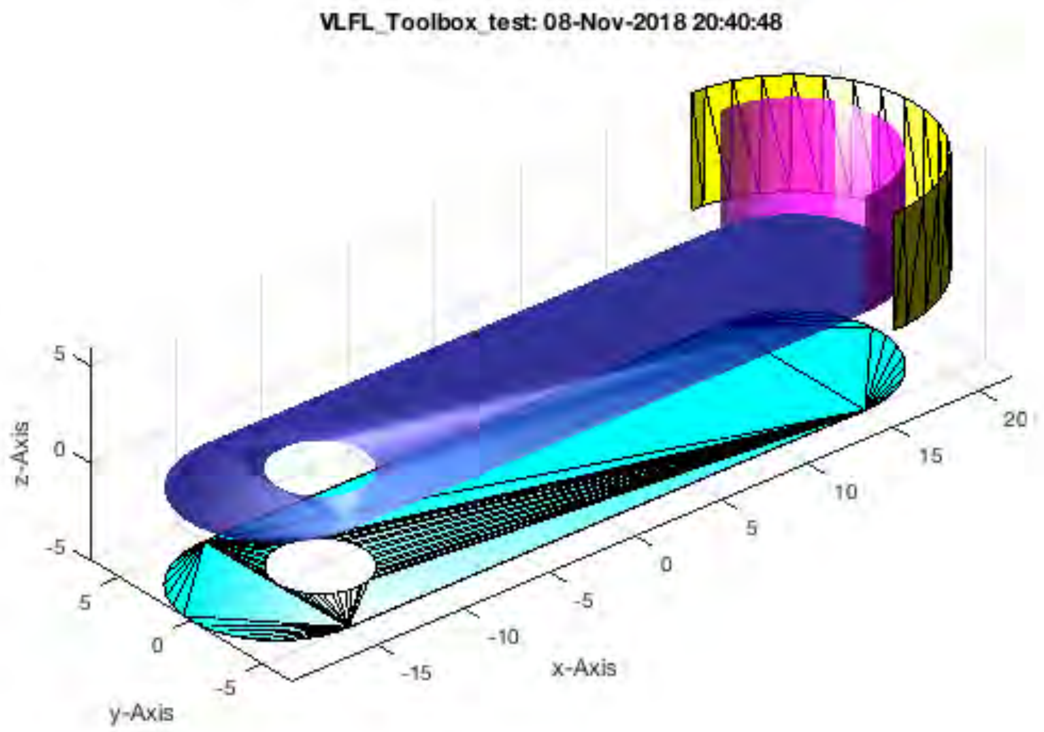


4. Create corresponding surfaces parallel to mounting faces

It is useful to create correspondig surface parallel to mounting faces, which can be smaller or larger than the original one. In the next example it is shown how to create a parallel surface in distance 5mm for a planar surface (#1) and a spherical surface (#5).

- VLtransN(VL,FL,shrink, distance) - helps to create corresonding surfaces

```
VLFLplotlight(1,0.8); view(-40,30);
[VL,~,~,FL]=VLtransN(SG.VL,SG.FL(ML==5,:),0,2);
VLFLplot(VL,FL,'y');

[VL,~,~,FL]=VLtransN(SG.VL,SG.FL(ML==1,:),0,5);
VLFLplot(VL,FL,'c');
```

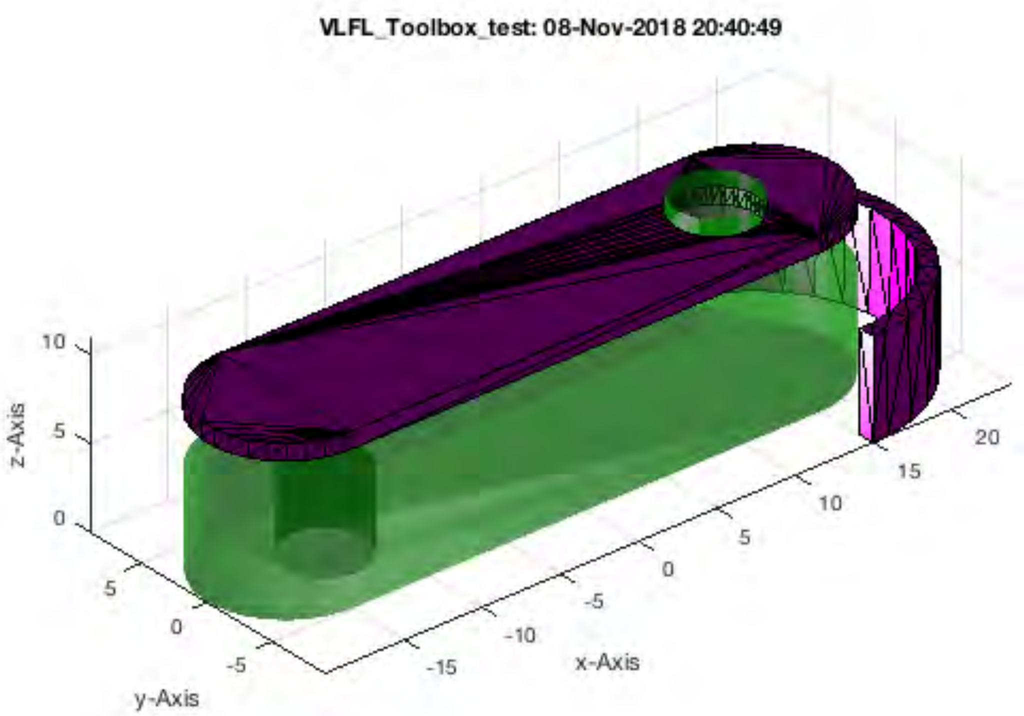


5. Create solids using the parallel surfaces and a plate thickness

Often we want to create a plate solid parallel to the mounting face.

- SGofSurface(VL,FL,thickness, distance, streching) - creates solids parallel to mounting faces.

```
close all; SGfigure; view (-30,30);
SGplot(SG); VLFLplotlight(1,0.5); view(-40,30);
SG2=SGofSurface(SG.VL,SG.FL(ML==2,:),1,3);
SGplot(SG2,'m');
SG2=SGofSurface(SG.VL,SG.FL(ML==5,:),1,3);
SGplot(SG2,'m');
```

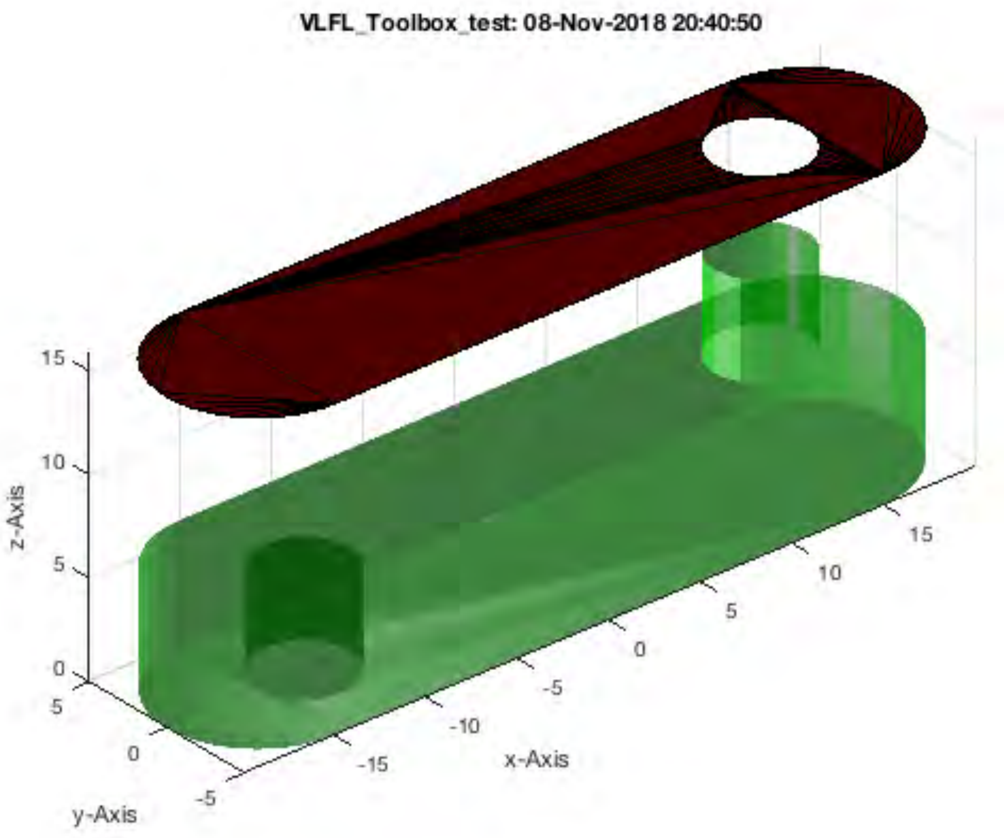


6. Finding the 2D CPL of a planar 3D Surface

Many procedures are based on the manipulation of CPL contours. Nevertheless not all planar surfacer are in the xy-plane. Therefor, there is a function that creates a CPL contour of a surface and returns also the transformation matrix for the back transformation.

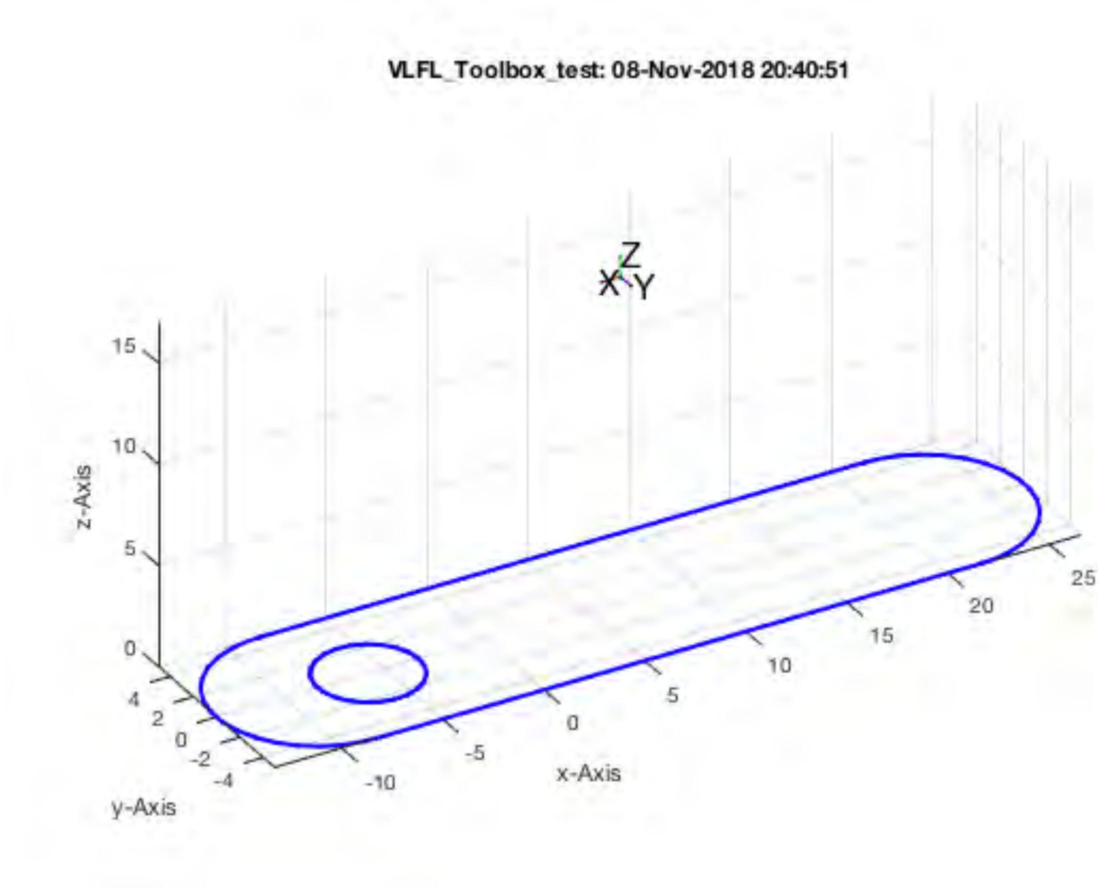
- [PL,T]=PLofVLFL(VL,FL) - returns a PL and a transformation matrix
- [CPL,T]=CPLofVLFL(VL,FL) - returns a CPL and a transformation matrix

```
close all; SGfigure; view (-30,30);
SGplot(SG); VLFLplotlight(1,0.5); view(-40,30);
[VL,~,~,FL]=VLtransN(SG.VL,SG.FL(ML==2,:),0,10);
VLFLplot(VL,FL);
```



Now show simply the isolated CPL of this mounting face

```
close all; SGfigure; view (-30,30); axis on; grid on;
[CPL,T]=CPLofVLFL(VL,FL);
CPLplot(CPL,'b.-',2);
plotT(T);
```

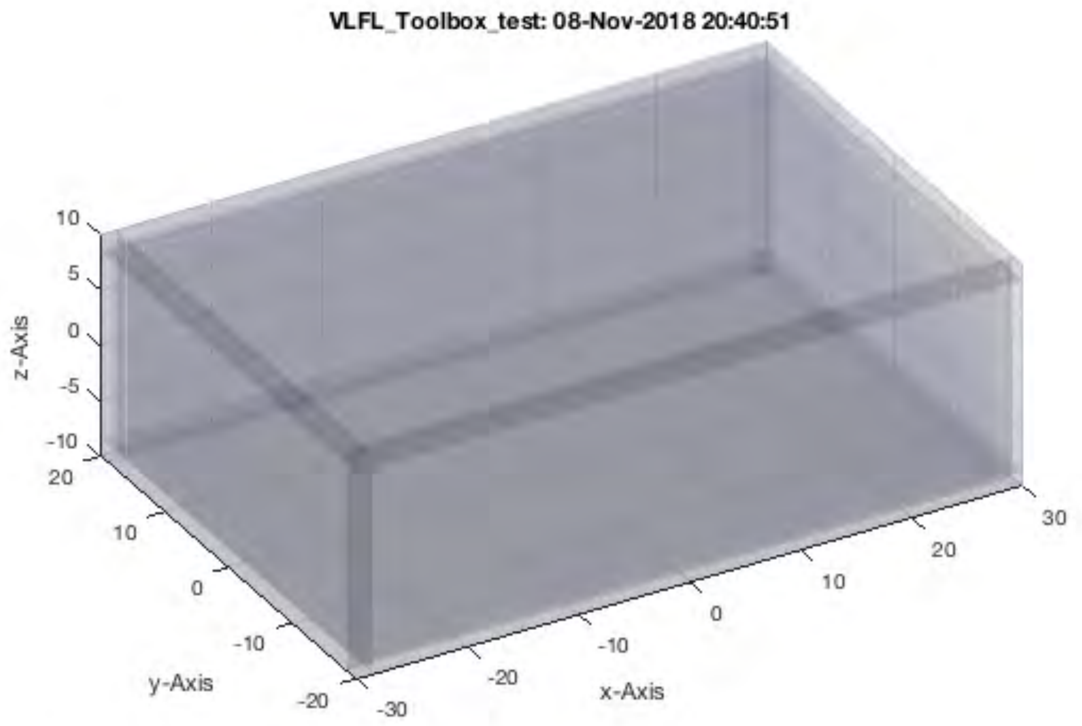


7. Replace a solid block by covering plates

By converting a solidblock into a hollow structure using covering plates, the weight an mass inertia of a solid is reduced.

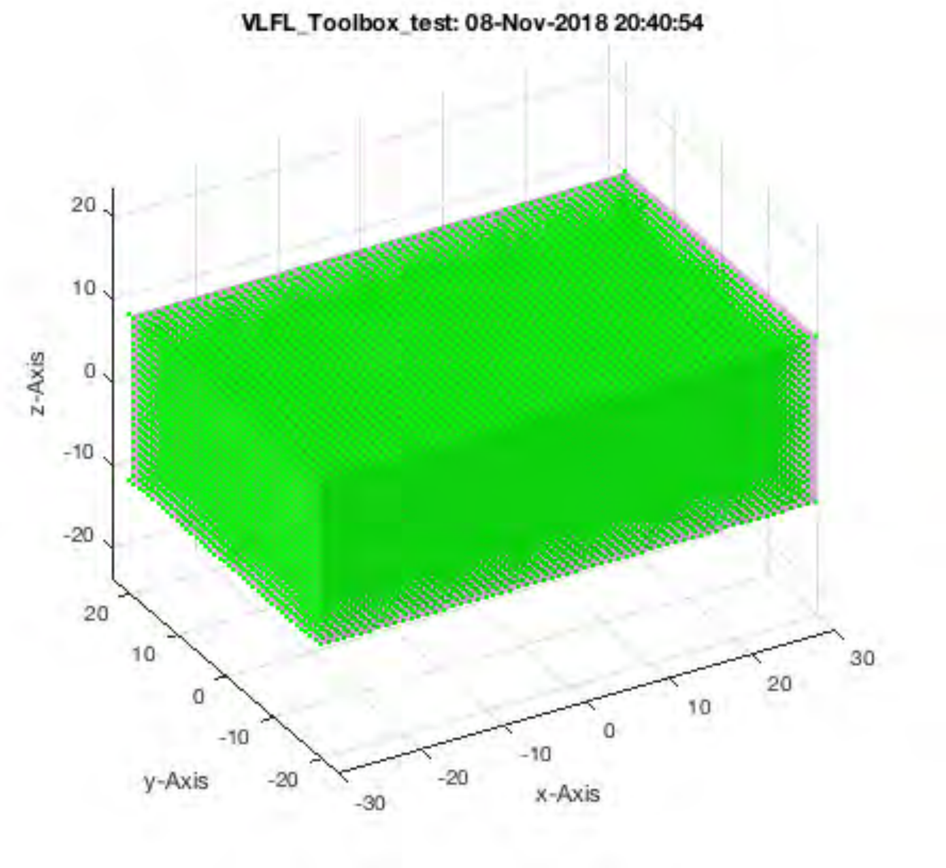
- SGplatesofSGML(SG,thickness) - convert a solid into a plate structure
- SGweight (SG,sepecific weight,resolution) - slowly calculates the weight

```
close all; SGfigure; view (-30,30);

SGN=SGplatesofSGML(SGbox([60,40,20]),1.5);
SGplot(SGN,'w'); VLFLplotlight(1,0.2);
```



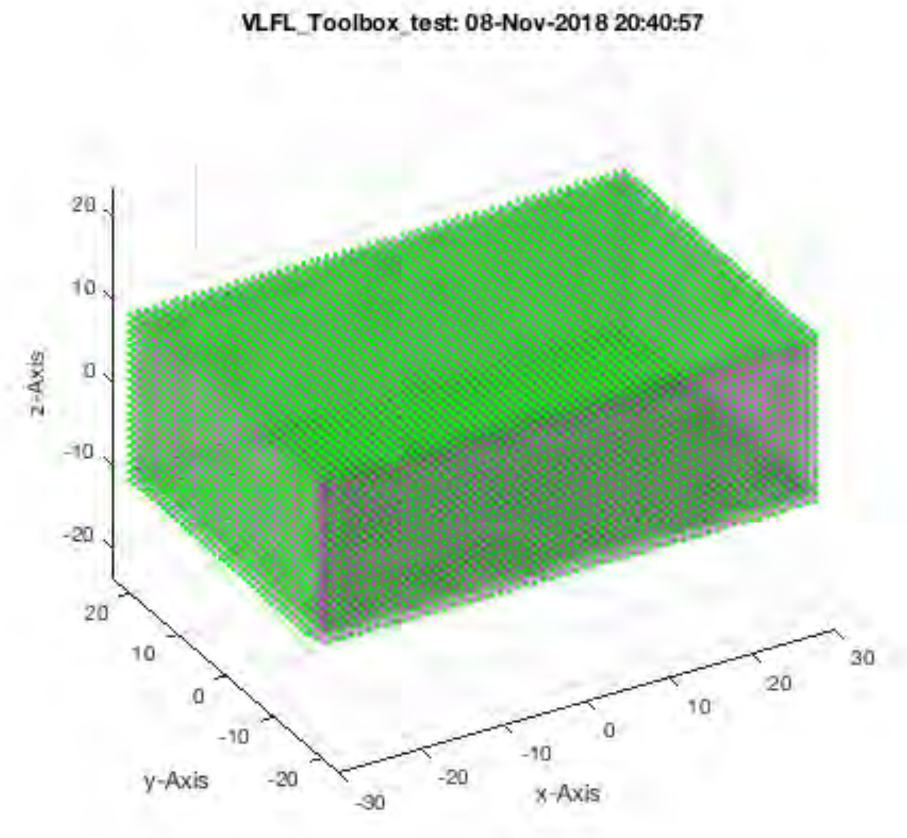

```
SGweight(SGbox([60,40,20]),[],1);
```

```
Using a resolution of 1.0 mm^3 (n=62307) and a specific weight of 1.15 milligramm per mm^3,
 the overall weight is ca. 58 gramm.
Elapsed time is 2.401770 seconds.
```




```
SGweight(SGN,[],1);
```

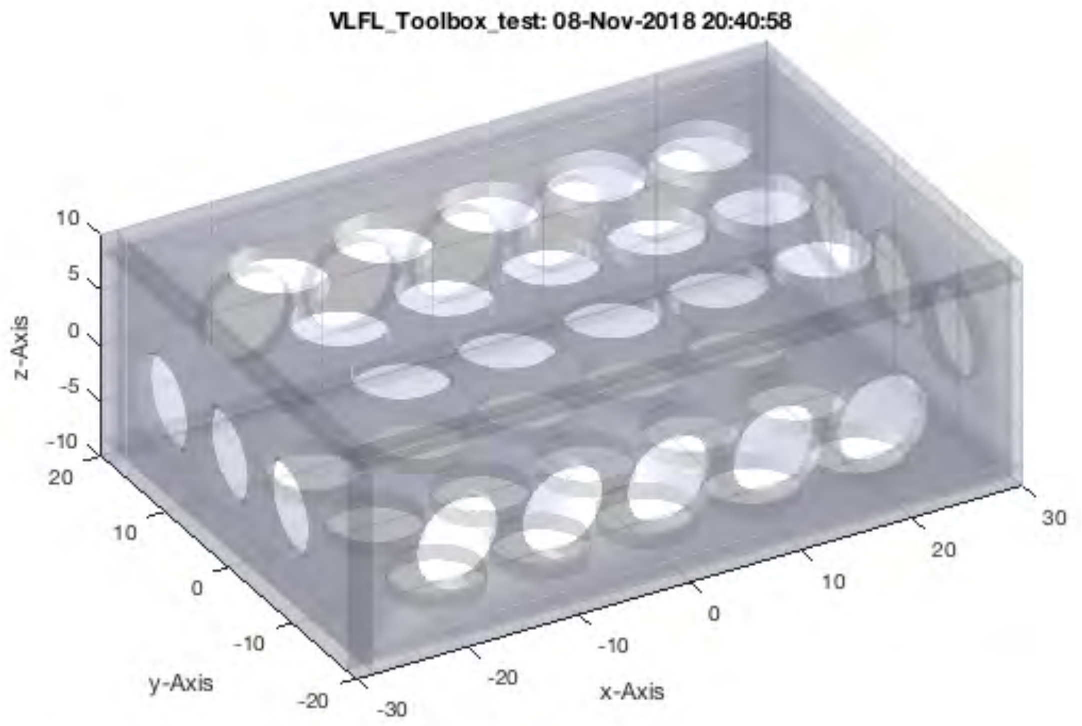
```
Using a resolution of 1.0 mm^3 (n=62307) and a specific weight of 1.15 milligramm per mm^3,
 the overall weight is ca. 11 gramm.
Elapsed time is 2.345526 seconds.
```



8. Replace a solid block by covering plates with punched contours

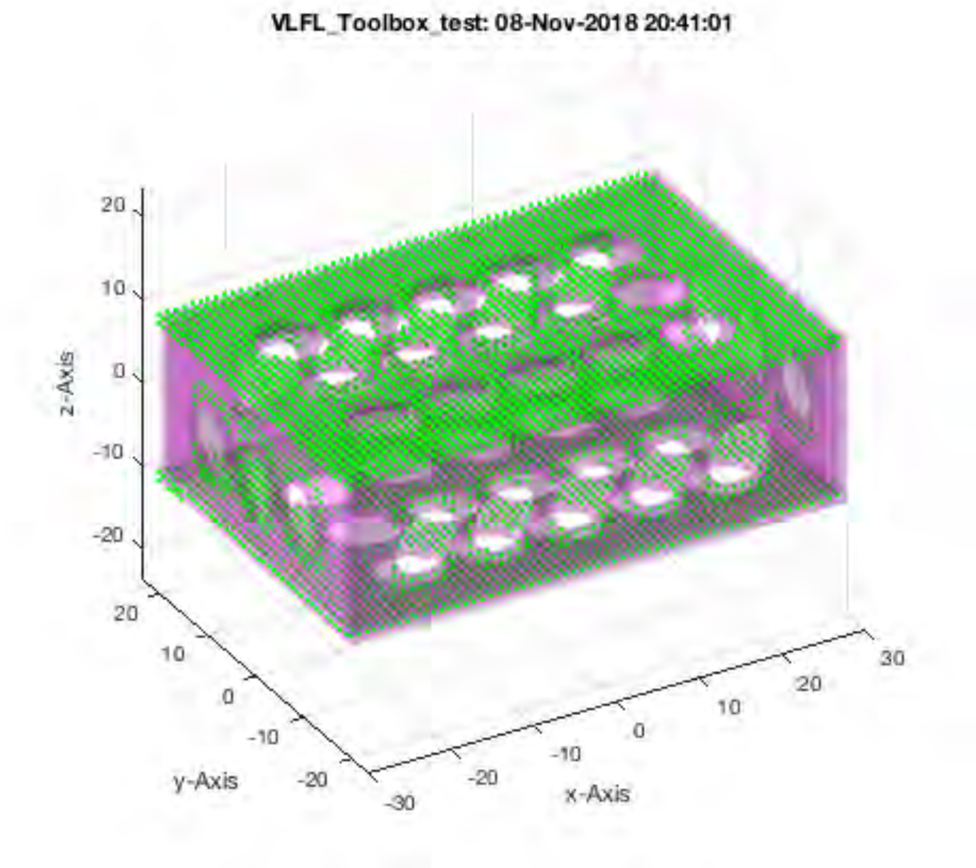
- SGplatesofSGML(SG,thickness,CPL) - convert a solid into a punched plate structure

```
close all; SGfigure; view (-30,30);
SGN=SGplatesofSGML(SGbox([60,40,20]),1.5,PLcircle(4));
SGplot(SGN,'w'); VLFLplotlight(1,0.2);
```




```
SGweight(SGN,[],1);
```

```
Using a resolution of 1.0 mm^3 (n=62307) and a specific weight of 1.15 milligramm per mm^3,
 the overall weight is ca. 7 gramm.
Elapsed time is 2.897552 seconds.
```



Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:41:02!
Executed 08-Nov-2018 20:41:04 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
simulink
video_and_image_blockset
===========================================================================================
==========
```

- *Tim Lueth, tested and compiled on OSX 10.7.5 with Matlab 2014b on 2015-09-11*
- _________, executed and published on 64 Bit PC using Windows with Matlab 2015a on 2015-xx-xx_

Published with MATLAB® R2018a

Tutorial 14: Manipulation Functions for Closed Polygons and Laser Cutting (SVG)

2015-09-20: 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)

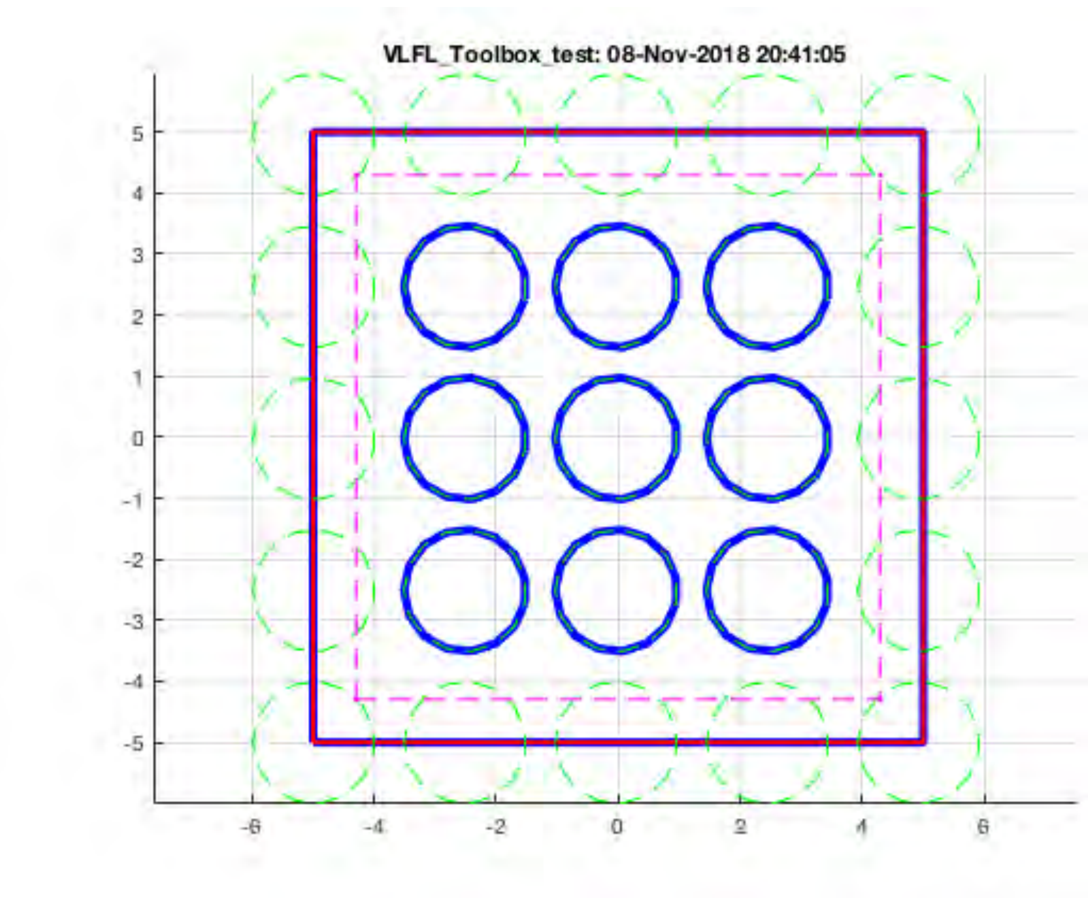
Motivation for this tutorial: (Originally SolidGeometry 2.6.1 required)

2. Fill a contour with copies of pattern

As it was shown already in the function SGplatesofSGML, it often makes sense to fill a contour with another pattern. This can be done by using one of the following functions:

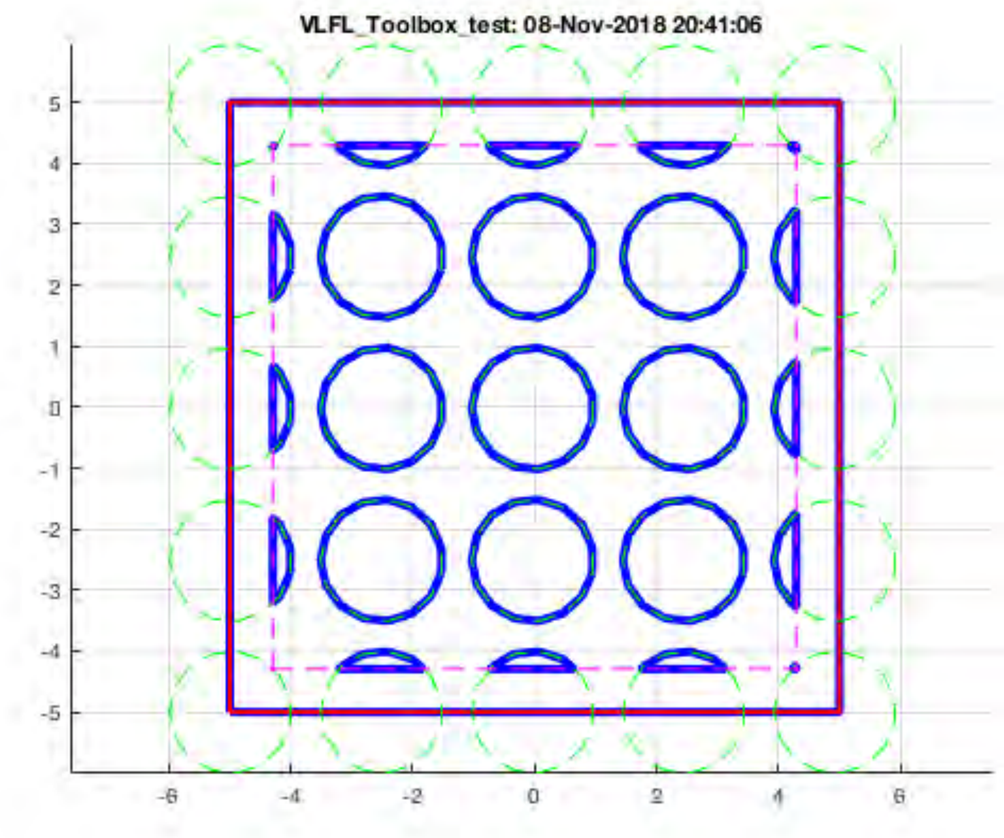
- **CPLfillPattern(CPLA, CPLB,w,d)** - fills a contour CPLA with copies of the pattern CPLB with a distance to the outer contour w and distance between the patterns of d

```
SGfigure; view(0,90); axis on;
CPLfillPattern(PLsquare(10,10),PLcircle(1),1);
```



This can also be done with cutted pattern instead of complete pattern

```
SGfigure; view(0,90); axis on;
CPLfillPattern(PLsquare(10,10),PLcircle(1),1,[],true);
```



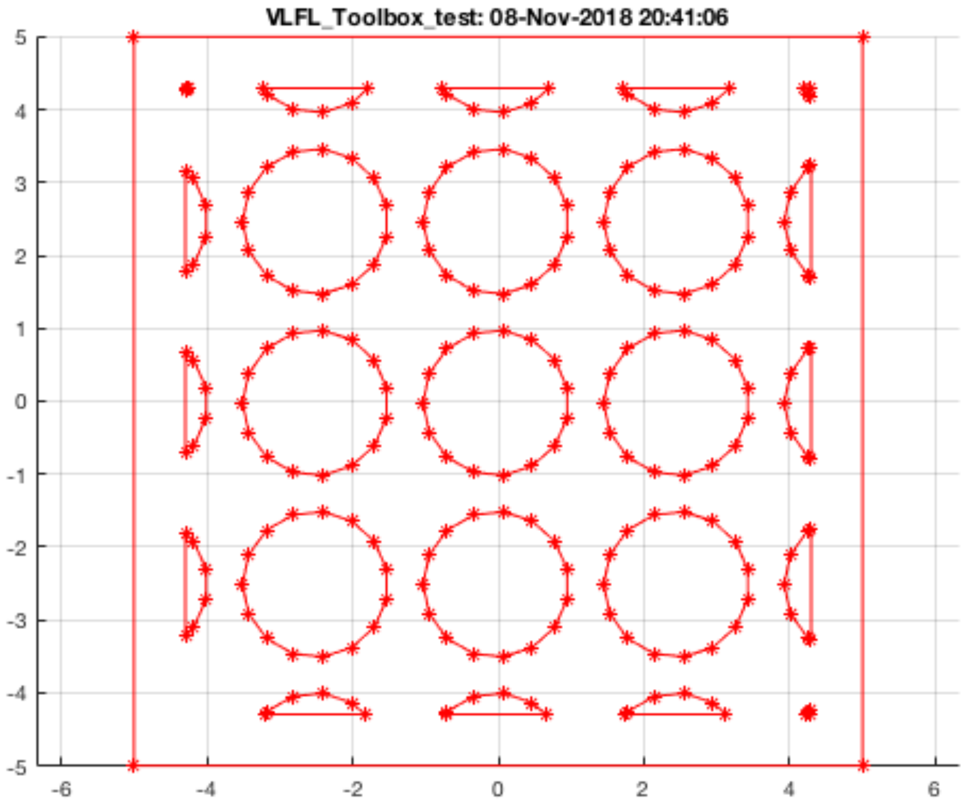
3. Writing a contour as SVG-File for laser-cutting

Especially for laser cutting or platting of contours, the SVG file-format is very popular. Handling of SVG Files is possible using the following functions:

- **CPLwriteSVG (CPL,Filename)** - writes the contours in a SVG-File

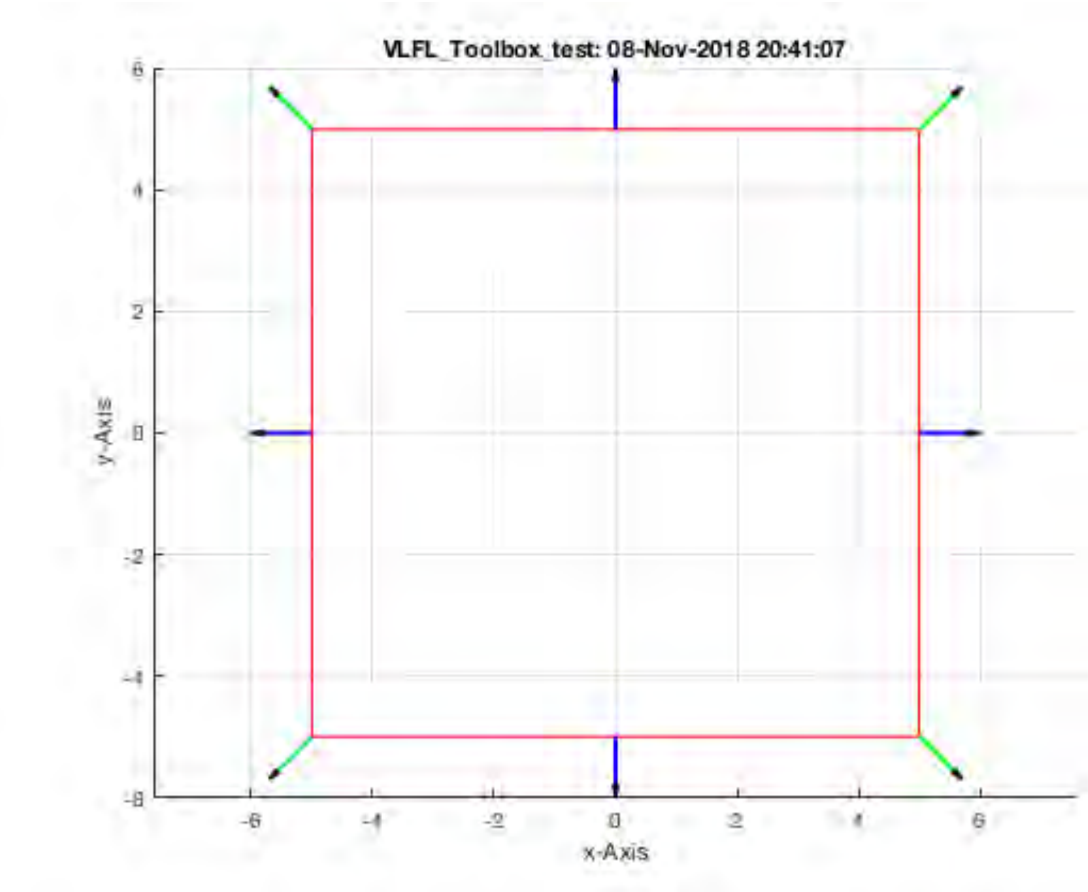
```
SGfigure; view(0,90); axis on;
A=CPLfillPattern(PLsquare(10,10),PLcircle(1),1,[],true);
CPLplot(A);
CPLwriteSVG(A,'VLFL_EXP14');
```

```
WRITING SVG FILE /Users/lueth/Desktop/Toolbox_test/VLFL_EXP14.SVG in ASCII MODE
completed.
```



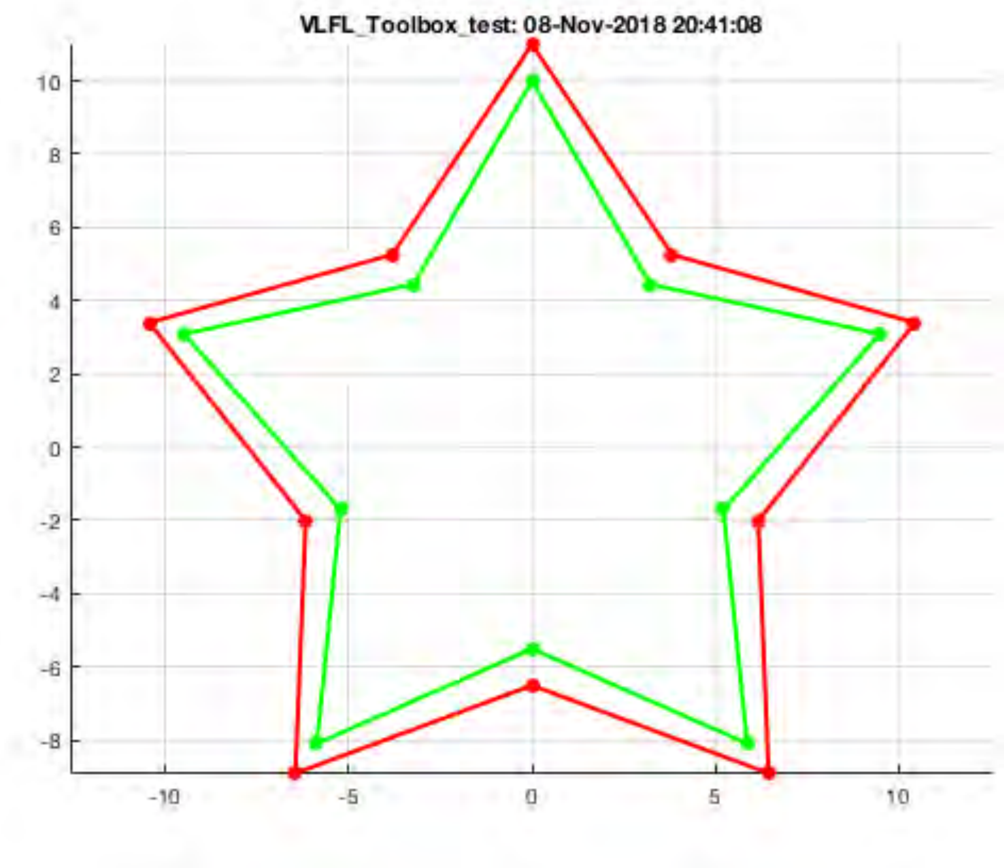
4. Calculating the normal vectors of edges and points

```
SGfigure; view(0,90);
CPLedgeNormal(PLsquare(10,10)); axis on;
```



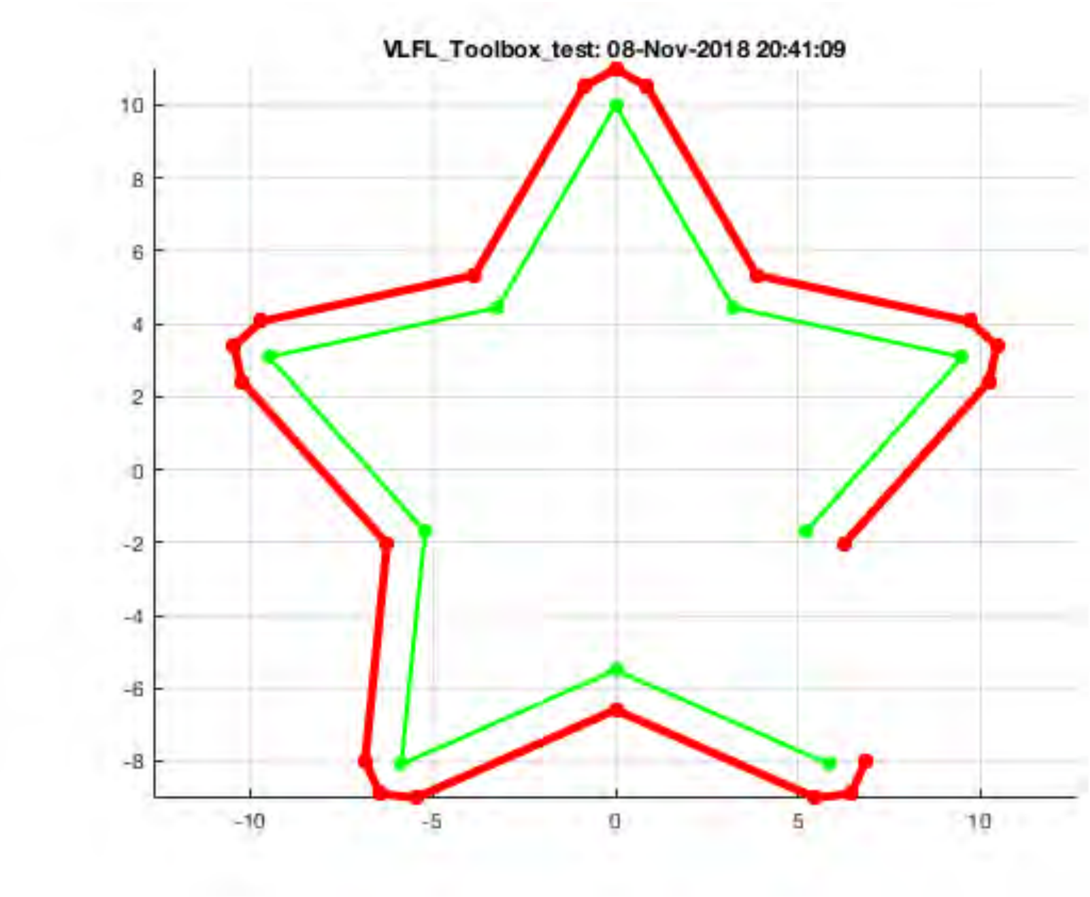
5. Growing with same number of points

```
SGfigure; view(0,90);
CPLgrow(PLstar(10,10),1); axis on;
```



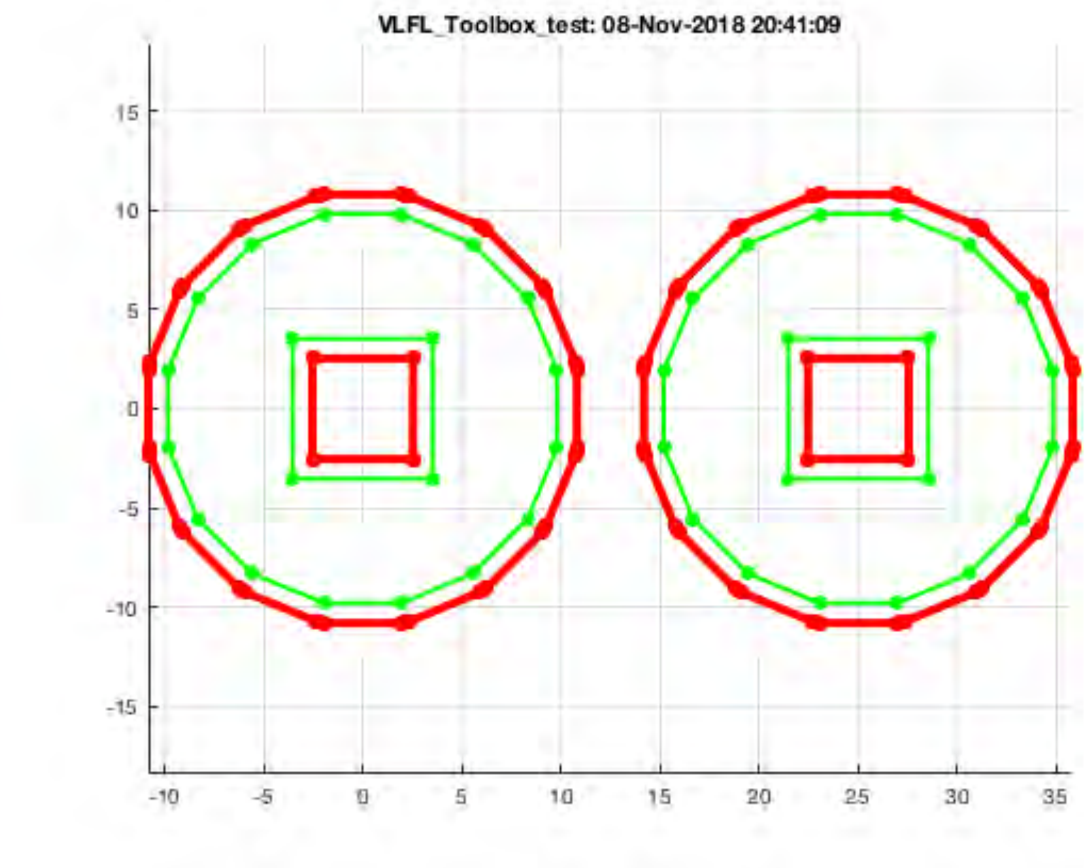
6. Growing with correct distance to edges

```
SGfigure; view(0,90); axis on;
CPLgrowEdge(PLstar(10,10),1);
```



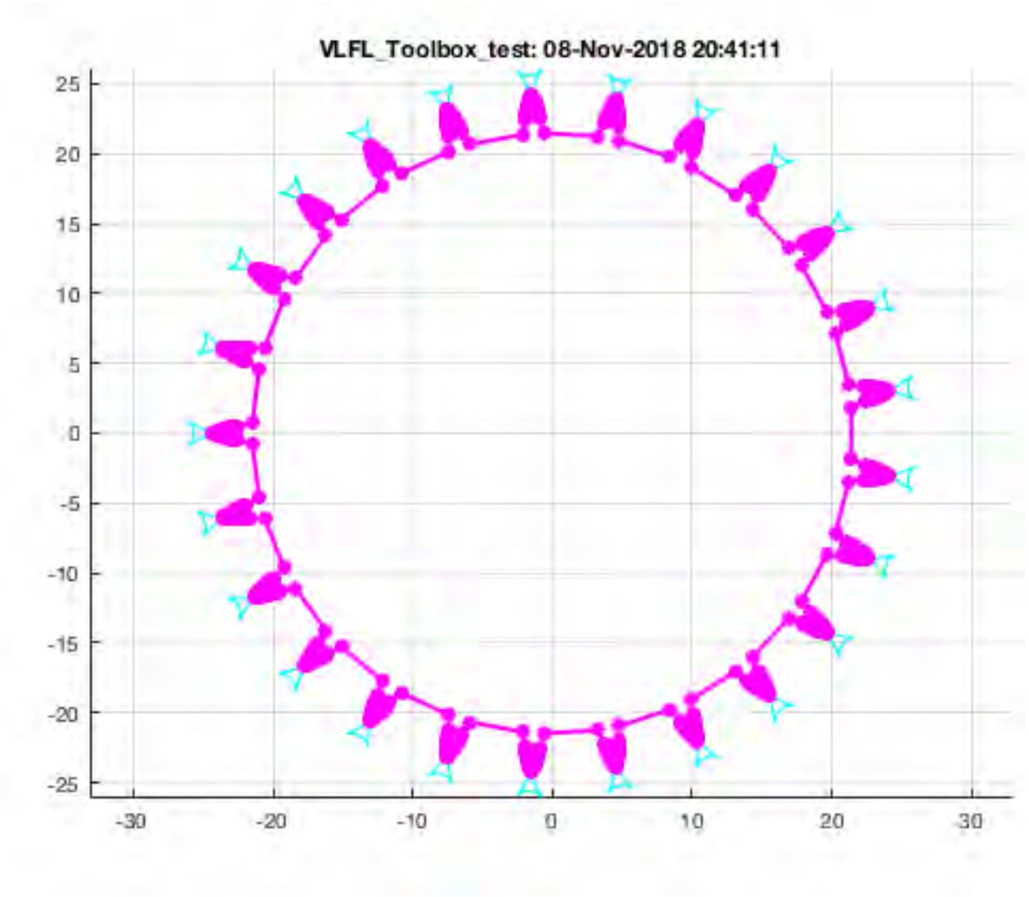
Another example using CPLsample

```
SGfigure; view(0,90);
CPLgrowEdge(CPLsample(12),1); axis on;
```



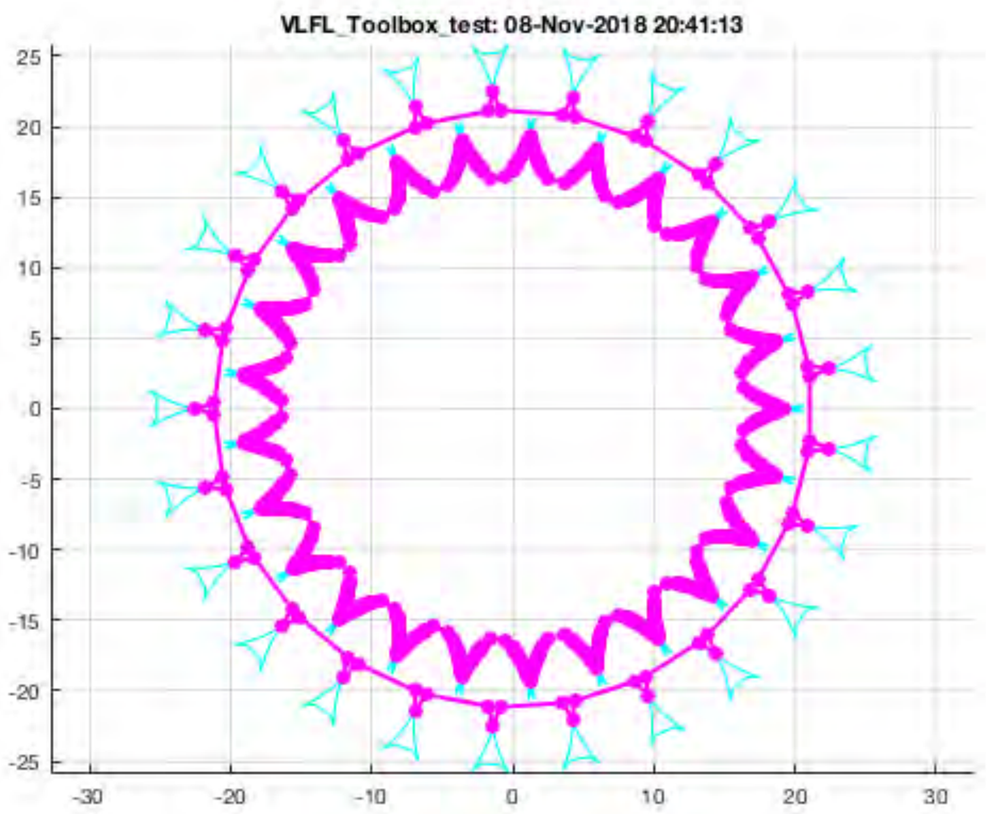
Growing may have no problems

```
SGfigure; view(0,90);
CPLgrow(CPLofPL(PLgearDIN(2,25)),0.5); axis on;

% *Growing may have problems*
SGfigure; view(0,90);
CPLgrow(CPLofPL(PLgearDIN(2,25)),1.5); axis on;

% *Growing problems can be solved using CPLoutercontour*
SGfigure; view(0,90);
CPLoutercontour(CPLgrow(CPLofPL(PLgearDIN(2,25)),1.5));
```

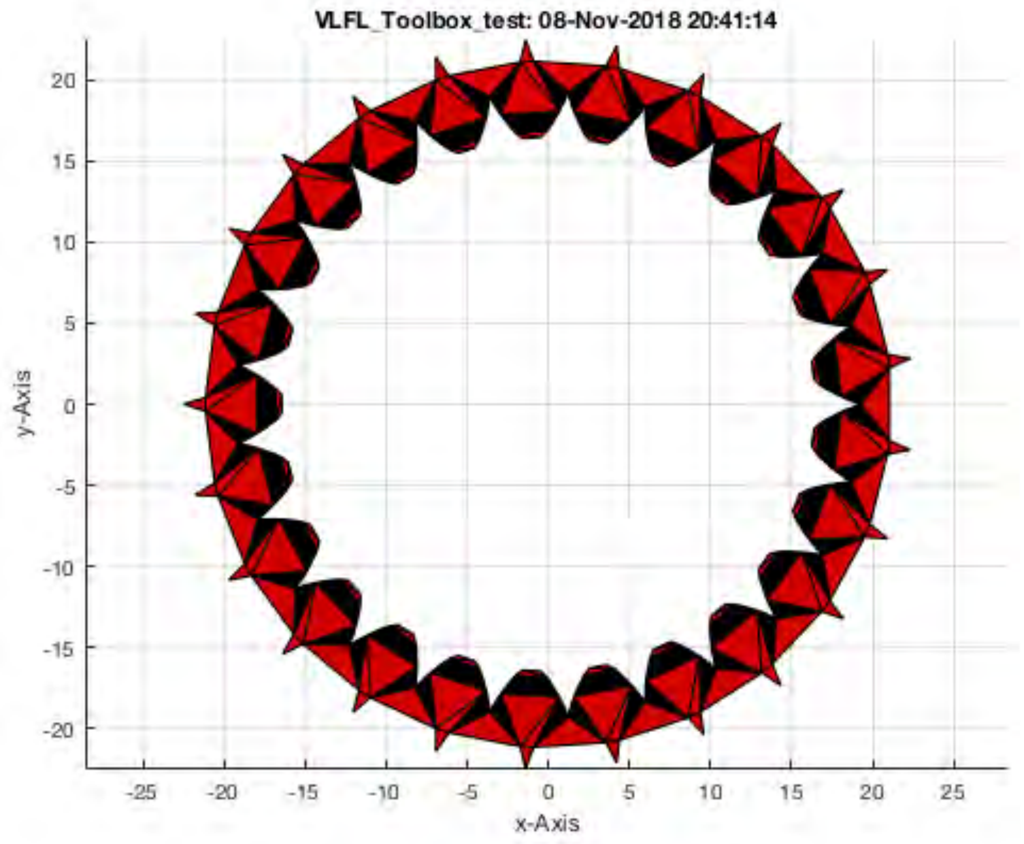


Another example using CPLoutercontour

```
SGfigure; view(0,90);
CPLoutercontour(CPLsample(25),1); axis on;
```



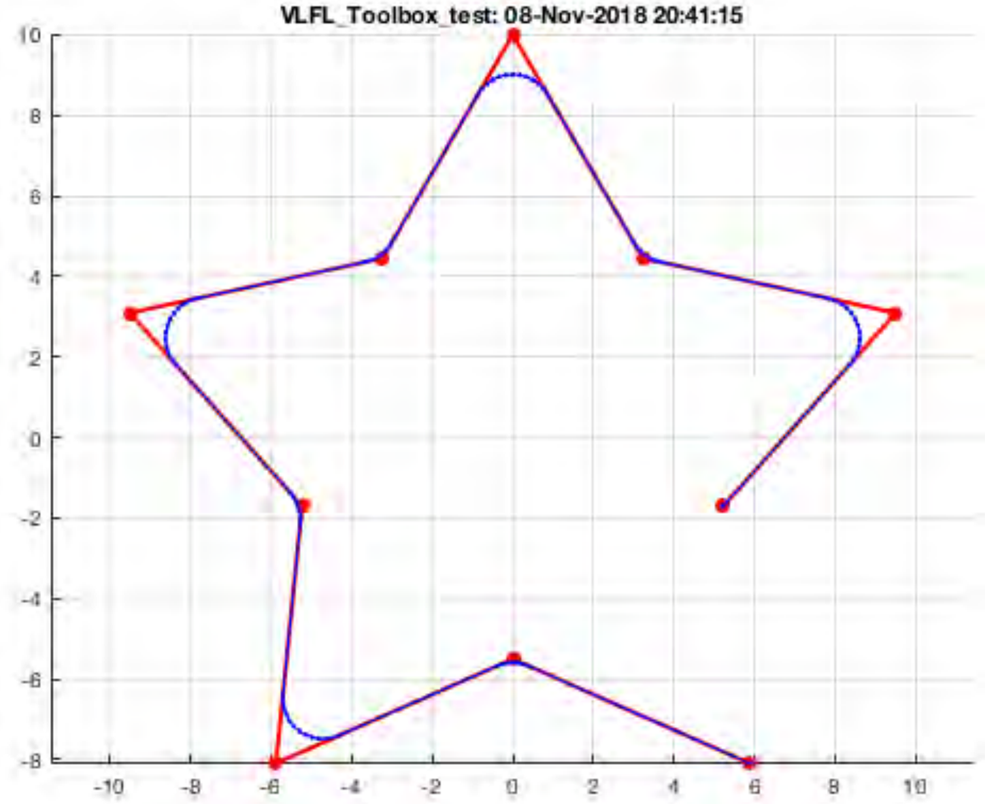

```
PLFLofCPLdelaunay(CPLoutercontour(CPLsample(25),1));
```



7. Rounded edges inside a contour

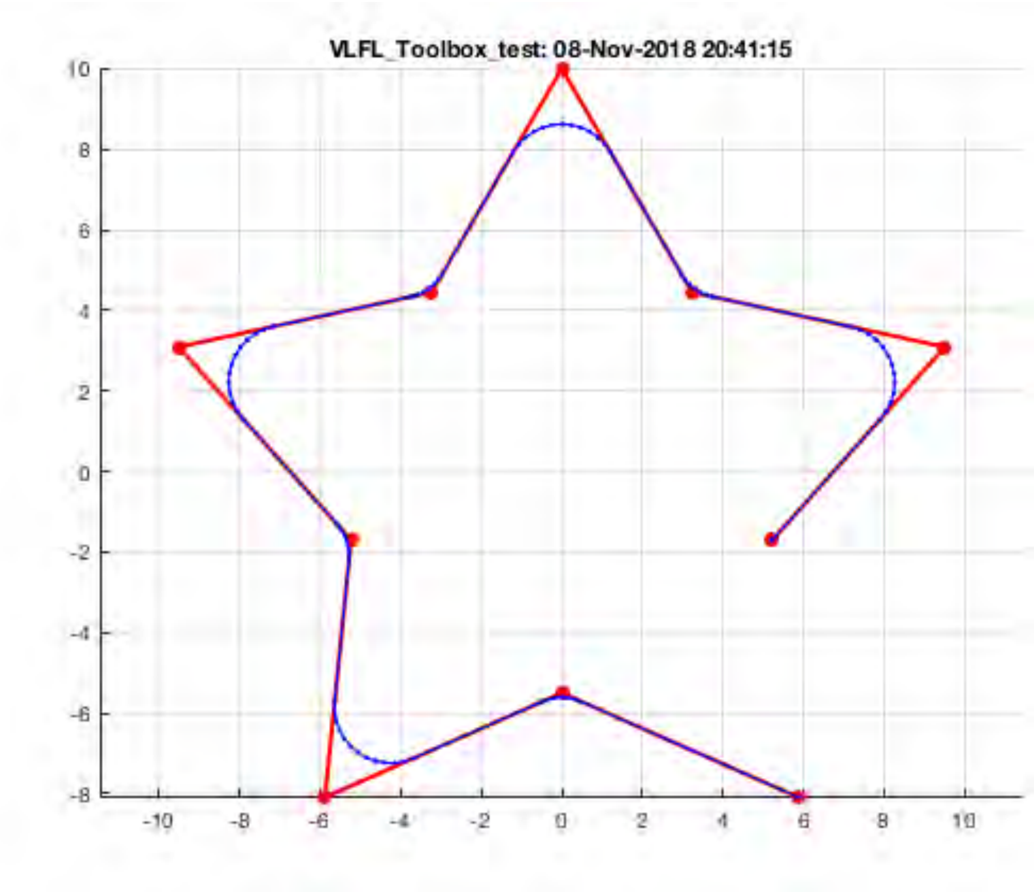
Another method to change the shape of a contour is to round the edges.

```
SGfigure; view(0,90);
PLradialEdges(PLstar(10,10));axis on;
```



Another example using radius=2

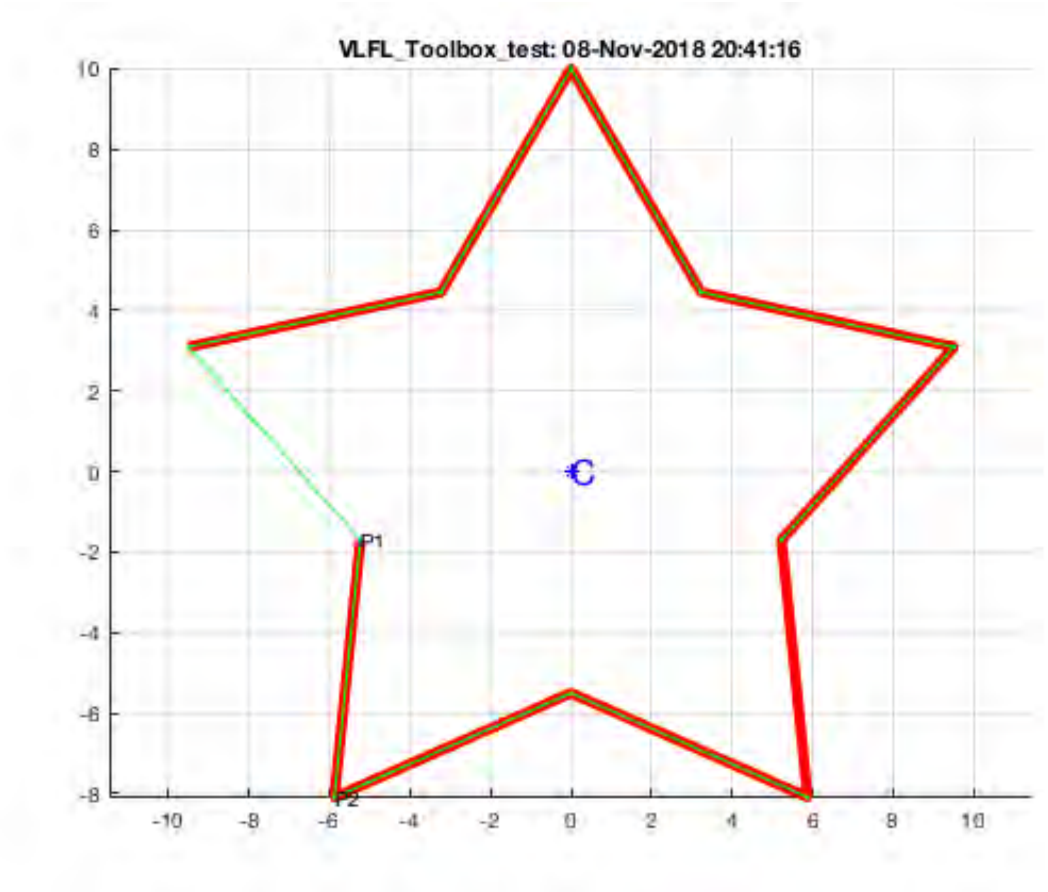
```
SGfigure; view(0,90); axis on;
PLradialEdges(PLstar(10,10),2); axis on;
```



8. Sort CPLs around its center

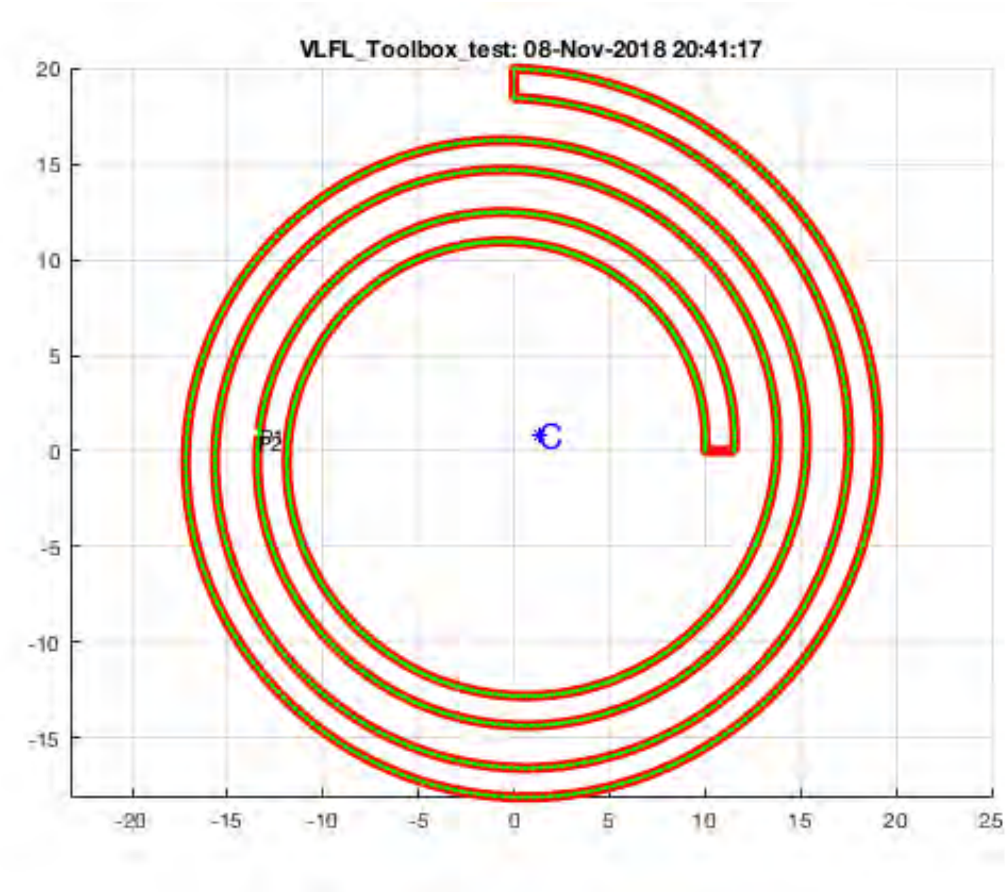
Find the minmal angle value of a star

```
SGfigure; view(0,90);
CPLsortC(PLstar(10,10),'min'); axis on;
```



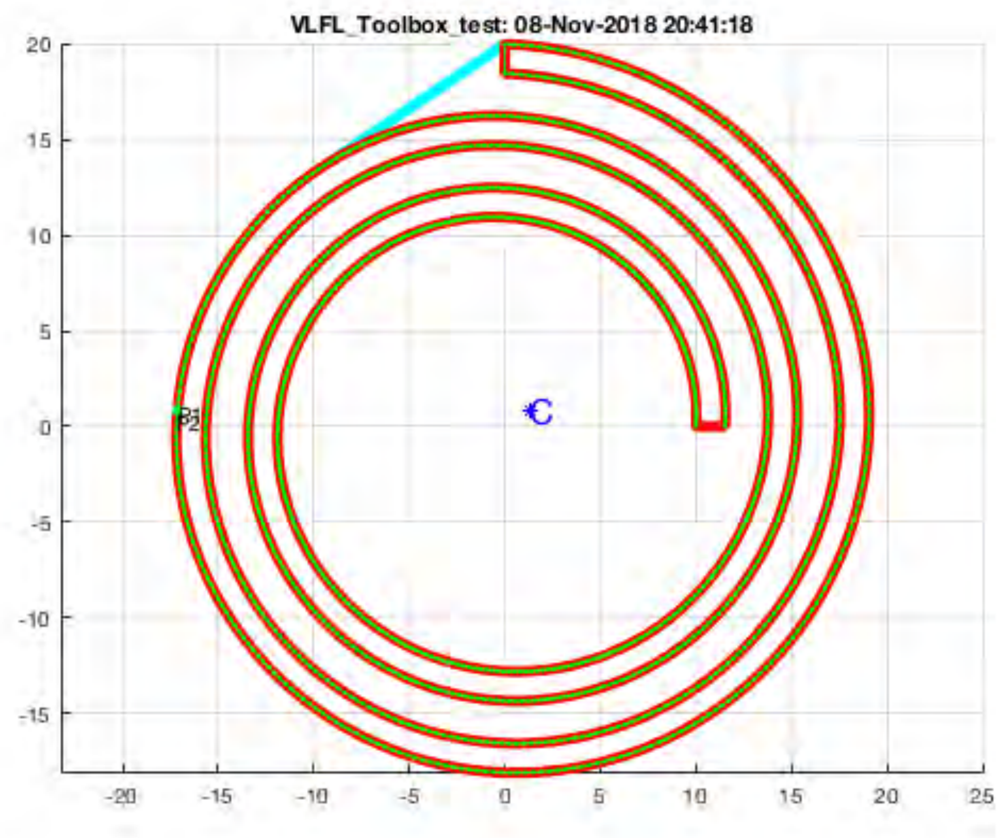
Find the minmal angle value of a spiral

```
SGfigure; view(0,90);
CPLsortC(CPLspiral(10,20,4*pi+pi/2),'min');
```



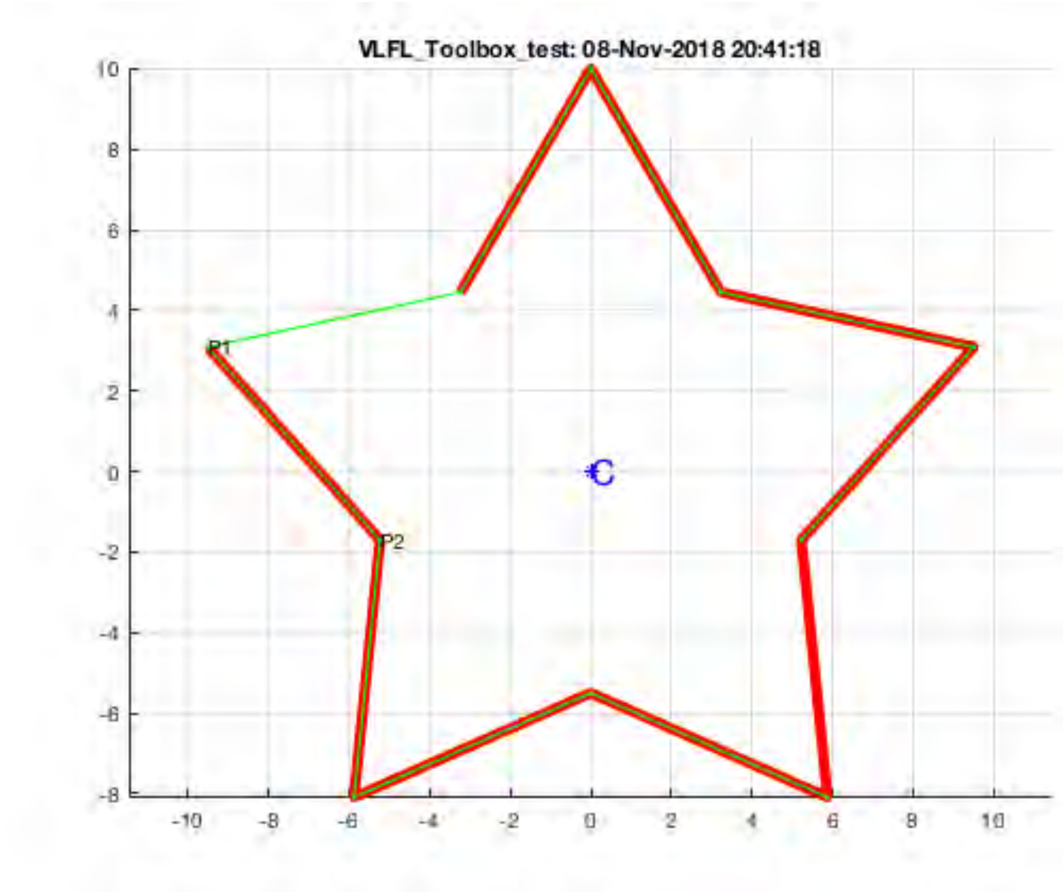
Find the minmal angle value of convex hull of a spiral

```
SGfigure; view(0,90);
CPLsortC(CPLspiral(10,20,4*pi+pi/2),'cmin');
```



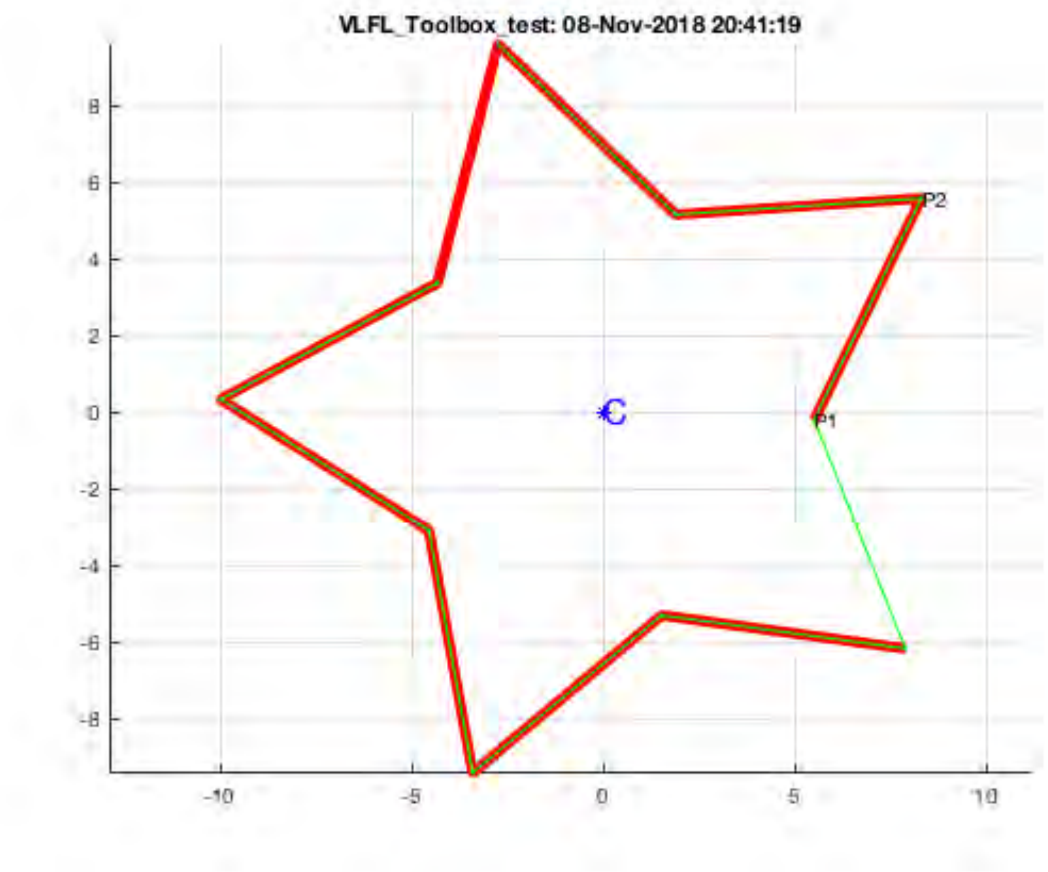
Find the maximum angle value of a star

```
SGfigure; view(0,90);
CPLsortC(PLstar(10,10),'max'); axis on;
```



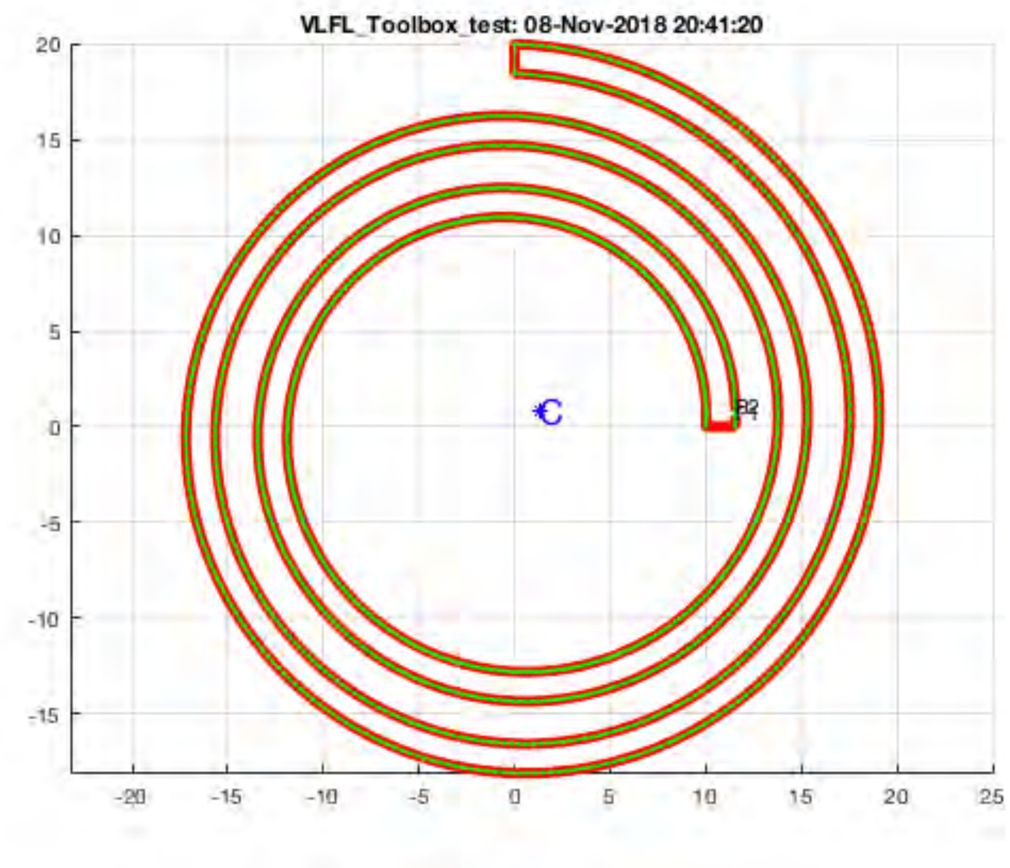
Find the angle value nearest to zero of a star

```
SGfigure; view(0,90);
CPLsortC(PLtrans(PLstar(10,10),rotdeg(160)),'zero'); axis on;
```



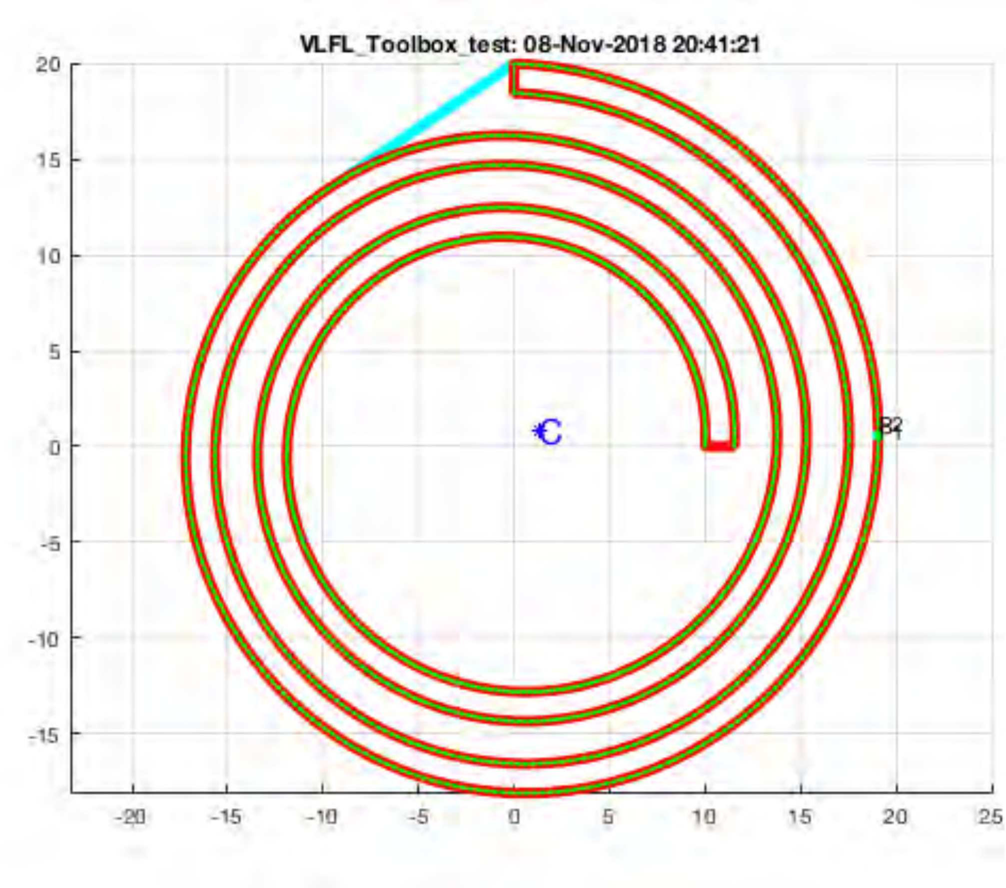
Find the angle value nearest to zero of a spiral

```
SGfigure; view(0,90);
CPLsortC(CPLspiral(10,20,4*pi+pi/2),'zero');
```



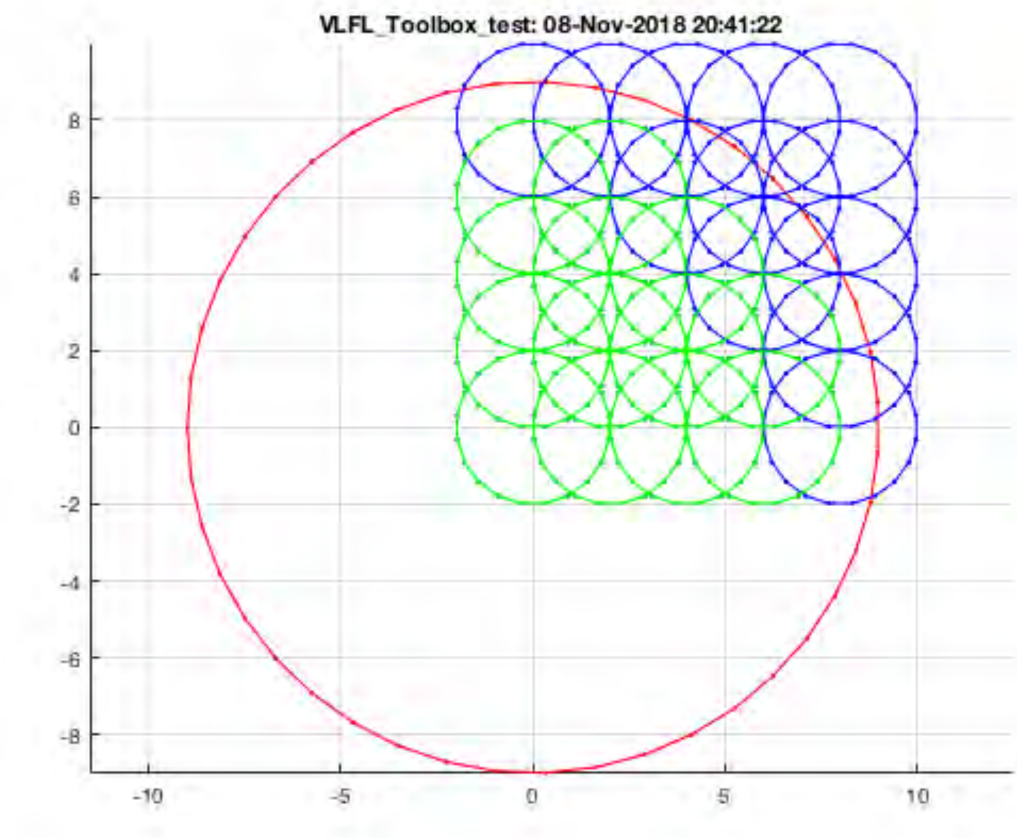
Find the angle value nearest to zero of convex hull of a spiral

```
SGfigure; view(0,90);
CPLsortC(CPLspiral(10,20,4*pi+pi/2),'czero');
```



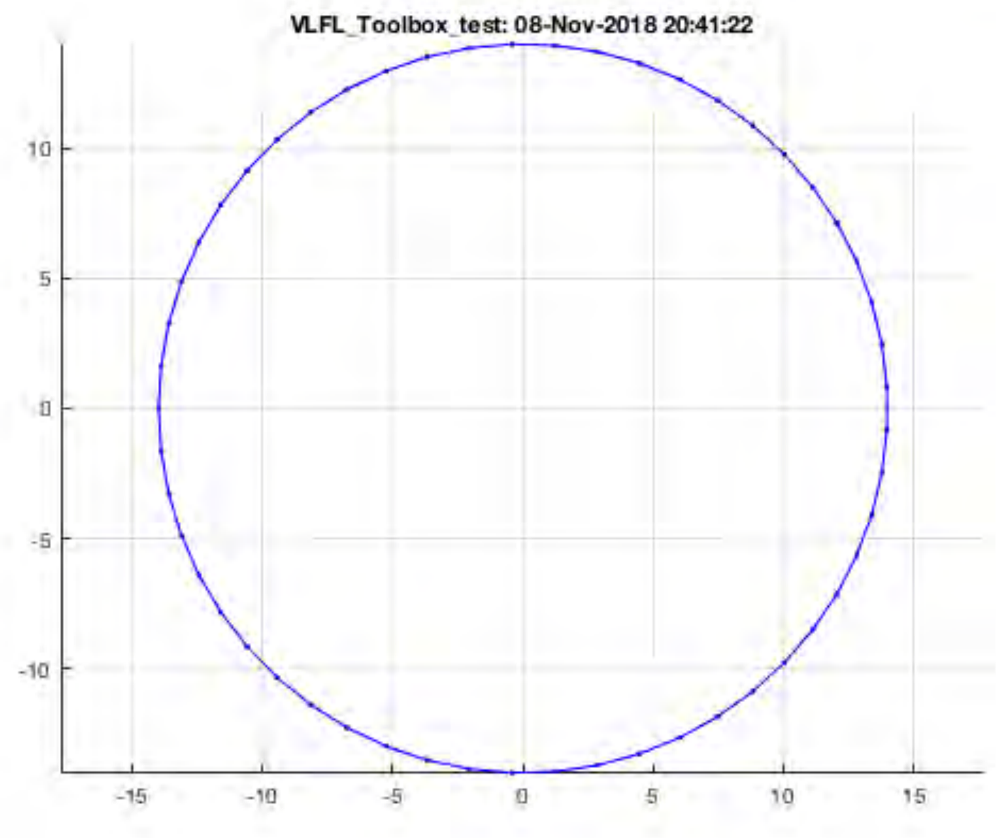
9. Informations on contours inside of others

```
SGfigure; view(0,90);
CPLinsideCPL(PLcircle(9),CPLcopypattern(PLcircle(2),[5 5],[2 2])); axis on;
```



Identical contours are not inside each other

```
CPLinsideCPL(PLcircle(14),CPLsortC(PLcircle(14)));
```



10. Order contours for the sequential plot with a laser cutter

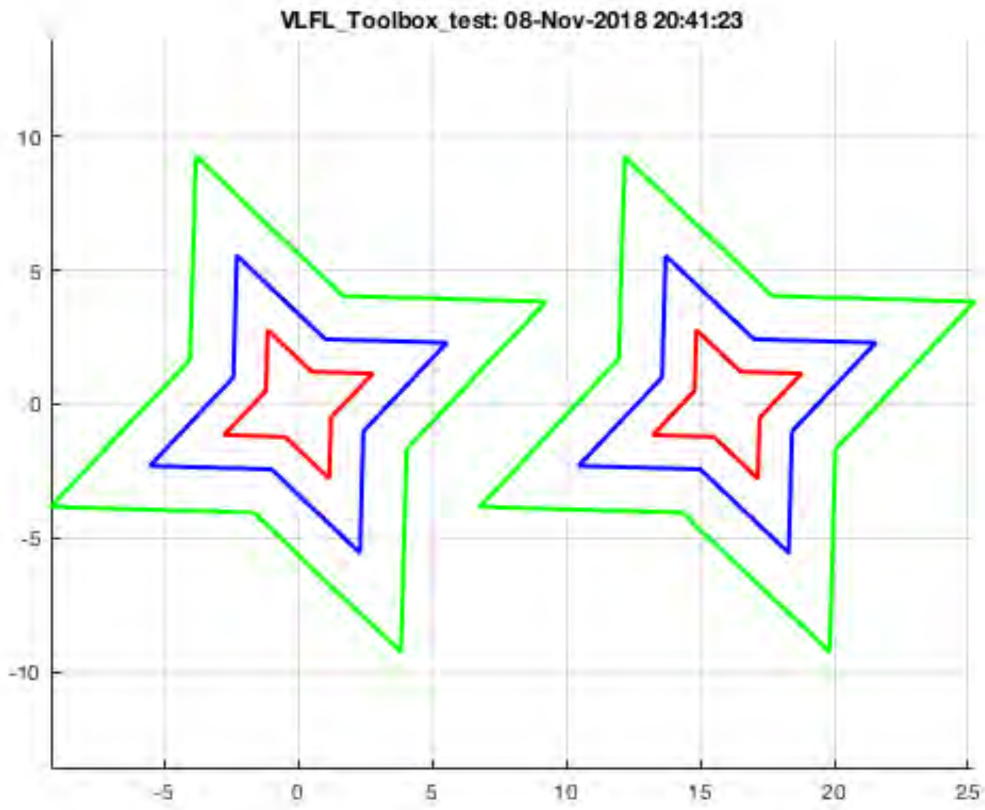
The "level" starts with zero runs from outer to inner. In case of a laser cutter it is necessary to cut the inner contours first.

- ***CPLwriteSVG*** - writes a CPL ans SVG on disk
- ***svgpolylineofCPL*** - plots an SVG file
- ***separateNaN*** - separates CPLs and CVLs
- ***selectNaN*** - creates a new set of selectex CPLs/CVLs
- ***CPLsortinout*** - sorts contours to inner and outer

```
SGfigure; view(0,90);
[ci,CC]=CPLsortinout(CPLsample(14))
CPLsortinout(CPLsample(14));
```

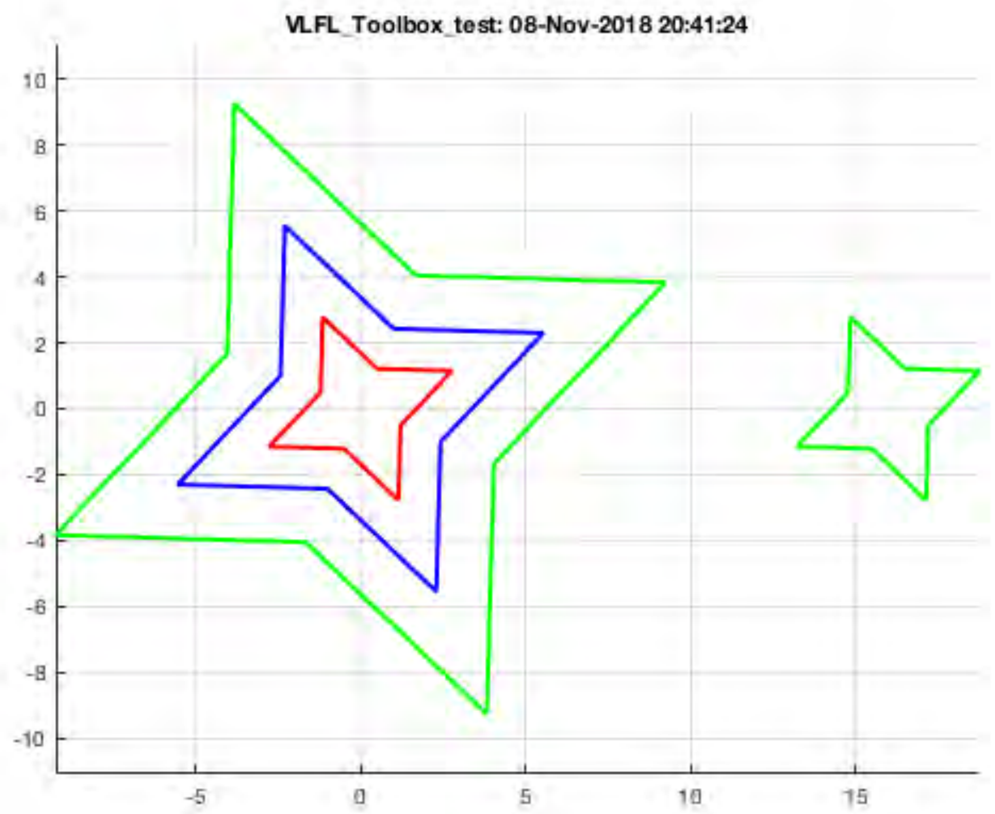
```
ci =

     0
     1
     2
     0
     1
     2


CC =
```

```
   NaN      1      1     -1     -1     -1
    -1    NaN      1     -1     -1     -1
    -1     -1    NaN     -1     -1     -1
    -1     -1     -1    NaN      1      1
    -1     -1     -1     -1    NaN      1
    -1     -1     -1     -1     -1    NaN
```



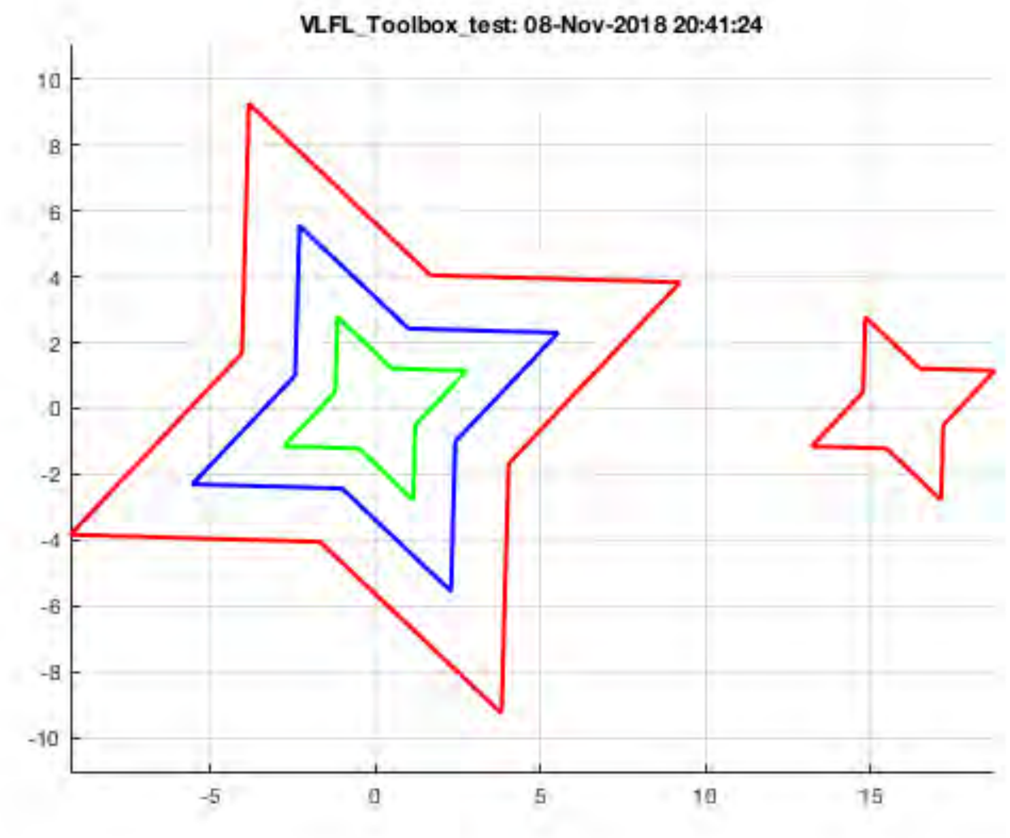
Show a selction from inner to outer for laser cutting

```
CPLsortinout(selectNaN(CPLsample(14),[1,2,3,6]));
```



Now change the order direction from outer to inner

```
CPLsortinout(selectNaN(CPLsample(14),[1,2,3,6]),false);
```



Now write is als a cutter file

```
CPLwriteSVG(CPLsample(14),'VLFL_EXP14_cutter','',true);
```

```
WRITING SVG FILE /Users/lueth/Desktop/Toolbox_test/VLFL_EXP14_cutter.SVG in ASCII MODE
completed.
```

Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:41:25!
Executed 08-Nov-2018 20:41:27 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
simulink
video_and_image_blockset
```

==
==========

- *Tim Lueth, tested and compiled on OSX 10.7.5 with Matlab 2014b on 2015-09-20*
- __________, executed and published on 64 Bit PC using Windows with Matlab 2015a on 2015-xx-xx_

Published with MATLAB® R2018a

Tutorial 15: Create a Solid by 2 Closed Polygons

2015-10-03: 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

Motivation for this tutorial: (Originally SolidGeometry 2.7 required)

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).

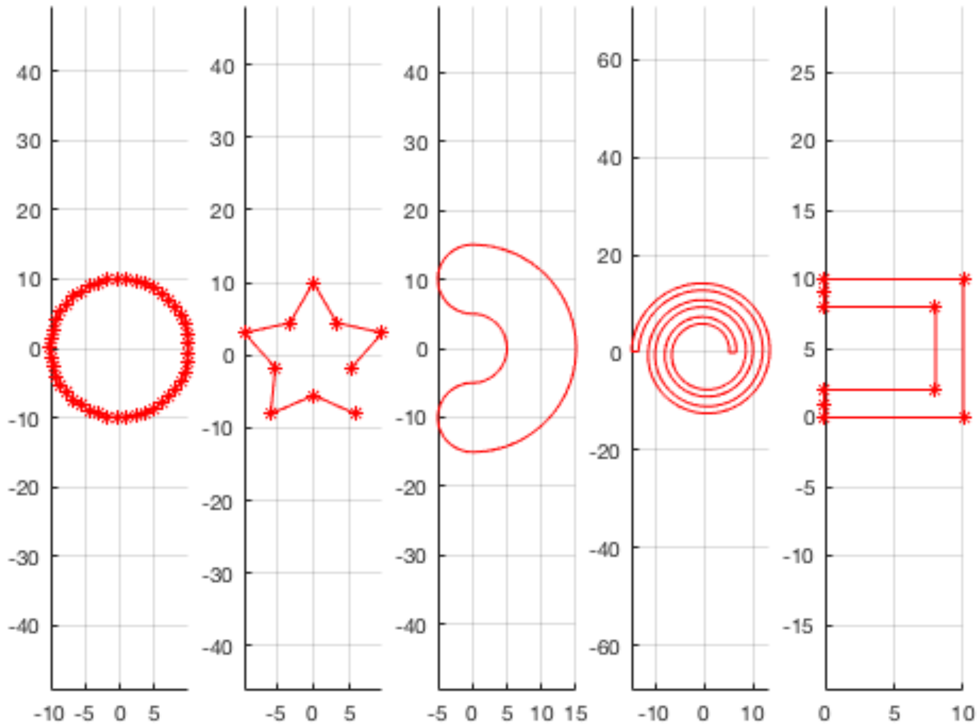
Even **some concave shaped polygons**, such as stars, **have at least monotonic increasing angle sum** and can be connected by this strategy.

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;
```

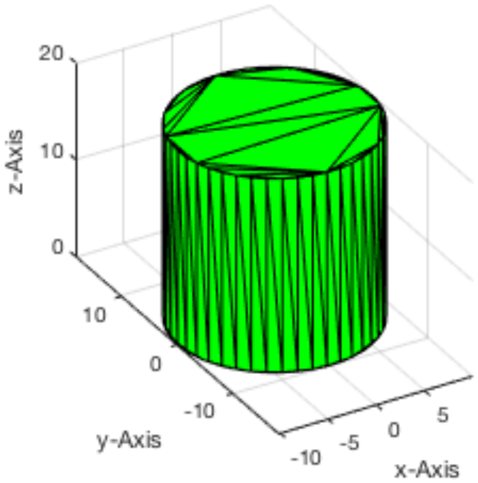


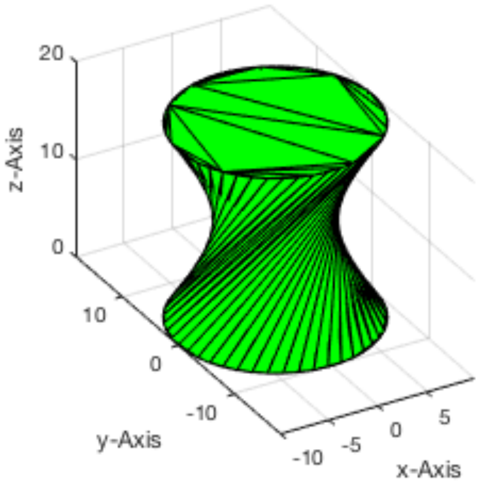
3. Solid surface generation and the importance of start point of the contour (turning)

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

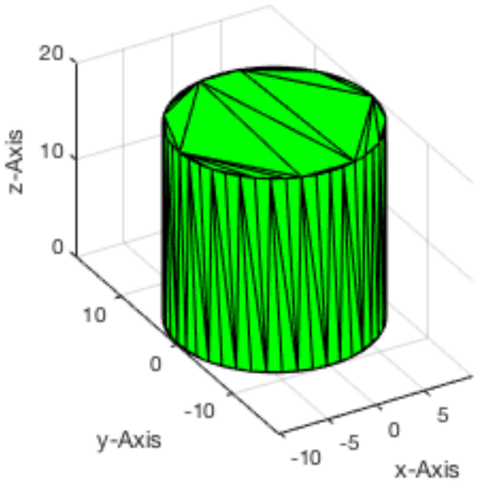
```
SGfigure; view(-30,30);
subplot(1,2,1);
SG=SGof2CPLz(PLcircle(10),PLcircle(10),20); SGplot(SG,'g'); view(-30,30);
subplot(1,2,2);
SG=SGof2CPLz(PLcircle(10),PLtransR(PLcircle(10),rotdeg(120)),20,[],'none');
SGplot(SG,'g'); view(-30,30);
```

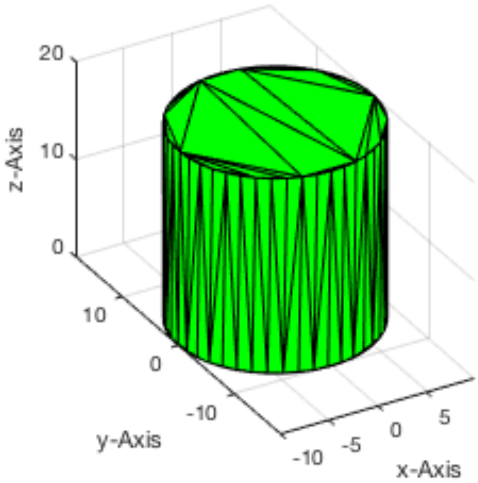




One input paramter of SGof2CPLz allows to select the turning angle adjustment to 'none', 'rot', or 'miny'. Please read the documentation of 'czero'=rot (minimal angle near zero of the convex hull) of CPLsortC and PLminyx (Point with the minimal y and minmal x value) to understand 'miny'. In addition 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

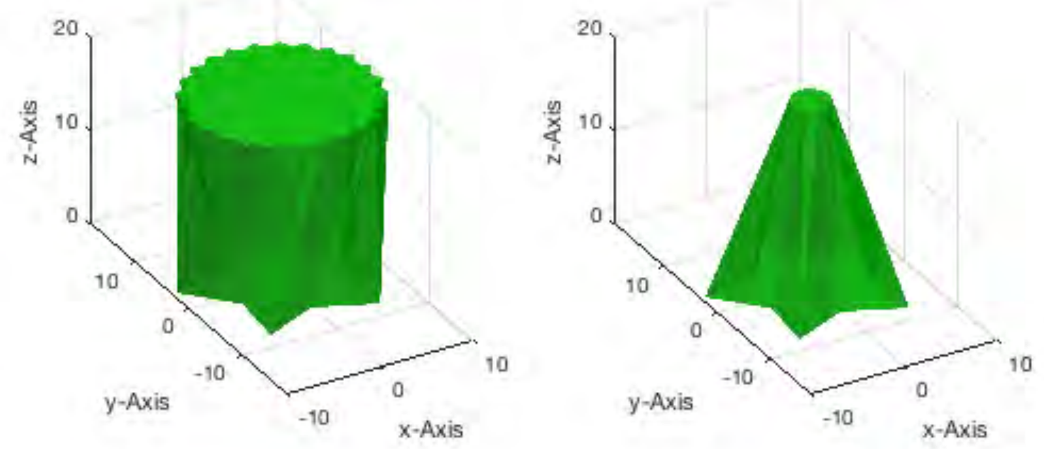
Currently, there are four strategies for the contour point assignment during contour connections: SGof2CPLZ(PLA,PLB,z,assignment,turning);

- **'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);
```

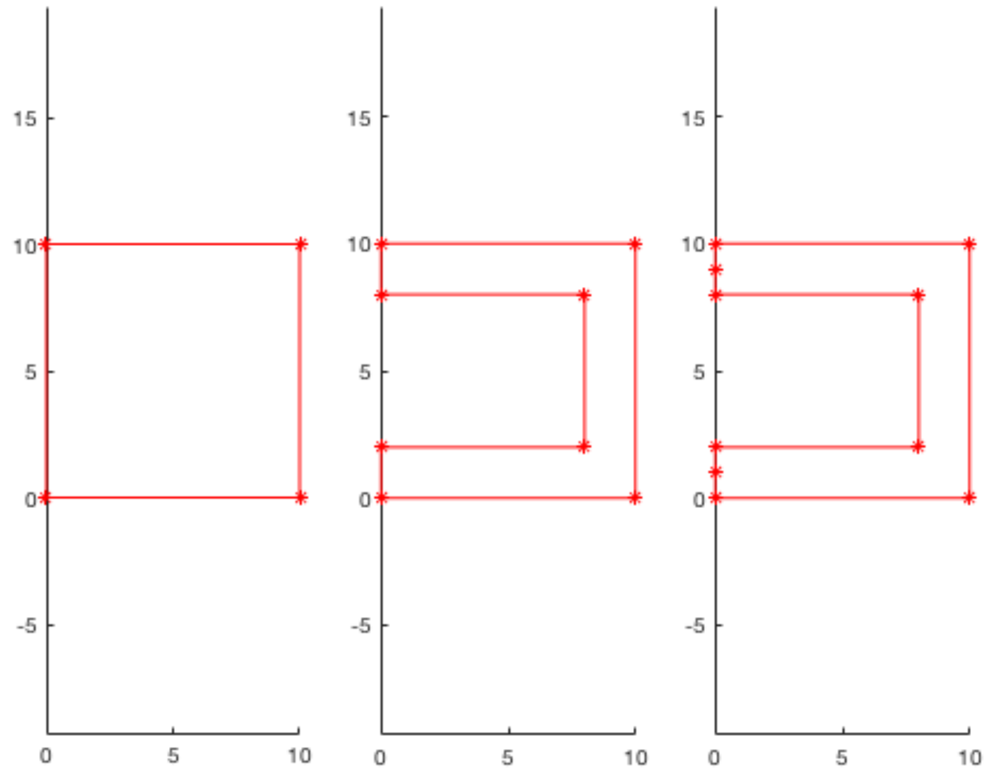
```
VLFLplotlight(1,0.7);
```



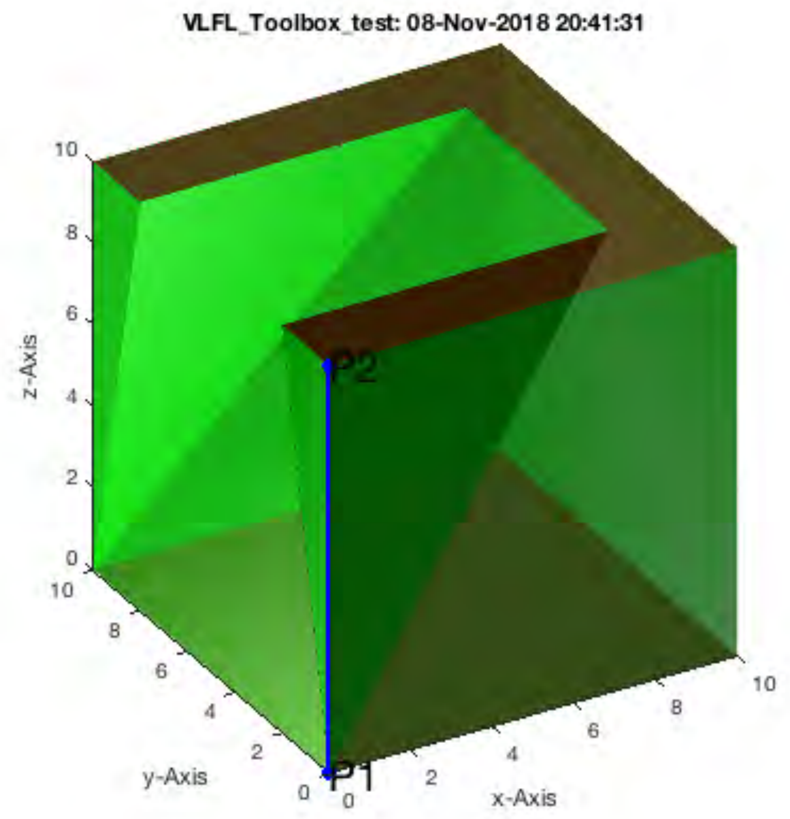
5. Solid surface generation for polygons with a small number of points

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

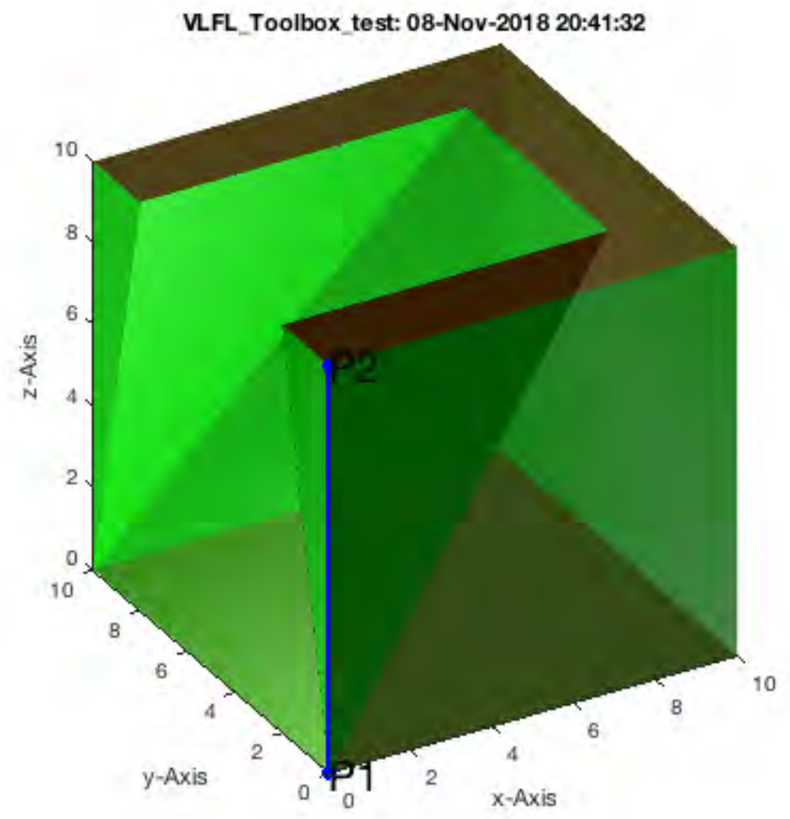
```
SGfigure;
PLU0=[0 0;10 0; 10 10; 0 10; 0 0];
PLU1=[0 0;10 0; 10 10; 0 10; 0 8; 8 8; 8 2; 0 2; 0 0];
PLU2=[0 0;10 0; 10 10; 0 10; 0 9; 0 8; 8 8; 8 2; 0 2; 0 1; 0 0];
subplot(1,3,1); CPLplot(PLU0);  view(0,90); axis equal;
subplot(1,3,2); CPLplot(PLU1);  view(0,90); axis equal
subplot(1,3,3); CPLplot(PLU2);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



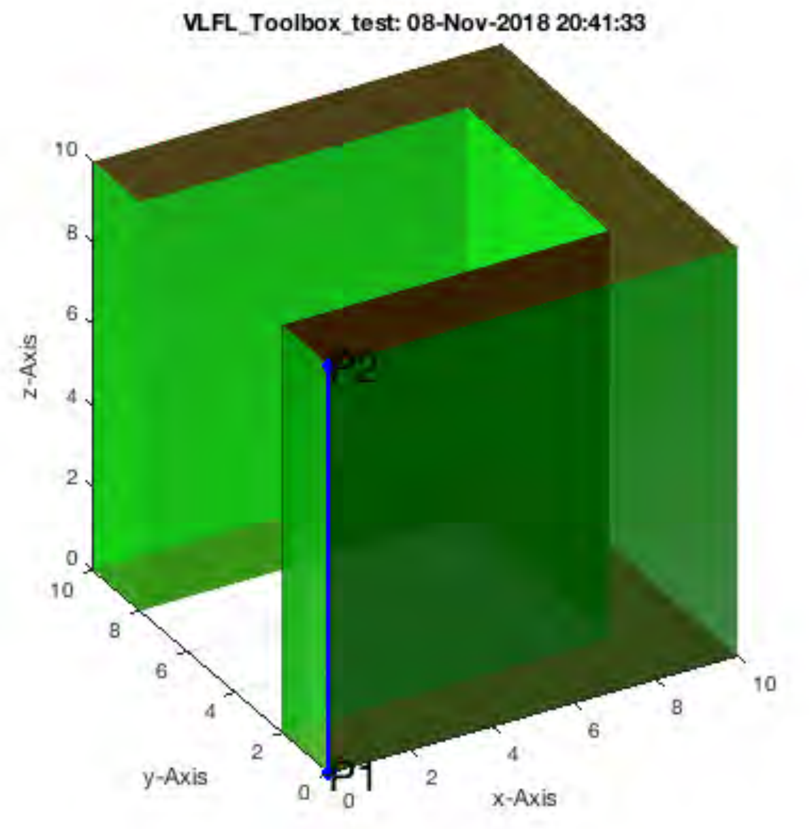

```
SGof2CPLz(PLU0,PLU2,10); VLFLplotlight(1,0.7);
```




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



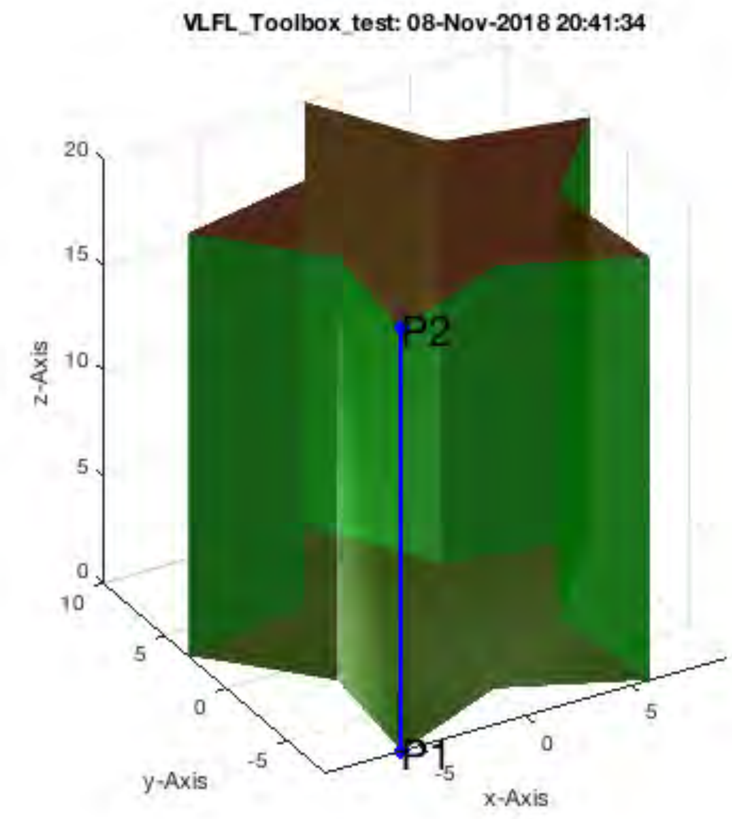

```
SGof2CPLz(PLU1,PLU2,10); VLFLplotlight(1,0.7);
```



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.

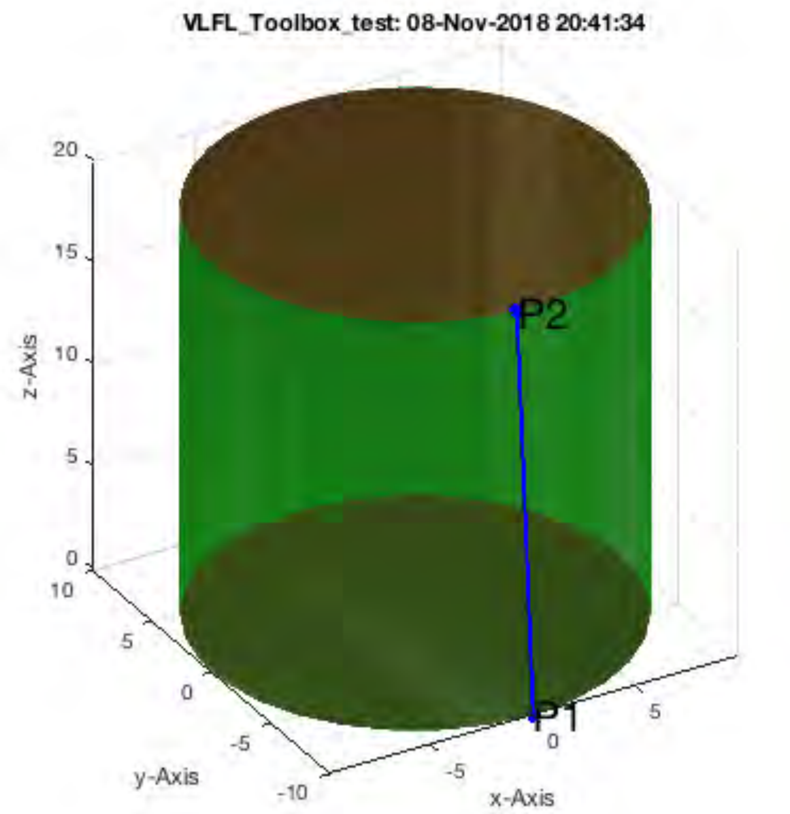
```
SGof2CPLz(PLstar(10,10),PLstar(10,10),20); VLFLplotlight(1,0.7);
```



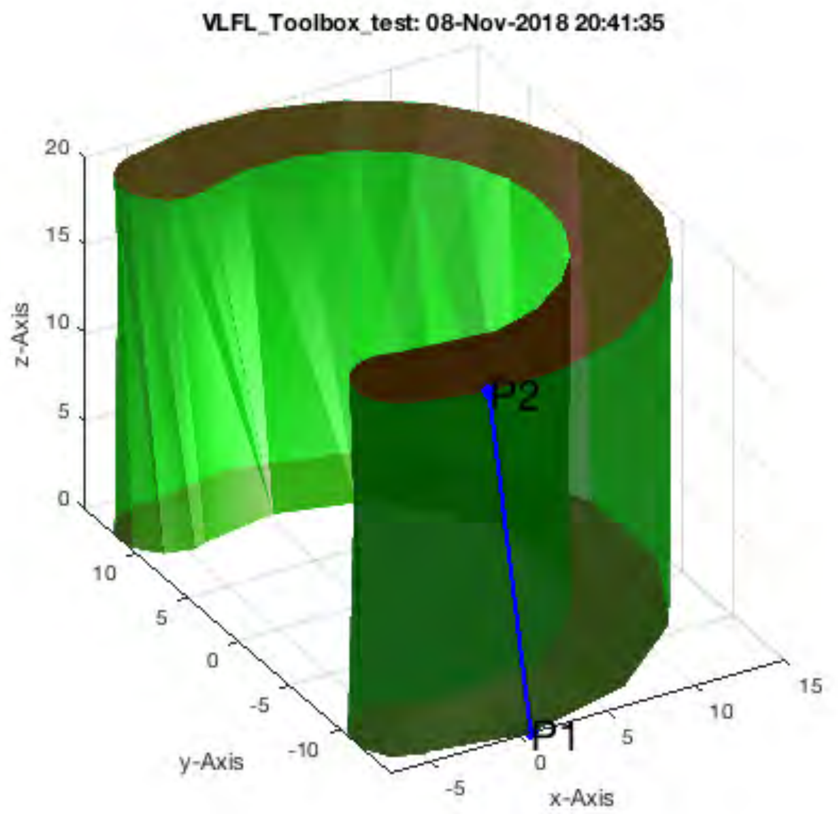
7. Two contours of the same shape with different number of points

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);
```



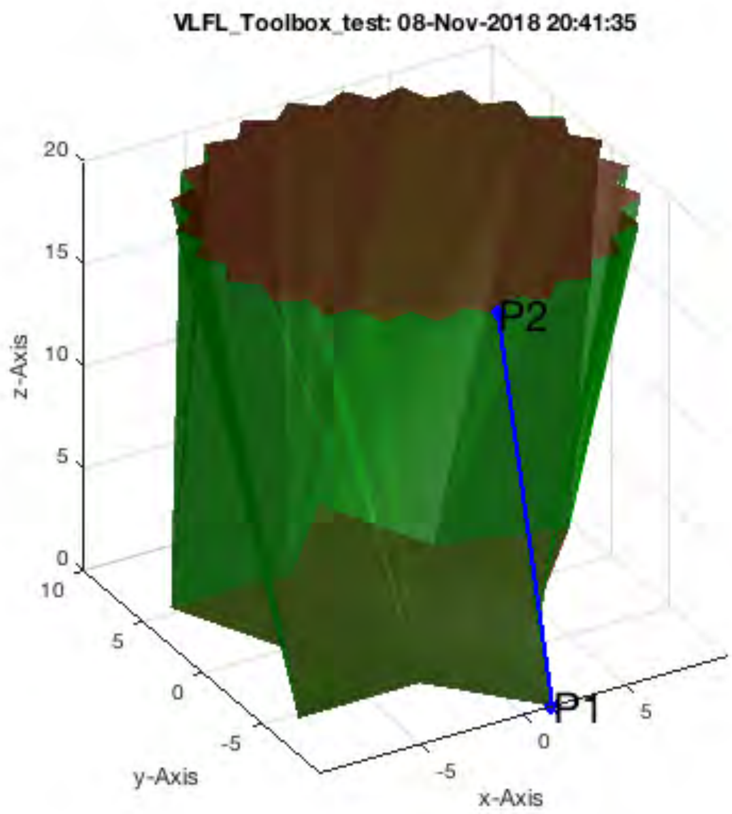

```
SGof2CPLz(PLkidney(10,15,pi/0.8,10),PLkidney(10,15,pi/0.8,15),20);
VLFLplotlight(1,0.7);
```



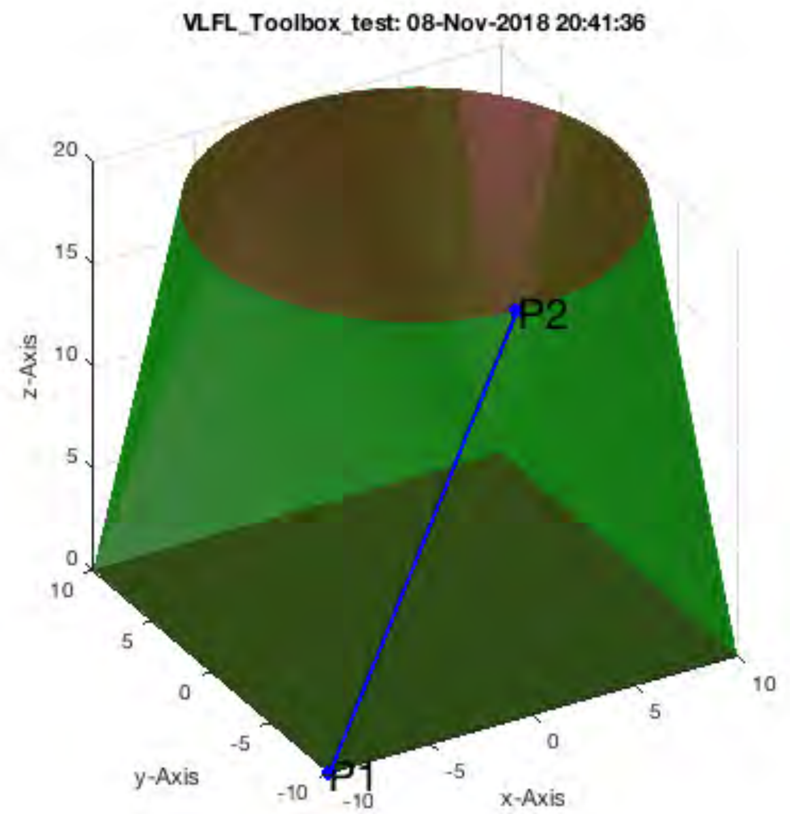
8. Two contours of the similar shape with different number of points

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.

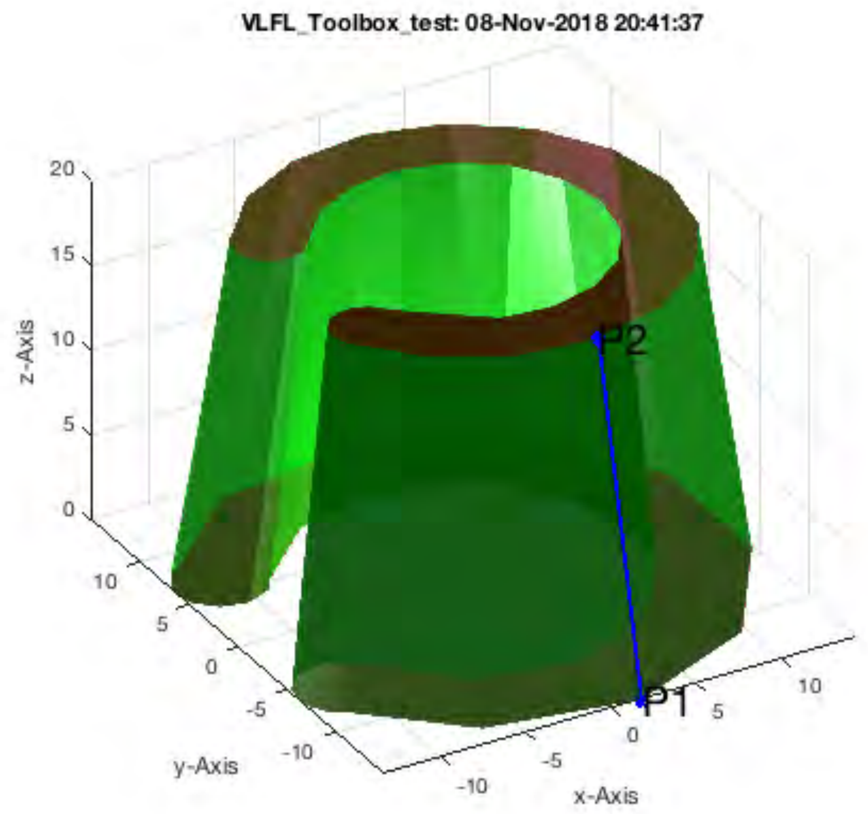
```
SGof2CPLz(PLstar(10,11),PLstar(10,50),20); VLFLplotlight(1,0.7);
```




```
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);
```

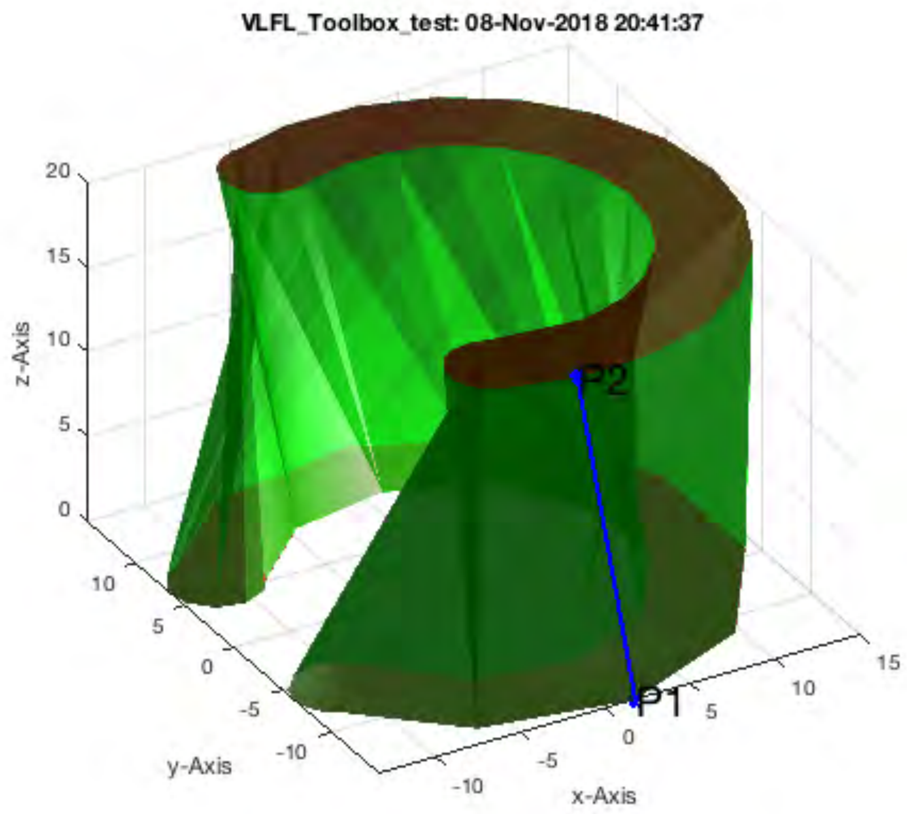


In this case we see that the kidney has different size but the same bending angle of pi/.6.

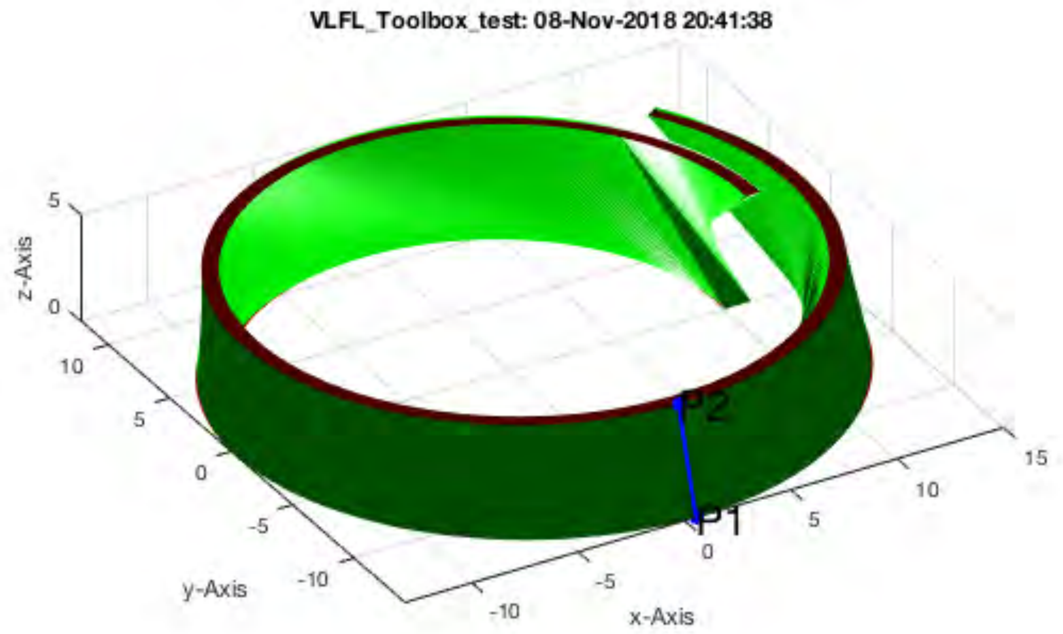
9. Two contours of the similar shape with different sum of boundary bending angle sum

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);
```

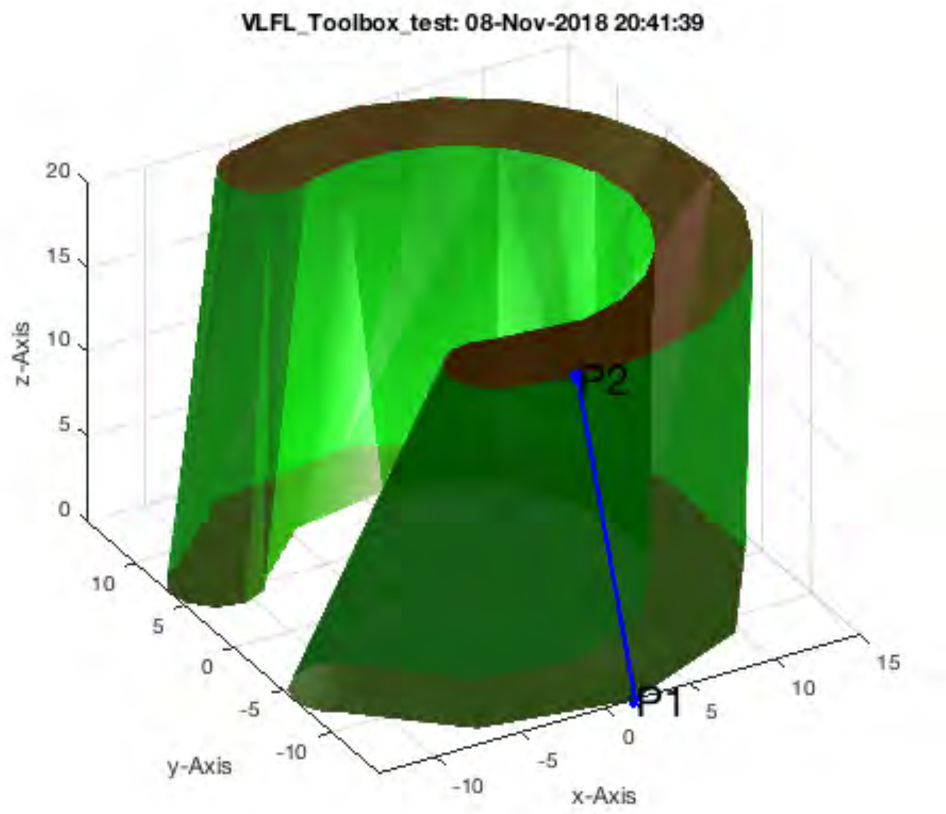



```
SGof2CPLz(CPLspiral(10,15,2*pi),CPLspiral(11,14,2.2*pi),5);
VLFLplotlight (1,1);
```

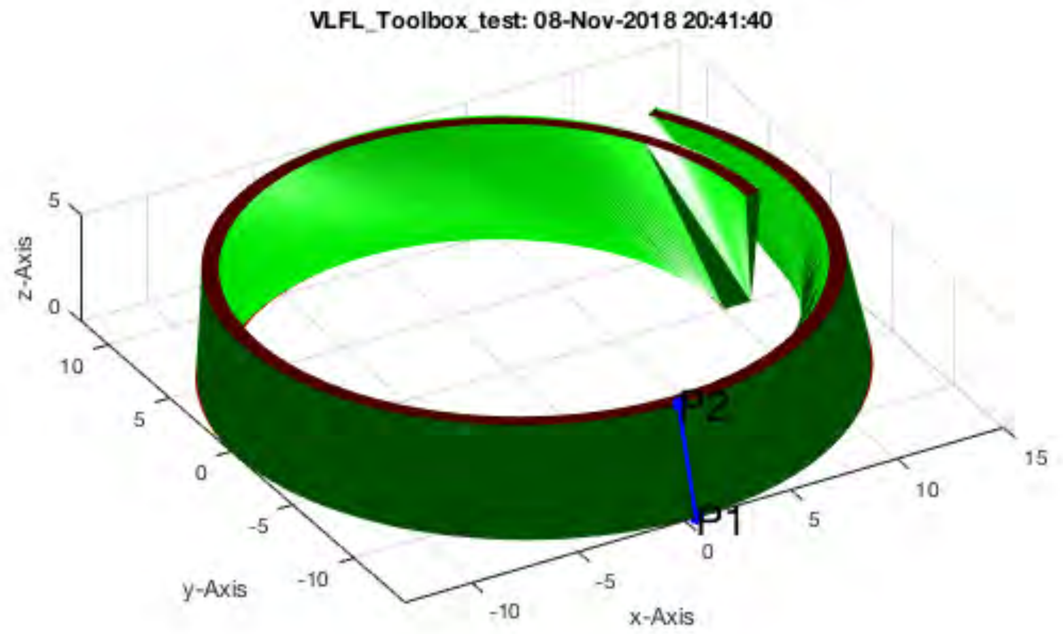


In this case we see malformed solids.

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

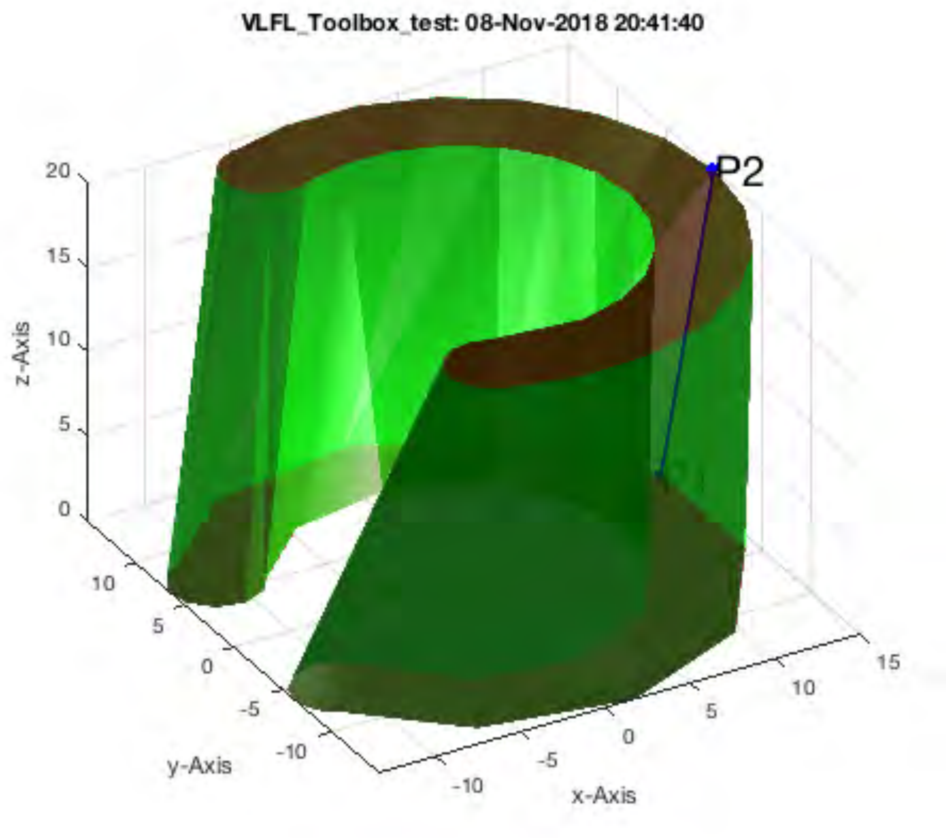



```
SGof2CPLz(CPLspiral(10,15,2*pi),CPLspiral(11,14,2.2*pi),5,'angle');
VLFLplotlight (1,1);
```

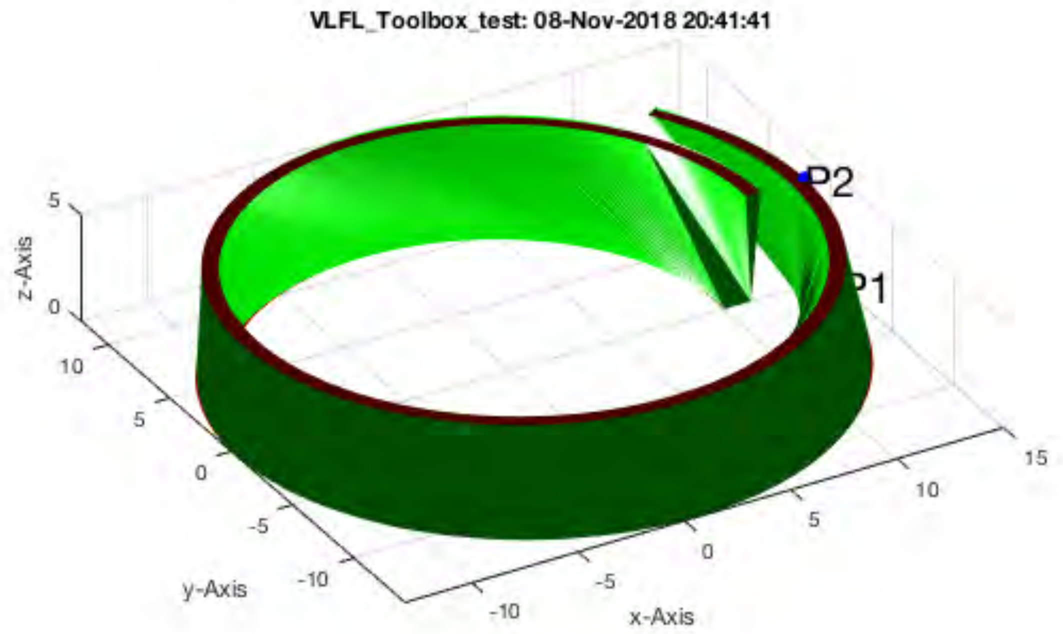


By using 'angle' instead of 'length', the solids are more what we expected.

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

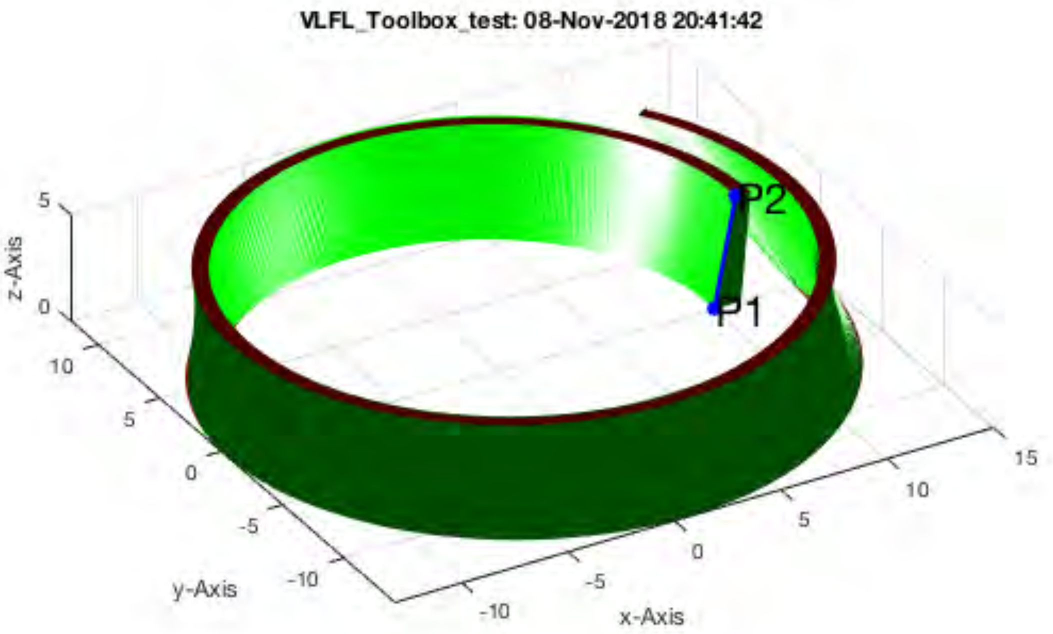



```
SGof2CPLz(CPLspiral(10,15,2*pi),CPLspiral(11,14,2.2*pi),5,'angle','rot');
VLFLplotlight (1,1);
```



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.  (2018-Nov-08), 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 11-Aug-2073 20:41:42!
Executed 08-Nov-2018 20:41:44 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
simulink
video_and_image_blockset
===========================================================================================
==========
```

- *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® R2018a

Tutorial 16: Create Tube-Style Solids by Succeeding Polygons

2015-10-04: 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

Motivation for this tutorial: (Originally SolidGeometry 2.7 required)

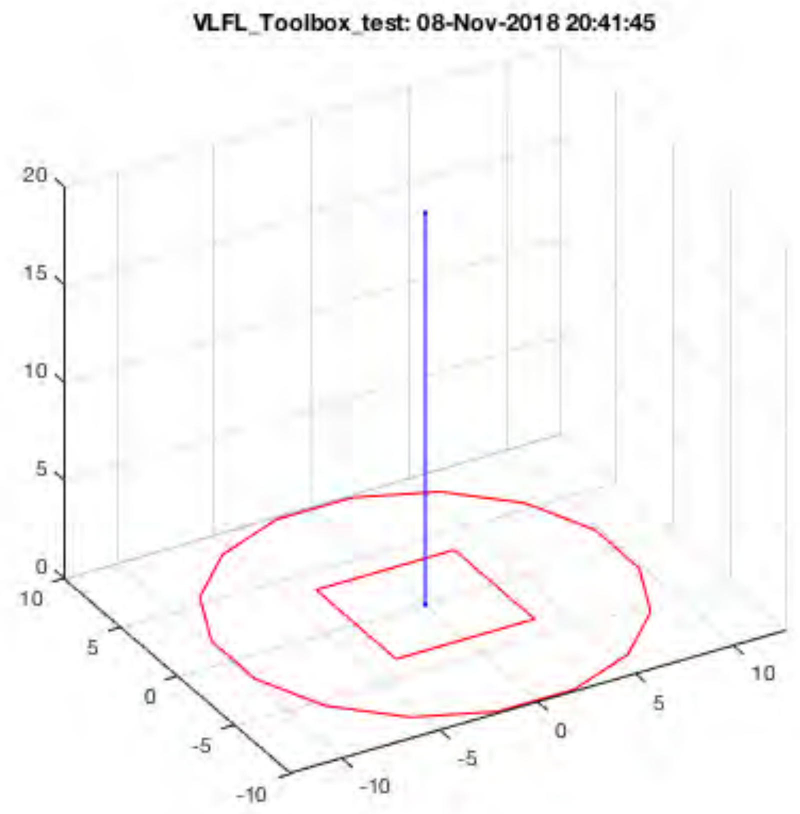
2. Tube generation by repeating identical CPL along a 3D path

For robotics design,very often we have a wish to extrude a CPL not only in an orthognal z direction to the xy-plane, but in an any desired direction even along a path in 3D space. Intutively we expect a result, but this is not easy to achieve automatically. Anyway, for those tasks we have two functions:

- SGcontourtube - repeats a CPL along a path in 3D
- FLofCVL

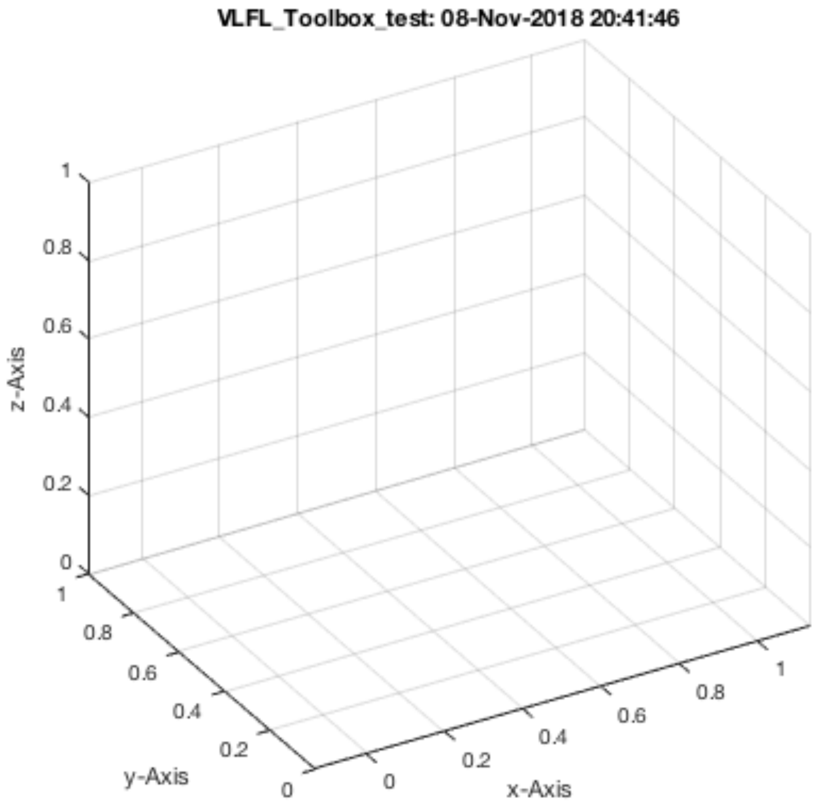
Let us start with a planar contour in the x/y-plane, and an orthogonal path

```
SGfigure; axis on ; grid on;
CPL=CPLsample(8);
CPLplot(CPL,'r-');

VL=[0 0 0; 0 0 20];
VLplot(VL,'b.-'); view(-30,30);
```



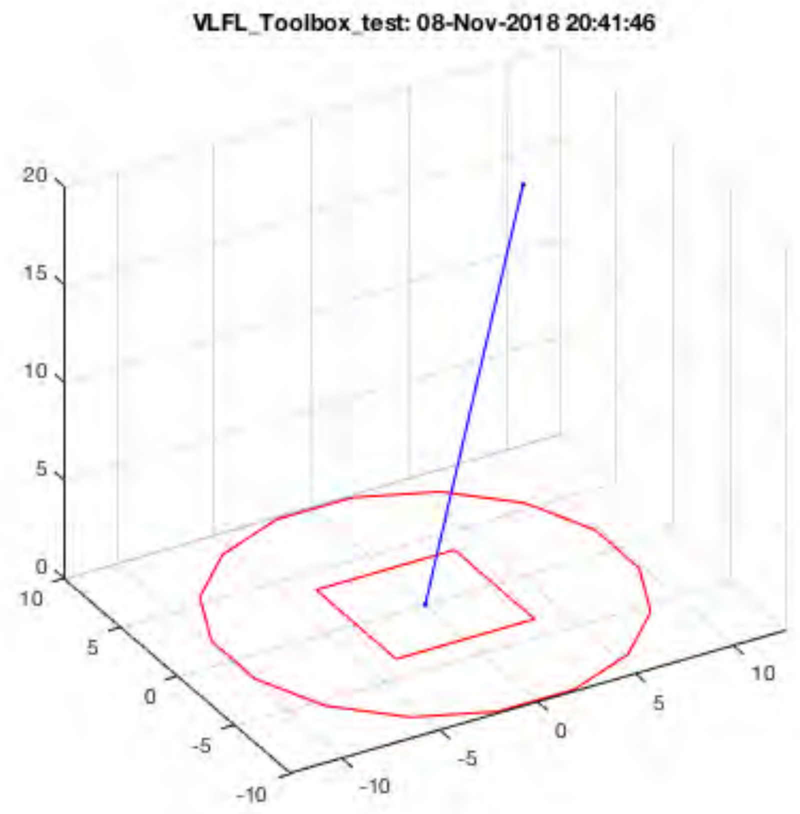
The final result looks as expected

```
SG=SGcontourtube(CPL,VL); SGfigure(SG); VLFLplotlight(1,1);view (-30,30);
```



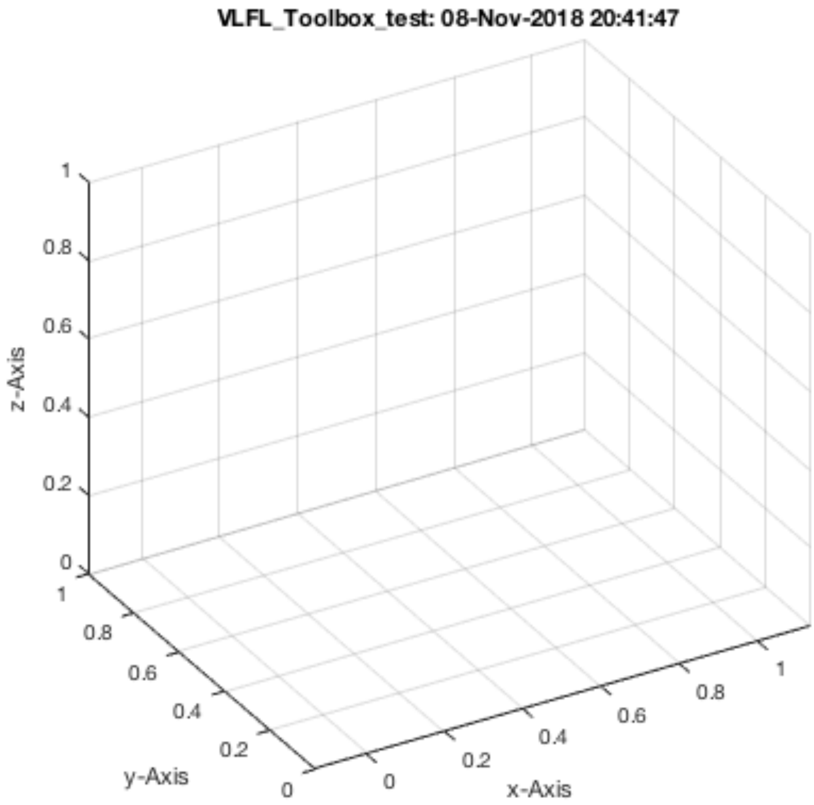
Now we change the path a little

```
SGfigure;  axis on ; grid on;
CPLplot(CPL,'r-');
VL=[0 0 0; 5 0 20];
VLplot(VL,'b.-'); view(-30,30);
```



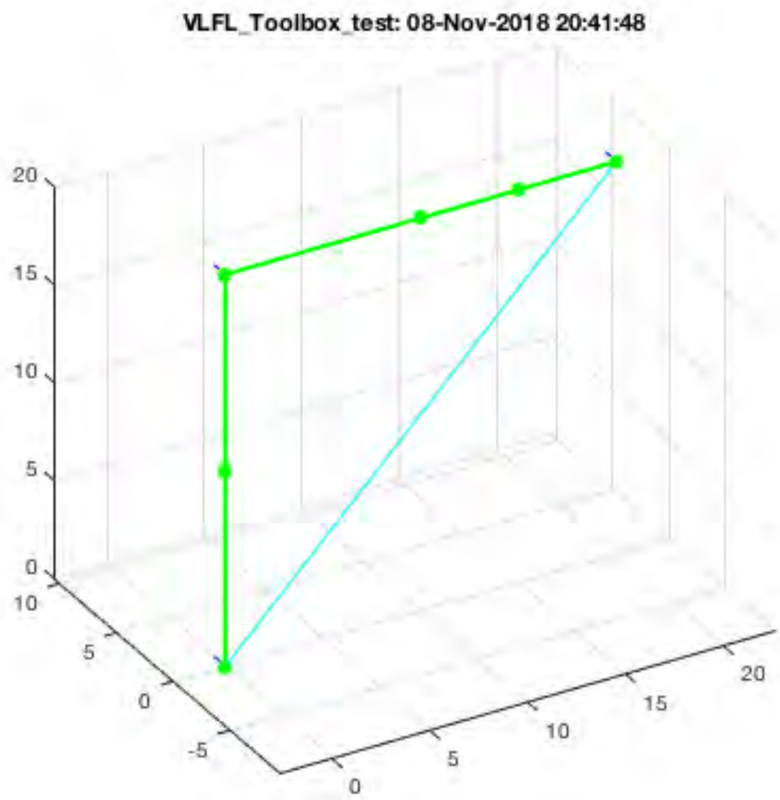
The final result may not look as expected

```
SG=SGcontourtube(CPL,VL);
SGfigure; axis on ; grid on;
SGplot(SG,'m');
VLFLplotlight(1,1);view (-30,30);
```



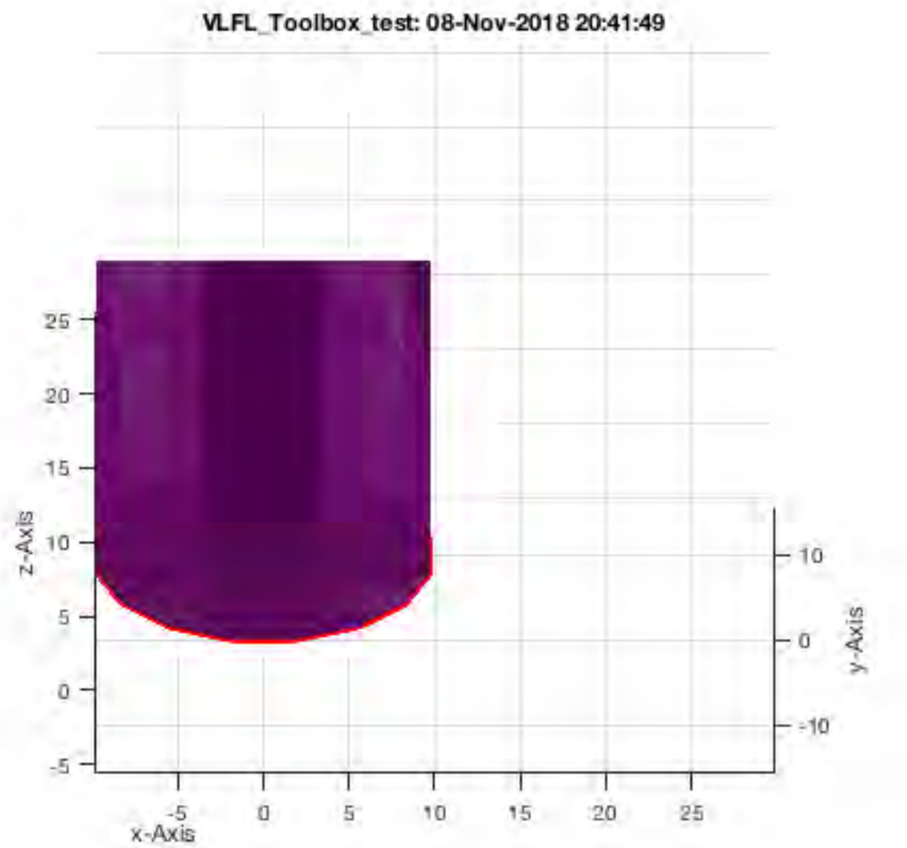
Now we change the path, still a planar one and anylyze the angle situation The cynan colored path explain that the vertex list has to be closed to analyze the first an last angle. Furthermore we see that more than one point has no defined orthogonal vector since it sits on a straight line

```
SGfigure; CPLplot(CPL,'r-');  axis on ; grid on;
VL=[0 0 0; 0 0 10; 0 0 20; 10 0 20; 15 0 20; 20 0 20];
VLangle(VL);
```



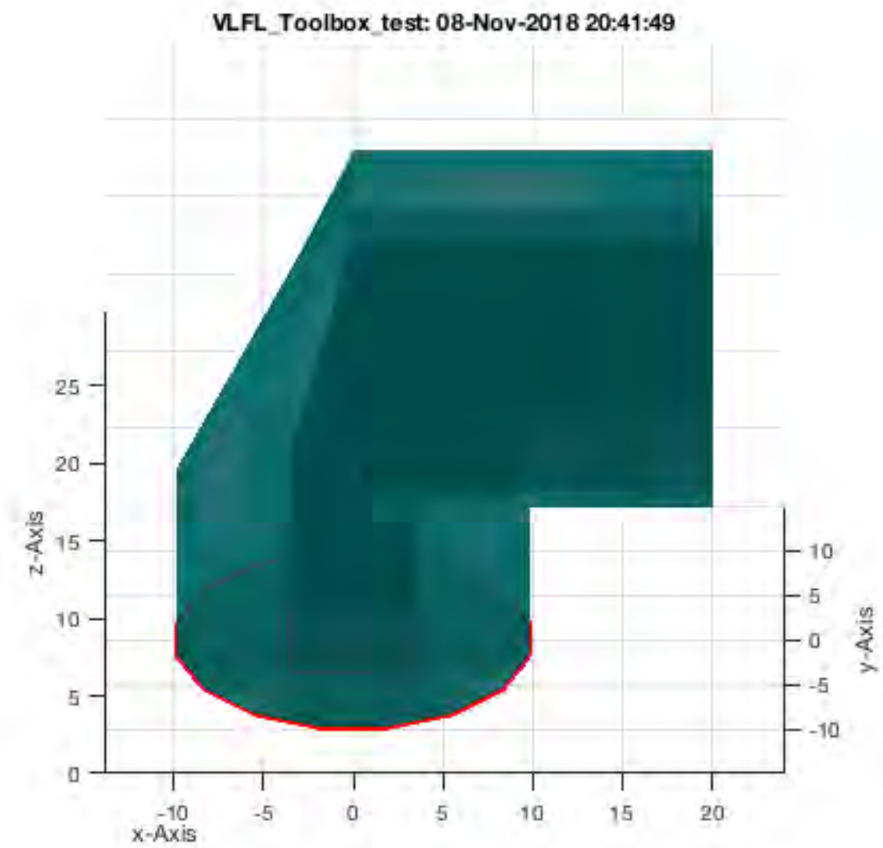
The result is not may be we expected

```
SGfigure; CPLplot(CPL,'r-',2);  axis on ; grid on;
SG=SGcontourtube(CPL,VL); SGplot(SG,'m'); VLFLplotlight(1,0.8);view (0,30);
```

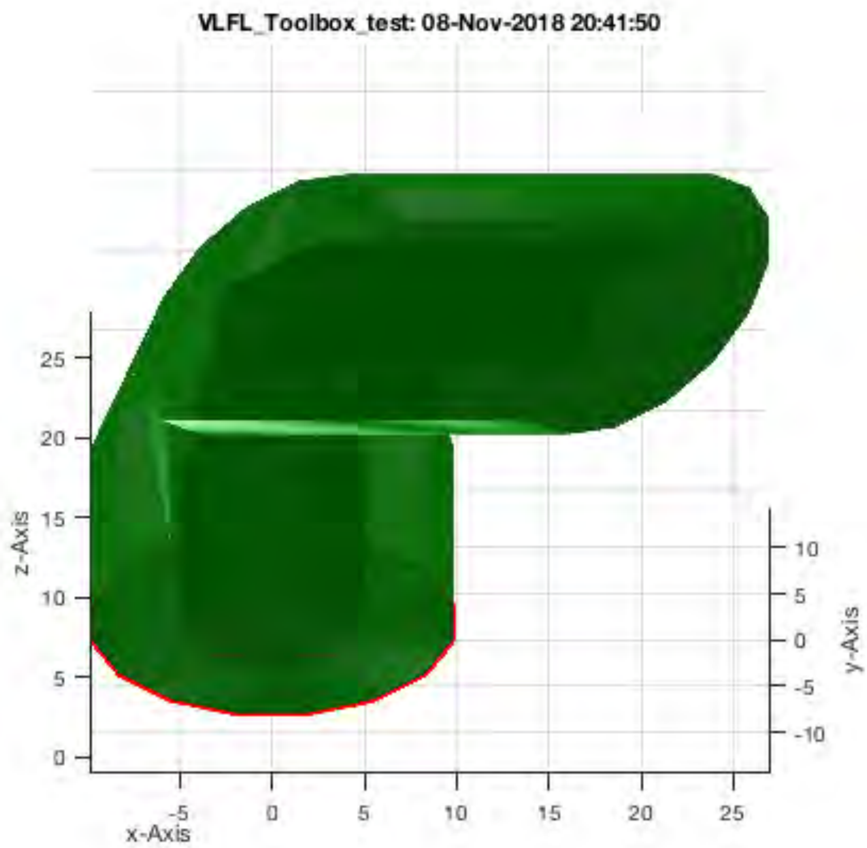


The result can be adjusted by defining the ex-vector at the start point

```
SGfigure; CPLplot(CPL,'r-',2);  axis on ; grid on;
SG=SGcontourtube(CPL,VL,[0 1 0]); SGplot(SG,'c'); VLFLplotlight(1,0.8);view (0,30);
```




```
SGfigure; CPLplot(CPL,'r-',2);  axis on ; grid on;
SG=SGcontourtube(CPL,VL,[1 1 0]); SGplot(SG,'g'); VLFLplotlight(1,0.8);view (0,30);
```



One lazy approach is delivered by SGofCPLCVLR without a given radius

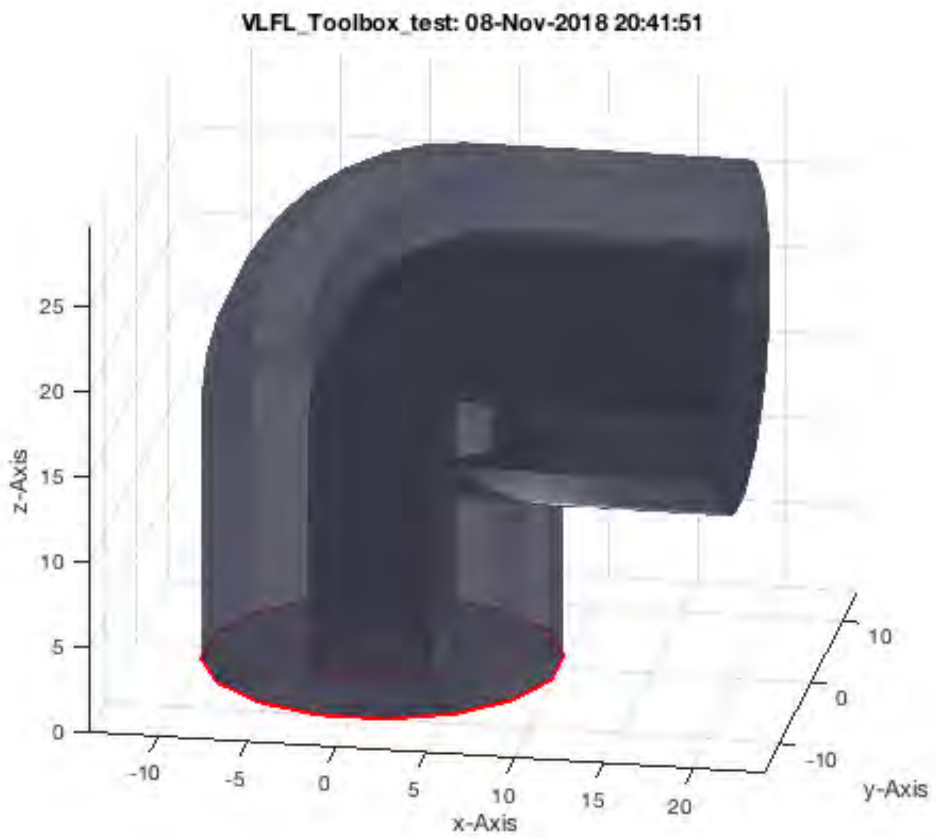
```
SGfigure; CPLplot(CPL,'r-',2);  axis on ; grid on;
SG=SGofCPLCVLR(CPL,VL); SGplot(SG,'y'); VLFLplotlight(1,0.8);view (10,20);
```

```
Warning: 1st Euler angle does not fit to vertex list path direction
Warning: Last Euler angle does not fit to vertex list path direction
```

One lazy approach is delivered by SGofCPLCVLR including a radius 5

```
SGfigure; CPLplot(CPL,'r-',2);  axis on ; grid on;
SG=SGofCPLCVLR(CPL,VL,5); SGplot(SG,'w'); VLFLplotlight(1,0.8);view (10,20);
```

```
VLradialEdges: Radius 5.00 reduced to 4.76
```

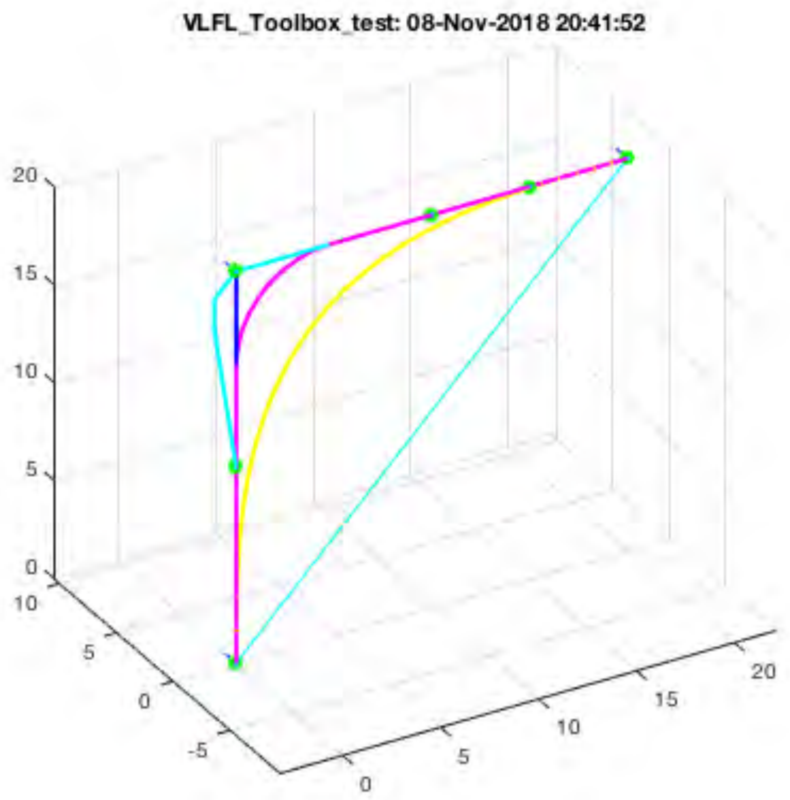


3. Tube generation using a 3D path with Bezier-curves or radial edges

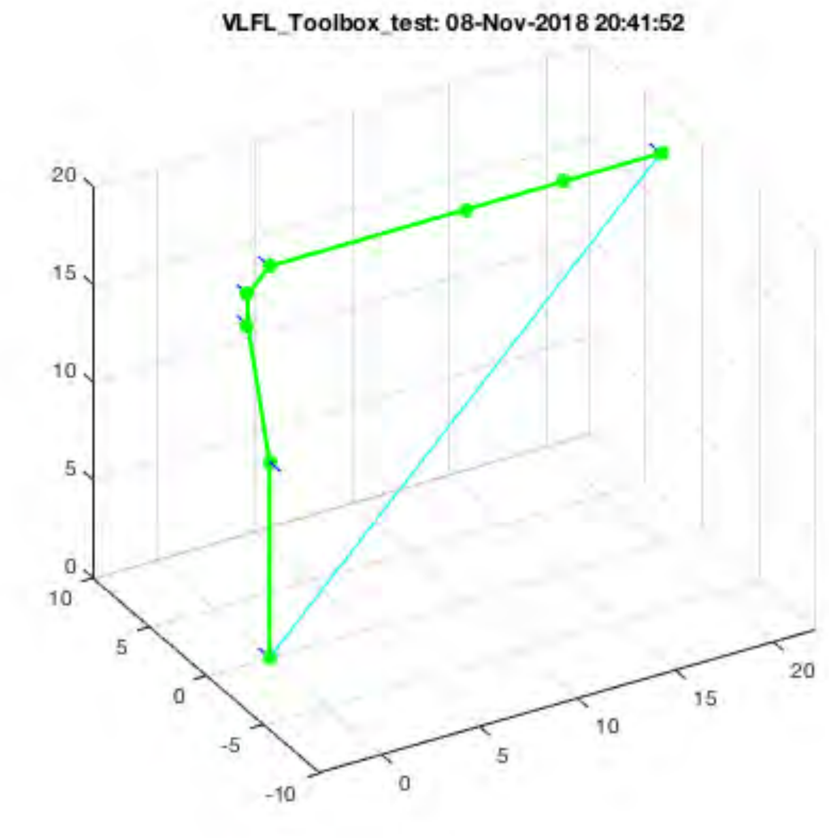
Create a 2D closed polygon line to by copied in 3D space

```
SGfigure; CPLplot(CPL,'r-');  axis on ; grid on;
VL=[0 0 0; 0 0 10; 0 0 20; 10 0 20; 15 0 20; 20 0 20];
VLangle(VL);

VLB=VLBezierC(VL,30);
VLR=VLRadiusC(VL,pi/4,2);
VLr=VLradialEdges(VL,5);
VLplot(VL,'b.-',2); view (-30,30);
VLplot(VLB,'y.-',2); view (-30,30);
VLplot(VLR,'c.-',2); view (-30,30);
VLplot(VLr,'m.-',2); view (-30,30);
```

```
VLradialEdges: Radius 5.00 reduced to 4.76
```

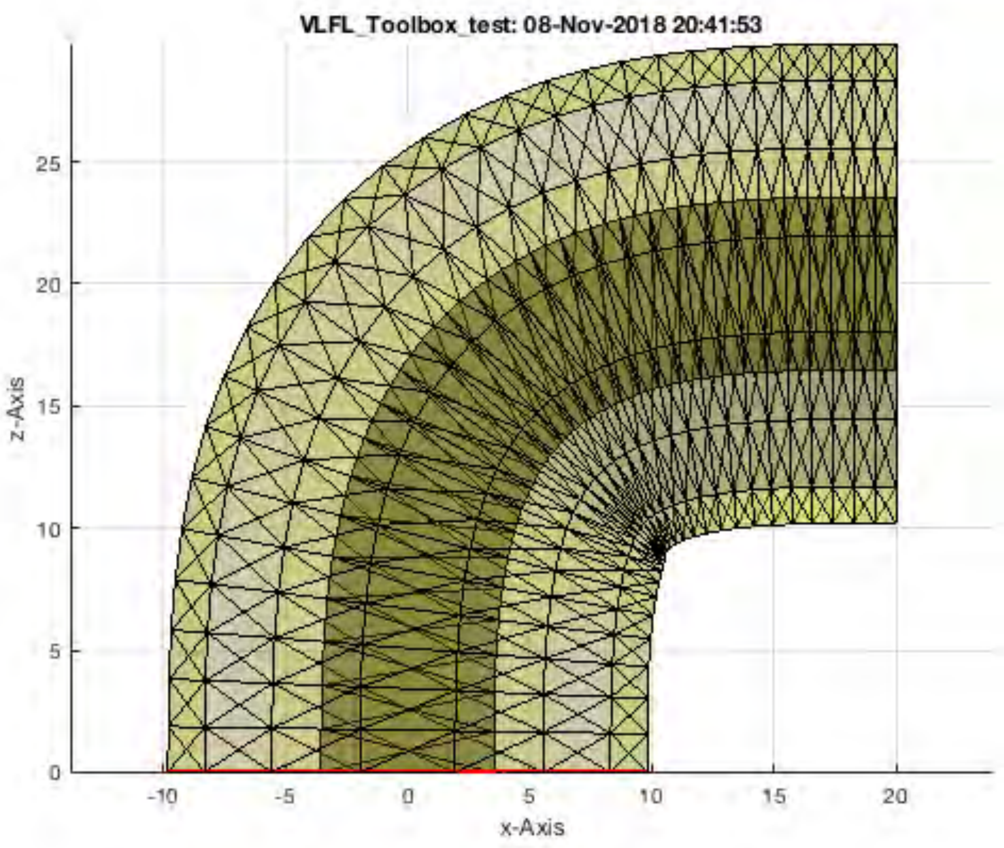

VLFL_Toolbox_test: 08-Nov-2018 20:41:52
20
15
10
5
0
10
5
0
-5
0
5
10
15
20

```
VLangle(VLR);
```



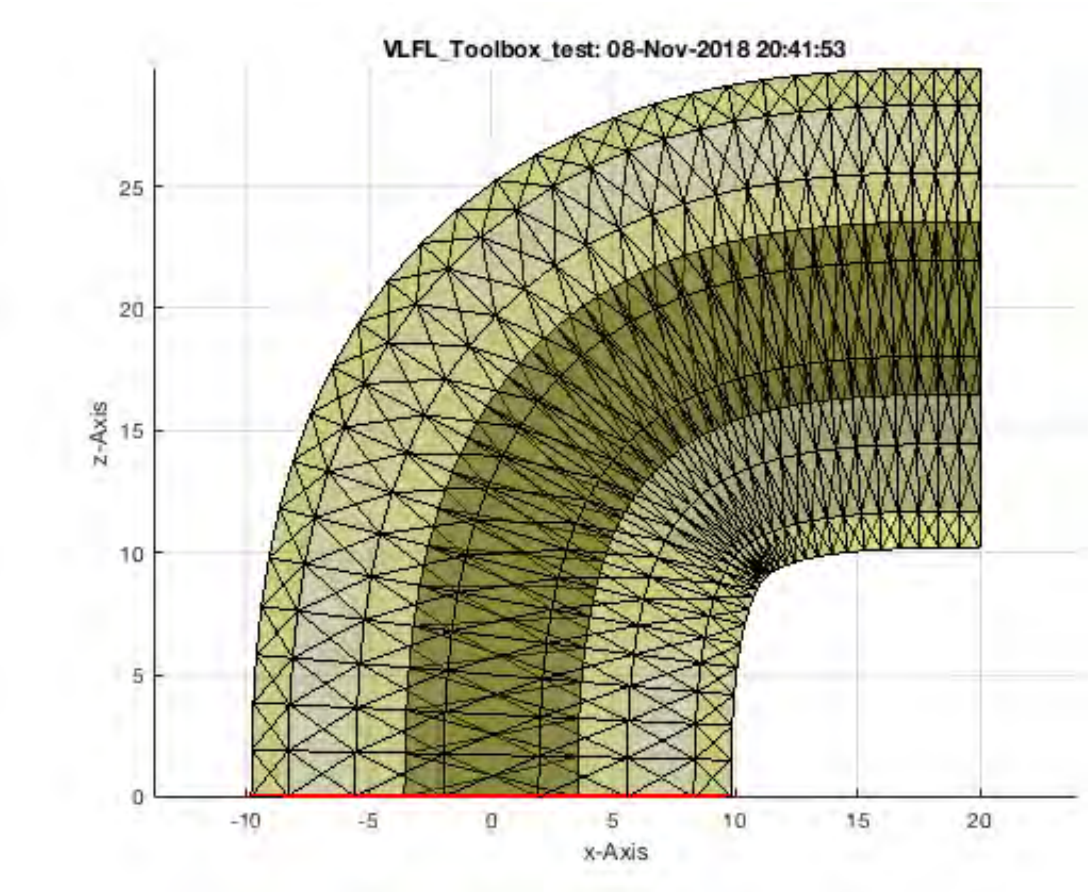
The Bezier-curve tube

```
SGfigure; CPLplot(CPL,'r-',2);  axis on ; grid on;
SG=SGcontourtube(CPL,VLB); SGplot(SG,'y'); VLFLplotlight(0,0.3);view (0,0);
```



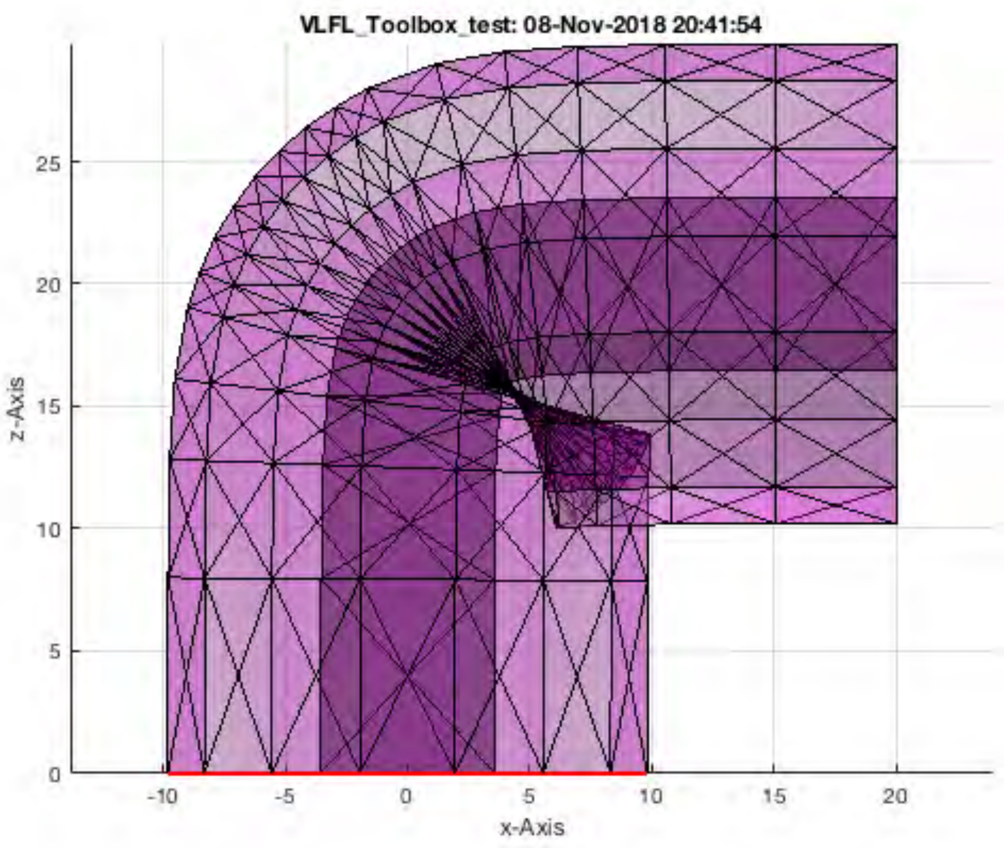
The Bezier-curve tube

```
SGfigure; CPLplot(CPL,'r-',2);  axis on ; grid on;
SG=SGofCPLCVLR(CPL,VLB); SGplot(SG,'y'); VLFLplotlight(0,0.3);view (0,0);
```



The Radial-curve tube

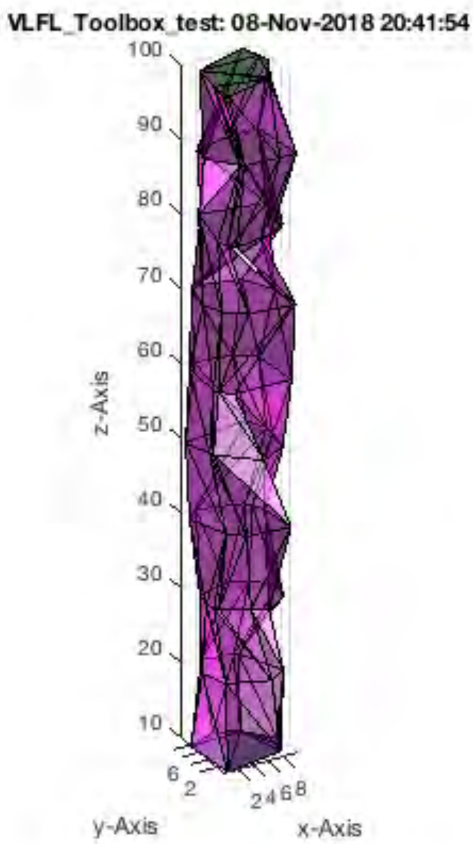
```
SGfigure; CPLplot(CPL,'r-',2);  axis on ; grid on;
SG=SGofCPLCVLR(CPL,VLr); SGplot(SG,'m'); VLFLplotlight(0,0.3);view (0,0);
```



4. Creating solids by closed polygons in different height: z-coordinate

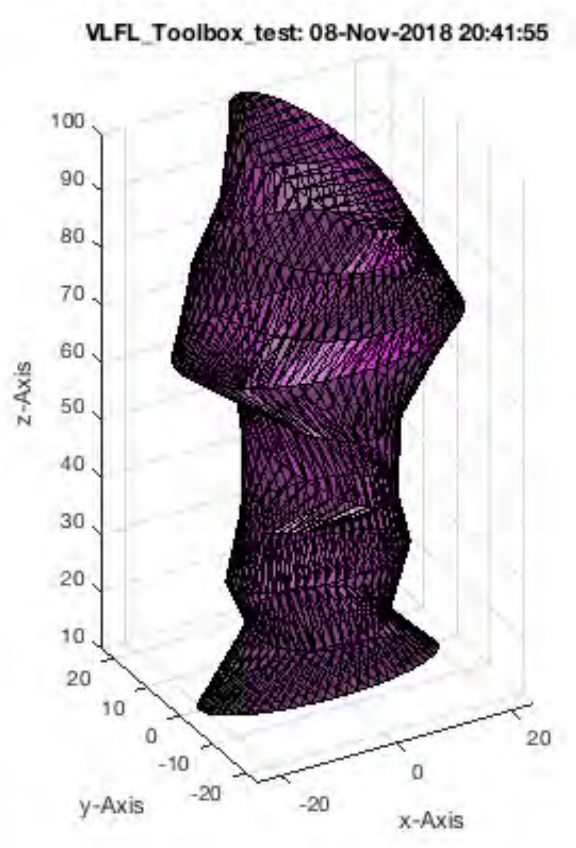
The connection of contours in different z-values works currently only with ONE contour per z-value

```
SGfigure; view(-30,30); axis on; grid on;
VL=[]; for i=1:10; VL=[VL;VLaddz(PLconvexhull(10*rand(20,2)),i*10)]; end;
[FLB,FLW,FLT]=FLofCVL(VL);
VLFLplot(VL,FLB,'b'); VLFLplot(VL,FLW,'m'); VLFLplot(VL,FLT,'g'); VLFLplotlight(0,0.5)
```



Same us but this time with ellipoids

```
SGfigure; view(-30,30); axis on; grid on;
VL=[]; for i=1:10; VL=[VL;VLaddz(PLcircle(5+20*rand,[],[],5+20*rand),i*10)]; end;
[FLB,FLW,FLT]=FLofCVL(VL); FL=[FLB;FLW;FLT];
VLFLplot(VL,FL,'m'); VLFLplotlight(0,0.5)
```



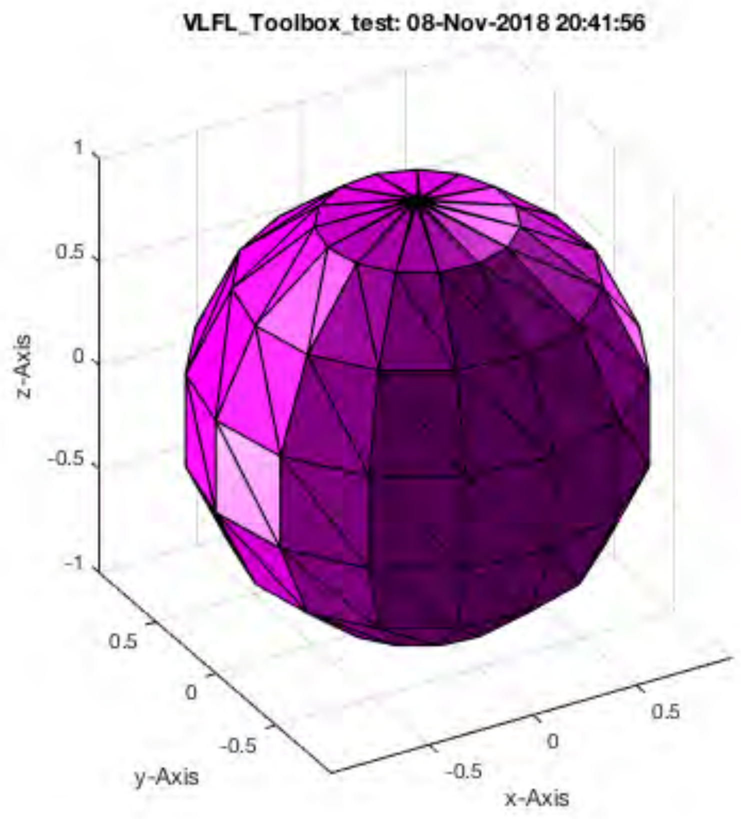
5. Creating a sphere with minmal number of points

For the creation of spherical joints, we need sphered shaped geometries. Those spheres consist of circular point lists in different z-height. The number of points of each polygon, the number of polygons an the z-resolution depend on the size of the sphere.

A sphere with just 1mm radius and a resolution of 50µm (default) has only hundreds of facets.

```
SGsphere(1)
```

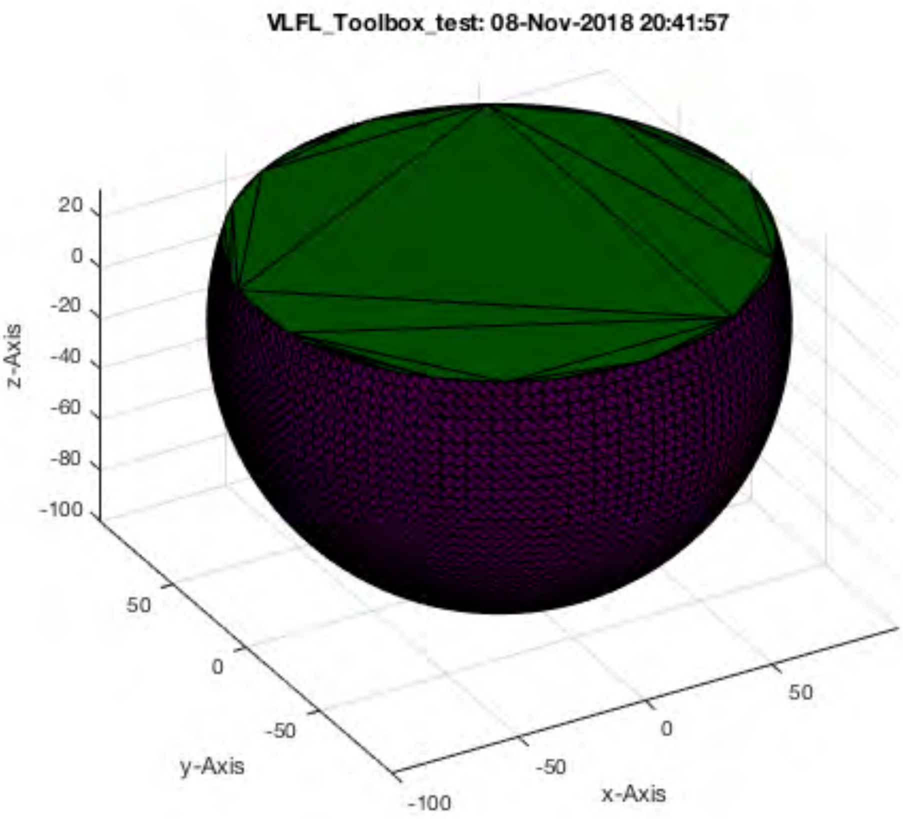
```
ans =

  struct with fields:

    VL: [120×3 double]
    FL: [236×3 double]
```



Asphere with 100mm radius and a resolution of 50µm (default) has then thausands of facets.

```
SGsphere(100,[],pi/10)
```

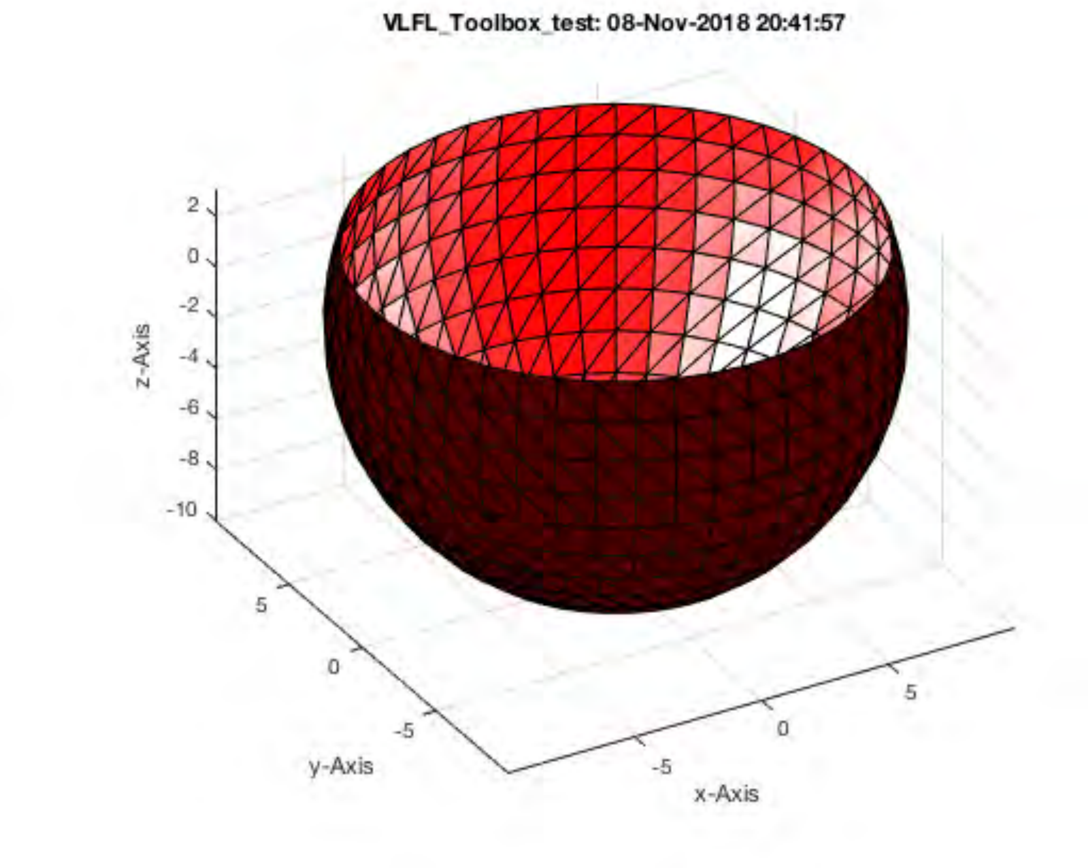
```
ans =

  struct with fields:

    VL: [6063×3 double]
    FL: [12122×3 double]
```



6. Creating a spherical joint

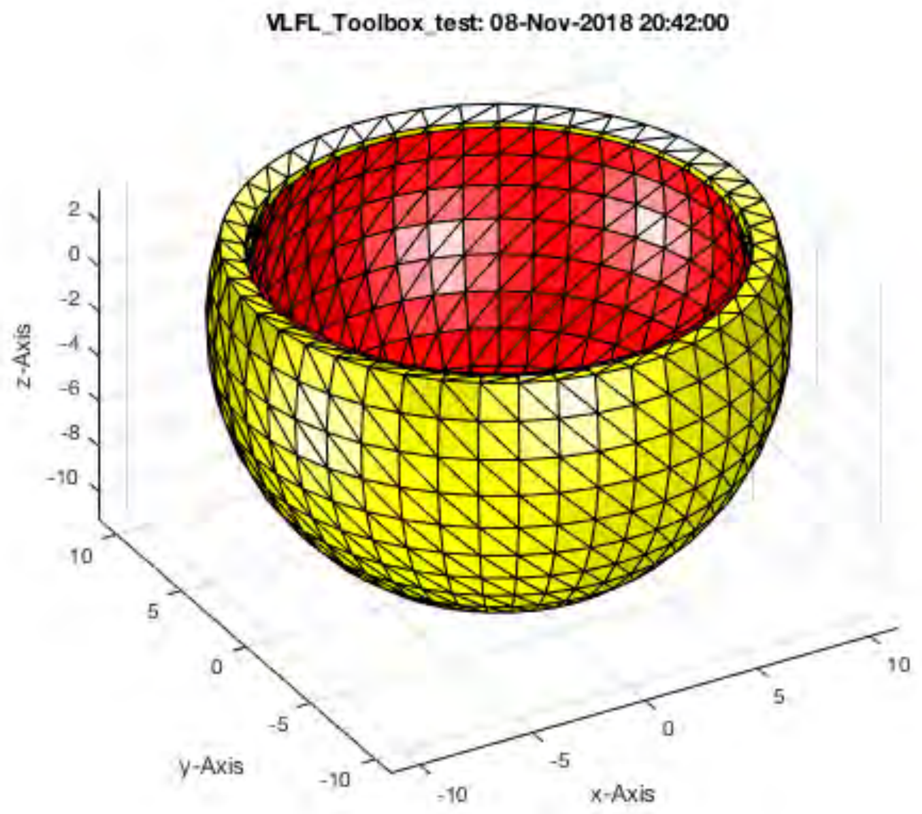
First we have to create a sphere and separate the spherical surface

```
SGfigure; view(-30,30);
[~,~,SG]=MLofSG(SGsphere(10,[],pi/10));
VLFLplot(SG.VL,SG.FL(SG.ML(:,1)==1,:));
```



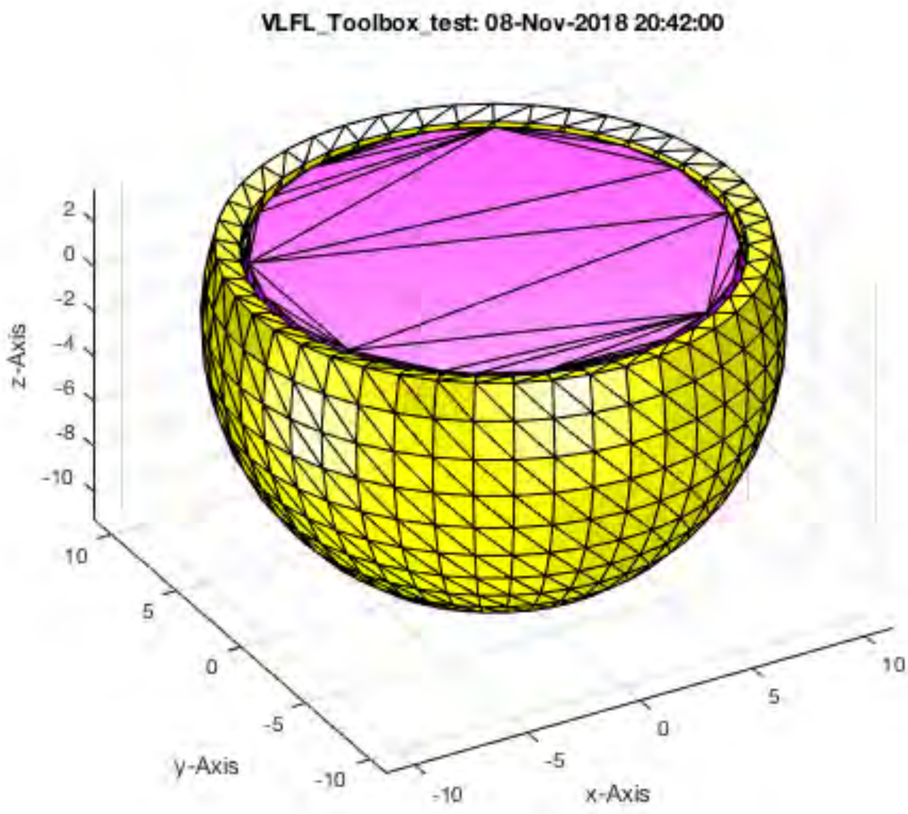
Now create the surface for the joint from the spherical surfacewith tickness 1 as bearing

```
SGofSurface(SG.VL,SG.FL(SG.ML(:,1)==1,:),1);
```



Now fill in the sphere ball as joint

```
SGplot(SG,'m');
```



Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:42:01!
Executed 08-Nov-2018 20:42:03 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
robotics_system_toolbox
simulink
video_and_image_blockset
===========================================================================================
==========
```

- *Tim Lueth, tested and compiled on OSX 10.7.5 with Matlab 2014b on 2015-10-12*
- _________, executed and published on 64 Bit PC using Windows with Matlab 2015a on 2015-xx-xx_

Published with MATLAB® R2018a

Tutorial 17: Filling and Bending of Polygons and Solids

2017-03-29: 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:

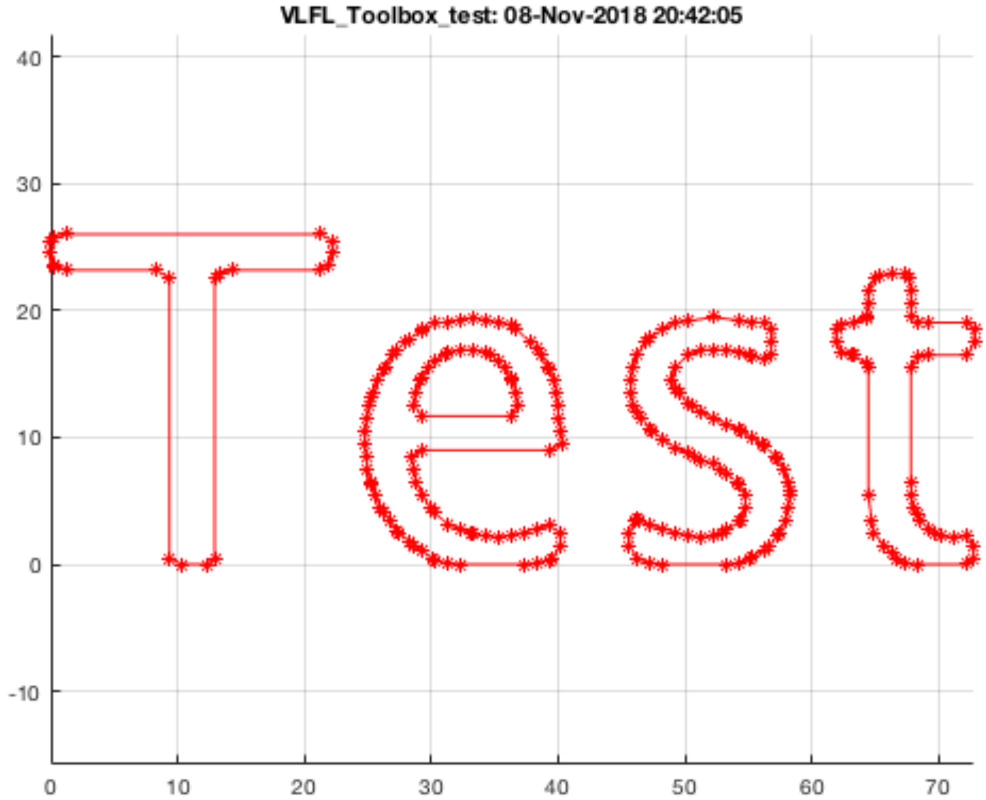
- Tutorial 01: First Steps Using the VLFL-Toolbox for Solid Object Design
- Tutorial 02: Using the VLFL-Toolbox for STL-File Export and Import
- Tutorial 03: Closed 2D Contours and Boolean Operations in 2D
- Tutorial 04: 2½D Design Using Boolean Operators on Closed Polygon Lists (CPL)
- Tutorial 05: 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

Motivation for this tutorial: (Originally SolidGeometry 3.7 required)

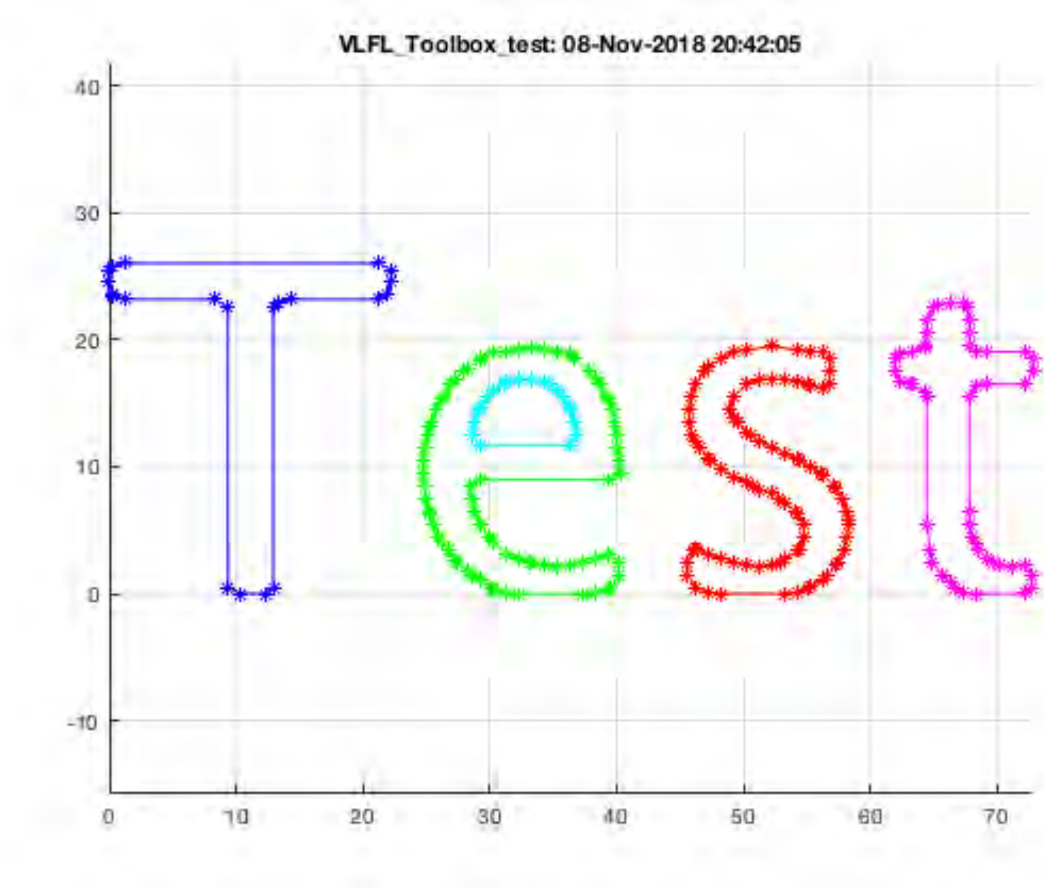
For robotics design,very often we have a wish to extrude a CPL not only in an orthognal z direction to the xy-plane, but in an any desired direction even along a path in 3D space. Intutively we expect a result, but this is not easy to achieve automatically. Anyway, for those tasks we have two functions:

1. Creating Closed Polygon Line

```
CPL=CPLoftext('Test');
SGfigure; view(0,90); CPLplot(CPL);
```

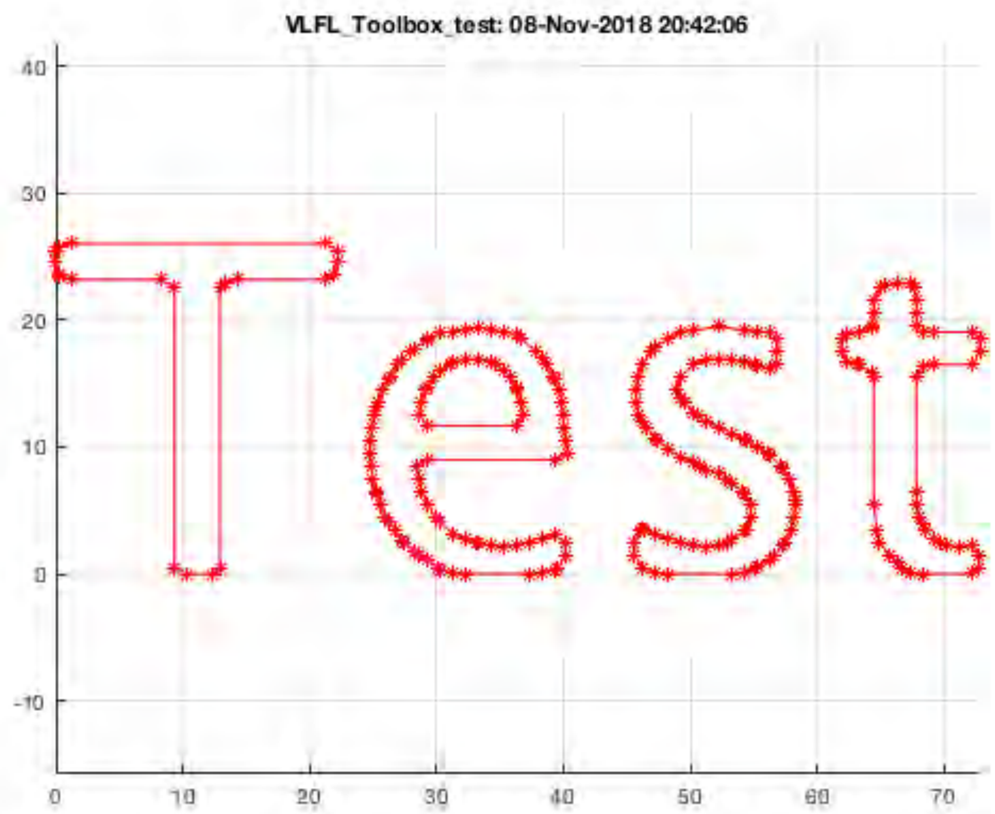



```
SGfigure; view(0,90); CVLplot(CPL);
```

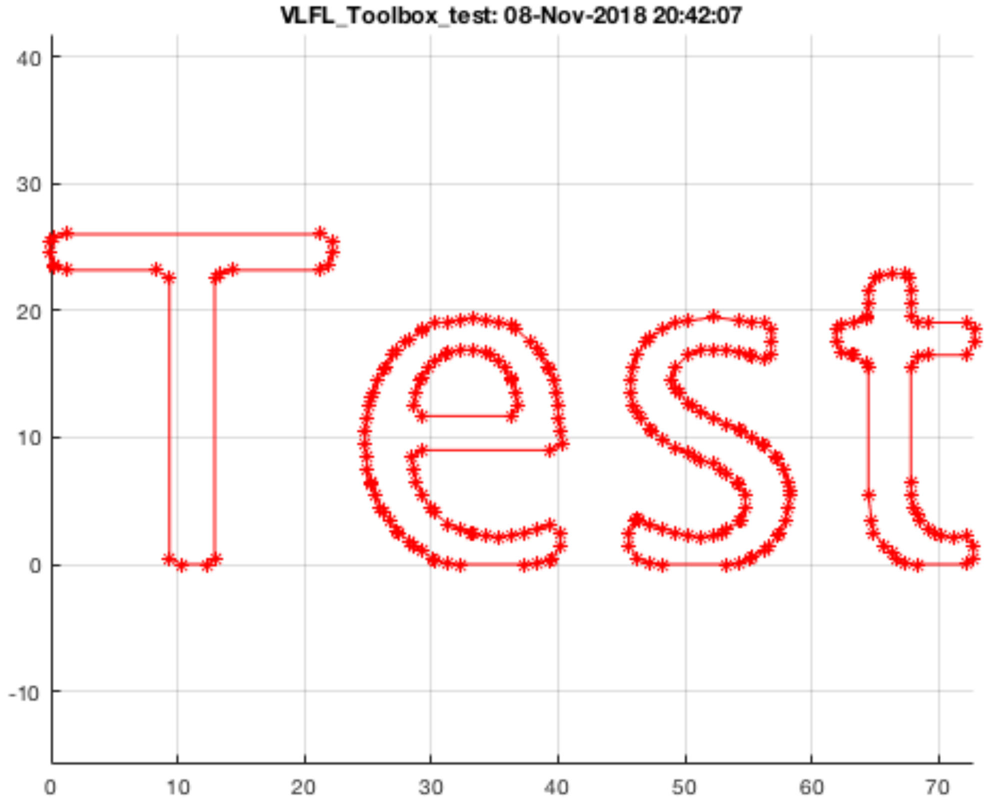


2. Converting CPL into PL EL

```
[PL,EL]=PLELofCPL(CPL);
SGfigure; view(0,90); PLELplot(PL,EL);
```

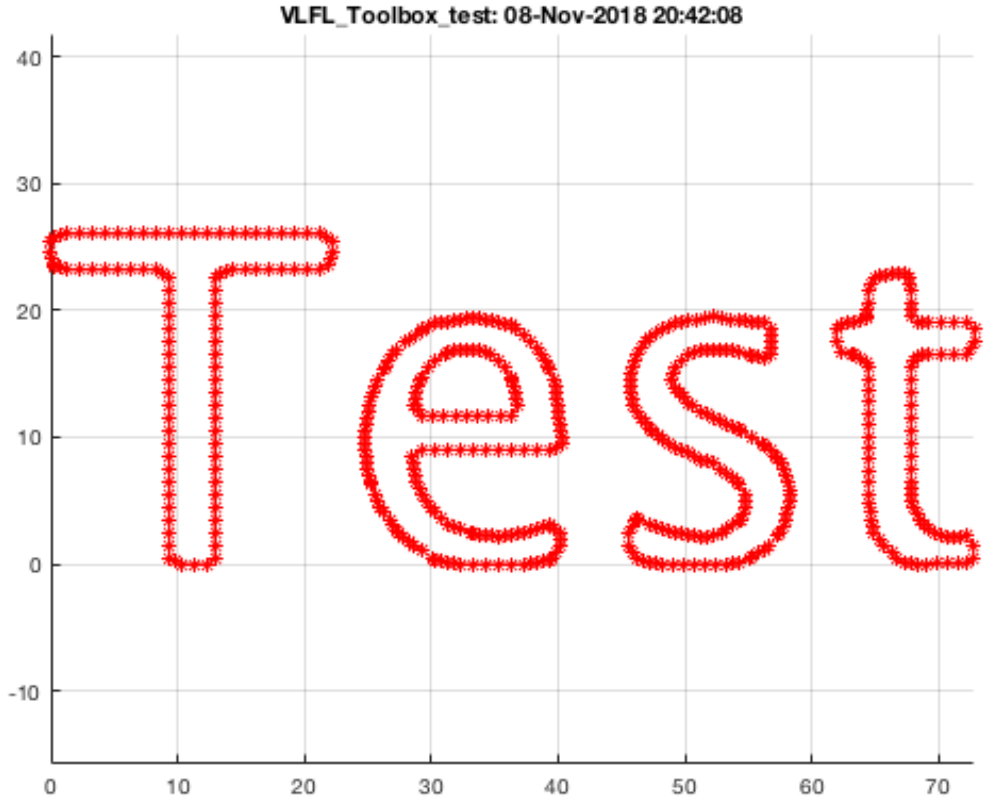



```
CPL=CPLofPLEL(PL,EL);
SGfigure; view(0,90); CPLplot(CPL);
```

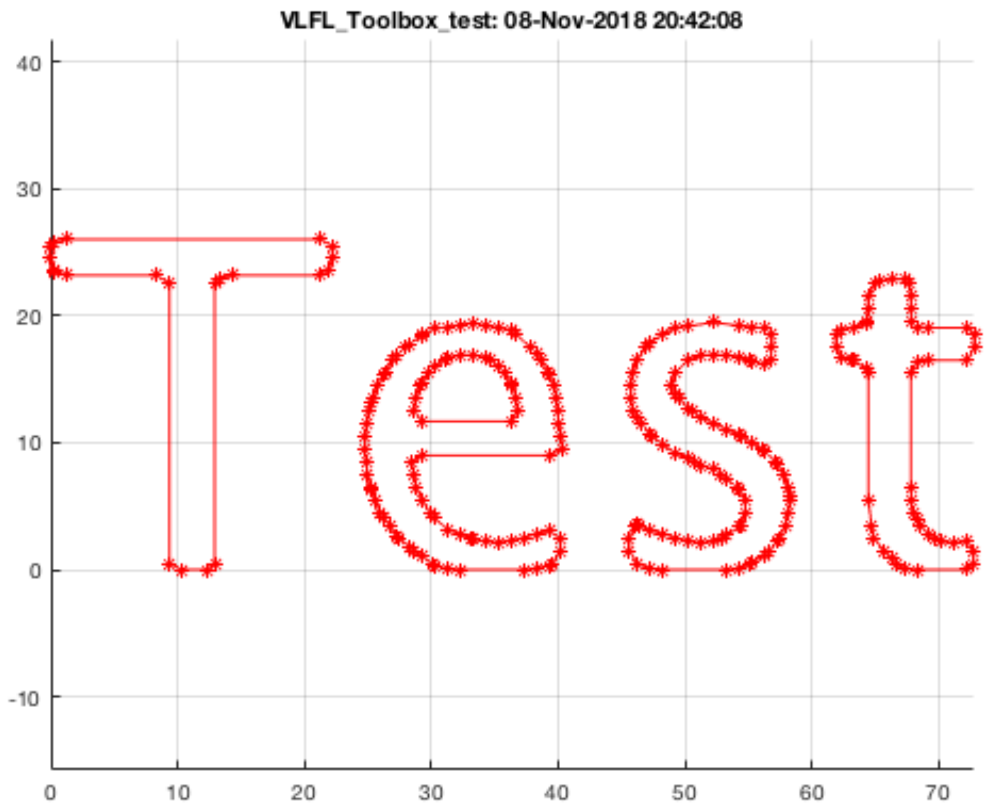


3. Adding and removing points on the contour

```
CPLN=CPLaddauxpoints(CPL,1);
SGfigure; view(0,90); CPLplot(CPLN);
```

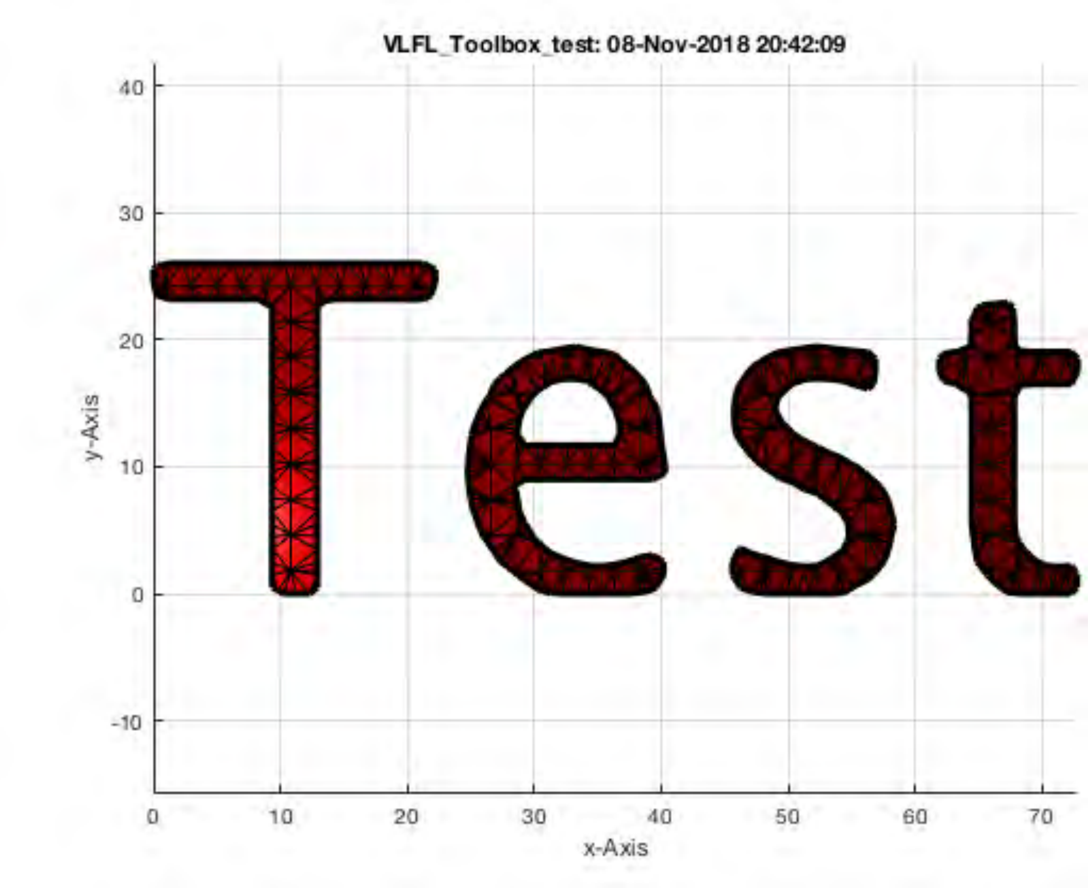



```
CPLB=CPLremstraight(CPL);
SGfigure; view(0,90); CPLplot(CPLB);
```



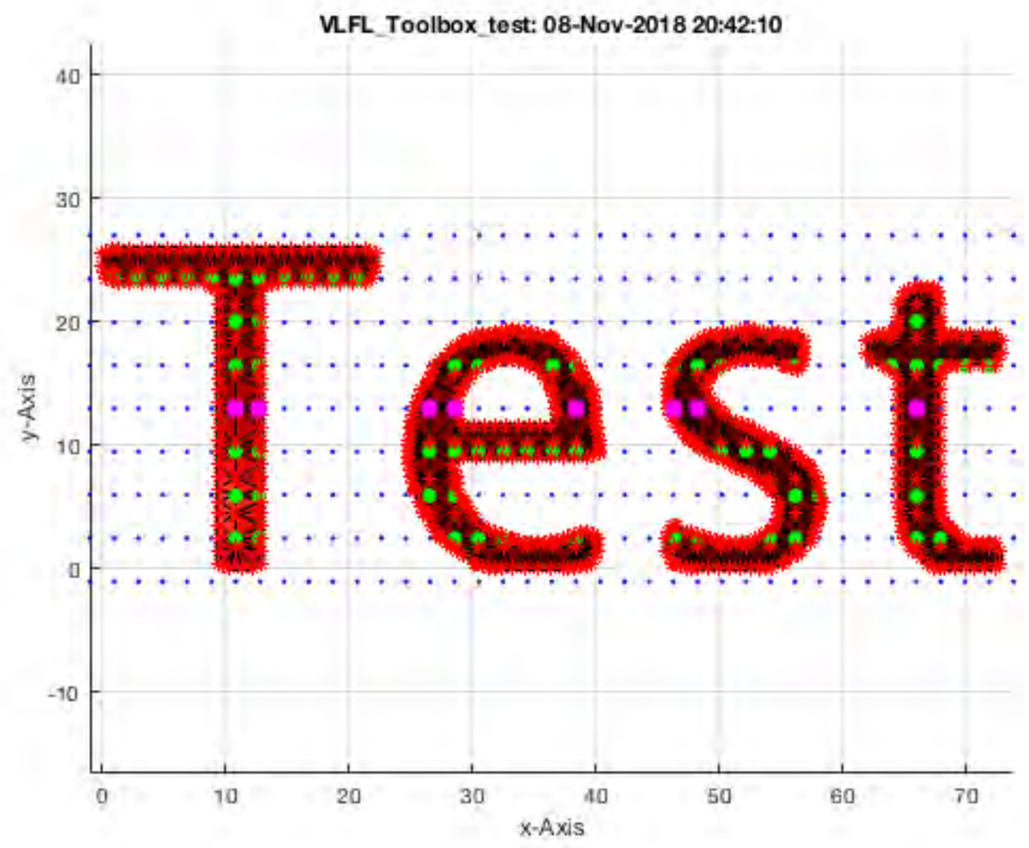
4. Adding and removing points inside of the contour

```
[PL,FL,EL]=PLFLofCPLdelaunayGrid(CPL,1,2,3);
SGfigure; view(0,90); VLFLplot(PL,FL); VLELplots(PL,EL,'k',3);
```



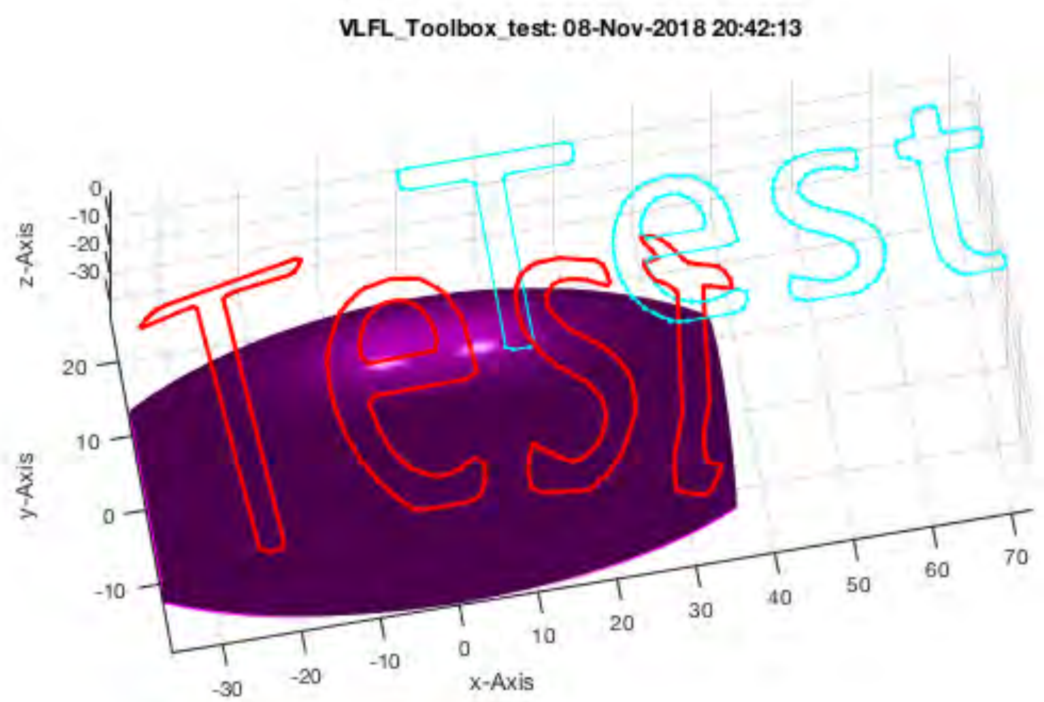
5. Calculate Grid Points

```
GPL=GPLauxgridpointsPLEL(PL,EL,2,4);
insidePLELdelaunay(PL,EL,GPL);
```

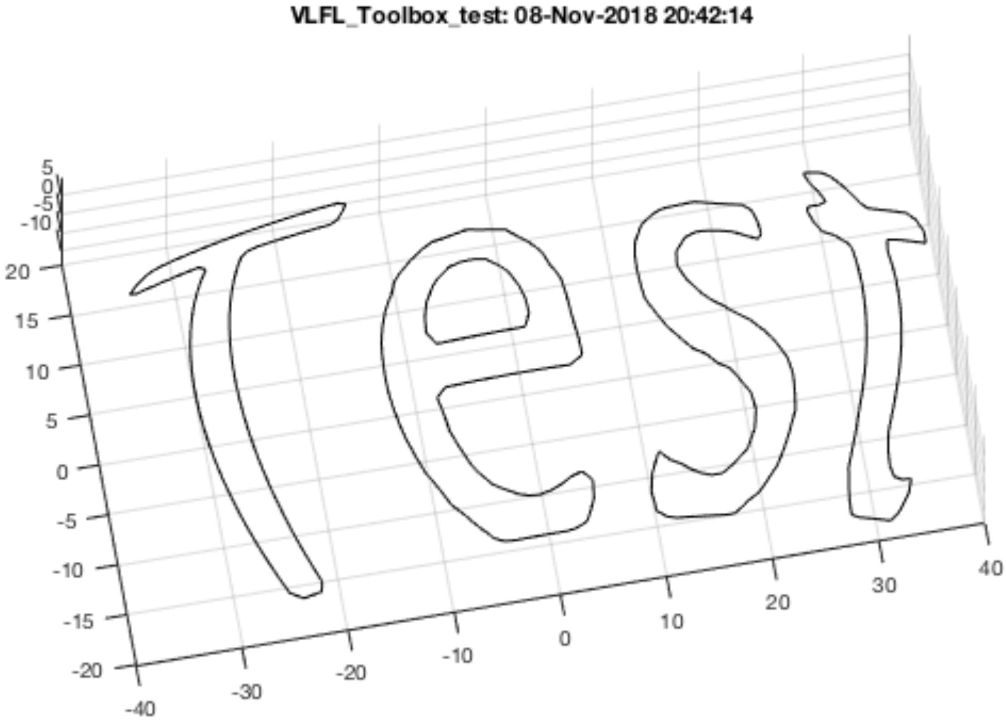


6. Bending of contours

```
% Without auxiliary points

BPL=PLbending(CPL,50,10,20);
PLbending(CPL,50,10,20);

% With auxiliary points
```

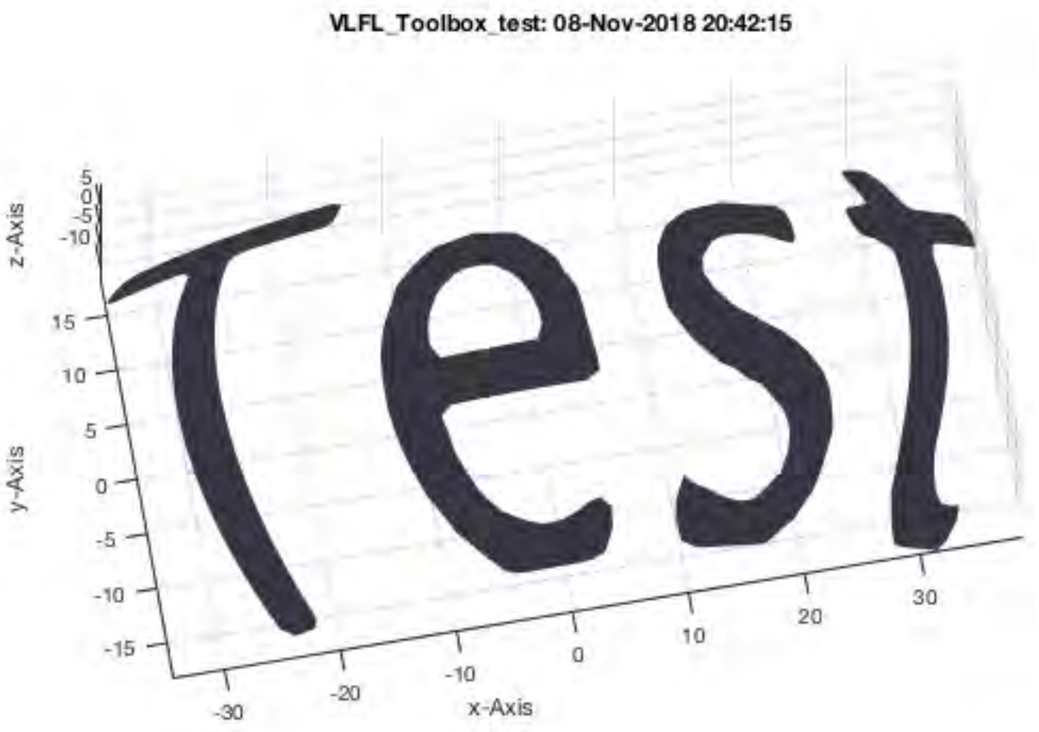



```
[PL,FL,EL]=PLFLofCPLdelaunayGrid(CPL,1,2,3);
NPL=PLbending(PL,50,10,20);
SGfigure; view(-10,70); VLELplot(NPL,EL,'k-');
```



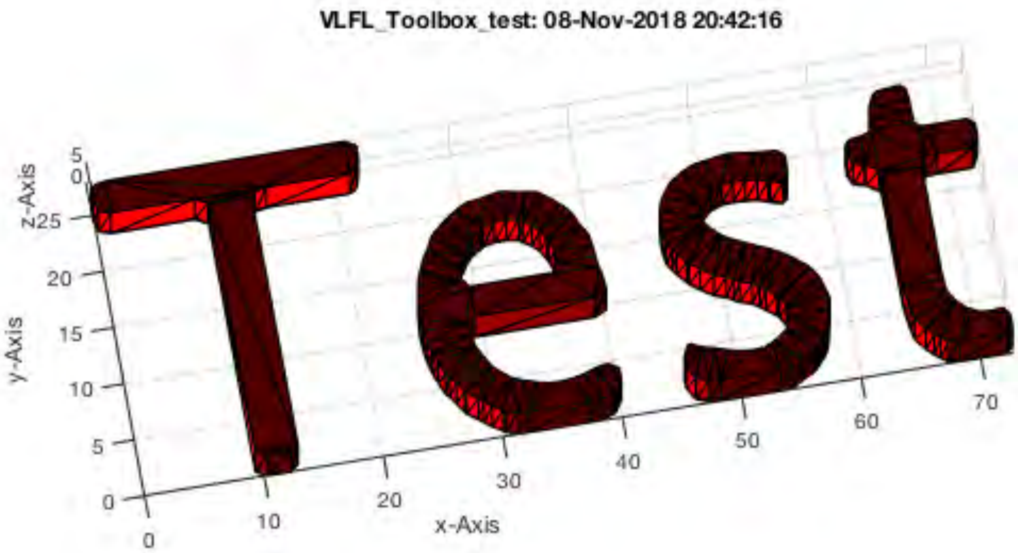
7. Bending of closed contour surfaces

```
SGfigure; view(-10,70); VLFLplot(NPL,FL,'k-'); VLFLplotlight(1,1);
```

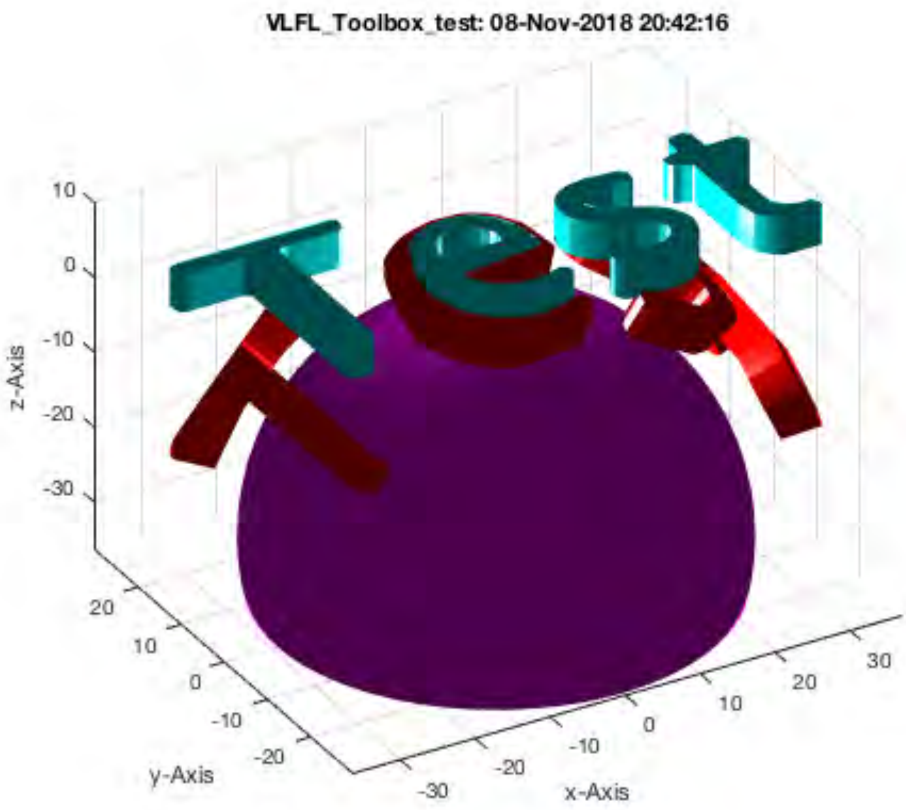


8. Bending of solid geometries

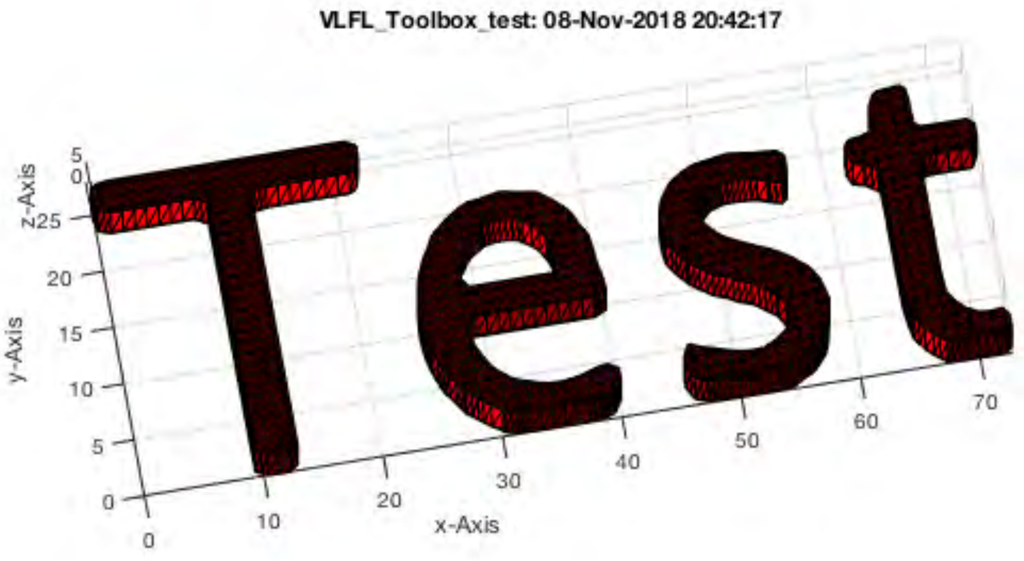
```
% Without auxiliary points

SG=SGofCPLzdelaunayGrid (CPL,5);
SGfigure; view(-10,70); SGplot(SG); VLFLplotlight(0,1);
```



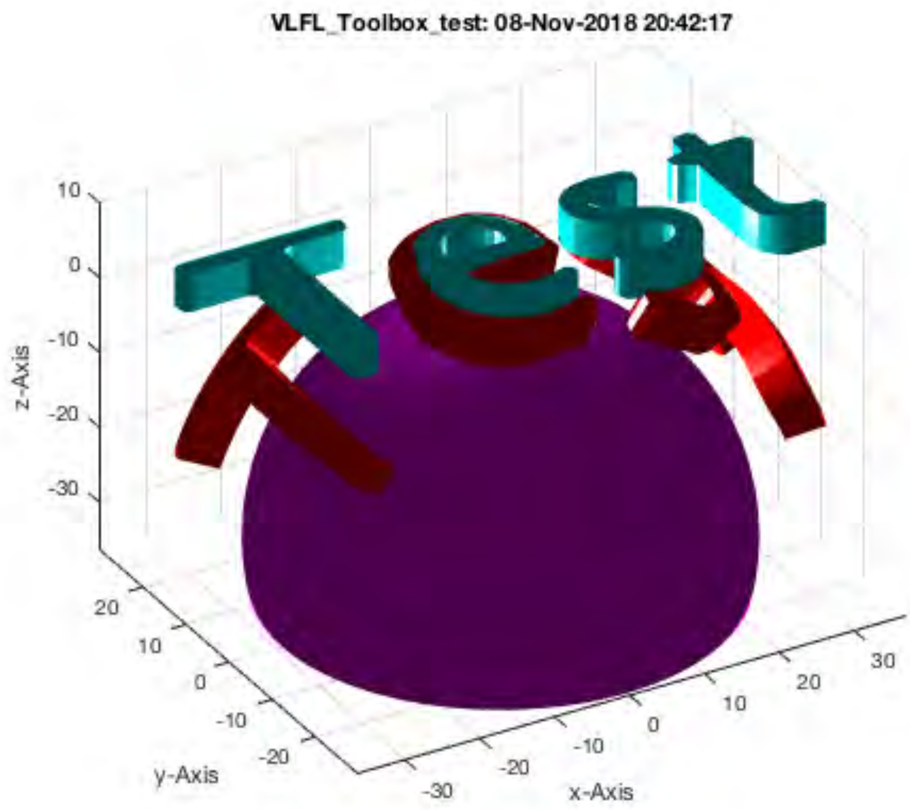

```
SGbending(SG,30,5,30); VLFLplotlight(1,1);
```




```
% With auxiliary points
SG=SGofCPLzdelaunayGrid (CPL,5,1,1,1);
SGfigure; view(-10,70); SGplot(SG); VLFLplotlight(0,1);
```




```
SGbending(SG,30,5,30); VLFLplotlight(1,1);
```



Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:42:18!
Executed 08-Nov-2018 20:42:20 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
robotics_system_toolbox
simulink
video_and_image_blockset
===========================================================================================
==========
```

- *Tim Lueth, tested and compiled on OSX 10.11.6 with Matlab 2016b on 2017-03-29*
- _________, executed and published on 64 Bit PC using Windows with Matlab 2015a on 2015-xx-xx_

Published with MATLAB® R2018a

Tutorial 18: Analyzing and modifying STL files from CSG modeler (Catia)

2017-04-04: 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)

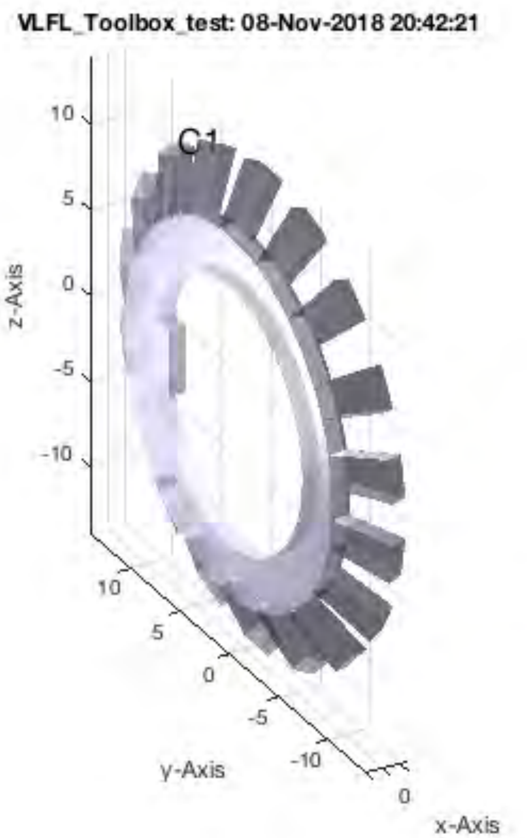
Motivation for this tutorial: (Originally SolidGeometry 3.8 required)

Often CSG modelers are used for mechanism construction and the subsequent STL export. This tutorial will show you how to use those STL files after reading them in as SG

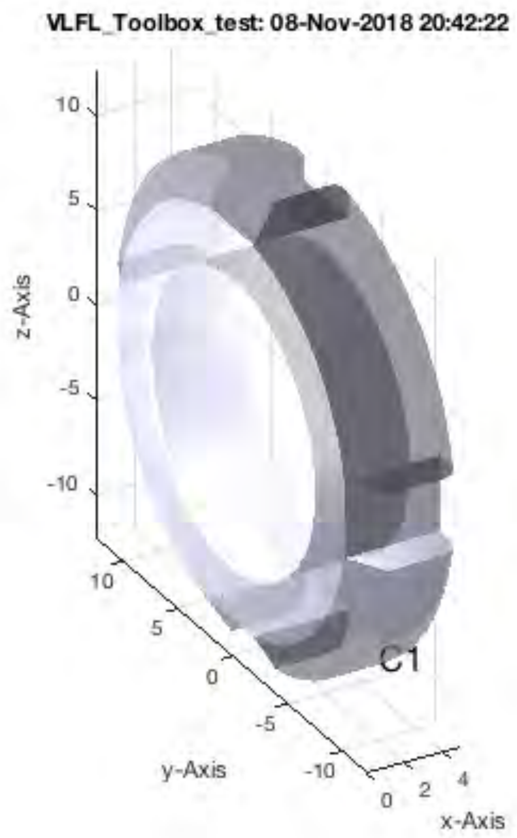
The mat-File 'FZG_Welle.mat' contains already read STL files of the TUM FZG institute. There are 6 Solids that contain overall 18 separate surfaces of a bearing for an axle. You can either load the data from the WWW page of the Technical University of Munich or after download use the load command.

```
% loadweb ('FZG_Welle.mat',true) % load the data from the TUM Mimed Page
load ('FZG_Welle.mat'); % load the data from the matlab path
FZG={SG1,SG2,SG3,SG4,SG5,SG6};
```

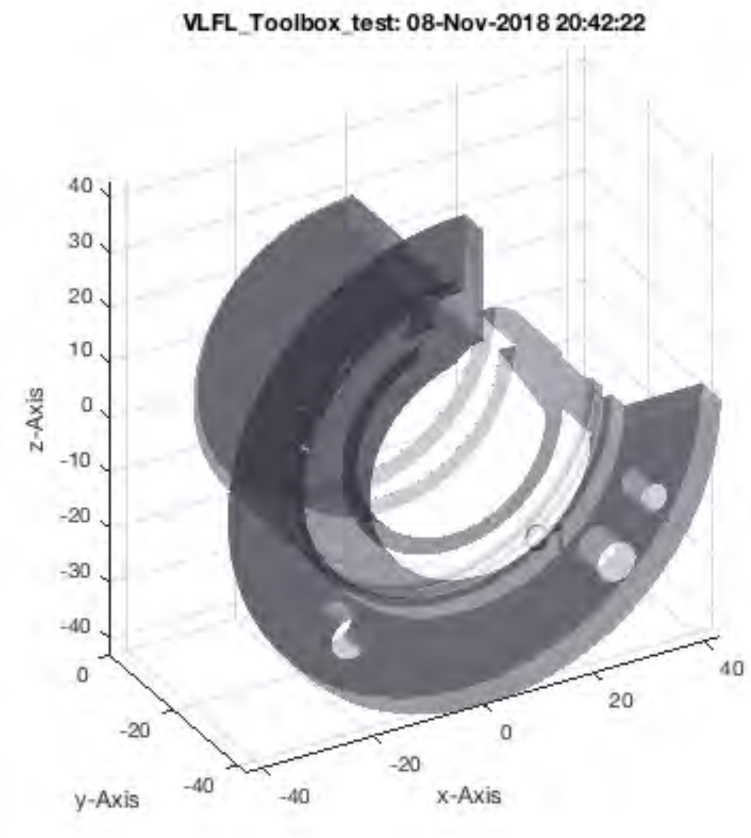
```
SGfigure; SGsurfaceplot(SG1); view(-30,30);
```



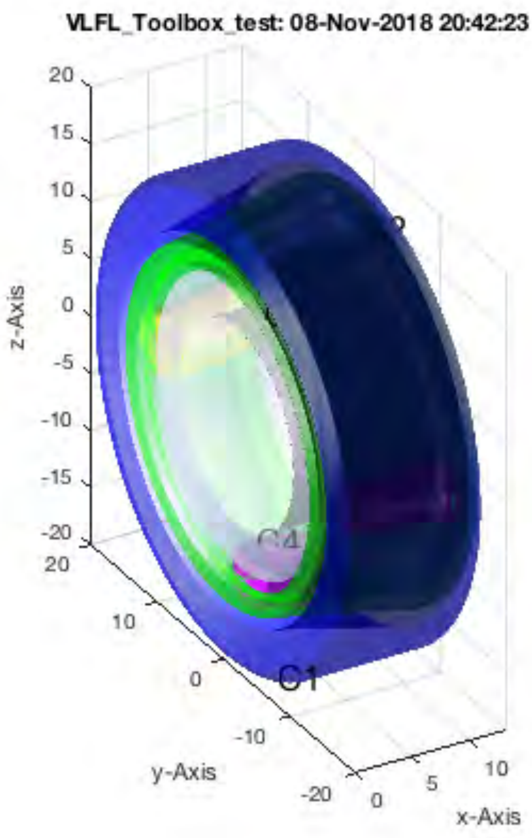

```
SGfigure; SGsurfaceplot(SG2); view(-30,30);
```



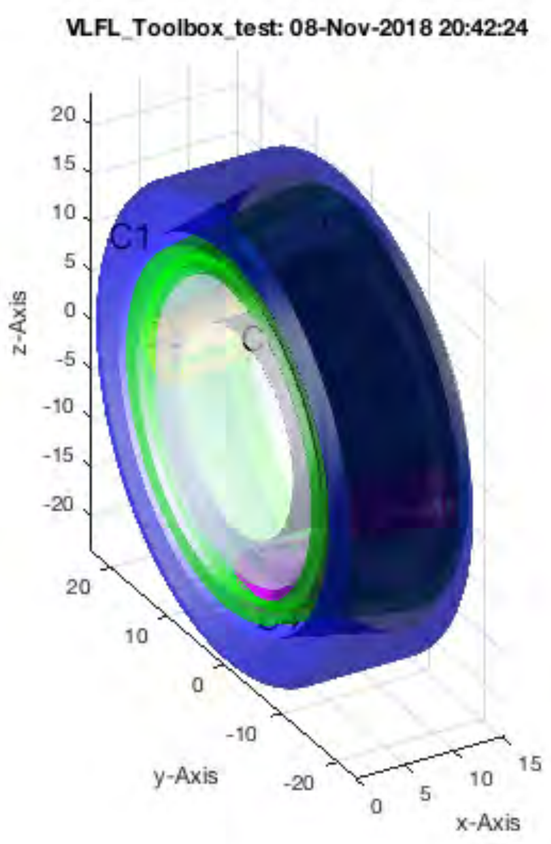

```
SGfigure; SGsurfaceplot(SG3); view(-30,30);
```



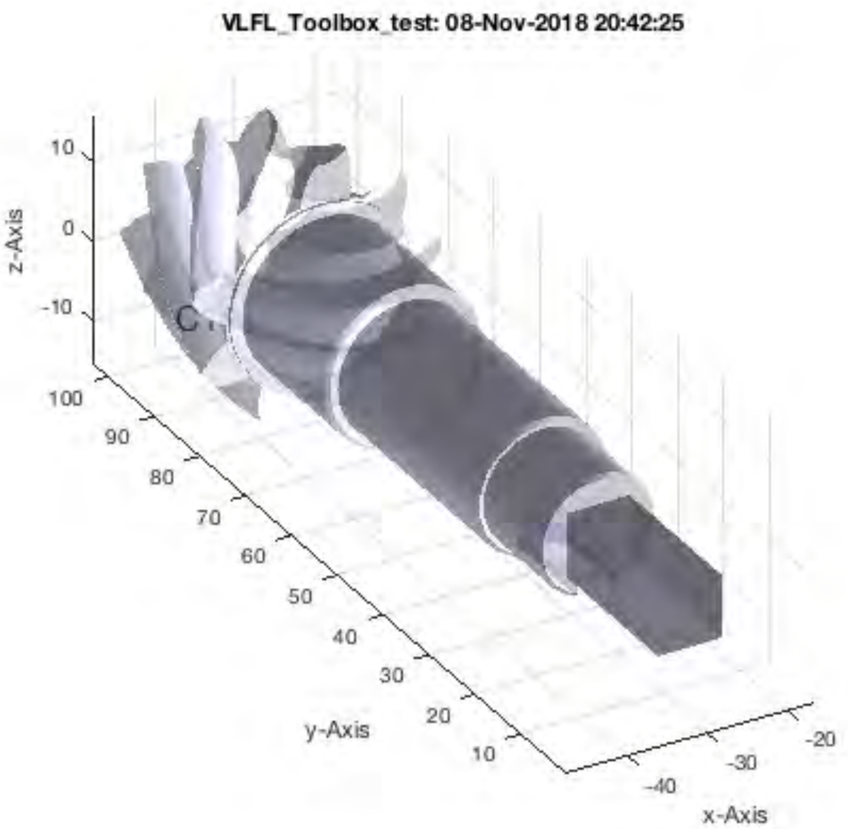

```
SGfigure; SGsurfaceplot(SG4); view(-30,30);
```




```
SGfigure; SGsurfaceplot(SG5); view(-30,30);
```

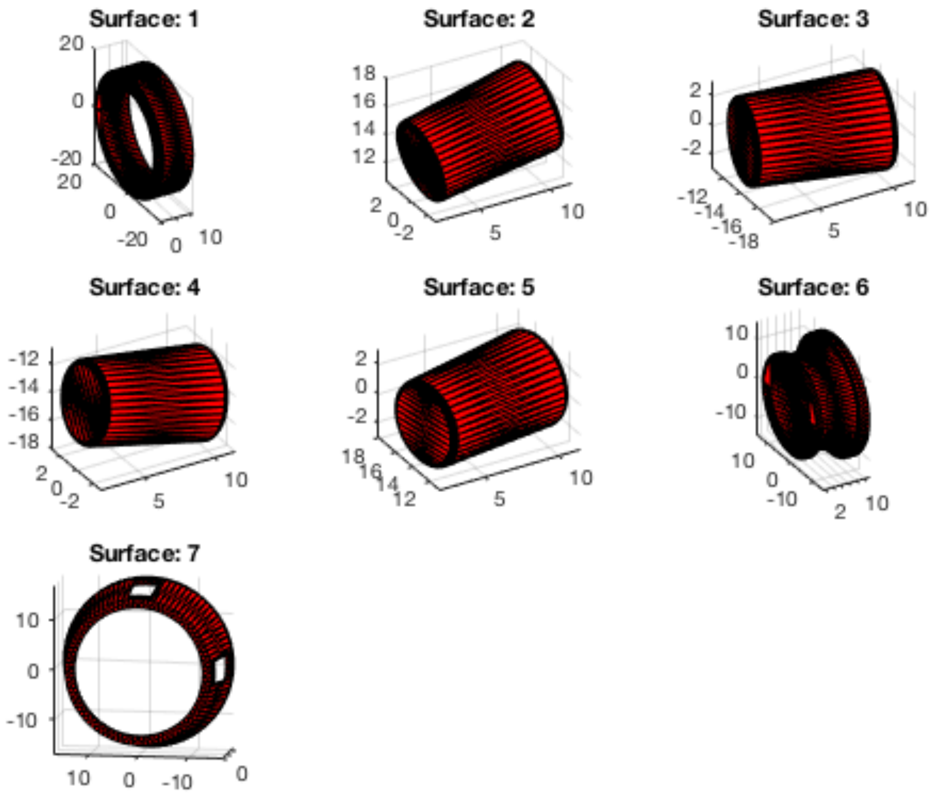



```
SGfigure; SGsurfaceplot(SG6); view(-30,30);
```



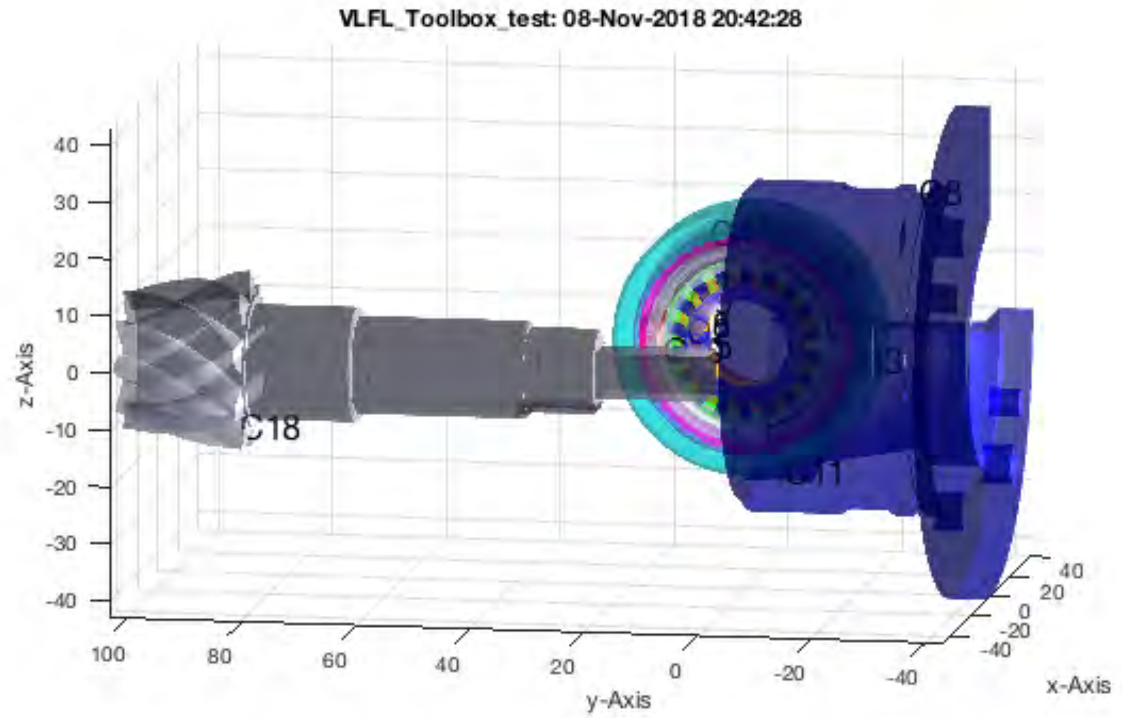
1. Show all separated surfaces that are part of a Solid

```
SGseparate(SG4); view(-80,10);
```



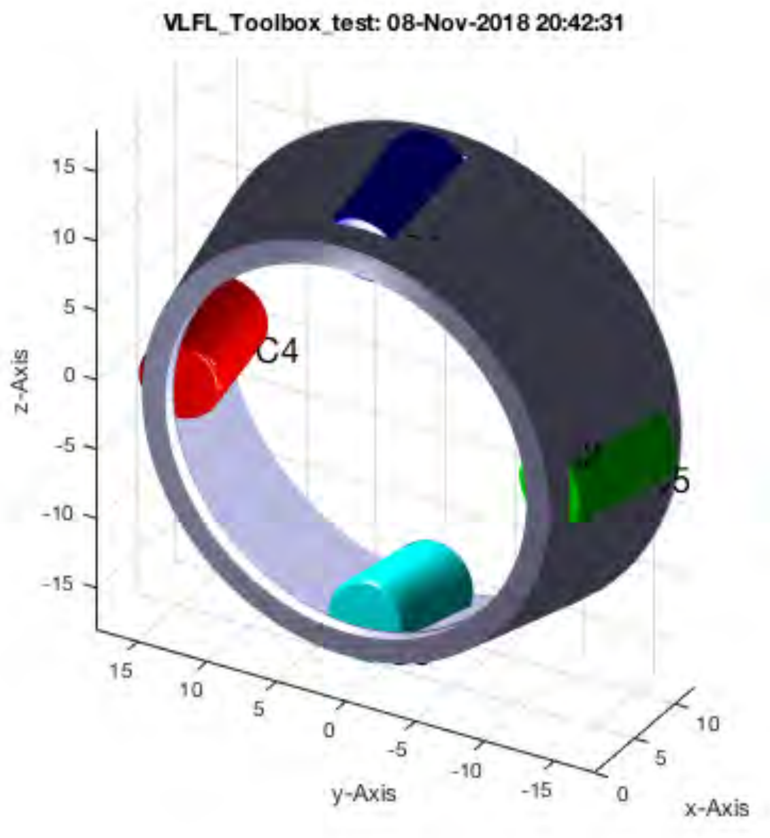
Show all surfaces in different colors

```
SGsurfaces(FZG); view(-80,10);
```



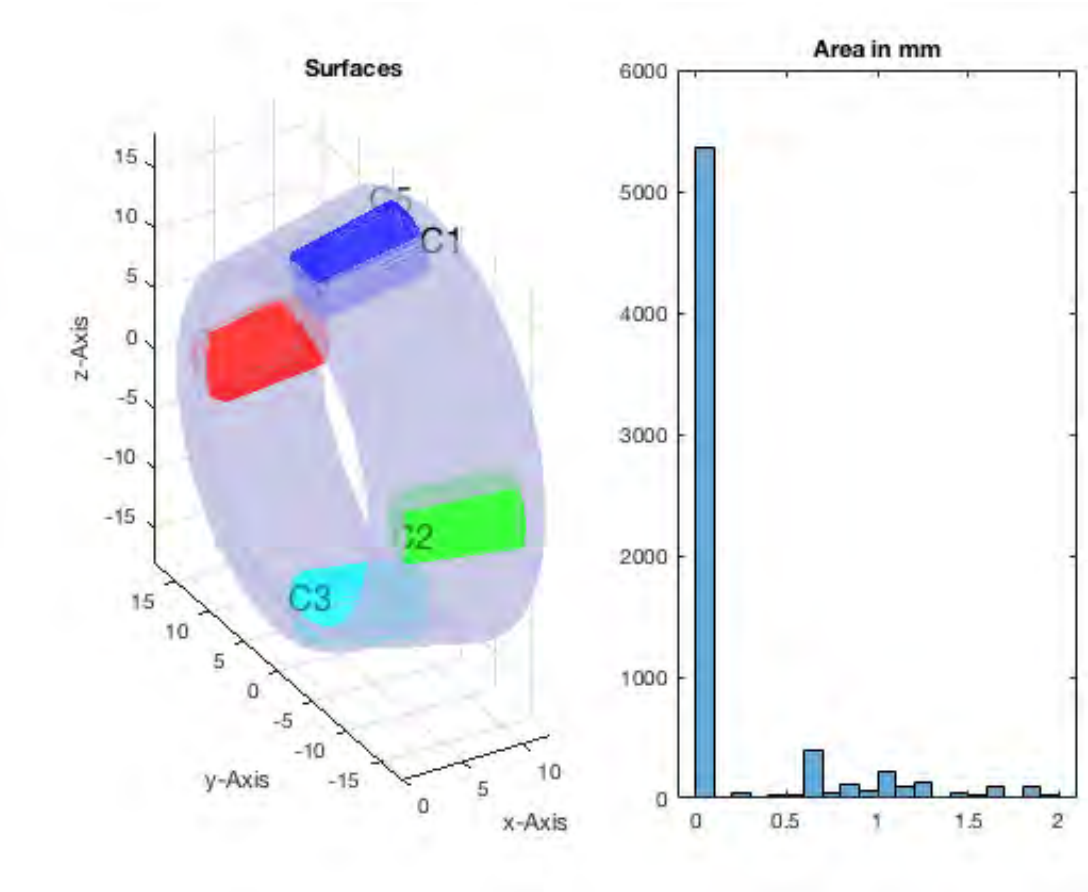
2.Select some of the surfaces

```
SGsurfaces(SG4,[2 3 4 5 7]); view(-60,30); VLFLplotlight(1,1);
```



3. Show the size of the surfaces as histogram

```
SGsurfacehistogram(SG4,[2 3 4 5 7]);
```



4. Show just a single solid

```
B=SGsurfaces(SG4)
```

```
B =

  7×1 cell array

    {1×1 struct}
    {1×1 struct}
    {1×1 struct}
    {1×1 struct}
    {1×1 struct}
    {1×1 struct}
    {1×1 struct}
```

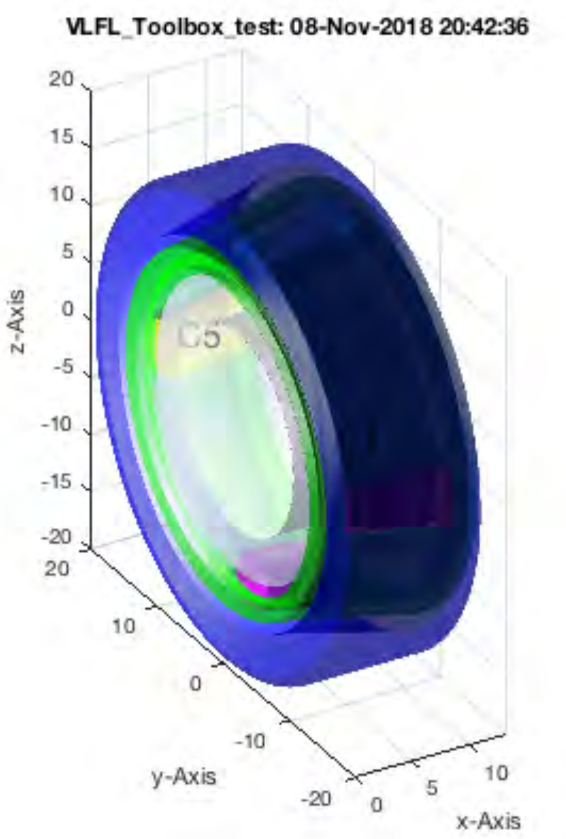
5. Shrink all convex parts

Now reduce al convex solids by 0.3 mm SG=SGreadSTL('30204-a.stl')

```
B=SGsurfaces(SG4)
for i=1:length(B)
    if SGisconvex(B{i})
        B{i}=SGgrow(B{i},-0.3);
%         B{i}=SGofVLdelaunay(B{i}.VL); % Just to show convex solids
```

```
    end
end


SGsurfaces(B);
```

```
B =

  7×1 cell array

    {1×1 struct}
    {1×1 struct}
    {1×1 struct}
    {1×1 struct}
    {1×1 struct}
    {1×1 struct}
    {1×1 struct}
```



6. Print all surfaces in different STL files

```
SGwriteMultipleSTL(B)
```

```
SGwritemultipleSTL: Writing 7 STL files in /Users/lueth/Desktop/Toolbox_test/EXP-2018-11-08
/
```

Show the written files on disk

```
dir ([desktopdir expname])  %
```

```
.                         EXP-2018-11-08_0003.stl  EXP-2018-11-08_0007.stl
..                        EXP-2018-11-08_0004.stl
EXP-2018-11-08_0001.stl  EXP-2018-11-08_0005.stl
EXP-2018-11-08_0002.stl  EXP-2018-11-08_0006.stl
```

Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:42:37!
Executed 08-Nov-2018 20:42:39 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
robotics_system_toolbox
simulink
video_and_image_blockset
===========================================================================================
==========
```

- *Tim Lueth, tested and compiled on OSX 10.11.6 with Matlab 2016b on 2017-03-29*
- _________, executed and published on 64 Bit PC using Windows with Matlab 2015a on 2015-xx-xx_

Published with MATLAB® R2018a

Tutorial 19: Creating drawing templates and dimensioning from polygon lines

2017-04-23: 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 3.8 required)

```
%
% function VLFL_EXP19
```

Motivation

Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:42:39!
Executed 08-Nov-2018 20:42:41 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
robotics_system_toolbox
simulink
video_and_image_blockset
===========================================================================================
==========
```

- *Tim Lueth, tested and compiled on OSX 10.11.6 with Matlab 2016b on 2017-03-29*
- _________, executed and published on 64 Bit PC using Windows with Matlab 2015a on 2015-xx-xx_

Published with MATLAB® R2018a

Tutorial 20: Programmatically Interface to SimMechanics Multi-Body Toolbox

2016-11-19: 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:

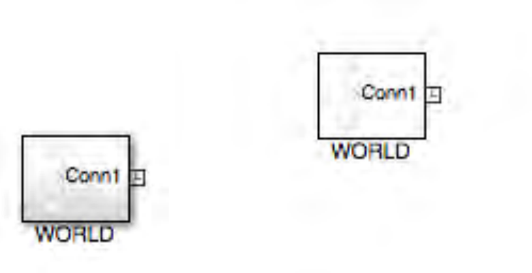
- Tutorial 01: First Steps Using the VLFL-Toolbox for Solid Object Design
- Tutorial 02: Using the VLFL-Toolbox for STL-File Export and Import
- Tutorial 03: Closed 2D Contours and Boolean Operations in 2D
- Tutorial 04: 2½D Design Using Boolean Operators on Closed Polygon Lists (CPL)
- Tutorial 05: 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

Motivation for this tutorial: (Originally SolidGeometry 3.0 required)

2. Creating a new SimMechanics System

```
smbNewSystem ('SG_LIB_EXP_20');            % Creates the mechansim diagramm
smbDrawNow;
```

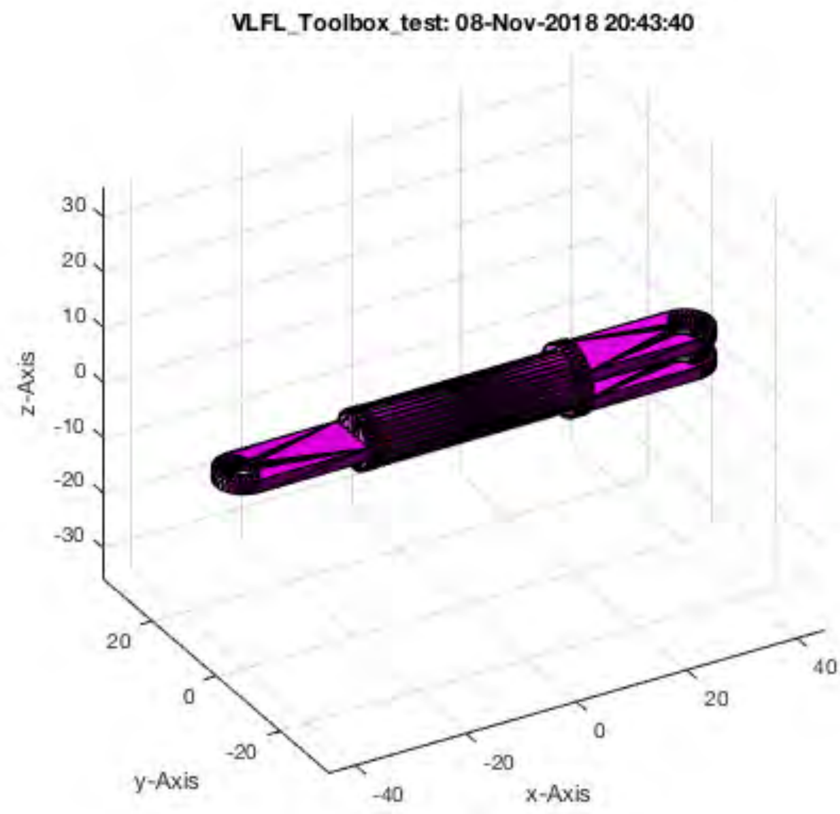
```
Creating temporary directory '/Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_20/'
```



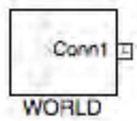
3. Create two links with length 50 and 80 and one or two mounting holes

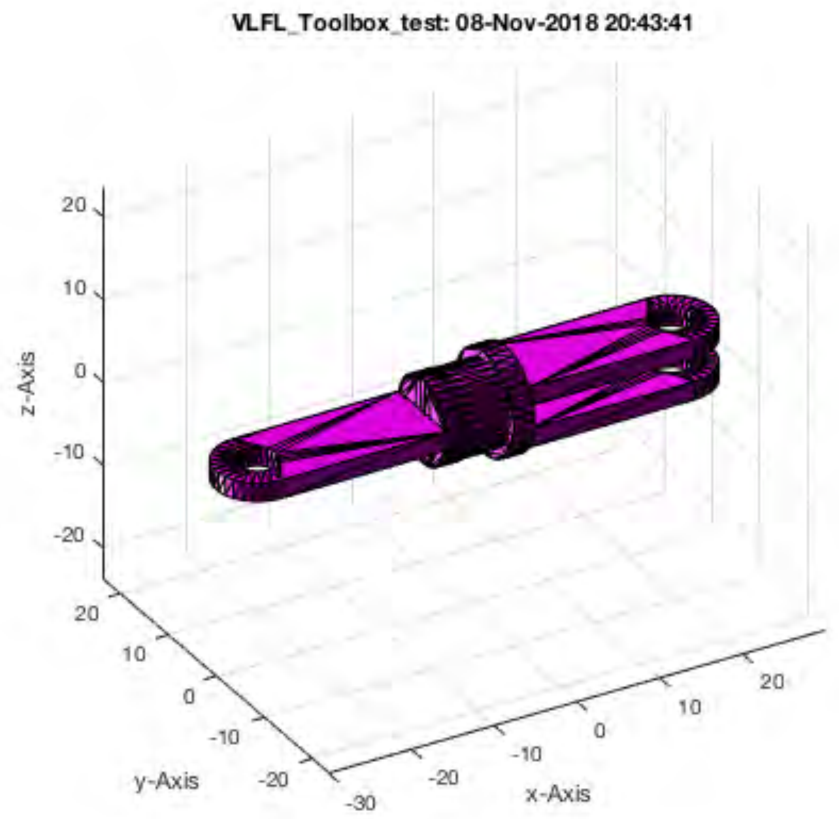
```
SG1=SGmodelLink(80,'',1,2);                % Creates a long rod with flange
SG2=SGmodelLink(50,'',1,2);                % Creates a short rod with flange
```

```
SGfigure(SG1); view(-30,30);
```



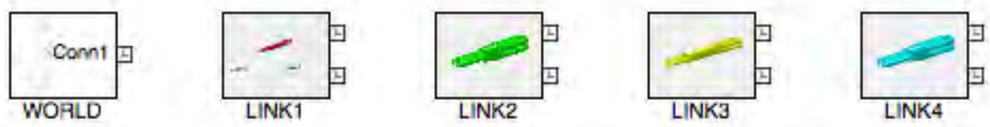

```
SGfigure(SG2); view(-30,30);
```





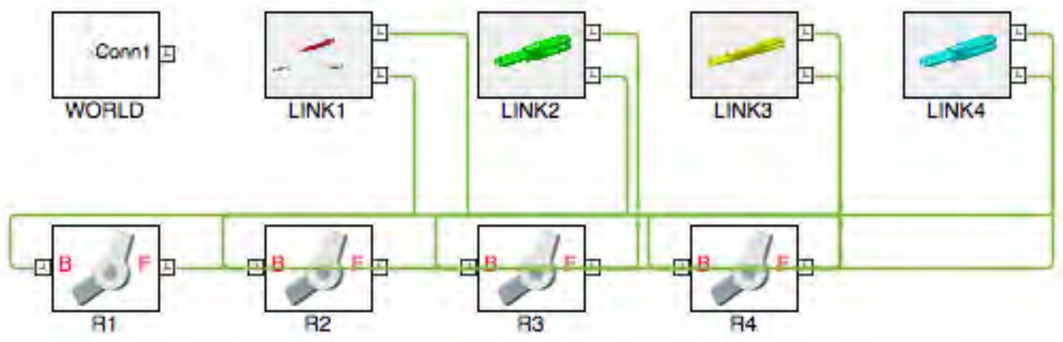
4. Create SimMechanics models for the four links in different colors

```
smbCreateSG (SG1,'LINK1','r');                % Add long rod as LINK1
smbCreateSG (SG2,'LINK2','g');                % Add short rod as LINK2
smbCreateSG (SG1,'LINK3','y');                % Add long rod as LINK3
smbCreateSG (SG2,'LINK4','c');                % Add short rod as LINK4
smbDrawNow;
```



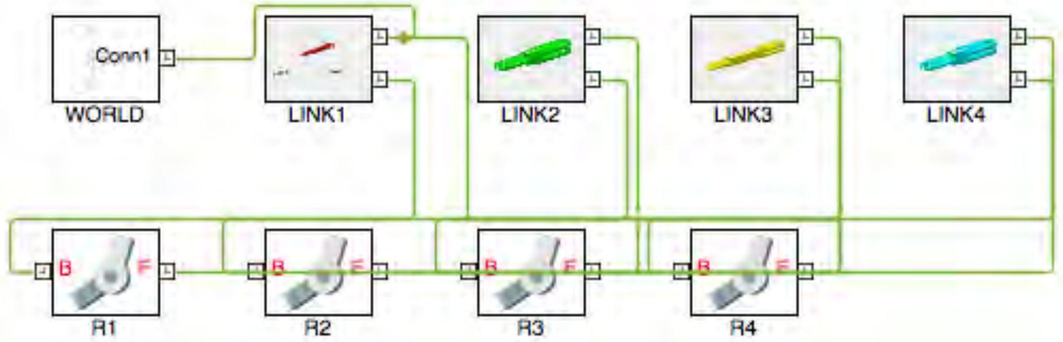
5. Create SimMechanics models for the four joint and connect them with the links

```
smbCreateJoint ('R','R1','LINK1.F','LINK2.B'); % Add a RR Joint
smbCreateJoint ('R','R2','LINK2.F','LINK3.B'); % Add a RR Joint
smbCreateJoint ('R','R3','LINK3.F','LINK4.B'); % Add a RR Joint
smbCreateJoint ('R','R4','LINK4.F','LINK1.B'); % Add a RR Joint
smbDrawNow;
```



6. Connect the base frame of link1 to the world coordinate system

```
smbCreateConnection('WORLD.ORIGIN','LINK1.B'); % Connect Linkage to World Frame
smbDrawNow;
```



7. Run the Simulation of the Simulink/SimMechanics diagram for 1 second

```
smbSimulate(1);          % Simulate for 1 second
```

8. Create a Video of the Simualatin for 5 seconds

```
smbVideoSimulation(5);  % Show a 5 seconds video
```

```
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
```

```
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
```

Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:44:32!
Executed 08-Nov-2018 20:44:34 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
```

```
simulink
video_and_image_blockset
==========================================================================================
==========
```

- *Tim Lueth, tested and compiled on OSX 10.11.6 with Matlab 2016 on 2016-12-09*
- _________, executed and published on 64 Bit PC using Windows with Matlab 2015a on 2015-xx-xx_

Published with MATLAB® R2018a

Tutorial 21: Programmatically Convert Joints into Drives (SimMechanics)

2016-11-19: 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

Motivation for this tutorial: (Originally SolidGeometry 3.0 required)

2. Creating a new SimMechanics System

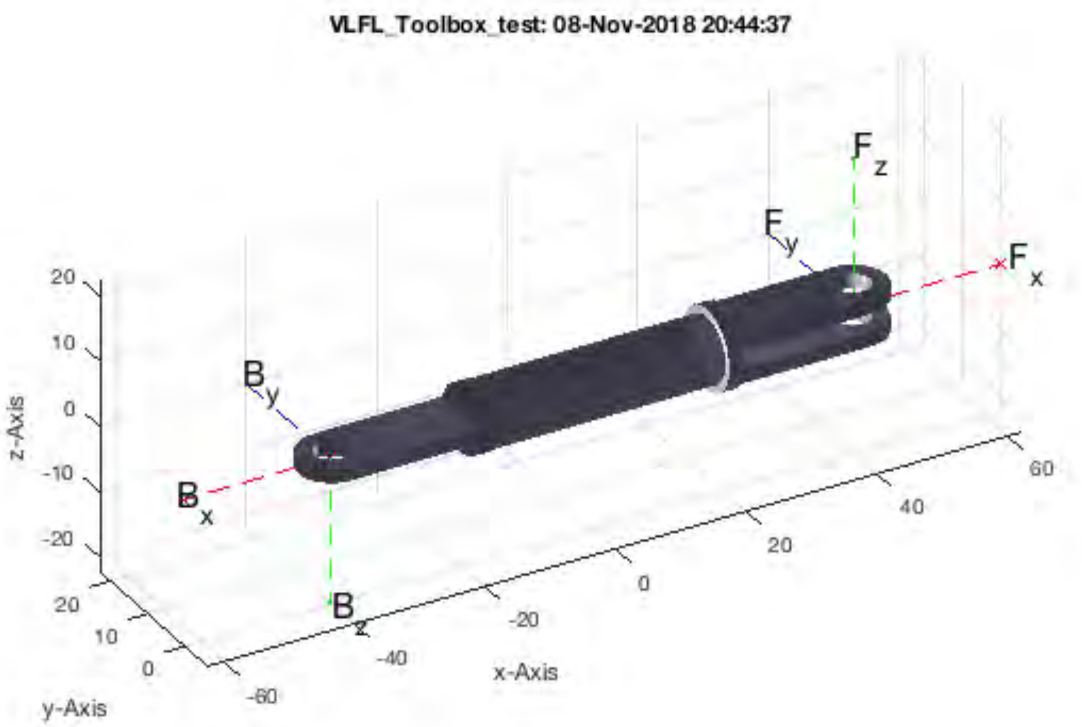
```
smbNewSystem ('SG_LIB_EXP_21');          % Creates the mechansim diagramm
```

```
Creating temporary directory '/Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_21/'
```

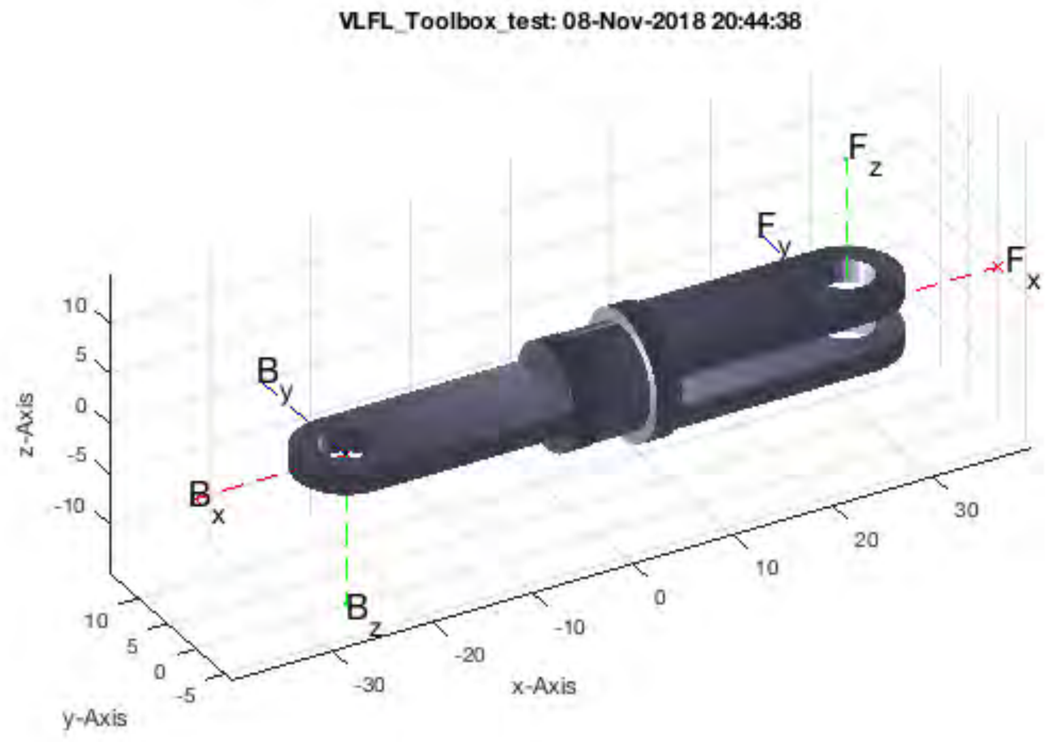
3. Create two links with length 50 and 80 and one or two mounting holes

```
SG1=SGmodelLink(80,'',1,2);              % Creates a long rod with flange
SG2=SGmodelLink(50,'',1,2);              % Creates a short rod with flange
```

```
SGfigure; view(-30,30); axis on;
SGT(SG1);
```

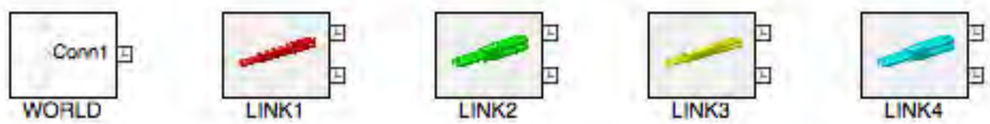



```
SGfigure; view(-30,30); axis on;
SGT (SG2);
```



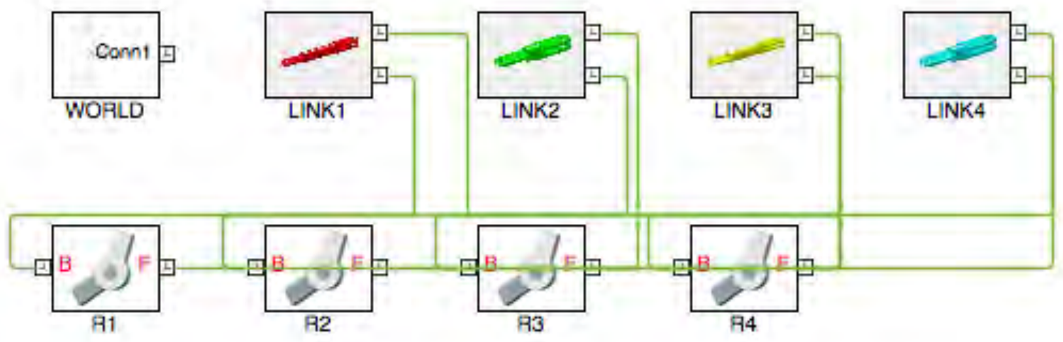
4. Create SimMechanics models for the four links in different colors

```
smbCreateSG (SG1,'LINK1','r');             % Add long rod as LINK1
smbCreateSG (SG2,'LINK2','g');             % Add short rod as LINK2
smbCreateSG (SG1,'LINK3','y');             % Add long rod as LINK3
smbCreateSG (SG2,'LINK4','c');             % Add short rod as LINK4
smbDrawNow;
```



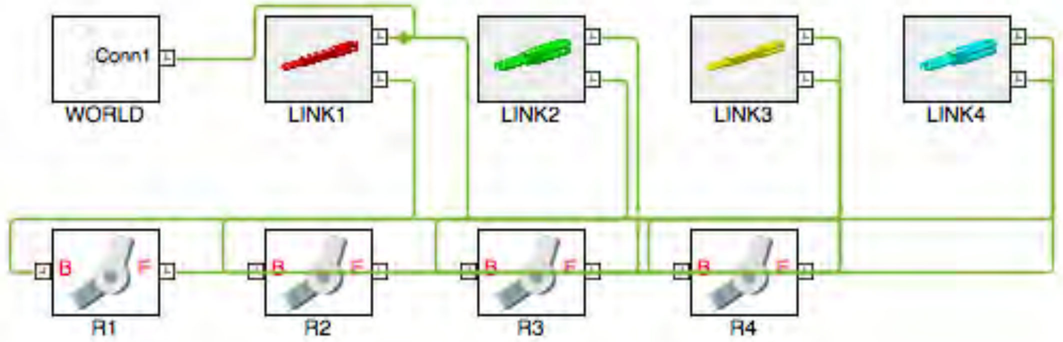
5. Create SimMechanics models for the four joint and connect them with the links

```
smbCreateJoint ('R','R1','LINK1.F','LINK2.B'); % Add a RR Joint
smbCreateJoint ('R','R2','LINK2.F','LINK3.B'); % Add a RR Joint
smbCreateJoint ('R','R3','LINK3.F','LINK4.B'); % Add a RR Joint
smbCreateJoint ('R','R4','LINK4.F','LINK1.B'); % Add a RR Joint
smbDrawNow;
```



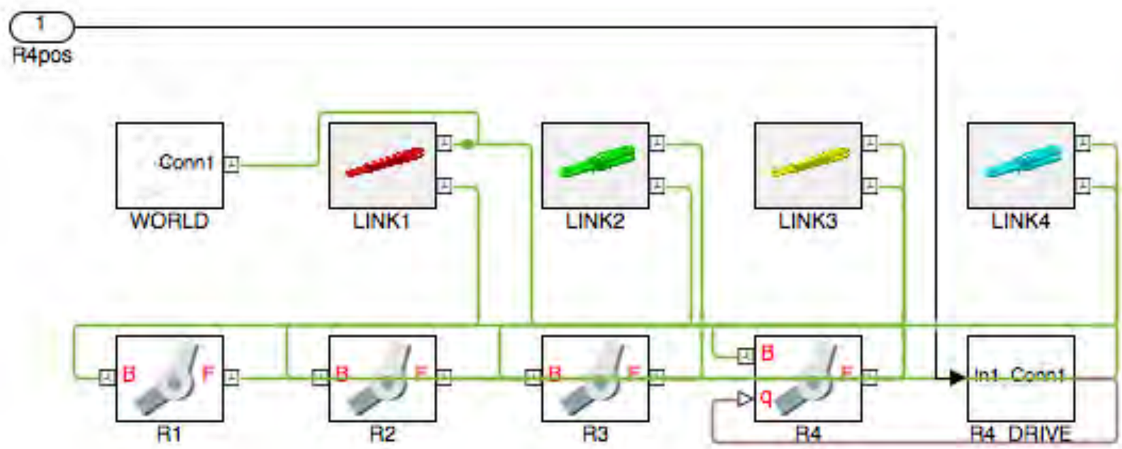
6. Connect the base frame of link1 to the world coordinate system

```
smbCreateConnection('WORLD.ORIGIN','LINK1.B'); % Connect Linkage to World Frame
smbDrawNow;
```



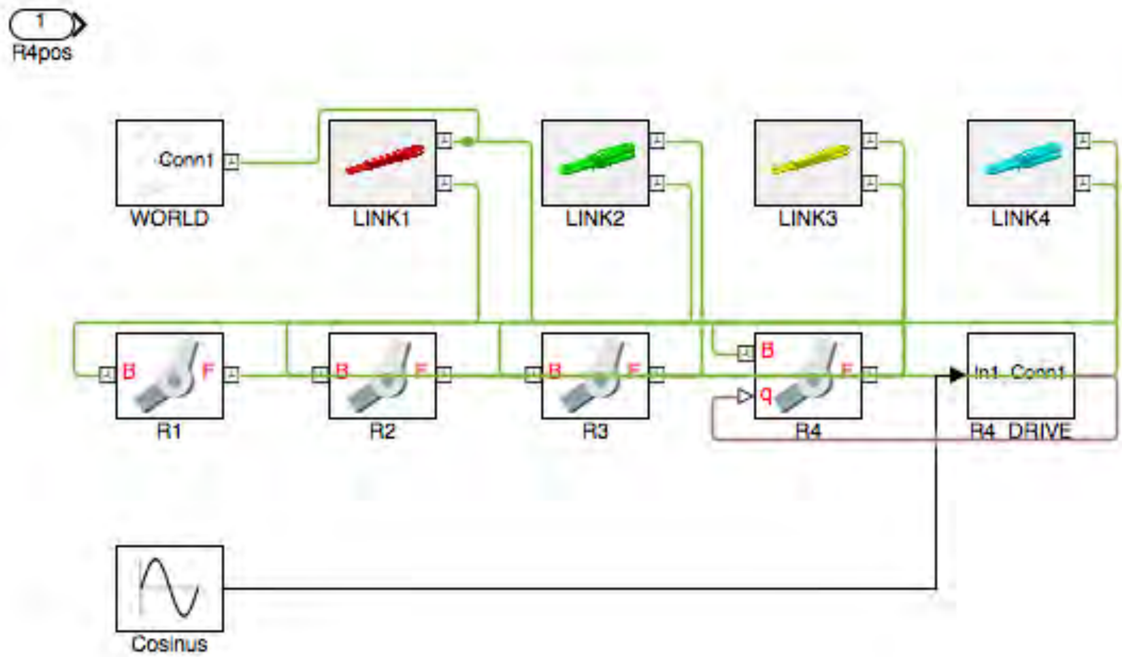
7. Create a SimMechanics model for a motor/drive and use a Cosinus Rotation

```
smbCreateDrive ('R4');                      % Convert Joint R4 into a Drive
smbDrawNow;
```



8. Create a Simulink models for a cosinus signal

```
smbCreateSineWave ('Cosinus','R4_DRIVE/1');     % Connect a Sinus Generator to Drive
smbDrawNow;
```

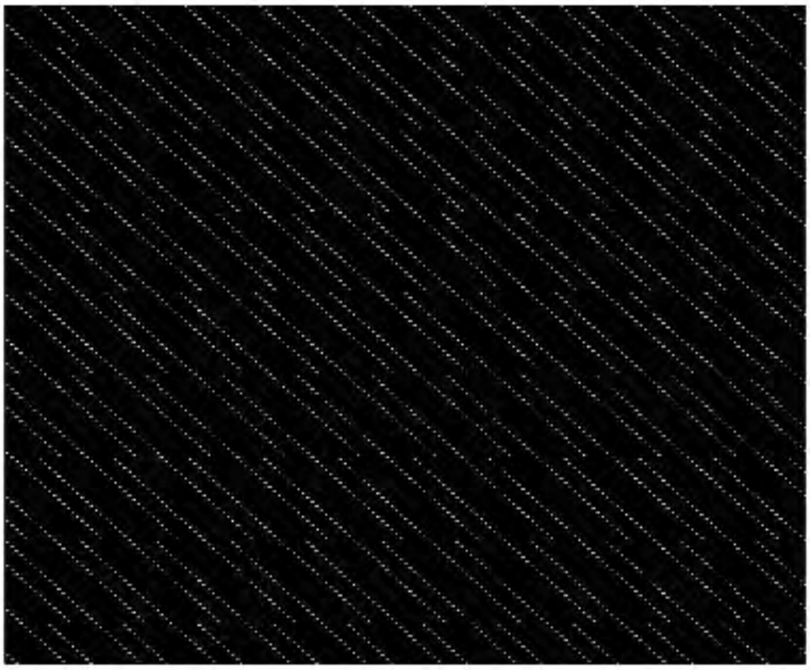


9. Create a Video of the Simualati0n for 10 seconds

```
smbVideoSimulation;  % Show a 5 seconds video
```

```
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
```

```
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
```



Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:45:07!
Executed 08-Nov-2018 20:45:09 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
```

===
==========

- *Tim Lueth, tested and compiled on OSX 10.11.6 with Matlab 2016 on 2016-12-09*
- _________, executed and published on 64 Bit PC using Windows with Matlab 2015a on 2015-xx-xx_

Published with MATLAB® R2018a

Tutorial 22: Adding Simulink Signals to Record Frame Movements

2016-12-18: 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:

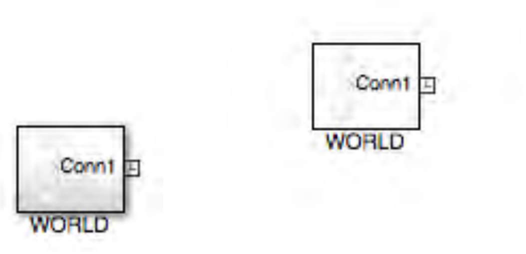
- Tutorial 01: First Steps Using the VLFL-Toolbox for Solid Object Design
- Tutorial 02: Using the VLFL-Toolbox for STL-File Export and Import
- Tutorial 03: Closed 2D Contours and Boolean Operations in 2D
- Tutorial 04: 2½D Design Using Boolean Operators on Closed Polygon Lists (CPL)
- Tutorial 05: 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

Motivation for this tutorial: (Originally SolidGeometry 3.1 required)

2. Creating a new SimMechanics System

```
smbNewSystem ('SG_LIB_EXP_22');          % Creates the mechansim diagramm
smbDrawNow;
```

```
Creating temporary directory '/Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_22/'
```



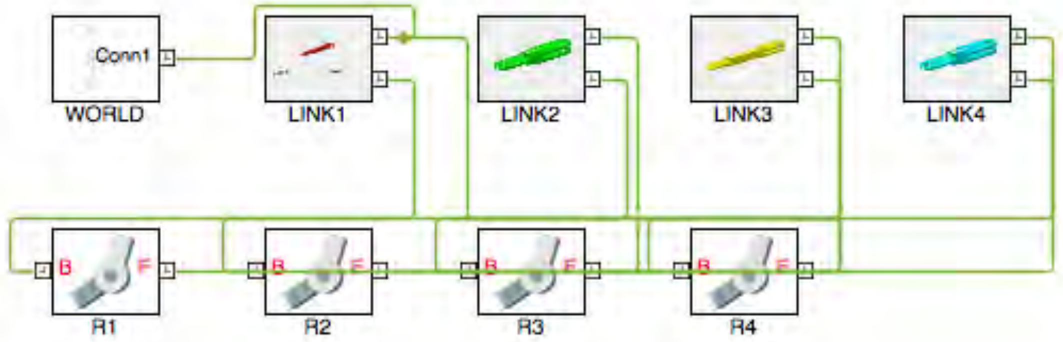
3. Create two links with length 50 and 80 and one or two mounting holes

```
SG1=SGmodelLink(80,'',1,2);              % Creates a long rod with flange
SG2=SGmodelLink(50,'',1,2);              % Creates a short rod with flange
```

4. Create SimMechanics models for the four links and four joints in different colors

```
smbCreateSG (SG1,'LINK1','r');            % Add long rod as LINK1
smbCreateSG (SG2,'LINK2','g');            % Add short rod as LINK2
smbCreateSG (SG1,'LINK3','y');            % Add long rod as LINK3
```

```
smbCreateSG (SG2,'LINK4','c');                % Add short rod as LINK4
smbCreateJoint ('R','R1','LINK1.F','LINK2.B'); % Add a RR Joint
smbCreateJoint ('R','R2','LINK2.F','LINK3.B'); % Add a RR Joint
smbCreateJoint ('R','R3','LINK3.F','LINK4.B'); % Add a RR Joint
smbCreateJoint ('R','R4','LINK4.F','LINK1.B'); % Add a RR Joint
smbCreateConnection('WORLD.ORIGIN','LINK1.B'); % Connect Linkage to World Frame
smbDrawNow;
```



5. Create a video of the movements

smbVideoSimulation(3); % Show a 3 seconds video

6. Analyze the simulation for 3 Seconds

The result of a simulation is a strucutre that contains SimMultiBody states (xout) and recorded Simulink signals (sim). If there are no Simulink signals, sout is empty.

```
simOut=smbSimulate(3)
```

```
simOut = 

  Simulink.SimulationOutput:
                 simlog: [1x1 simscape.logging.Node]
                   tout: [241x1 double]
                   xout: [1x1 Simulink.SimulationData.Dataset]

     SimulationMetadata: [1x1 Simulink.SimulationMetadata]
           ErrorMessage: [0x0 char]
```

The states contain the parameter = angles/velocity of the joints

```
xout = simOut.get('xout')
```

```
xout =
```

```
Simulink.SimulationData.Dataset 'xout' with 8 elements

                       Name                   BlockPath
                       ____________________   ________________

    1  [1x1 State]     SG_LIB_EXP_22.R1.Rz.q  SG_LIB_EXP_22/R1
    2  [1x1 State]     SG_LIB_EXP_22.R1.Rz.w  SG_LIB_EXP_22/R1
    3  [1x1 State]     SG_LIB_EXP_22.R2.Rz.q  SG_LIB_EXP_22/R2
    4  [1x1 State]     SG_LIB_EXP_22.R2.Rz.w  SG_LIB_EXP_22/R2
    5  [1x1 State]     SG_LIB_EXP_22.R3.Rz.q  SG_LIB_EXP_22/R3
    6  [1x1 State]     SG_LIB_EXP_22.R3.Rz.w  SG_LIB_EXP_22/R3
    7  [1x1 State]     SG_LIB_EXP_22.R4.Rz.q  SG_LIB_EXP_22/R4
    8  [1x1 State]     SG_LIB_EXP_22.R4.Rz.w  SG_LIB_EXP_22/R4

  - Use braces { } to access, modify, or add elements using index.
```

There is no Simulink signals yet

```
sout = simOut.get('sout')
```

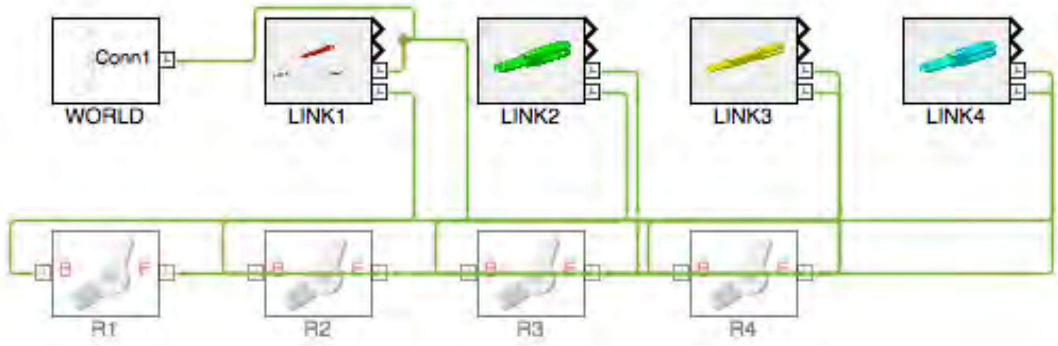
```
sout =

     []
```

7. Create Simulink signals for all the frames of the four links

```
smbAddFrameSensor ('LINK1.RF');
smbAddFrameSensor ('LINK2.RF');
smbAddFrameSensor ('LINK3.RF');
smbAddFrameSensor ('LINK4.RF');
```

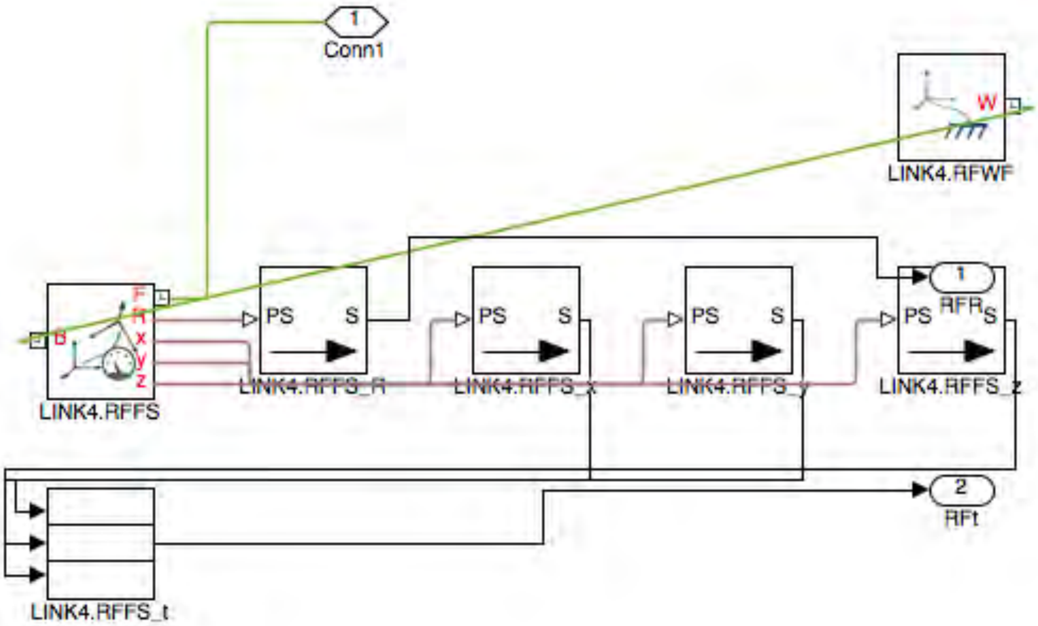
Now, all links have simulink signals and signal output for R and T of the reference frame

```
smbDrawNow;
```



The model of link4 is extendend by a transformation sensor

```
smbDrawNow ('LINK4.RF_T');
```

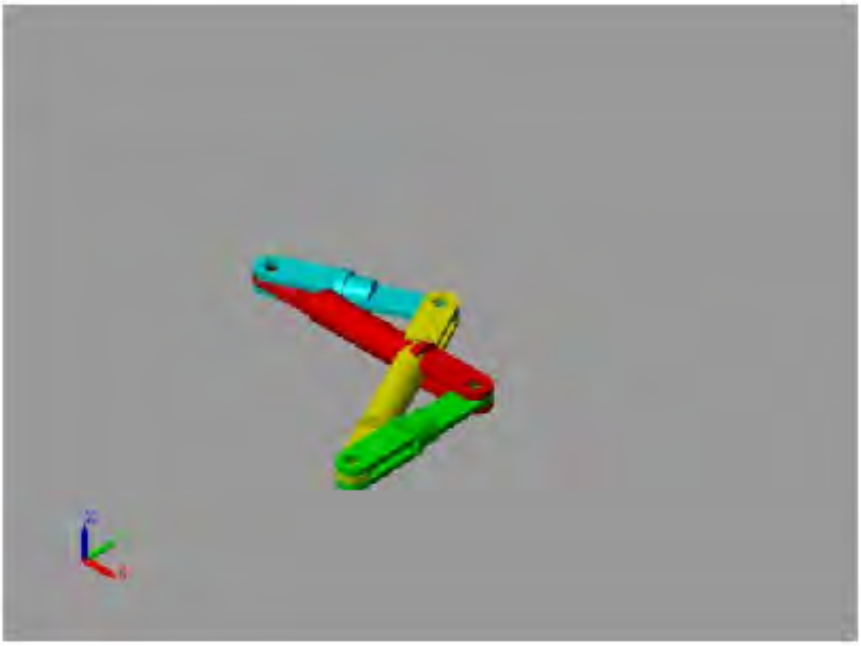


8. Simulate and record those signals too

```
simOut=smbSimulate(3)
smbVideoSimulation(3);
```

```
simOut = 

  Simulink.SimulationOutput:
                simlog: [1x1 simscape.logging.Node]
```

```
                    sout: [1x1 Simulink.SimulationData.Dataset]
                    tout: [241x1 double]
                    xout: [1x1 Simulink.SimulationData.Dataset]

      SimulationMetadata: [1x1 Simulink.SimulationMetadata]
            ErrorMessage: [0x0 char]

.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
```



The states contain the parameter = angles/velocity of the joints

```
xout = simOut.get('xout')
```

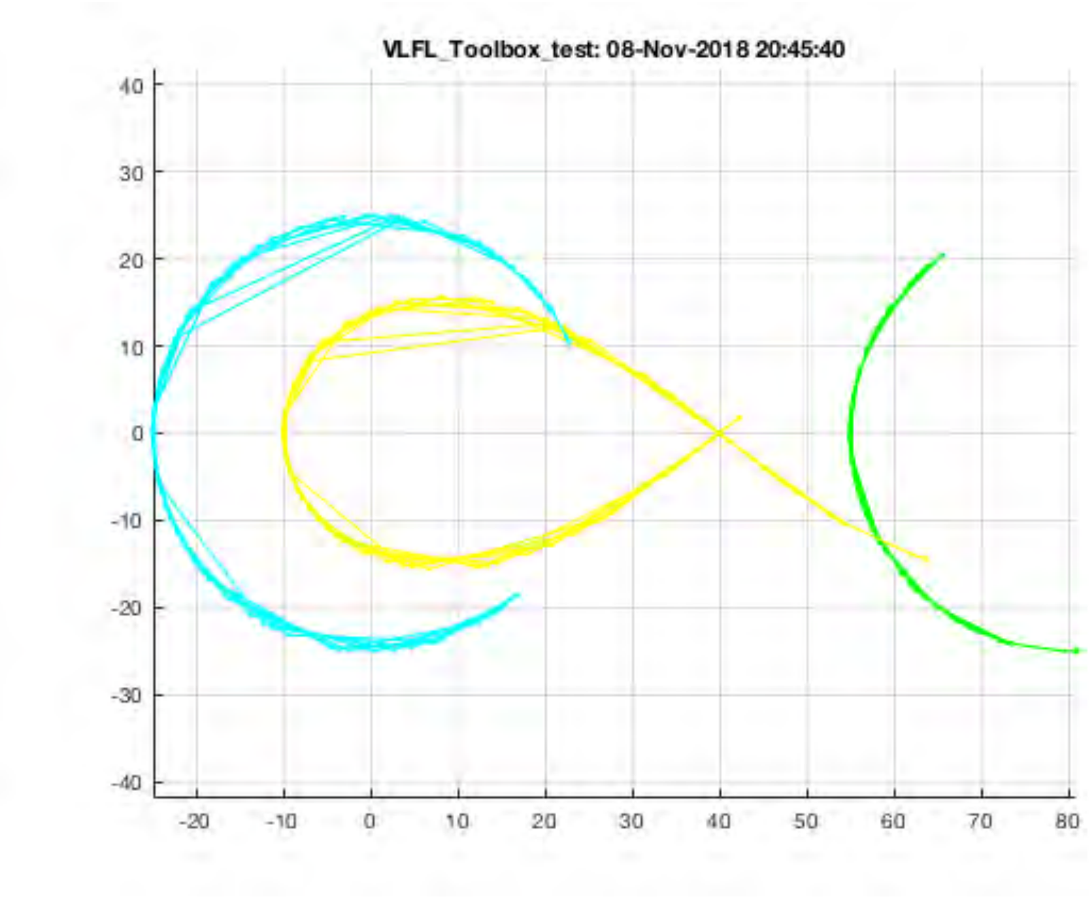
```
xout =

Simulink.SimulationData.Dataset 'xout' with 8 elements

                        Name                  BlockPath
                        ____________________  ________________
```

```
     1  [1x1 State]      SG_LIB_EXP_22.R1.Rz.q  SG_LIB_EXP_22/R1
     2  [1x1 State]      SG_LIB_EXP_22.R1.Rz.w  SG_LIB_EXP_22/R1
     3  [1x1 State]      SG_LIB_EXP_22.R2.Rz.q  SG_LIB_EXP_22/R2
     4  [1x1 State]      SG_LIB_EXP_22.R2.Rz.w  SG_LIB_EXP_22/R2
     5  [1x1 State]      SG_LIB_EXP_22.R3.Rz.q  SG_LIB_EXP_22/R3
     6  [1x1 State]      SG_LIB_EXP_22.R3.Rz.w  SG_LIB_EXP_22/R3
     7  [1x1 State]      SG_LIB_EXP_22.R4.Rz.q  SG_LIB_EXP_22/R4
     8  [1x1 State]      SG_LIB_EXP_22.R4.Rz.w  SG_LIB_EXP_22/R4

  - Use braces { } to access, modify, or add elements using index.
```

TheSimulink signals are related to the reference rotation and translation

```
sout = simOut.get('sout')
T1=smbTofSimOut(simOut,'LINK1.RF'); VL1=squeeze(T1(1:3,4,:))';
T2=smbTofSimOut(simOut,'LINK2.RF'); VL2=squeeze(T2(1:3,4,:))';
T3=smbTofSimOut(simOut,'LINK3.RF'); VL3=squeeze(T3(1:3,4,:))';
T4=smbTofSimOut(simOut,'LINK4.RF'); VL4=squeeze(T4(1:3,4,:))';
SGfigure; axis on; view(0,90); grid on;
VLplot(VL1,'r.-');
VLplot(VL2,'g.-');
VLplot(VL3,'y.-');
VLplot(VL4,'c.-');
drawnow;
```

```
sout = 

Simulink.SimulationData.Dataset 'sout' with 8 elements

                         Name                                BlockPath                          
                         __________________________________  ______________________________ 

     1  [1x1 Signal]     SG_LIB_EXP_22/LINK1/LINK1.RF_T.RFR  SG_LIB_EXP_22/LINK1/LINK1.RF_T
     2  [1x1 Signal]     SG_LIB_EXP_22/LINK1/LINK1.RF_T.RFt  SG_LIB_EXP_22/LINK1/LINK1.RF_T
     3  [1x1 Signal]     SG_LIB_EXP_22/LINK2/LINK2.RF_T.RFR  SG_LIB_EXP_22/LINK2/LINK2.RF_T
     4  [1x1 Signal]     SG_LIB_EXP_22/LINK2/LINK2.RF_T.RFt  SG_LIB_EXP_22/LINK2/LINK2.RF_T
     5  [1x1 Signal]     SG_LIB_EXP_22/LINK3/LINK3.RF_T.RFR  SG_LIB_EXP_22/LINK3/LINK3.RF_T
     6  [1x1 Signal]     SG_LIB_EXP_22/LINK3/LINK3.RF_T.RFt  SG_LIB_EXP_22/LINK3/LINK3.RF_T
     7  [1x1 Signal]     SG_LIB_EXP_22/LINK4/LINK4.RF_T.RFR  SG_LIB_EXP_22/LINK4/LINK4.RF_T
     8  [1x1 Signal]     SG_LIB_EXP_22/LINK4/LINK4.RF_T.RFt  SG_LIB_EXP_22/LINK4/LINK4.RF_T

  - Use braces { } to access, modify, or add elements using index.
```



Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:45:41!
Executed 08-Nov-2018 20:45:43 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
============================================================================================
==========
```

- *Tim Lueth, tested and compiled on OSX 10.11.6 with Matlab 2016b on 2016-12-18*
- __________, executed and published on 64 Bit PC using Windows with Matlab 2015a on 2015-xx-xx_

Published with MATLAB® R2018a

Tutorial 23: Automatic Creation of a Missing Link and 3D Print of a Complete Model

2016-12-19: 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)

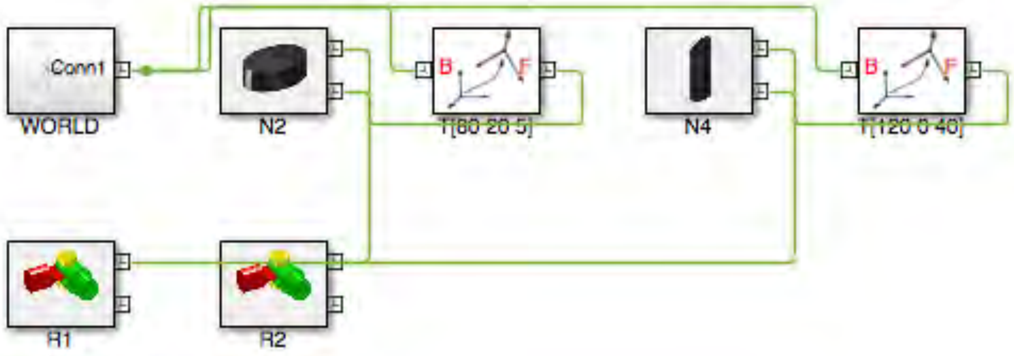
Motivation for this tutorial: (Originally SolidGeometry 3.1 required)

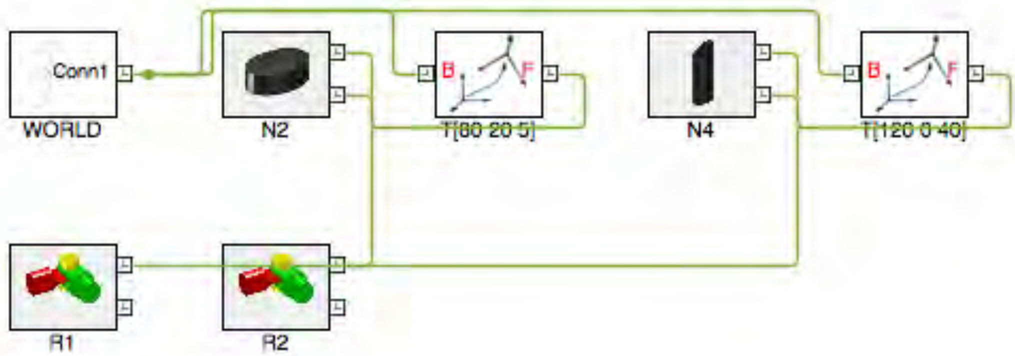
2. Open a system and create several fixed nodes and attach revolute joints

```
function VLFL_EXP23
```

```
smbNewSystem ('SG_LIB_EXP_23');

smbCreateSGNode ([80  20  5],'N2');
smbCreateSGNode ([120 0 40],'N4','',rot(0,-pi/8,0));
A=SGmodelJoint('R',pi/2);
smbCreateSGJoint('R','R1', A,'N4.F');
smbCreateSGJoint('R','R2',A,'N2.F');
smbDrawNow;
```

```
Creating temporary directory '/Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_23/'
```



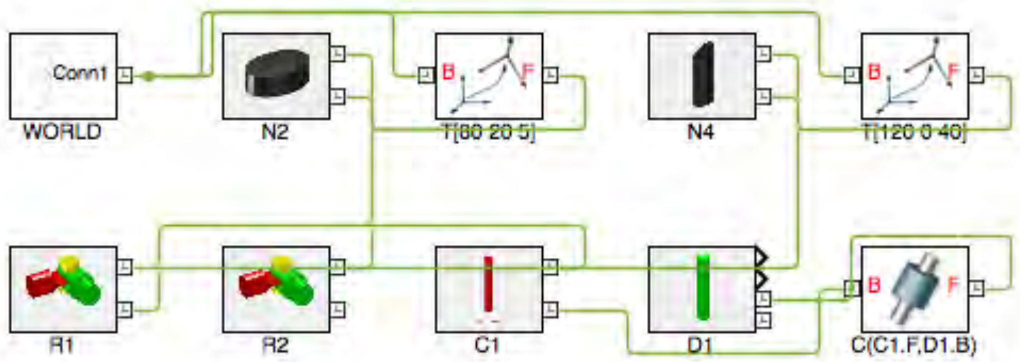


3. Create a cylindric joint from two solids an attach it to revolute joint

```
Ro=5;
Ri=3;
slot=0.3;

C1=SGofCPLz([PLcircle(Ro);NaN NaN;PLcircle(Ri+slot)],30);
% C1=SGTset(C1,'B',TofSG(C1,'bottom','roty',pi));
C1=SGTset(C1,'B',TofSG(C1,'incenter','right',-1,'roty',pi/2));
C1=SGTset(C1,'F',TofSG(C1,'bottom'));
smbCreateSG(C1,'C1','r','R1_M');
D1=SGofCPLz(PLcircle(3),30);
D1=SGTset(D1,'B',TofSG(D1,'incenter'));
D1=SGTset(D1,'F',TofSG(D1,'top'));
```

4. Attach two frame sensor to record the movement of the falling cylinder

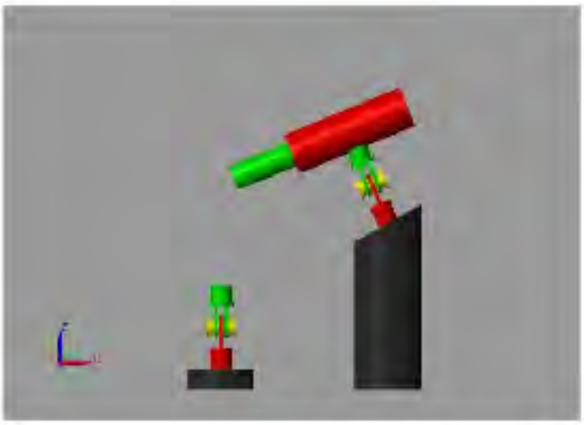
```
smbCreateSG(D1,'D1','g');
smbCreateConnection('C1.F','D1.B','C');
smbAddFrameSensor('R2_M.F');
smbAddFrameSensor('D1.F');
smbDrawNow;
```



5. Show the Simulation

```
simOut=smbSimulate(0.1);
smbVideoSimulation(1);
```

```
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
```



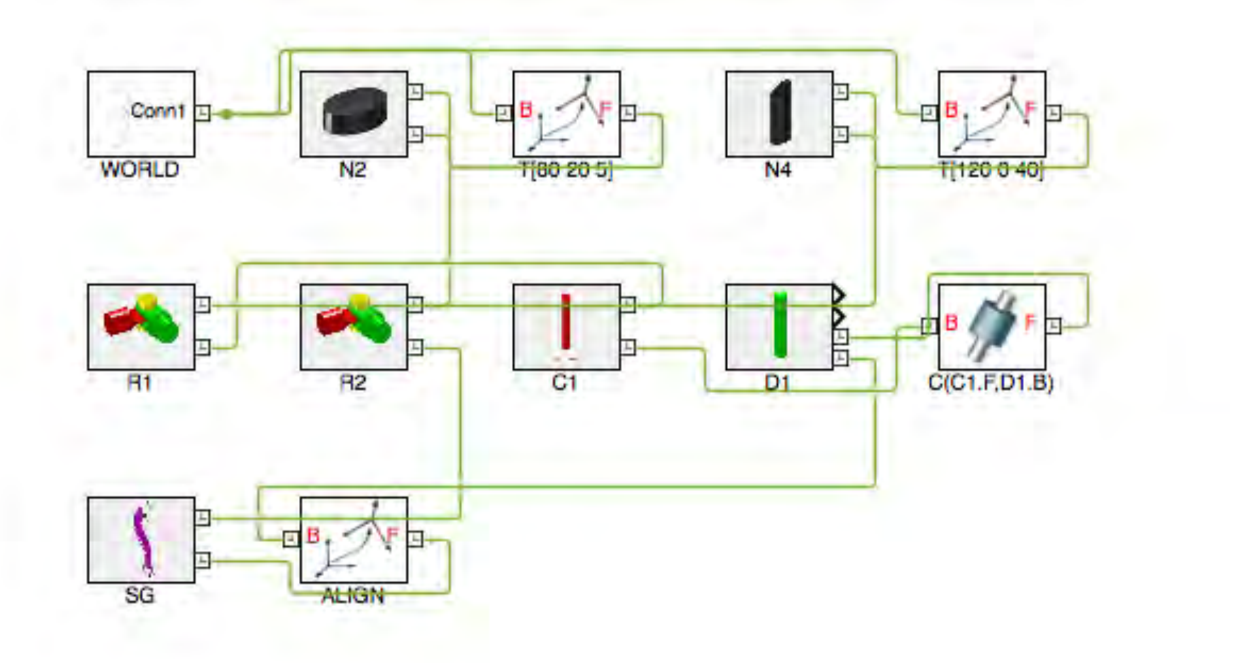
6. Now create a solid between the revolute joint and cylindric joint

```
[T,ta]=smbTofSimOut(simOut,'R2_M.F'); T1=squeeze(T(:,:,1));
[T,tb]=smbTofSimOut(simOut,'D1.F');   T2=squeeze(T(:,:,1));
SG=SGof2T(T1,T2*TofR(rot(0,pi,0)),'',4); % Radius 4
SGTplot(SG);
```

```
VLradialEdges: Radius 4.00 reduced to 2.73
```

7. Now connect the new solid in the model

```
smbCreateSG(SG,'SG','m');
smbCreateConnection('R2_M','SG.B');
smbCreateConnection('D1.F','SG.F','align');
smbDrawNow;
```

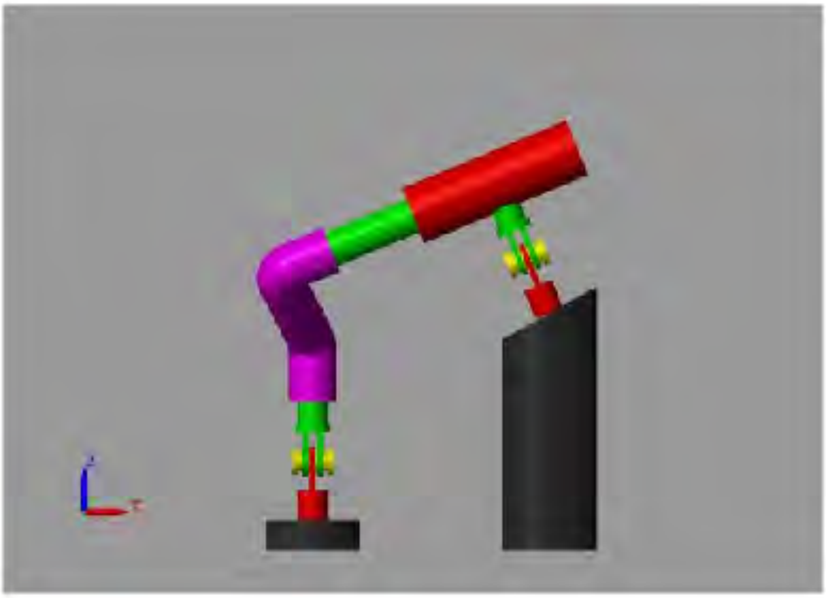


8. Show the Simulation: The Mechnism has no Movement anymore

```
smbVideoSimulation(1);
```

```
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
```

```
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
```



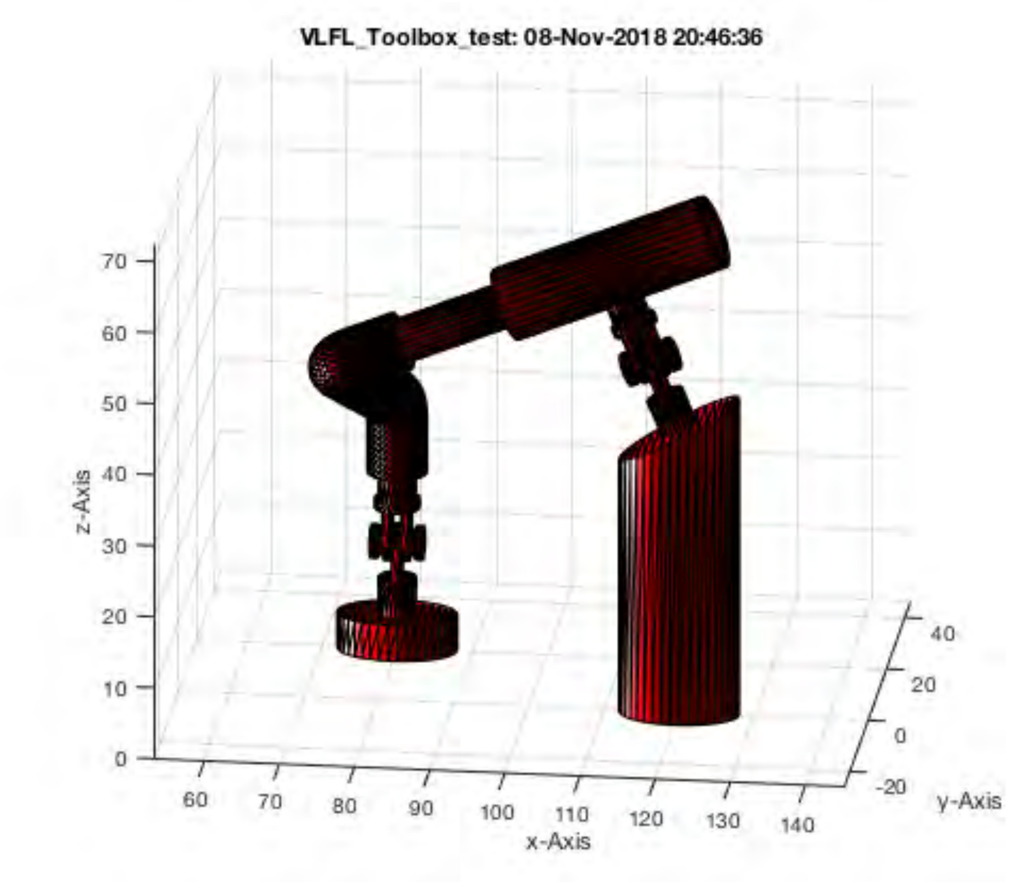
9. Now Create a Solid Model of Movement Status at Time = 0.1 Seconds

```
SG=smbFullModelSimulation(0.1);
SGfigure; SGplot(SG); view (7,20);
```

```
CREATING A FULL SOLID-MOVEMENT SIMULATION-MODEL 'SG_LIB_EXP_23' THAT RUNS At LEAST 0.10 SEC
ONDS
===========================================================================================
========
Adding frame sensors for all solids of the model
Add frame sensors for 'C1.SG'
Add frame sensors for 'D1.SG'
Add frame sensors for 'N2.SG'
Add frame sensors for 'N4.SG'
Add frame sensors for 'R1.FIX1.SG'
Add frame sensors for 'R1_M.SG'
Add frame sensors for 'R1_S.SG'
Add frame sensors for 'R2.FIX1.SG'
Add frame sensors for 'R2_M.SG'
Add frame sensors for 'R2_S.SG'
Add frame sensors for 'SG.SG'
===========================================================================================
========

simOut =

  Simulink.SimulationOutput:
```

```
                  simlog: [1x1 simscape.logging.Node]
                    sout: [1x1 Simulink.SimulationData.Dataset]
                    tout: [51x1 double]
                    xout: [1x1 Simulink.SimulationData.Dataset]

      SimulationMetadata: [1x1 Simulink.SimulationMetadata]
            ErrorMessage: [0x0 char]

LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_23/sbm_temp_C1.st
l
Header:
Number of facets: 232
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_23/sbm_temp_D1.st
l
Header:
Number of facets: 96
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_23/sbm_temp_N2.st
l
Header:
Number of facets: 156
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_23/sbm_temp_N4.st
l
Header:
Number of facets: 156
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_23/sbm_temp_R1.FI
X1.stl
Header:
Number of facets: 240
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_23/sbm_temp_R1_M.
stl
Header:
Number of facets: 456
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_23/sbm_temp_R1_S.
stl
Header:
Number of facets: 448
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_23/sbm_temp_R2.FI
X1.stl
Header:
Number of facets: 240
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_23/sbm_temp_R2_M.
stl
Header:
Number of facets: 456
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_23/sbm_temp_R2_S.
stl
Header:
Number of facets: 448
```

```
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_23/sbm_temp_SG.st
l
Header:
Number of facets: 3824
0..

CREATED A SOLID GEOMETRY OF THE FULL SIMULATION-MODEL 'SG_LIB_EXP_23' AT TIME: 0.10 SECONDS
==========================================================================================
========
```



Write the STL file on disk for 3D printing

```
SGwriteSTL(SG);
```

Final Remarks

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:46:37!
Executed 08-Nov-2018 20:46:39 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
```

```
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
===========================================================================================
==========
```

Published with MATLAB® R2018a

Tutorial 24: Automatic Creation of a Joint Limitations

2016-12-25: 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

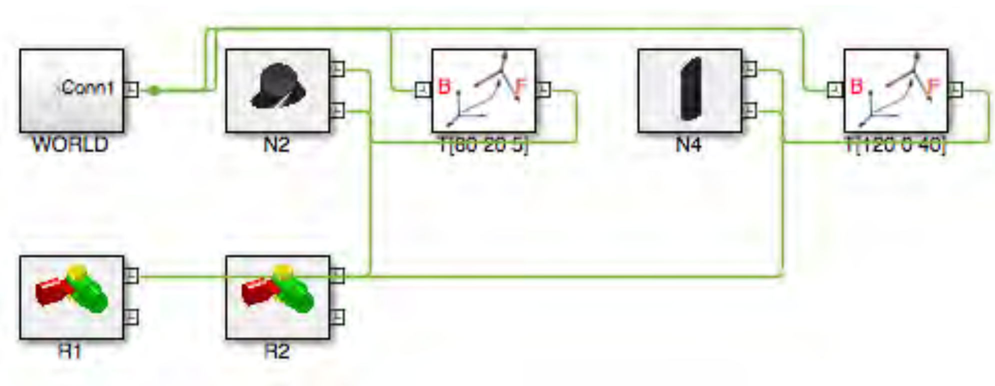
Motivation for this tutorial: (Originally SolidGeometry 3.2 required)

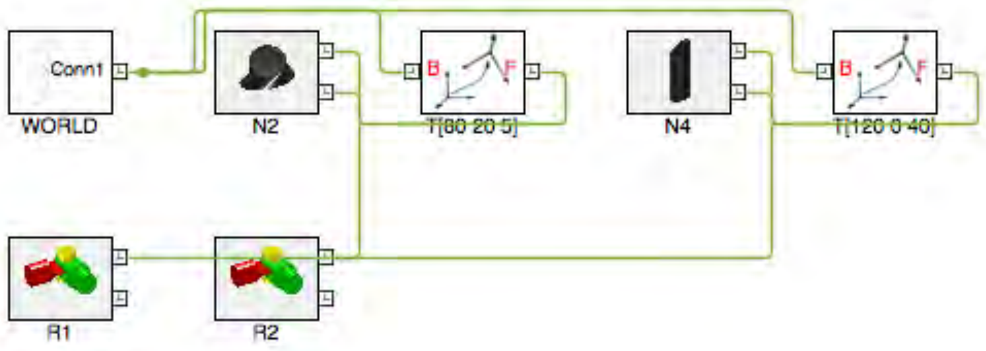
2. Open a system and create several fixed nodes and attach revolute joints

function VLFL_EXP24

```
smbsys='SG_LIB_EXP_24';
smbNewSystem (smbsys);

smbCreateSGNode ([80  20  5],'N2','',rot(0,0,pi/3));
smbCreateSGNode ([120 0 40],'N4','',rot(0,-pi/8,0));
A=SGmodelJoint('R',pi/2);
smbCreateSGJoint('R','R1', A,'N4.F');
smbCreateSGJoint('R','R2',A,'N2.F');
smbDrawNow;
```

```
Creating temporary directory '/Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_24/'
```



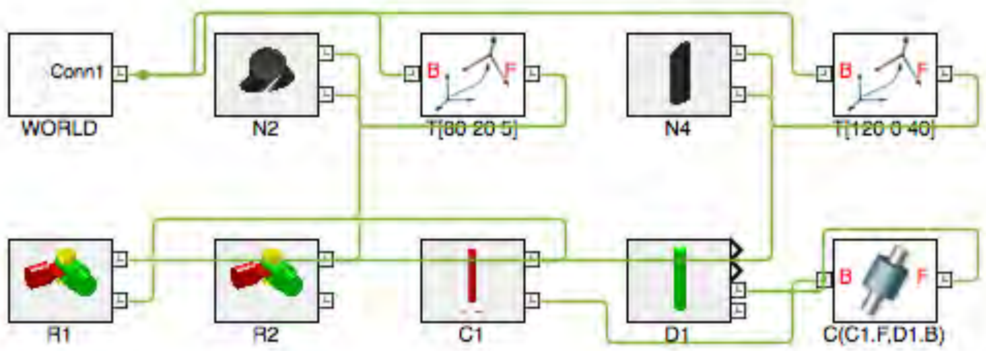


3. Create a cylindric joint from two solids an attach it to revolute joint

```
Ro=5;
Ri=3;
slot=0.3;

C1=SGofCPLz([PLcircle(Ro);NaN NaN;PLcircle(Ri+slot)],30);
% C1=SGTset(C1,'B',TofSG(C1,'bottom','roty',pi));
C1=SGTset(C1,'B',TofSG(C1,'incenter','right',-1,'roty',pi/2));
C1=SGTset(C1,'F',TofSG(C1,'bottom'));
smbCreateSG(C1,'C1','r','R1_M');
D1=SGofCPLz(PLcircle(3),30);
D1=SGTset(D1,'B',TofSG(D1,'incenter'));
D1=SGTset(D1,'F',TofSG(D1,'top'));
```

4. Attach two frame sensor to record the movement of the falling cylinder

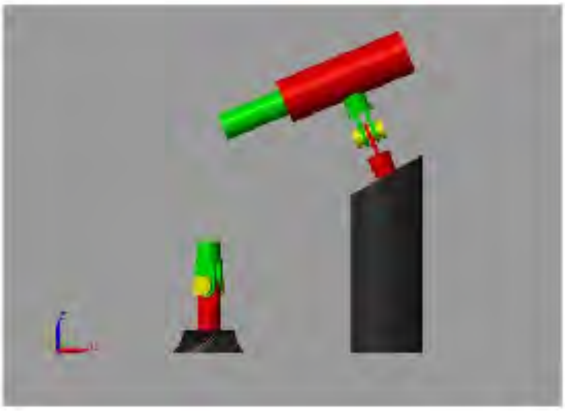
```
smbCreateSG(D1,'D1','g');
smbCreateConnection('C1.F','D1.B','C');
smbAddFrameSensor('R2_M.F');
smbAddFrameSensor('D1.F');
smbDrawNow;
```



5. Show the Simulation

```
simOut=smbSimulate(0.1);
smbVideoSimulation(4);
```

```
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
```

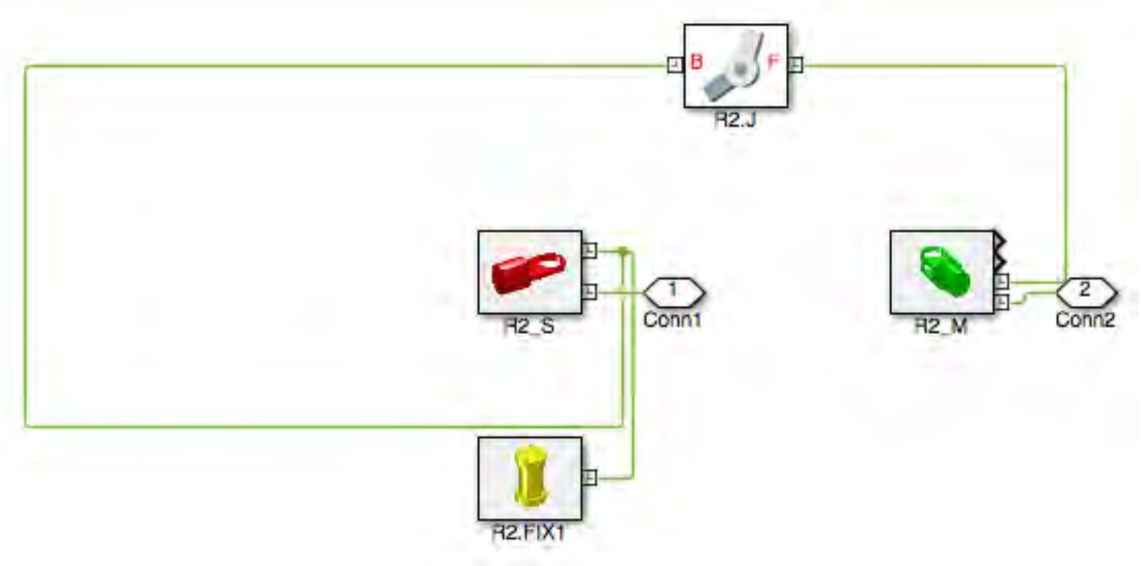


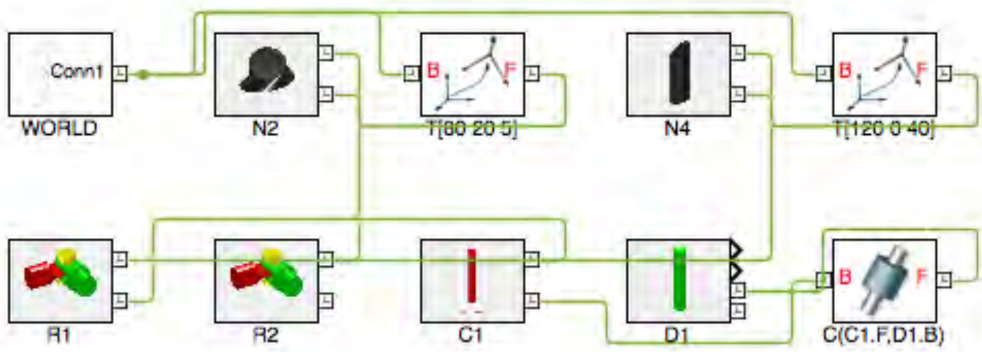
6. Install additional block funktion for joint restrictions

```
smbPSLibInstall
open_system(smbPSBlockname);
open_system(smbsys,'tab');
open_system(smbWhich('R2'),'tab');smbDrawNow;
```

```
Create /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_24/+mechPS_Tim_Lueth/PS_force_friction
_rot.ssc
Create /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_24/+mechPS_Tim_Lueth/PS_force_friction
_rot.svg
Create /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_24/+mechPS_Tim_Lueth/PS_force_hardstop
_rot.ssc
Create /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_24/+mechPS_Tim_Lueth/PS_force_hardstop
_rot.svg
Create /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_24/+mechPS_Tim_Lueth/PS_force_friction
_trans.ssc
```

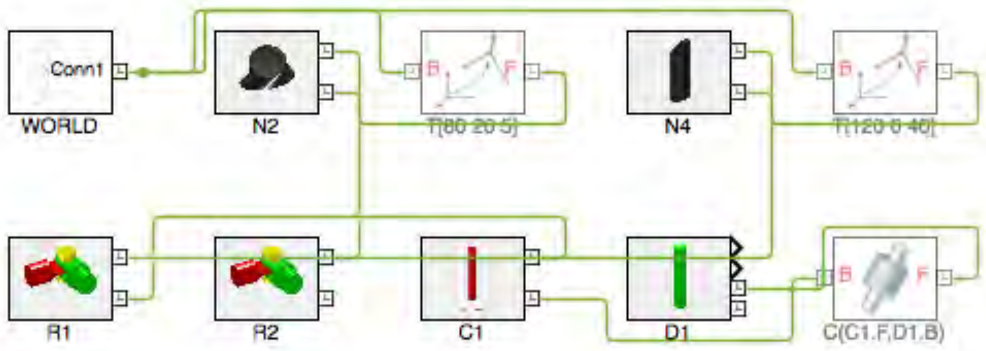
```
Create /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_24/+mechPS_Tim_Lueth/PS_force_friction
_trans.svg
Create /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_24/+mechPS_Tim_Lueth/PS_force_hardstop
_trans.ssc
Create /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_24/+mechPS_Tim_Lueth/PS_force_hardstop
_trans.svg
Generating Simulink library 'mechPS_Tim_Lueth_lib' in the current directory '/Users/lueth/Desk
top/Toolbox_test/tmp_SG_LIB_EXP_24' ...
```





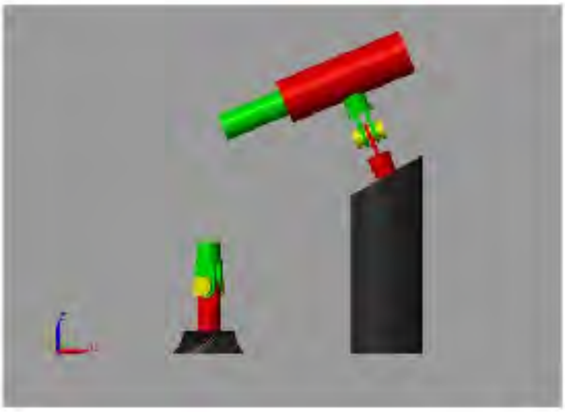
7. Create a stopp joint and copy all connections of an existing joint

```
smbCreateStopJointR ('R2stop.J',[-pi/2 +pi/2]);
smbCopyConnections  ('R2.J','R2stop.J');
smbDrawNow;
```



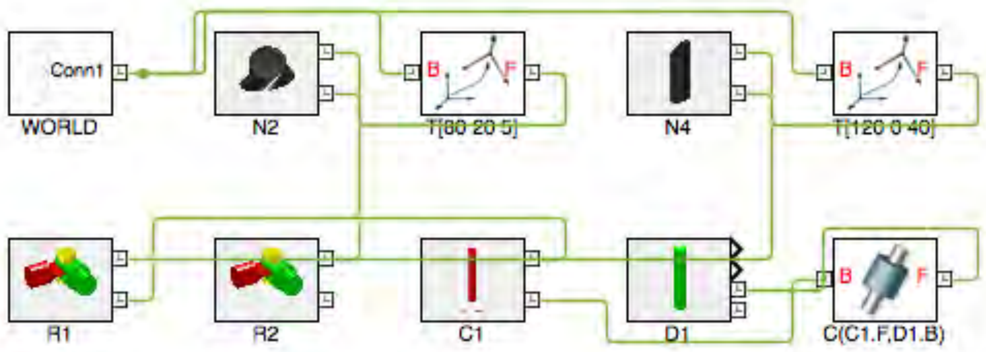

```
smbVideoSimulation(4);
```

```
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
```

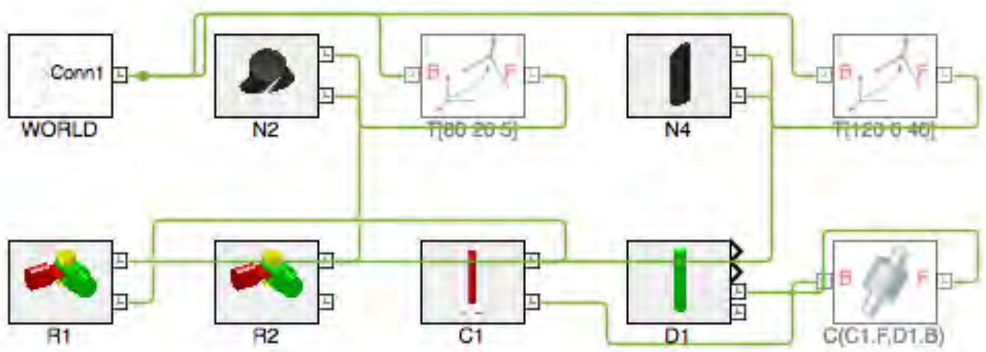


8. Create a stopp joint and replace an existing joint

```
delete_block(smbWhich('R2stop.J'));
smbDeleteUnconnectedLines;
smbDrawNow;
```




```
smbCreateStopJointR ('R2new.J',[-pi/2 +pi/2]);
smbCopyConnections  ('R2.J','R2new.J','replace');
smbDrawNow;
```



9. Final Remarks

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:47:41!
Executed 08-Nov-2018 20:47:43 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a MACI
64
======================================== Used Matlab products: ===============================
=======
antenna_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
```

```
simscape
simulink
video_and_image_blockset
=============================================================================================
=======
```

Published with MATLAB® R2018a

Tutorial 25: Automatic Creation of Video Titels, Endtitels and Textpages

2017-01-01: 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 3.2 required)

The creation of videos of the simulated multi body system, created with the SimMultiBody (2nd Generation) is essential for documentation of the results. Nevertheless, without video titles and end titles and some pages with text description, the videos cannot fulfill their purpose. Therefor it is essential also, to create text for explaining content and make comments on authors and creation date. Therefor we will show some examples:

- imageVideoFrames(xy,ptime); cuts images out of a video a defined positions
- imageVideoTitle(xy,STitles,cols,ptime), creates a 2 second title page video
- imageVideoEndtitle(xy,ETitles,cols), creates a 1 second end title page video
- imageVideoTextPage(xy,ETitles,cols), creates a 2 second text page video
- imageVideoWrite (v,I,t), creates a video by repeating an image t frames
- videoCopyFrames(v,vr), copies a video content into another video (no sound)
- videoCopyCutMovies (WName,RName,style), complex cutting function

3. Create a video clip for a text title

It is possible to start by defining the size of the video titles. If the function is called without am output parameter, automatically a video clip is created with this image

```
I=imageVideoTitle([640 480],{'Video Titel','$date'});
imshow(I.cdata);
```

It is also possible to name an existing video or to select it during function execution to define the size from an existing video. In addition, the background color and text color can be defined (in future also font name and font size), and furthermore times for creating a snapshot that becomes part of the title page. I=imageVideoTitle('',{'Video Titel','SubTitle','Author','$date'},['w' 'r'],[0 1 3]);

```
close all; figure; imshow(I.cdata);
```

Calling the function without an output parameter creates a small video clip in the desktopdir. This can later be used to add the original video including text pages, title page and end title page to a video clip that has sound. Matlab in 2016b does not support sound videos on MAC, only on PC platforms. imageVideoTitle('',{'Video Titel','SubTitle','Author','$date'},['w' 'r'],[0 1 3]);

4. Create an end title video clip

In similar manner, it is possible to define end titles. The creation date is added automatically. Please use the title page if you want to clarify the result was achieved earlier.

```
I=imageVideoEndtitle([640 480],{'Technical University of Munich','','www.tum.de'});
imshow(I.cdata);
imageVideoEndtitle([640 480],{'Technical University of Munich','','www.tum.de'}); % write video clip
```

```
Creating a new video file: '/Users/lueth/Desktop/Toolbox_test/imageVideoEndtitle.avi'
```

5. Create a text page title for a video

There are several reasons for adding text pages including latex equations too. This is also possible by a toolbox function. Again, the call without an output parameter would create a video clip.

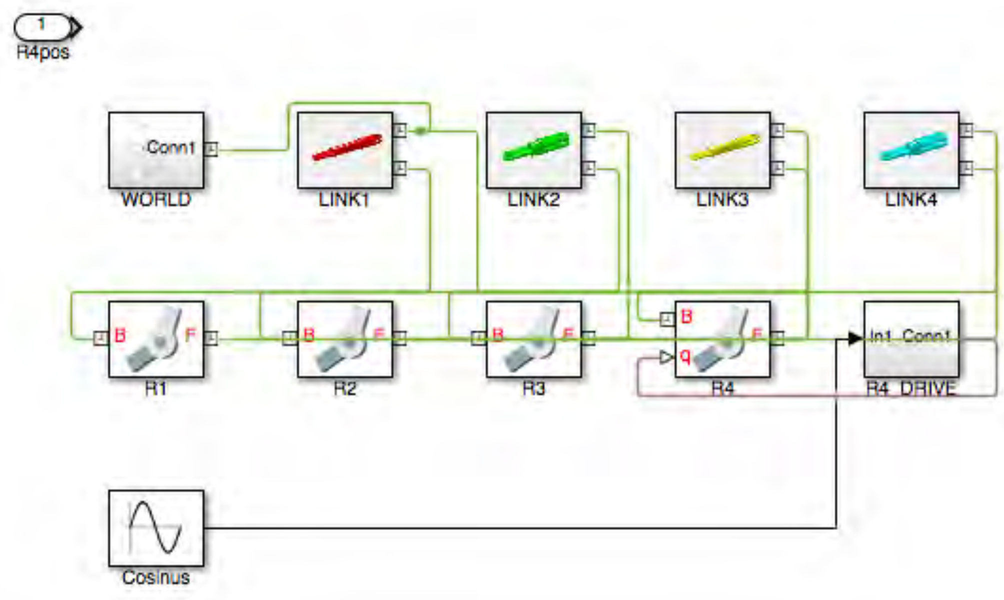
```
I=imageVideoTextPage([640 480],...
['It is also possible to name an existing video or to select it during function '...
'execution to define the size from an existing video. In addition, the '...
'background color and text color can be defined (in future also font name and '...
'font size), and furthermore times for creating a snapshot that becomes part of '...
'the title page.', char(13), '©']);
imshow(I.cdata);
```

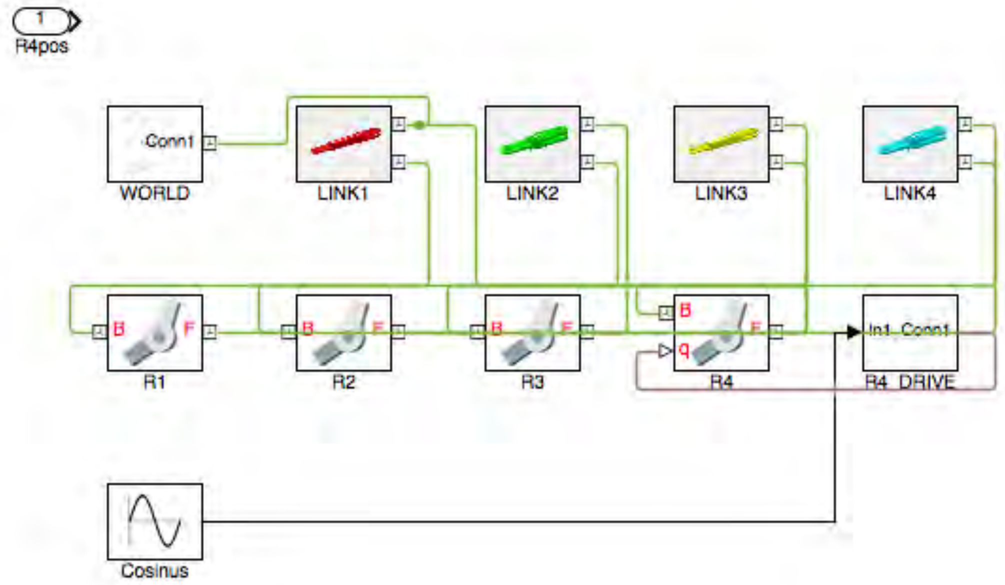
It is also possible to name an existing video or to select it during function execution to define the size from an existing video. In addition, the background color and text color can be defined (in future also font name and font size), and furthermore times for creating a snapshot that becomes part of the title page.
©

6. Now create a SimMultiBody fourbar linkage

```
smbNewSystem ('SG_LIB_EXP_25');           % Creates the mechansim diagramm
SG1=SGmodelLink(80,'',1,2);               % Creates a long rod with flange
SG2=SGmodelLink(50,'',1,2);               % Creates a short rod with flange
smbCreateSG (SG1,'LINK1','r');              % Add long rod as LINK1
smbCreateSG (SG2,'LINK2','g');              % Add short rod as LINK2
smbCreateSG (SG1,'LINK3','y');              % Add long rod as LINK3
smbCreateSG (SG2,'LINK4','c');              % Add short rod as LINK4
smbCreateJoint ('R','R1','LINK1.F','LINK2.B'); % Add a RR Joint
smbCreateJoint ('R','R2','LINK2.F','LINK3.B'); % Add a RR Joint
smbCreateJoint ('R','R3','LINK3.F','LINK4.B'); % Add a RR Joint
smbCreateJoint ('R','R4','LINK4.F','LINK1.B'); % Add a RR Joint
smbCreateConnection('WORLD.ORIGIN','LINK1.B'); % Connect Linkage to World Frame
smbCreateDrive ('R4');                         % Convert Joint R4 into a Drive
smbCreateSineWave ('Cosinus','R4_DRIVE/1');    % Connect a Sinus Generator to Drive
smbDrawNow;
```

```
Creating temporary directory '/Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_25/'
```



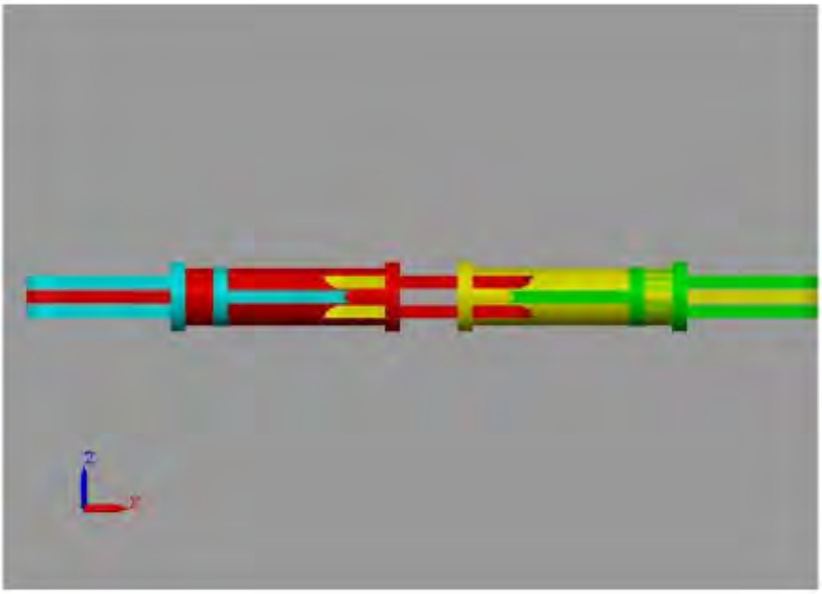


7. Create a video simulation and creates header and titles

```
[I,FN]=smbVideoSimulation (10);
```

```
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
```

```
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
```

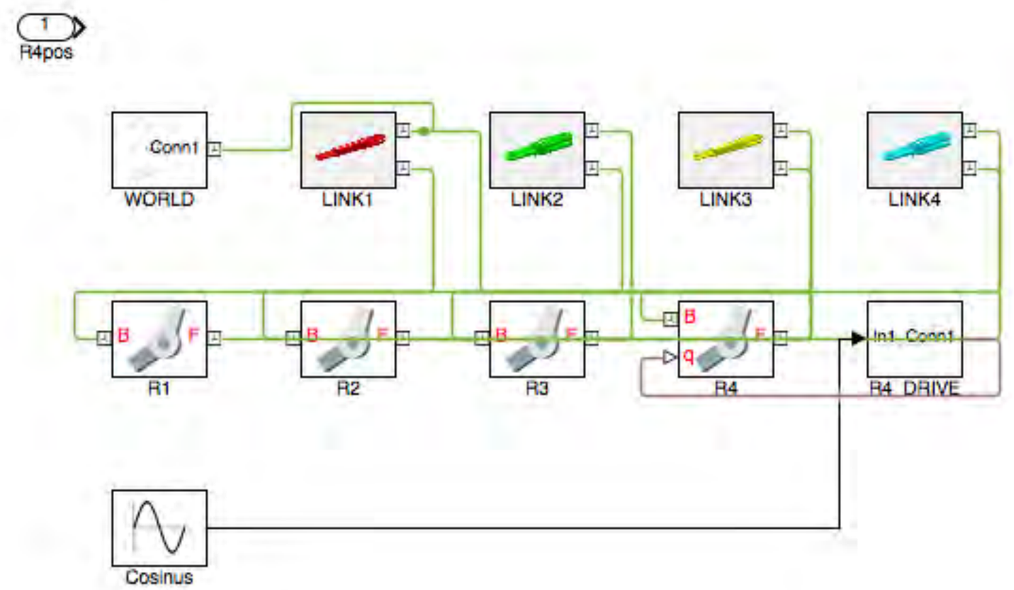


Now we create four small video clips in the desktopdir

```
imageVideoTitle(FN,{'SG-Lib Tutorial #25','Creating Videos Titles for SimMultiBody Videos','Tim C. Lueth
','$date'},'',[2 5]);
imageVideoEndtitle(FN,{'Technical University of Munich','','www.tum.de'});
imageVideoTextPage(FN,{...
['This video was created by using Mathwork''s SimMultiBody environment using the '...
'SG-Library of Tim C. Lueth. The fourbar linkage in the simulation has the '...
'following dimensions:©', char(13), ' '...
'L1= ', sprintf('%.2f mm',80), char(13), ' '...
'L2= ', sprintf('%.2f mm',50),char(13), ' '...
'L3= ', sprintf('%.2f mm',80),char(13), ' '...
```

```
'L4= ', sprintf('%.2f mm',50),char(13), '']});
imageVideoImagePage(FN,smbDrawNow);
```

```
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Creating a new video file: '/Users/lueth/Desktop/Toolbox_test/imageVideoTitle.avi'
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Creating a new video file: '/Users/lueth/Desktop/Toolbox_test/imageVideoEndtitle.avi'
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Creating a new video file: '/Users/lueth/Desktop/Toolbox_test/imageVideoTextPage.avi'
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Creating a new video file: '/Users/lueth/Desktop/Toolbox_test/imageVideoImagePage.avi'
```



8. Create Video Headers and Explaination

```
[I,FN]=smbVideoSimulation (10,'Video SG_LIB_EXP_25');
IT=imageVideoTitle(FN,{'SG-Lib Tutorial #25','Creating Videos Titles for SimMultiBody Videos','Tim C. Lu
eth','$date'},'',[2 5]);
IE=imageVideoEndtitle(FN,{'Technical University of Munich','','www.tum.de'});
ID=imageVideoTextPage(FN,{...
['This video was programmatically created by using Mathwork''s SimMultiBody environment using the '...
'SG-Library of Tim C. Lueth. The fourbar linkage in the simulation has the '...
'following dimensions:©', char(13), ' '...
'L1= ', sprintf('%.2f mm',80), char(13), ' '...
'L2= ', sprintf('%.2f mm',50),char(13), ' '...
'L3= ', sprintf('%.2f mm',80),char(13), ' '...
```

```
'L4= ', sprintf('%.2f mm',50),char(13), '']});
IM=imageVideoImagePage(FN,smbDrawNow);
videoWriteClipMovie(smbFilename('Video comp SG_LIB_EXP_25.avi'),IT,2,ID,5,IM,5,smbFilename('Video SG_LIB
_EXP_25.avi'),IE,1);
```

```
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Creating a new video file (NO SOUND/2016b): '/Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_25/Video c
omp SG_LIB_EXP_25.avi'
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100%
```

Final Remarks

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:49:06!
Executed 08-Nov-2018 20:49:08 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a MACI64
======================================== Used Matlab products: ======================================
antenna_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
=====================================================================================================
```

Published with MATLAB® R2018a

Tutorial 26: Create Mechanisms using Universal Planar Links

2017-01-20: 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

Motivation for this tutorial: (Originally SolidGeometry 3.3 required)

```
% function VLFL_EXP26
```

Motivation for this tutorial

A mechanism consists of two basic elements: a) joints and b) links that connect these joints. In the "automatic construction" of mechanisms, it is helpful to limit one of the two elements. This has already been used in the previous tutorials. In this tutorial a new procedure is presented. They are "universal planar links". These consist of a simple joint member and two halves of a rotary joint. If two links are connected to each other at one of the end points, the two halves of the joints are connected to an axis of rotation due to a spatial overlap, and a swivel joint is automatically formed. If a member is not connected, an axis of rotation is still retained there. Each axis of rotation can be connected with "knobs or drive mechanisms relative to the joint and its angular range can be restricted, the links can be connected in fixed planes, allowing a collision-free movement considering the links as well as the consideration of drive elements. This tutorial now shows you how to use the universal planar links in a simple example.

```
% clear all;
```

1. Create a SimMultiBody System for a Fourbar-Linkage

```
smbNewSystem ('SG_LIB_EXP_26')    % Creates the mechansim diagramm

L1=75;
L2=60;
L3=50;
L4=50;

L1=75; A=SGmodelLink2(L1,0,1,'BL,FL'); A.col='r';
L2=60; B=SGmodelLink2(L2,0,1);         B.col='g';
L3=50; C=SGmodelLink2(L3,0,-1);        C.col='y';
```

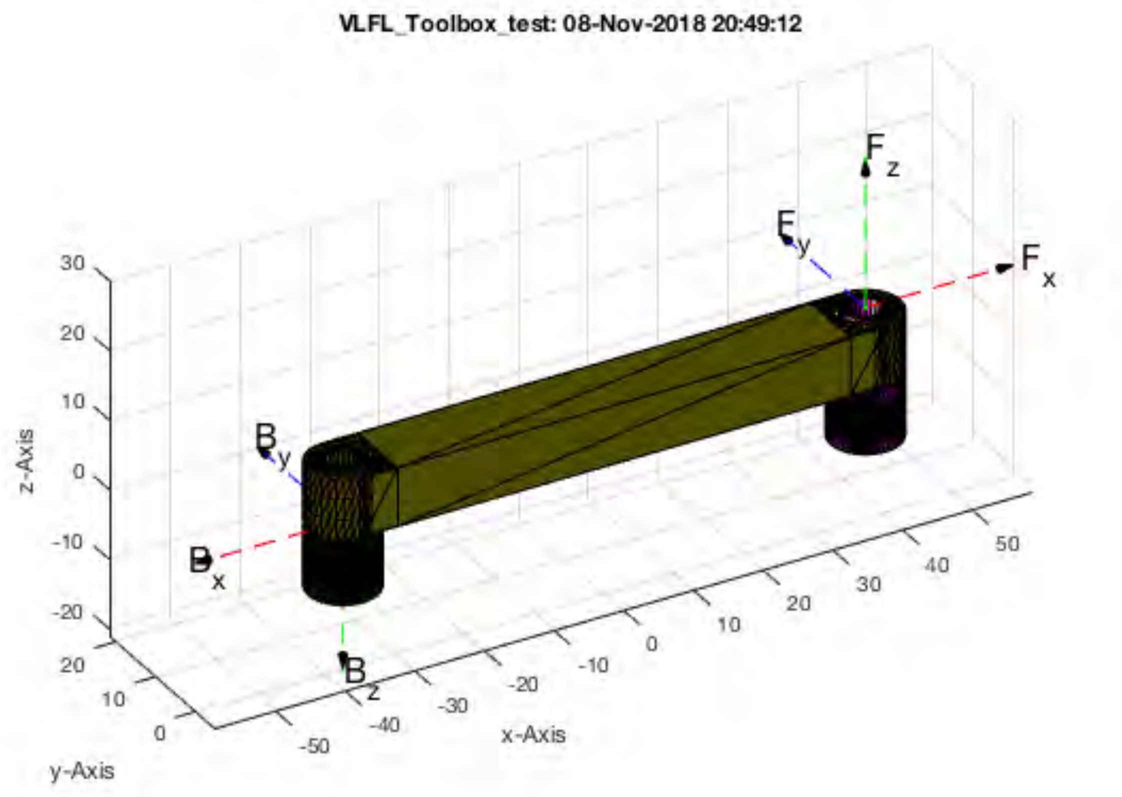
```
L4=50;D=SGmodelLink2(L4,0,-1);          D.col='m';
```

```
Creating temporary directory '/Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_26/'
```

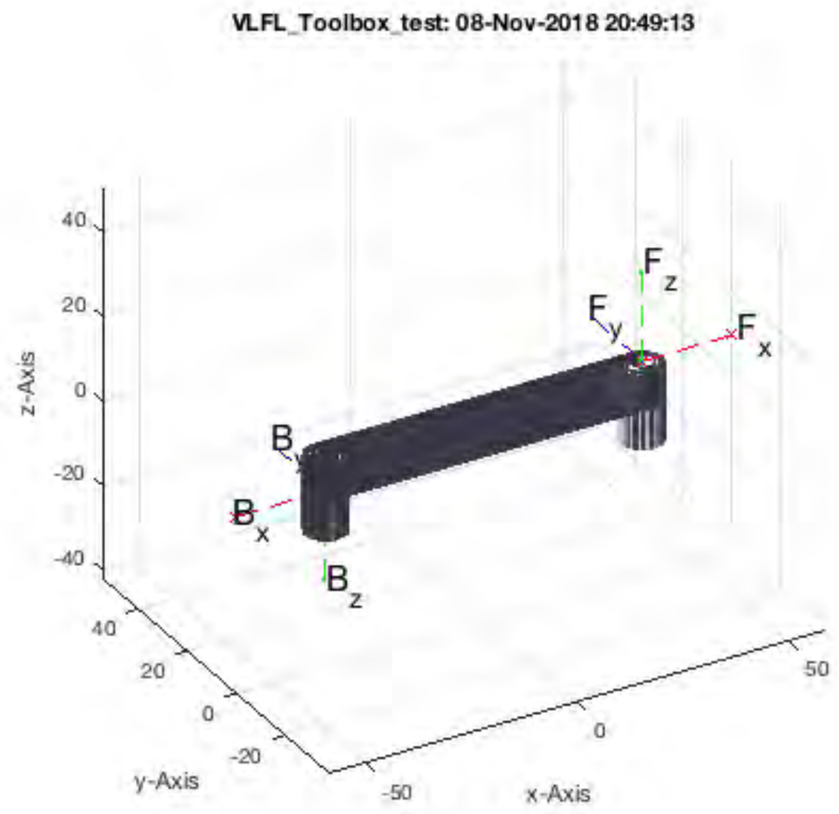
Show the components of the link

```
SGanalyzeGroupParts(A); SGTframeplot(A);
```

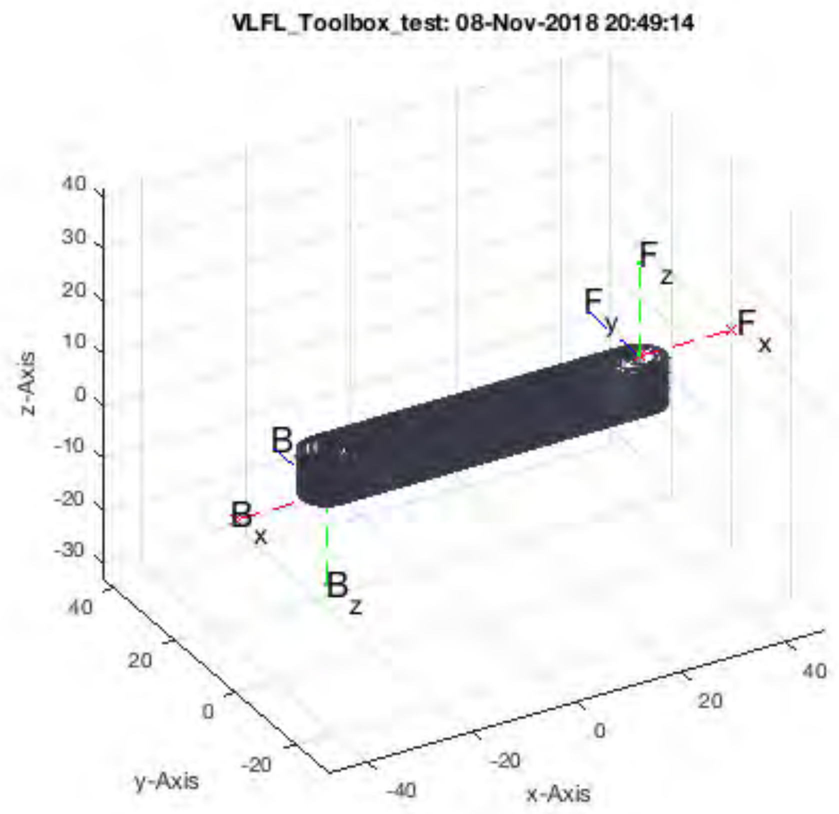
```
8% 12% 16% 20% 24% 28% 32% 36% 40% 44% 48% 52% 56% 60% 64% 68% 72% 76% 80% 84% 88% 92% 96%
100%
SGanalyzeGroupParts: 3 separated parts found.
```



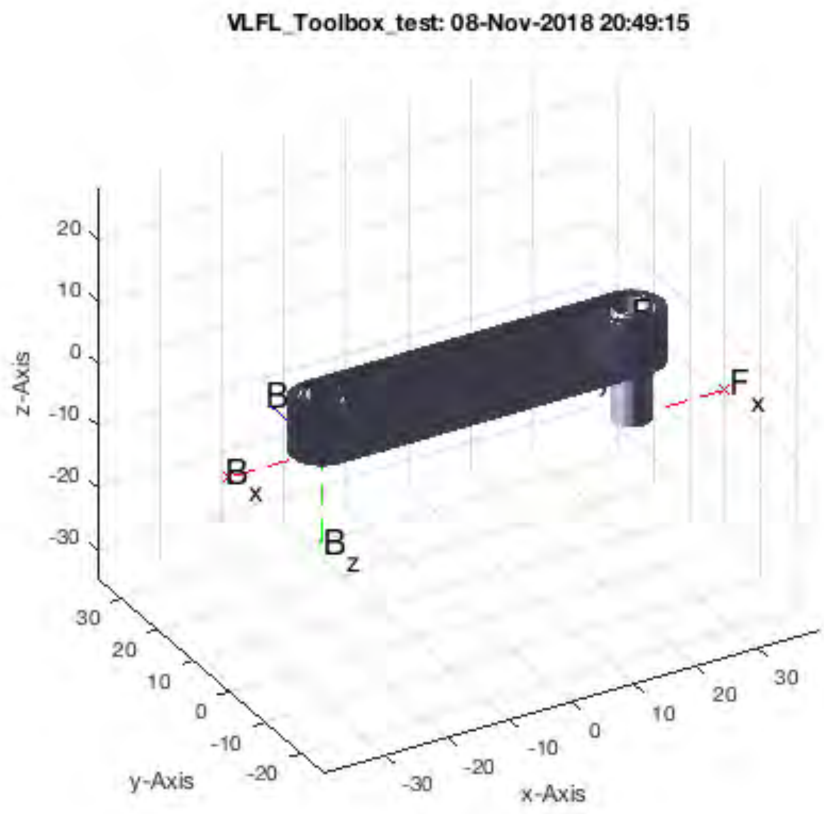

```
SGfigure; SGTplot(A); view(-30,30);
```



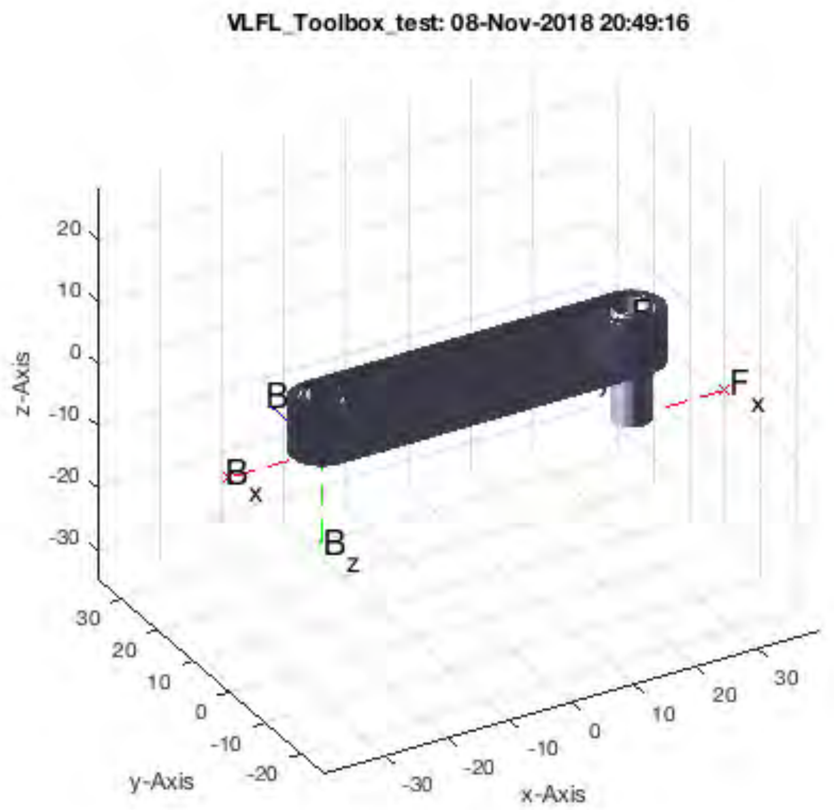

```
SGfigure; SGTplot(B); view(-30,30);
```



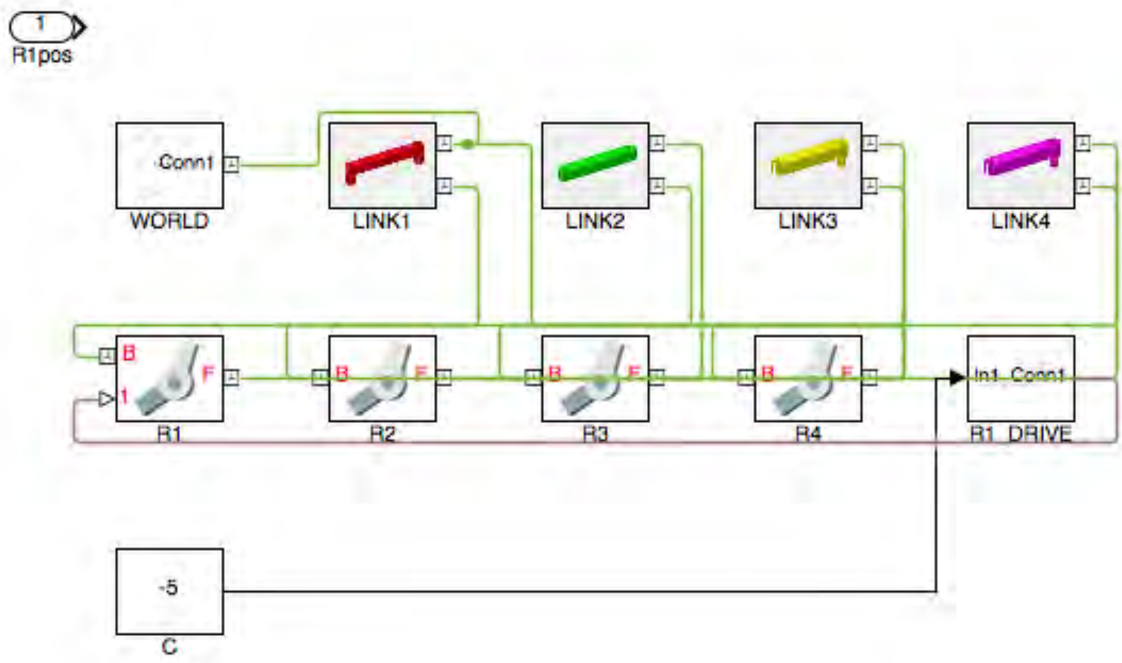

```
SGfigure; SGTplot(C); view(-30,30);
```




```
SGfigure; SGTplot(D); view(-30,30);
```




```
smbCreateSG (A,'LINK1','r');                % Add long rod as LINK1
smbCreateSG (B,'LINK2','g');                % Add short rod as LINK2
smbCreateSG (C,'LINK3','y');                % Add long rod as LINK3
smbCreateSG (D,'LINK4','m');                % Add short rod as LINK4
smbCreateJoint ('R','R1','LINK1.F','LINK2.B'); % Add a RR Joint
smbCreateJoint ('R','R2','LINK2.F','LINK3.B'); % Add a RR Joint
smbCreateJoint ('R','R3','LINK3.F','LINK4.B'); % Add a RR Joint
smbCreateJoint ('R','R4','LINK4.F','LINK1.B'); % Add a RR Joint

smbCreateConnection('WORLD.ORIGIN','LINK1.B'); % Connect Linkage to World Frame
smbCreateDrive ('R1');
smbSetJointInputTorque('R1');
smbCreateBlockConst('C','R1_DRIVE/1',-5)
ID=smbDrawNow;
smbSimulate(4);
```

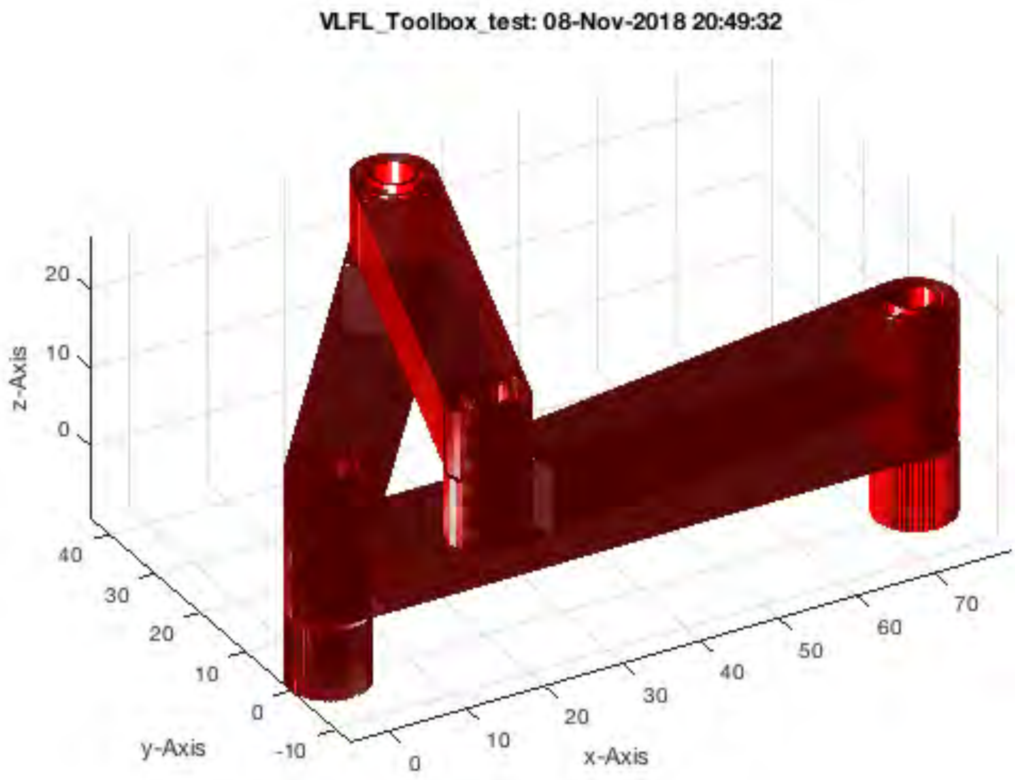


2. Now Create a Specific Configuration (Pose) and Write a STL-Files

```
SG=smbFullModelSimulation(5);
% SG=SGmagnifyVL(SG,'',[100 100 100]);
SGwriteSTL(SG,smbFilename('Universal Planar Link'));
```

```
CREATING A FULL SOLID-MOVEMENT SIMULATION-MODEL 'SG_LIB_EXP_26' THAT RUNS At LEAST 5.00 SEC
ONDS
==========================================================================================
========
Adding frame sensors for all solids of the model
Add frame sensors for 'LINK1.SG'
Add frame sensors for 'LINK2.SG'
Add frame sensors for 'LINK3.SG'
Add frame sensors for 'LINK4.SG'
==========================================================================================
========

simOut = 

  Simulink.SimulationOutput:
                 simlog: [1x1 simscape.logging.Node]
                   sout: [1x1 Simulink.SimulationData.Dataset]
                   tout: [1000x1 double]
                   xout: [1x1 Simulink.SimulationData.Dataset]

     SimulationMetadata: [1x1 Simulink.SimulationMetadata]
           ErrorMessage: [0x0 char]

LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_26/sbm_temp_LINK1
```

```
.stl
Header:
Number of facets: 3756
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_26/sbm_temp_LINK2
.stl
Header:
Number of facets: 2404
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_26/sbm_temp_LINK3
.stl
Header:
Number of facets: 2584
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_26/sbm_temp_LINK4
.stl
Header:
Number of facets: 2584
0..

CREATED A SOLID GEOMETRY OF THE FULL SIMULATION-MODEL 'SG_LIB_EXP_26' AT TIME: 5.00 SECONDS
===========================================================================================
========
1000..2000..3000..4000..5000..6000..7000..8000..9000..10000..11000..
```

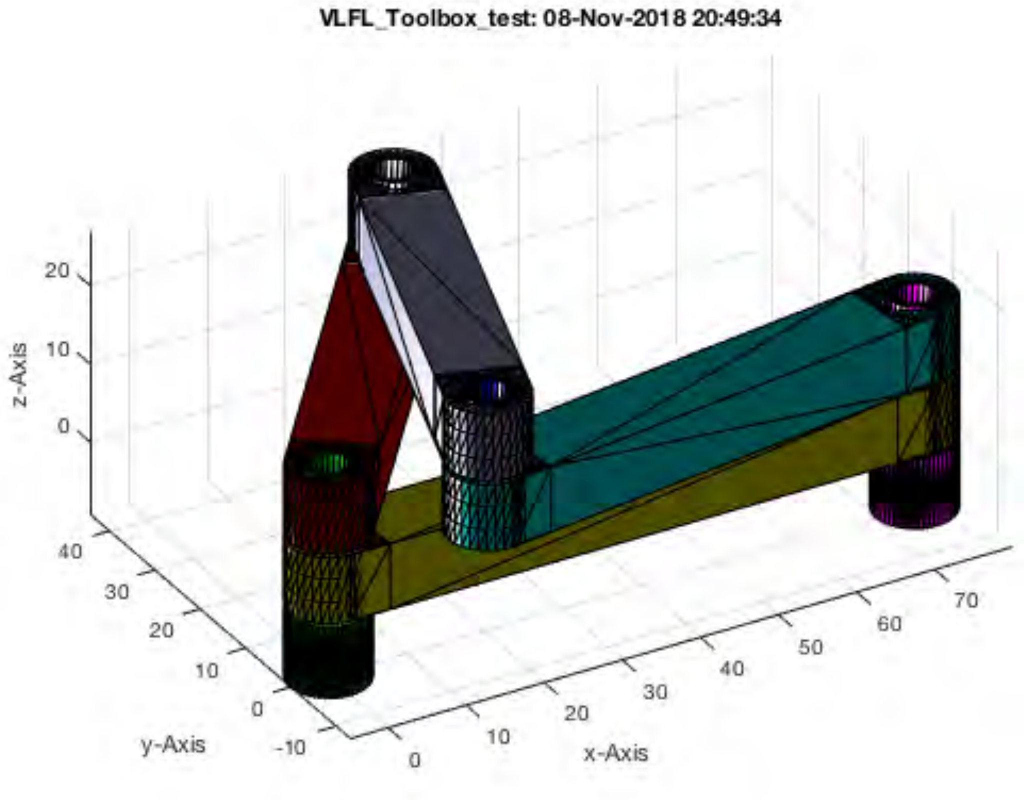


3. Now Analyze the Stucture and Group the Solids to Parts

```
SGN=SG;
```

```
SGfigure; view(-30,30); SGplot(SG,'m');  SG=SGanalyzeGroupParts(SG); SGplot(SG);
```

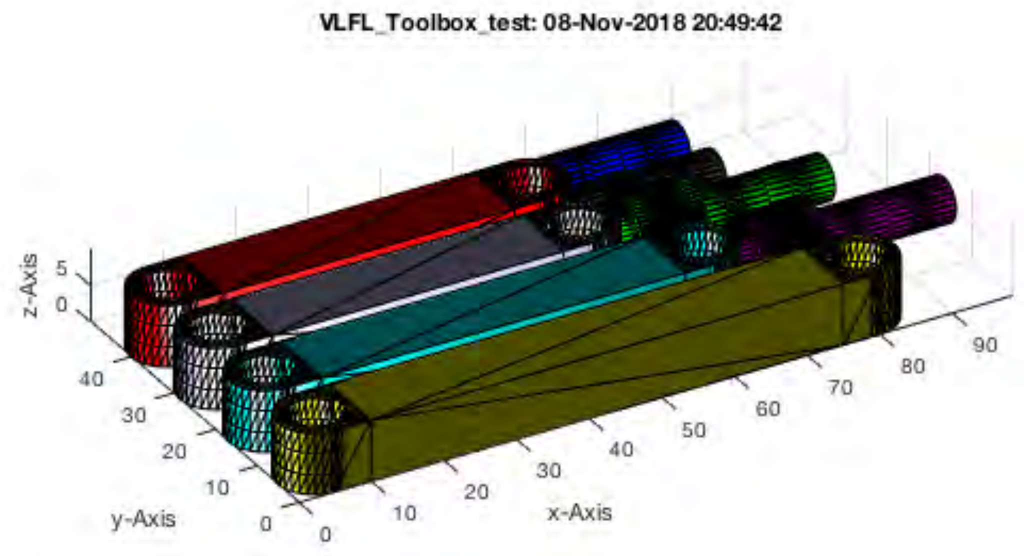
```
4% 8% 12% 16% 20% 24% 28% 32% 36% 40% 44% 48% 52% 56% 60% 64% 68% 72% 76% 80% 84% 88% 92% 9
6% 100%
```



4. Now Arrange all Parts for Printing as Separated Solids

```
[~,SG]=SGpacking(SG); SGfigure; view(-30,30); SGplot(SG);
```

```
Packing 8 objects (h=66):
```



5. Now Write the Separated Parts into Different STL Files

```
SGwriteSeparatedSTL(SG);
```

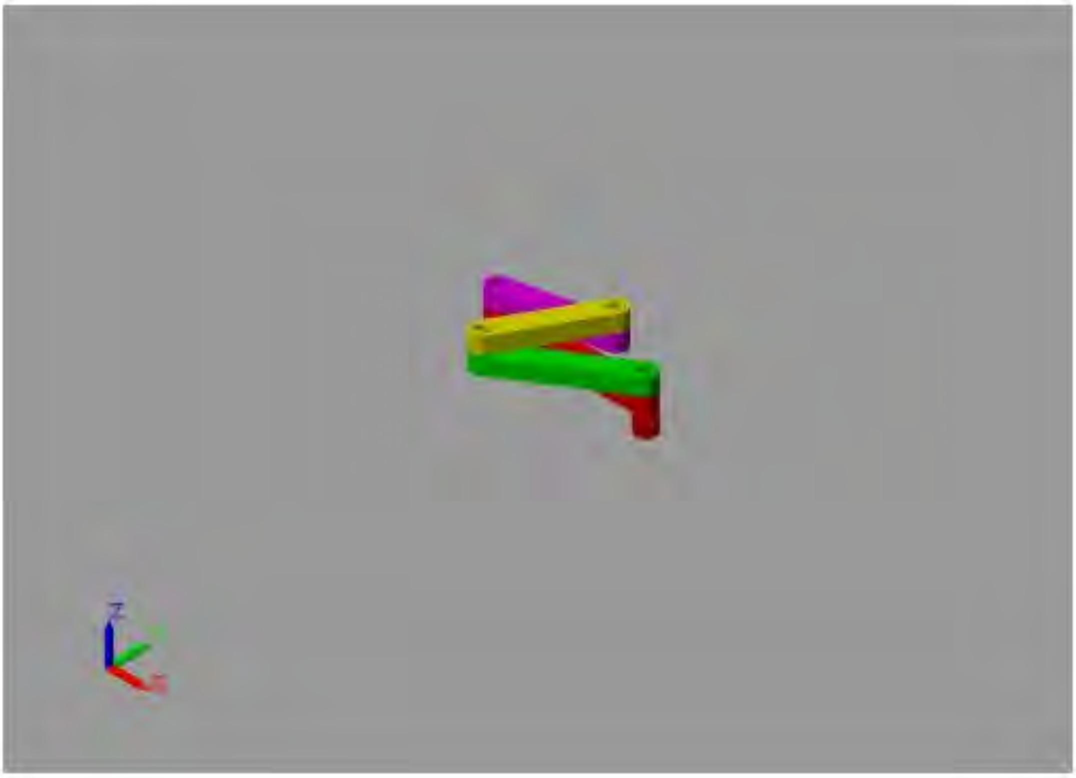
```
SGwriteSeparatedSTL: Writing 8 STL files in /Users/lueth/Desktop/Toolbox_test/EXP-2018-11-0
8/
```

6. Create a Video of the Linkage Simulation

```
[I1,FN]=smbVideoSimulation (4);     % Simulate for 1 second
IT=imageVideoTitle(FN,{'SG-Lib Tutorial #26','Universal Planar Links','Tim C. Lueth','$date
'},'',[0.1 0.2 0.3]);
IE=imageVideoEndtitle(FN);
videoWriteClipMovie(smbFilename('Universal Planar Links SimMultiBody.avi'),IT,2,ID,1,FN,IE,
1);
imshow(I1);
```

```
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
```

```
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Creating a new video file (NO SOUND/2016b): '/Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_E
XP_26/Universal Planar Links SimMultiBody.avi'
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100%
```



Final Remarks

```
close all
```

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:50:02!
Executed 08-Nov-2018 20:50:04 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
===========================================================================================
==========
```

Published with MATLAB® R2018a

Tutorial 27: Fourbar-Linkage: 2 Pose Syntheses and Linkage Export for 3D Printing

2017-01-05: 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 3.3 required)

```
function VLFL_EXP27
```

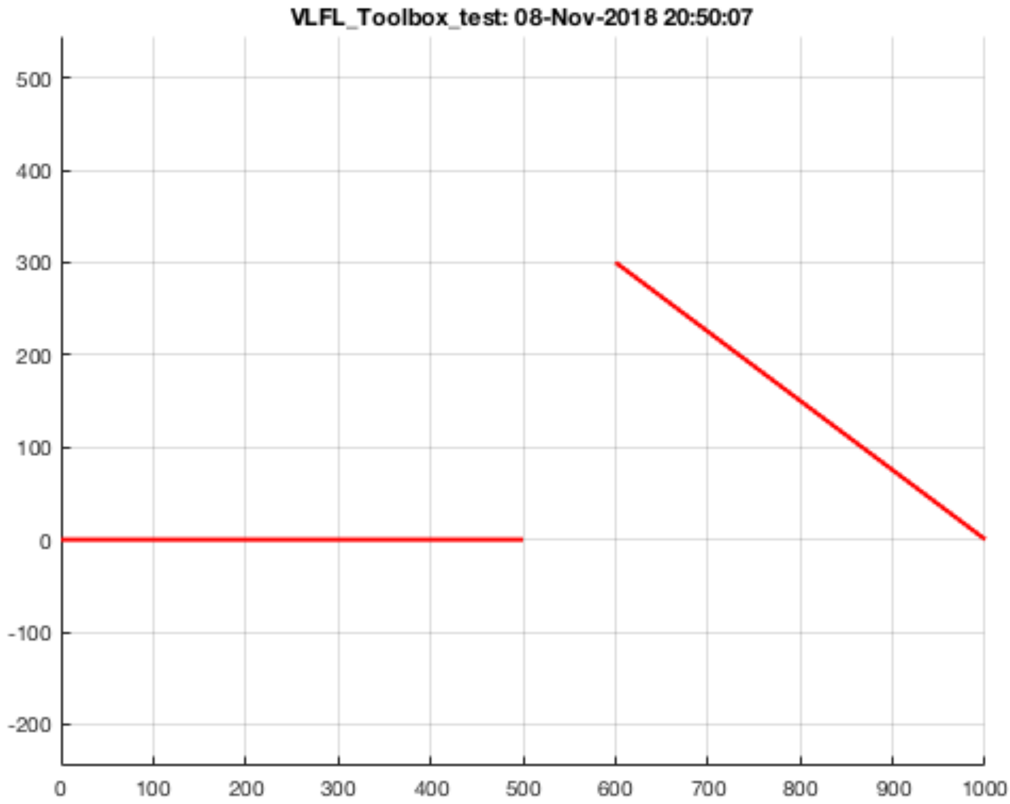
```
smbNewSystem ('SG_LIB_EXP_27');              % Creates the mechansim diagramm
```

```
Creating temporary directory '/Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_27/'
```

3. Show a Two Pose Problem

Poses are described by the start point and end point of a link. The first point of the coupling defines a point, the second point also the direction. In the simplest case, two poses for the design of a four-bar are given.

```
d=[0 0];
C1=[0 0];
D1=[500 0];
C2=[600 300];
D2=[1000 0];

SGfigure;
PLplot([C1;D1],'r-',2);
PLplot([C2;D2],'r-',2);
```



4. Find a General Solutions for the Two Pose Problem

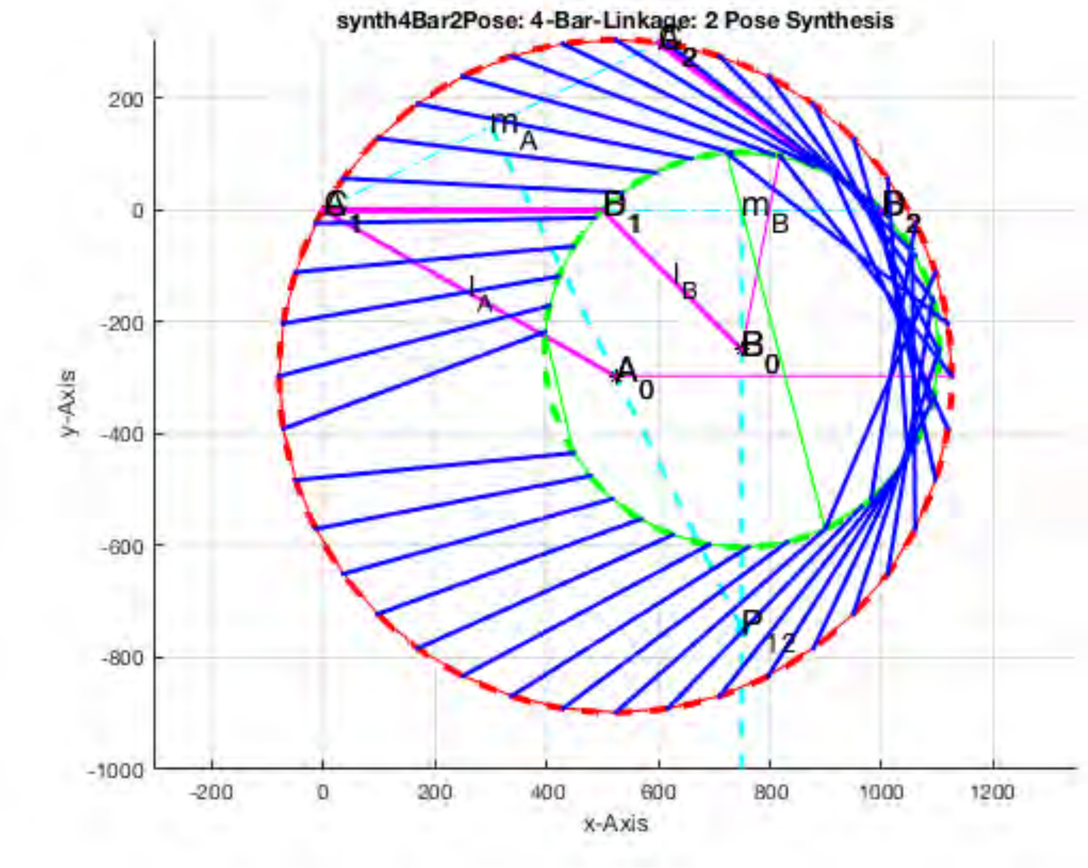
As a solution, there are two straight lines on each of which the frame point A0 or the frame point B0 may be located. The intersection point of these two lines is the pole point P12. In a special case, both frame points are located at this point and a triangle is formed from the four-bar. In any case, the rack points can be displaced such that other secondary conditions can also be fulfilled.

```
imageFigureMovie('record');
synth4Bar2Pose(C1,D1,C2,D2,d);
[~,FN2]=imageFigureMovie('write',smbFilename('synth4Bar2Pose.avi'));
```

```
ans = 

  Line with properties:

              Color: [0 0 1]
          LineStyle: '-'
          LineWidth: 3
             Marker: '.'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [0 500]
              YData: [0 0]
              ZData: [0 0]

  Use GET to show all properties
```

```
ans =

  Line with properties:

              Color: [0 0 1]
          LineStyle: '-'
          LineWidth: 3
             Marker: '.'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [600 1000]
              YData: [300 0]
              ZData: [0 0]

  Use GET to show all properties


ans =

  Line with properties:

              Color: [1 0 1]
          LineStyle: '-'
          LineWidth: 3
             Marker: '.'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [0 500]
              YData: [0 0]
              ZData: [0 0]

  Use GET to show all properties


ans =

  Line with properties:

              Color: [1 0 1]
          LineStyle: '-'
          LineWidth: 3
             Marker: '.'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [600 1000]
              YData: [300 0]
              ZData: [0 0]

  Use GET to show all properties


ans =

  Line with properties:
```

```
              Color: [1 0 1]
          LineStyle: '-'
          LineWidth: 2
             Marker: '.'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [523.6068 0]
              YData: [-297.2136 0]
              ZData: [0 0]

  Use GET to show all properties


ans = 

  Line with properties:

              Color: [1 0 1]
          LineStyle: '-'
          LineWidth: 2
             Marker: '.'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [750 500]
              YData: [-250 0]
              ZData: [0 0]

  Use GET to show all properties

imageFigureSaveMovie: Writing figure movie with 42 frames in file: /Users/lueth/Desktop/Too
lbox_test/tmp_SG_LIB_EXP_27/synth4Bar2Pose.avi.
```



5. Find a special Solution for the 4Bar-Linkage wit A0 and B0 on same level

If A0 and B0 are to lie on the same plane, B0+k*(P12-B0) must correspond to the Y coordinate of the Y coordinate of A0 in the y coordinate

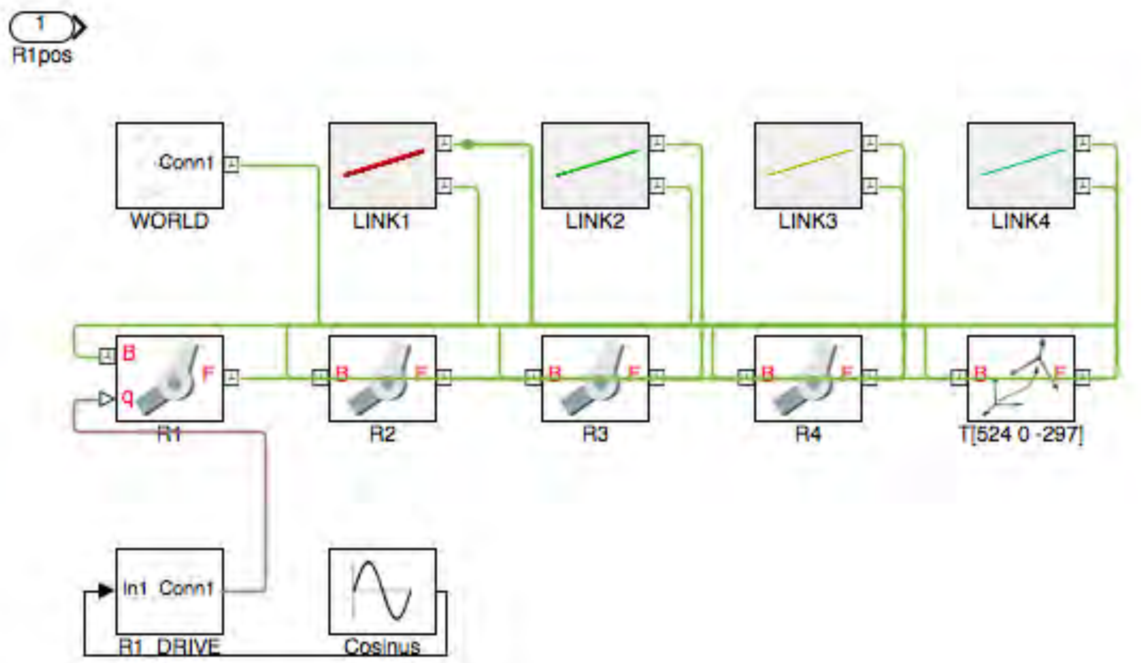
```
[A0,B0,A1,B1,P12]=synth4Bar2Pose(C1,D1,C2,D2,d);
db=P12-B0
k=(A0(2)-B0(2))/db(2)
B0=B0+k*db
A0
L1=norm(B0-A0)
L2=norm(B1-B0)
L3=norm(A1-B1)
L4=norm(A0-A1)
```

```
db =

    0.0000 -500.0000


k =

    0.0944


B0 =
```

```
  750.0000 -297.2136


A0 =

  523.6068 -297.2136


L1 =

  226.3932


L2 =

  388.3760


L3 =

   500


L4 =

  602.0797
```

6. Create a SimMultiyBody System for the calculated solution

```
A=SGmodelLink(L1,'',1,2); A=SGmodelLink2(L1,0,1);
B=SGmodelLink(L2,'',1,2); B=SGmodelLink2(L2,0,1);
C=SGmodelLink(L3,'',1,2); C=SGmodelLink2(L3,0,-1);
D=SGmodelLink(L4,'',1,2); D=SGmodelLink2(L4,0,-1);

smbCreateSG (A,'LINK1','r');              % Add long rod as LINK1
smbCreateSG (B,'LINK2','g');              % Add short rod as LINK2
smbCreateSG (C,'LINK3','y');              % Add long rod as LINK3
smbCreateSG (D,'LINK4','c');              % Add short rod as LINK4
smbCreateJoint ('R','R1','LINK1.F','LINK2.B'); % Add a RR Joint
smbCreateJoint ('R','R2','LINK2.F','LINK3.B'); % Add a RR Joint
smbCreateJoint ('R','R3','LINK3.F','LINK4.B'); % Add a RR Joint
smbCreateJoint ('R','R4','LINK4.F','LINK1.B'); % Add a RR Joint
smbCreateConnection('WORLD.ORIGIN','LINK1.B',TofP([A0(1) 0 A0(2)])); % Connect Linkage to W
orld Frame
smbCreateDrive ('R1');
smbCreateSineWave ('Cosinus','R1_DRIVE/1');
ID=smbDrawNow;
```



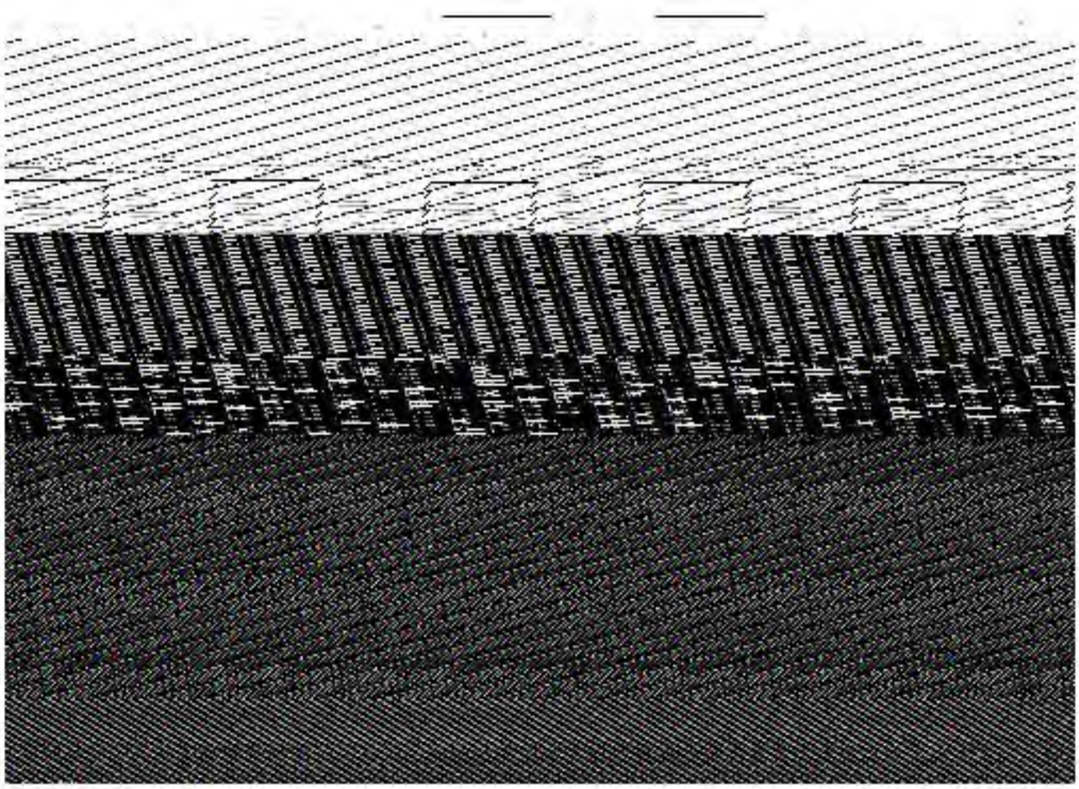
7. Show the Video of the Simulation

```
[I1,FN]=smbVideoSimulation (4);     % Simulate for 1 second
IT=imageVideoTitle(FN,{'SG-Lib Tutorial #27','2 Pose Syntheses','Tim C. Lueth','$date'},'',
[0.78 1.33]);
IE=imageVideoEndtitle(FN);
videoWriteClipMovie(smbFilename('2 Pose Syntheses SimMultiBody.avi'),IT,2,FN2,ID,1,FN,IE,1)
;
imshow(I1);
```

```
Warning: Solver is encountering difficulty in simulating model '<a
href="matlab:open_system ('SG_LIB_EXP_27')">SG_LIB_EXP_27</a>' at time
0.00020857922471329357. Simulink will continue to simulate with warnings. Please
check the model for errors.
Warning: Solver was unable to reduce the step size without violating minimum
step size of 7.4102226475251900E-019 for 1 consecutive times at time
2.0857922471329400E-004.  Solver will continue simulation with the step size
restricted to 7.4102226475251900E-019 and using an effective relative error
tolerance of 8.4742961193161300E-003, which is greater than the specified
relative error tolerance of 1.0000000000000000E-003. This usually may be caused
by the high stiffness of the system. Please check the system or increase the
solver <a
href="matlab:configset.internal.open('SG_LIB_EXP_27','MaxConsecutiveMinStep');">Number
of consecutive min steps</a> violation parameter.
Warning: Solver is encountering difficulty in simulating model '<a
href="matlab:open_system ('SG_LIB_EXP_27')">SG_LIB_EXP_27</a>' at time
0.0002085792247132943. Simulink will continue to simulate with warnings. Please
check the model for errors.
Warning: Solver was unable to reduce the step size without violating minimum
step size of 7.4102226475252200E-019 for 2 consecutive times at time
```

```
2.0857922471329400E-004.  Solver will continue simulation with the step size
restricted to 7.4102226475252200E-019 and using an effective relative error
tolerance of 1.2481106134613500E-003, which is greater than the specified
relative error tolerance of 1.0000000000000000E-003. This usually may be caused
by the high stiffness of the system. Please check the system or increase the
solver <a
href="matlab:configset.internal.open('SG_LIB_EXP_27','MaxConsecutiveMinStep');">Number
of consecutive min steps</a> violation parameter.
Warning: Solver is encountering difficulty in simulating model '<a
href="matlab:open_system ('SG_LIB_EXP_27')">SG_LIB_EXP_27</a>' at time
0.00020857922521788663. Simulink will continue to simulate with warnings. Please
check the model for errors.
Warning: Solver was unable to reduce the step size without violating minimum
step size of 7.4102226654519400E-019 for 1 consecutive times at time
2.0857922521788700E-004.  Solver will continue simulation with the step size
restricted to 7.4102226654519400E-019 and using an effective relative error
tolerance of 1.1695417848713900E-002, which is greater than the specified
relative error tolerance of 1.0000000000000000E-003. This usually may be caused
by the high stiffness of the system. Please check the system or increase the
solver <a
href="matlab:configset.internal.open('SG_LIB_EXP_27','MaxConsecutiveMinStep');">Number
of consecutive min steps</a> violation parameter.
Warning: Solver is encountering difficulty in simulating model '<a
href="matlab:open_system ('SG_LIB_EXP_27')">SG_LIB_EXP_27</a>' at time
0.00020857922521788736. Simulink will continue to simulate with warnings. Please
check the model for errors.
Warning: Solver was unable to reduce the step size without violating minimum
step size of 7.4102226654519600E-019 for 2 consecutive times at time
2.0857922521788700E-004.  Solver will continue simulation with the step size
restricted to 7.4102226654519600E-019 and using an effective relative error
tolerance of 1.0469936968904200E-002, which is greater than the specified
relative error tolerance of 1.0000000000000000E-003. This usually may be caused
by the high stiffness of the system. Please check the system or increase the
solver <a
href="matlab:configset.internal.open('SG_LIB_EXP_27','MaxConsecutiveMinStep');">Number
of consecutive min steps</a> violation parameter.
Warning: Solver is encountering difficulty in simulating model '<a
href="matlab:open_system ('SG_LIB_EXP_27')">SG_LIB_EXP_27</a>' at time
0.00020857922521788809. Simulink will continue to simulate with warnings. Please
check the model for errors.
Warning: Solver was unable to reduce the step size without violating minimum
step size of 7.4102226654519900E-019 for 3 consecutive times at time
2.0857922521788800E-004.  Solver will continue simulation with the step size
restricted to 7.4102226654519900E-019 and using an effective relative error
tolerance of 1.2481106164805300E-003, which is greater than the specified
relative error tolerance of 1.0000000000000000E-003. This usually may be caused
by the high stiffness of the system. Please check the system or increase the
solver <a
href="matlab:configset.internal.open('SG_LIB_EXP_27','MaxConsecutiveMinStep');">Number
of consecutive min steps</a> violation parameter.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
```

```
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Creating a new video file (NO SOUND/2016b): '/Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_E
XP_27/2 Pose Syntheses SimMultiBody.avi'
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100% Warning: A
value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100%
```

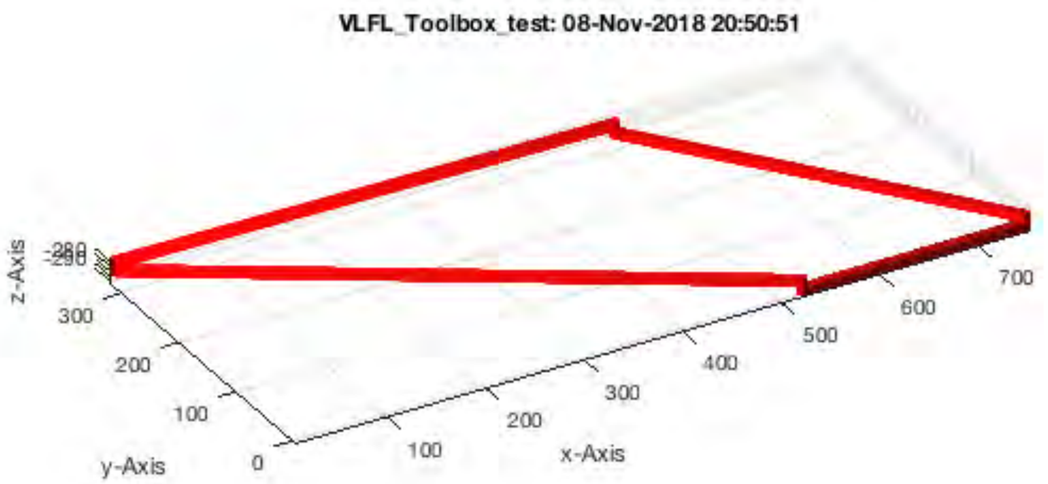


8. Now Create the Solid Geoemtry at time 0.78 seconds

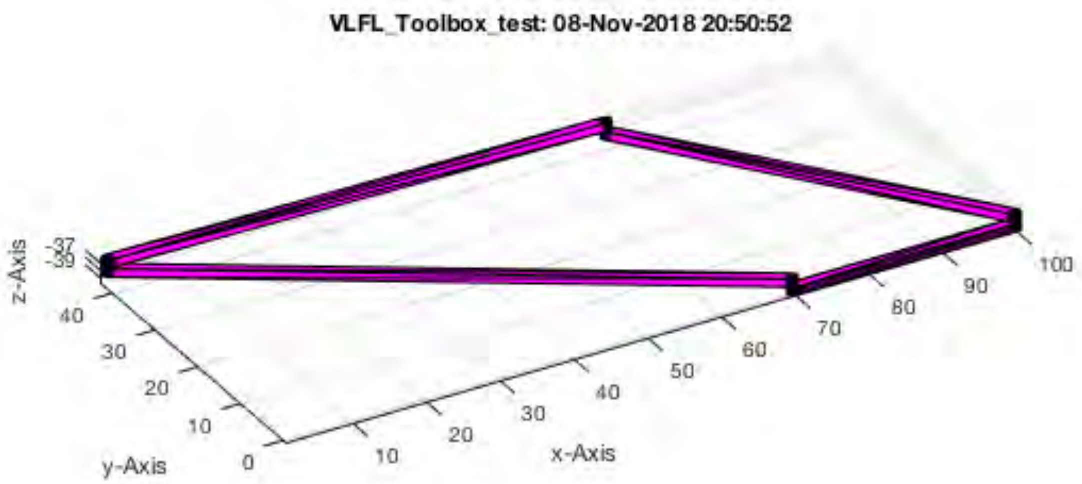
```
SG=smbFullModelSimulation(0.78);
SG=SGmagnifyVL(SG,'',[100 100 100]);
SGwriteSTL(SG,smbFilename('2-Pose-Synth'));
```

```
CREATING A FULL SOLID-MOVEMENT SIMULATION-MODEL 'SG_LIB_EXP_27' THAT RUNS At LEAST 0.78 SEC
ONDS
==========================================================================================
========
Adding frame sensors for all solids of the model
Add frame sensors for 'LINK1.SG'
Add frame sensors for 'LINK2.SG'
Add frame sensors for 'LINK3.SG'
Add frame sensors for 'LINK4.SG'
==========================================================================================
========
Warning: Solver is encountering difficulty in simulating model '<a
href="matlab:open_system ('SG_LIB_EXP_27')">SG_LIB_EXP_27</a>' at time
0.00015184117587706779. Simulink will continue to simulate with warnings. Please
check the model for errors.
Warning: Solver was unable to reduce the step size without violating minimum
step size of 5.3944822254360900E-019 for 1 consecutive times at time
1.5184117587706800E-004.  Solver will continue simulation with the step size
restricted to 5.3944822254360900E-019 and using an effective relative error
tolerance of 4.3035671698391400E-003, which is greater than the specified
relative error tolerance of 1.0000000000000000E-003. This usually may be caused
by the high stiffness of the system. Please check the system or increase the
solver <a
href="matlab:configset.internal.open('SG_LIB_EXP_27','MaxConsecutiveMinStep');">Number
of consecutive min steps</a> violation parameter.
Warning: Solver is encountering difficulty in simulating model '<a
href="matlab:open_system ('SG_LIB_EXP_27')">SG_LIB_EXP_27</a>' at time
0.00015184117697708603. Simulink will continue to simulate with warnings. Please
check the model for errors.
Warning: Solver was unable to reduce the step size without violating minimum
step size of 5.3944822645165900E-019 for 1 consecutive times at time
1.5184117697708600E-004.  Solver will continue simulation with the step size
restricted to 5.3944822645165900E-019 and using an effective relative error
tolerance of 6.1589696648945900E-003, which is greater than the specified
relative error tolerance of 1.0000000000000000E-003. This usually may be caused
by the high stiffness of the system. Please check the system or increase the
solver <a
href="matlab:configset.internal.open('SG_LIB_EXP_27','MaxConsecutiveMinStep');">Number
of consecutive min steps</a> violation parameter.
Warning: Solver is encountering difficulty in simulating model '<a
href="matlab:open_system ('SG_LIB_EXP_27')">SG_LIB_EXP_27</a>' at time
0.00015184117697708658. Simulink will continue to simulate with warnings. Please
check the model for errors.
Warning: Solver was unable to reduce the step size without violating minimum
step size of 5.3944822645166100E-019 for 2 consecutive times at time
1.5184117697708700E-004.  Solver will continue simulation with the step size
restricted to 5.3944822645166100E-019 and using an effective relative error
tolerance of 8.7521093125381700E-003, which is greater than the specified
relative error tolerance of 1.0000000000000000E-003. This usually may be caused
```

```
by the high stiffness of the system. Please check the system or increase the
solver <a
href="matlab:configset.internal.open('SG_LIB_EXP_27','MaxConsecutiveMinStep');">Number
of consecutive min steps</a> violation parameter.
Warning: Solver is encountering difficulty in simulating model '<a
href="matlab:open_system ('SG_LIB_EXP_27')">SG_LIB_EXP_27</a>' at time
0.00015184117697708712. Simulink will continue to simulate with warnings. Please
check the model for errors.
Warning: Solver was unable to reduce the step size without violating minimum
step size of 5.3944822645166300E-019 for 3 consecutive times at time
1.5184117697708700E-004.  Solver will continue simulation with the step size
restricted to 5.3944822645166300E-019 and using an effective relative error
tolerance of 1.2266067701044700E-003, which is greater than the specified
relative error tolerance of 1.0000000000000000E-003. This usually may be caused
by the high stiffness of the system. Please check the system or increase the
solver <a
href="matlab:configset.internal.open('SG_LIB_EXP_27','MaxConsecutiveMinStep');">Number
of consecutive min steps</a> violation parameter.

simOut =

  Simulink.SimulationOutput:
                 simlog: [1x1 simscape.logging.Node]
                   sout: [1x1 Simulink.SimulationData.Dataset]
                   tout: [189x1 double]
                   xout: [1x1 Simulink.SimulationData.Dataset]

     SimulationMetadata: [1x1 Simulink.SimulationMetadata]
           ErrorMessage: [0x0 char]

LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_27/sbm_temp_LINK1
.stl
Header:
Number of facets: 2404
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_27/sbm_temp_LINK2
.stl
Header:
Number of facets: 2404
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_27/sbm_temp_LINK3
.stl
Header:
Number of facets: 2584
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_27/sbm_temp_LINK4
.stl
Header:
Number of facets: 2584
0..

CREATED A SOLID GEOMETRY OF THE FULL SIMULATION-MODEL 'SG_LIB_EXP_27' AT TIME: 0.78 SECONDS
===========================================================================================
========
```




```
SGfigure; view(-30,30); SGplot(SG,'m'); % SGanalyzeGroupParts(SG);
```



Final Remarks

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:50:53!
Executed 08-Nov-2018 20:50:55 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
===========================================================================================
==========
```

Published with MATLAB® R2018a

Tutorial 28: Fourbar-Linkage: 3 Pose Syntheses and Linkage Export for 3D Printing

2017-01-08: 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

Motivation for this tutorial: (Originally SolidGeometry 3.3 required)

```
smbNewSystem ('SG_LIB_EXP_28',[+5 +5 +5]);           % Creates the mechansim diagramm
```

```
Creating temporary directory '/Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_28/'
```

3. Show a Two Pose Problem

Poses are described by the start point and end point of a link. The first point of the coupling defines a point, the second point also the direction. In the simplest case, two poses for the design of a four-bar are given.

```
d=[0 0];
C1=[ 0 0];
D1=[40 0];
C2=[60 30];
D2=[100 0];
C3=[90 20];
D3=[50 20];

SGfigure;
PLplot([C1;D1],'r-',2);
PLplot([C2;D2],'r-',2);
PLplot([C3;D3],'r-',2);
```

4. Find a General Solutions for the Three Pose Problem

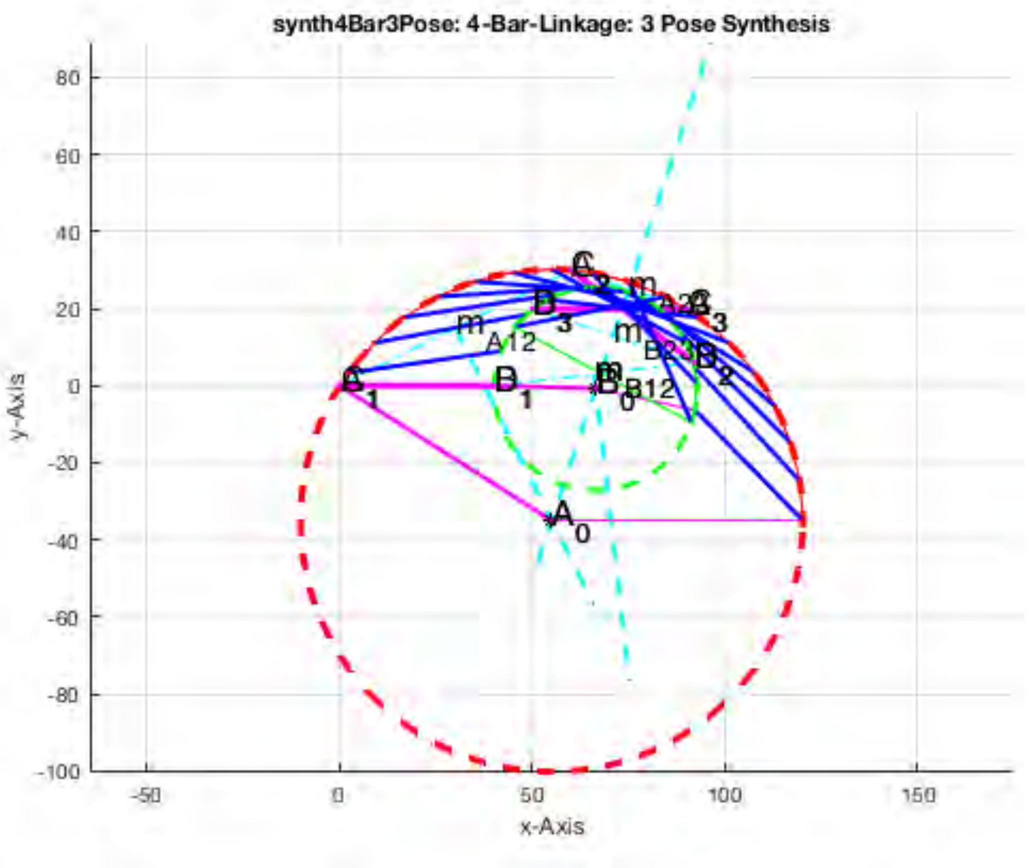
As a solution, there are two straight lines on each of which the frame point A0 or the frame point B0 may be located. The intersection point of these two lines is the pole point P12. In a special case, both frame points are located at this point and a triangle is formed from the four-bar. In any case, the rack points can be displaced such that other secondary conditions can also be fulfilled.

```
l=500

imageFigureMovie('record');
synth4Bar3Pose(C1,D1,C2,D2,C3,D3,d);
% exp_2017_01_08(C1,D1,C2,D2,C3,D3,l,d);
[~,FN2]=imageFigureMovie('write',smbFilename('synth4Bar3Pose.avi'));
```

```
l =

   500


ans =

  Line with properties:

              Color: [0 0 1]
          LineStyle: '-'
          LineWidth: 3
             Marker: '.'
```

```
          MarkerSize: 6
     MarkerFaceColor: 'none'
               XData: [0 40]
               YData: [0 0]
               ZData: [0 0]

  Use GET to show all properties


ans = 

  Line with properties:

               Color: [0 0 1]
           LineStyle: '-'
           LineWidth: 3
              Marker: '.'
          MarkerSize: 6
     MarkerFaceColor: 'none'
               XData: [60 92]
               YData: [30 6]
               ZData: [0 0]

  Use GET to show all properties


ans = 

  Line with properties:

               Color: [0 0 1]
           LineStyle: '-'
           LineWidth: 3
              Marker: '.'
          MarkerSize: 6
     MarkerFaceColor: 'none'
               XData: [90 50]
               YData: [20 20]
               ZData: [0 0]

  Use GET to show all properties


ans = 

  Line with properties:

               Color: [1 0 1]
           LineStyle: '-'
           LineWidth: 2
              Marker: '.'
          MarkerSize: 6
     MarkerFaceColor: 'none'
               XData: [55.0000 0]
               YData: [-35.0000 0]
               ZData: [0 0]
```

```
  Use GET to show all properties


ans = 

  Line with properties:

              Color: [1 0 1]
          LineStyle: '-'
          LineWidth: 2
             Marker: '.'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [66.4286 40]
              YData: [-0.7143 0]
              ZData: [0 0]

  Use GET to show all properties

imageFigureSaveMovie: Writing figure movie with 42 frames in file: /Users/lueth/Desktop/Too
lbox_test/tmp_SG_LIB_EXP_28/synth4Bar3Pose.avi.
```



5. Find a special Solution for the 4Bar-Linkage wit A0 and B0 on same level

If A0 and B0 are to lie on the same plane, B0+k*(P12-B0) must correspond to the Y coordinate of the Y coordinate of A0 in the y coordinate

```
[A0,B0,A1,B1]=synth4Bar3Pose(C1,D1,C2,D2,C3,D3,d);
```

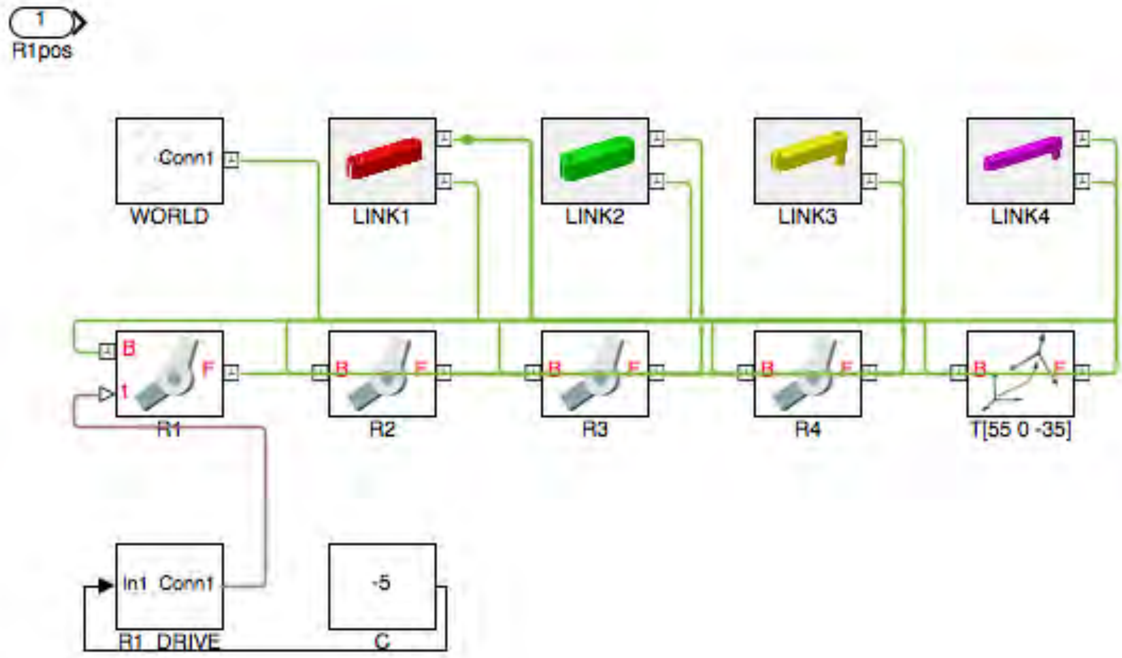
```
% [A0,B0,A1,B1]=exp_2017_01_08(C1,D1,C2,D2,C3,D3,l,d);
A0
B0

L1=norm(B0-A0)
L2=norm(B1-B0)
L3=norm(A1-B1)
L4=norm(A0-A1)

% L1=500; L2=400; L3=500; L4=400

L=[L1 L2 L3 L4]
LMax=find((L==max(L))); LMax=LMax(1);
LMin=find((L==min(L))); LMin=LMin(1);
l1=L(LMin)+L(LMax);
l2=sum(L)-l1;
l3=l1-l2
```

```
A0 =

   55.0000  -35.0000


B0 =

   66.4286   -0.7143


L1 =

   36.1403


L2 =

   26.4382


L3 =

    40


L4 =

   65.1920


L =

   36.1403   26.4382   40.0000   65.1920


l3 =

   15.4899
```

6. Create a SimMultiyBody System for the calculated solution

```
A=SGmodelLink(L1,'',1,2); A=SGmodelLink2(L1,0,1);
B=SGmodelLink(L2,'',1,2); B=SGmodelLink2(L2,0,1);
C=SGmodelLink(L3,'',1,2); C=SGmodelLink2(L3,0,-1);
D=SGmodelLink(L4,'',1,2); D=SGmodelLink2(L4,0,-1);

smbCreateSG (A,'LINK1','r');                % Add long rod as LINK1
smbCreateSG (B,'LINK2','g');                % Add short rod as LINK2
smbCreateSG (C,'LINK3','y');                % Add long rod as LINK3
smbCreateSG (D,'LINK4','m');                % Add short rod as LINK4
smbCreateJoint ('R','R1','LINK1.F','LINK2.B'); % Add a RR Joint
smbCreateJoint ('R','R2','LINK2.F','LINK3.B'); % Add a RR Joint
smbCreateJoint ('R','R3','LINK3.F','LINK4.B'); % Add a RR Joint
smbCreateJoint ('R','R4','LINK4.F','LINK1.B'); % Add a RR Joint
phi=atan2(B0(2)-A0(2),B0(1)-A0(1))
% phi=0;
smbCreateConnection('WORLD.ORIGIN','LINK1.B',TofR(rot(0,0,phi),[A0(1) 0 A0(2)])); % Connect
 Linkage to World Frame
smbCreateDrive ('R1');
smbSetJointInputTorque('R1');
smbCreateBlockConst('C','R1_DRIVE/1',-5)
ID=smbDrawNow;
smbSimulate(4);
```

```
phi =

    1.2490
```

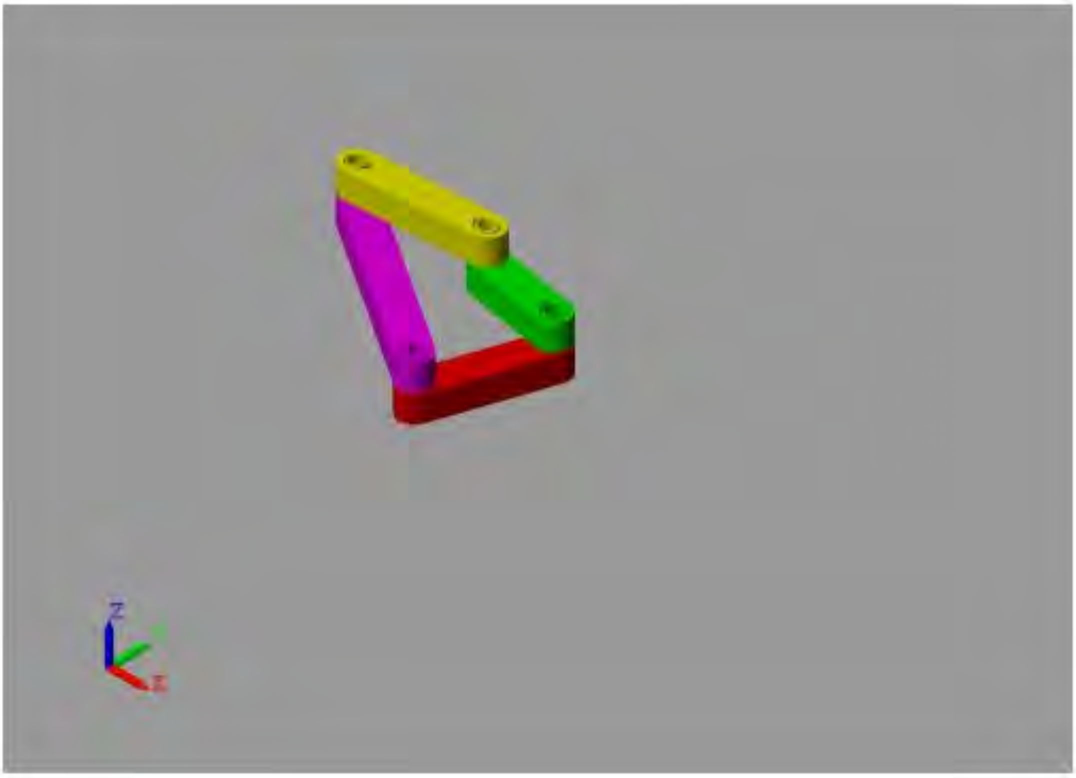


7. Show the Video of the Simulation

```
[I1,FN]=smbVideoSimulation (4);    % Simulate for 1 second
IT=imageVideoTitle(FN,{'SG-Lib Tutorial #27','3 Pose Syntheses','Tim C. Lueth','$date'},'',
[0.44 0.74 1.14]);
IE=imageVideoEndtitle(FN);
videoWriteClipMovie(smbFilename('3 Pose Syntheses SimMultiBody.avi'),IT,2,FN2,ID,1,FN,IE,1)
;
imshow(I1);
```

```
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
```

```
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Creating a new video file (NO SOUND/2016b): '/Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_E
XP_28/3 Pose Syntheses SimMultiBody.avi'
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100% Warning: A
value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100%
```

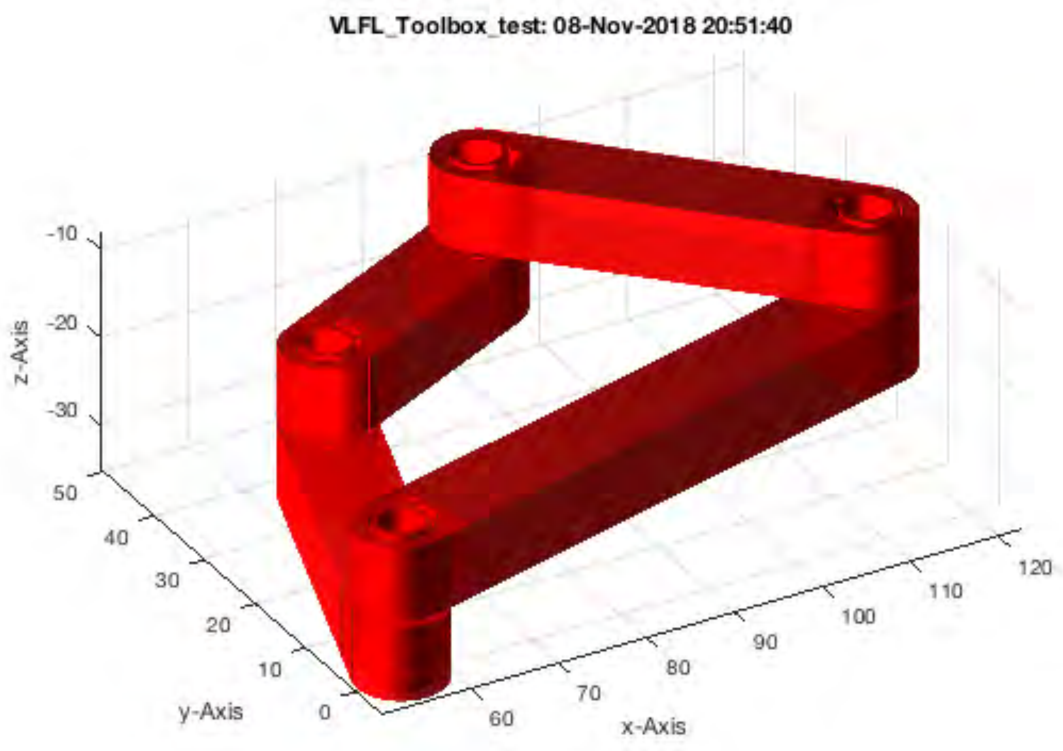


8. Now Create the Solid Geoemtry at time 0.78 seconds

```
SG=smbFullModelSimulation(0.74);
% SG=SGmagnifyVL(SG,'',[100 100 100]);
SGwriteSTL(SG,smbFilename('3-Pose-Synth'));
```

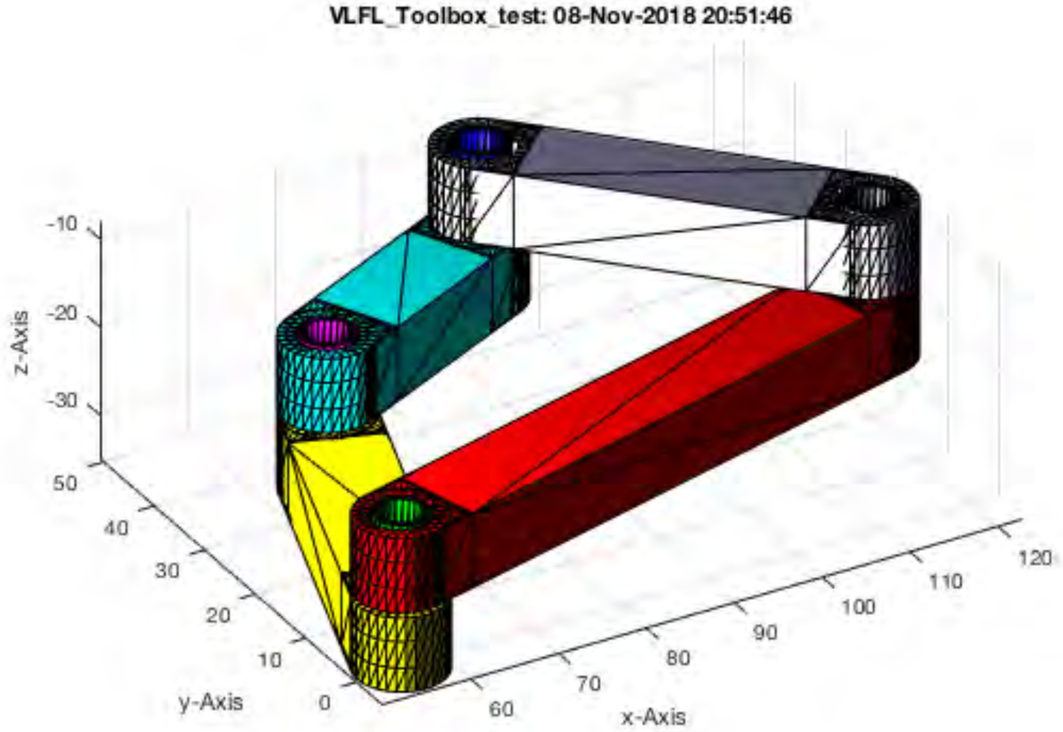
```
CREATING A FULL SOLID-MOVEMENT SIMULATION-MODEL 'SG_LIB_EXP_28' THAT RUNS At LEAST 0.74 SEC
ONDS
============================================================================================
========
Adding frame sensors for all solids of the model
```

```
Add frame sensors for 'LINK1.SG'
Add frame sensors for 'LINK2.SG'
Add frame sensors for 'LINK3.SG'
Add frame sensors for 'LINK4.SG'
===========================================================================================
========

simOut =

  Simulink.SimulationOutput:
                  simlog: [1x1 simscape.logging.Node]
                    sout: [1x1 Simulink.SimulationData.Dataset]
                    tout: [220x1 double]
                    xout: [1x1 Simulink.SimulationData.Dataset]

      SimulationMetadata: [1x1 Simulink.SimulationMetadata]
            ErrorMessage: [0x0 char]

LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_28/sbm_temp_LINK1
.stl
Header:
Number of facets: 2404
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_28/sbm_temp_LINK2
.stl
Header:
Number of facets: 2404
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_28/sbm_temp_LINK3
.stl
Header:
Number of facets: 2584
0..
LOADING BINARY STL-File: /Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_28/sbm_temp_LINK4
.stl
Header:
Number of facets: 2584
0..

CREATED A SOLID GEOMETRY OF THE FULL SIMULATION-MODEL 'SG_LIB_EXP_28' AT TIME: 0.74 SECONDS
===========================================================================================
========
```




```
SGfigure; view(-30,30); SGplot(SG,'m');  SGanalyzeGroupParts(SG);
```

```
4% 8% 12% 16% 20% 24% 28% 32% 36% 40% 44% 48% 52% 56% 60% 64% 68% 72% 76% 80% 84% 88% 92% 9
6% 100%
SGanalyzeGroupParts: 8 separated parts found.
```



Final Remarks

```
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:51:47!
Executed 08-Nov-2018 20:51:49 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
===========================================================================================
==========
```

Published with MATLAB® R2018a

Tutorial 29: Create a multi body simulation using several mass points

2017-03-17: 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

Motivation for this tutorial: (Originally SolidGeometry 3.6 required)

```
% function VLFL_EXP29
```

Motivation for this tutorial

Showing a finite element mass spring system

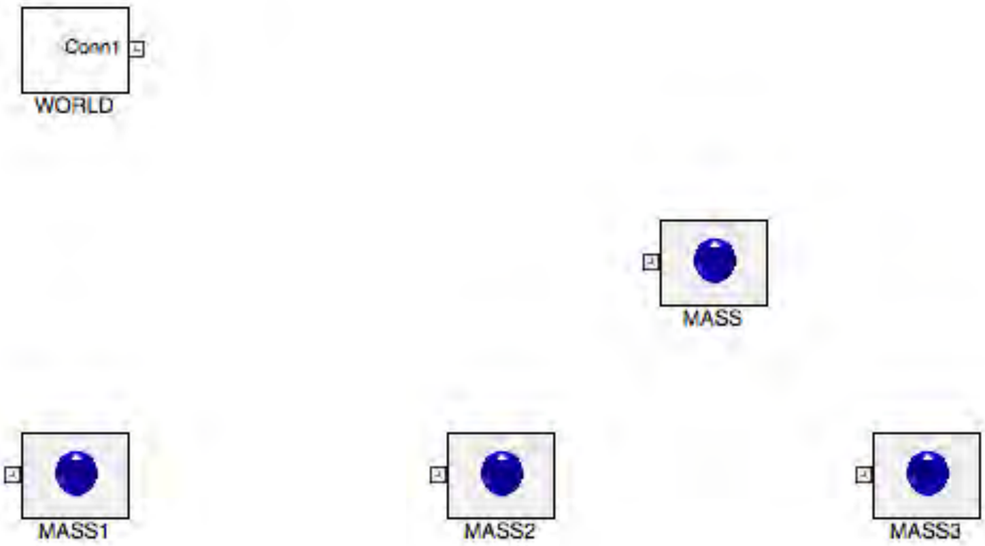
1. Create a SimMultiBody system for a Mass - Spring - Damper - System

```
smbNewSystem ('SG_LIB_EXP_29',[0 0 -9.81])    % Creates the mechansim diagramm
```

```
Creating temporary directory '/Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_EXP_29/'
```

2 Create four mass points

```
smbCreateSGMass;
smbCreateSGMass;
smbCreateSGMass;
smbCreateSGMass;
smbDrawNow;
```

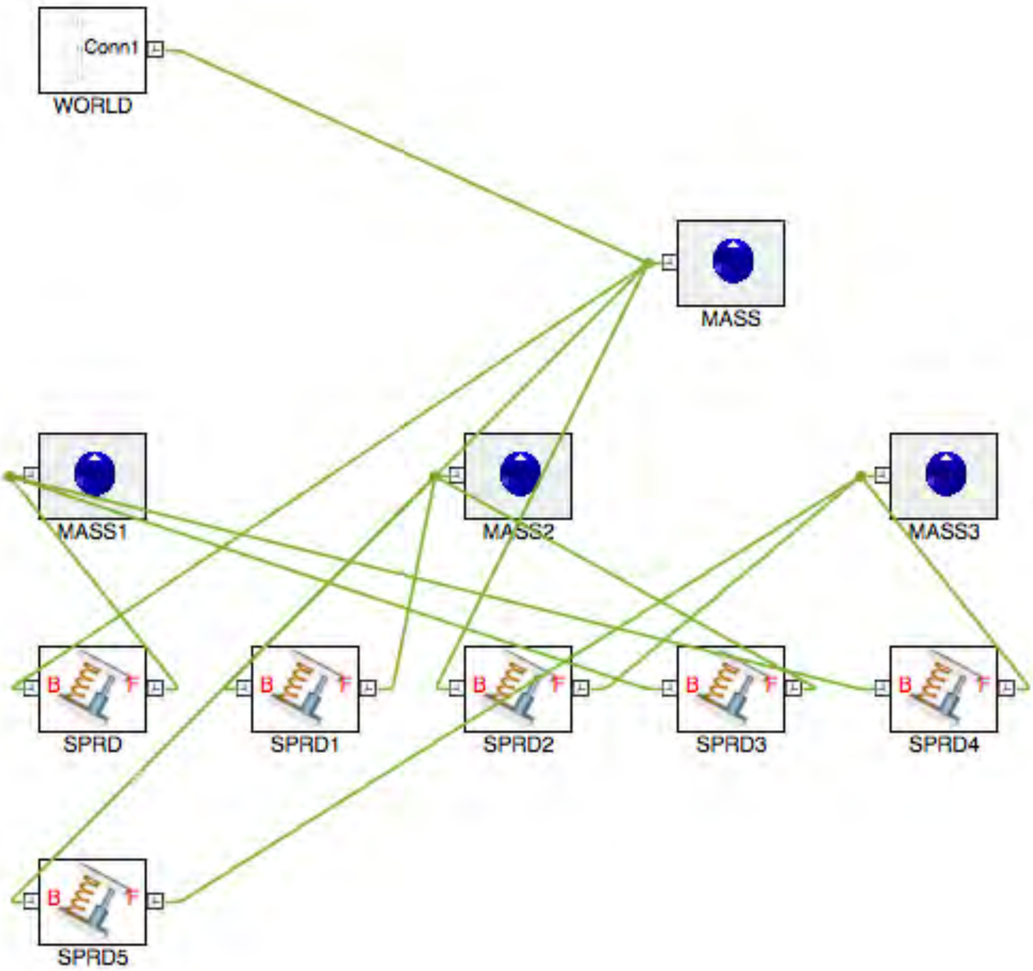


2 Create six springs between the masses

```
smbCreateSpring('MASS','MASS1');
smbCreateSpring('MASS','MASS2');
smbCreateSpring('MASS','MASS3');
smbCreateSpring('MASS1','MASS2');
smbCreateSpring('MASS1','MASS3');
smbCreateSpring('MASS2','MASS3');
```

3. Connect the mass - spring - damping system to the world coordinate system

```
smbAddLine( 'WORLD/RConn1', 'MASS/LConn1');
ID=smbDrawNow;
```



4. Show the Simulation

6. Create a Video of the Linkage Simulation

```
[I1,vname]=smbVideoSimulation (4);    % Simulate for 1 second
IT=imageVideoTitle(vname,{'SG-Lib Tutorial #29','Mass-Spring-Nets','Tim C. Lueth','$date'},
'',[0 4]);
IE=imageVideoEndtitle(vname);
videoWriteClipMovie(smbFilename('SG-Lib Tutorial #29-Mass-Spring-Nets.avi'),IT,2,ID,1,vname
,IE,1);
imshow(I1);
```

```
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
```

```
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
.Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
Creating a new video file (NO SOUND/2016b): '/Users/lueth/Desktop/Toolbox_test/tmp_SG_LIB_E
XP_29/SG-Lib Tutorial #29-Mass-Spring-Nets.avi'
Warning: A value of class "matlab.internal.video.PluginManager" was indexed with
no subscripts specified. Currently the result of this operation is the indexed
value itself, but in a future release, it will be an error.
5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100%
```

Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:52:16!
Executed 08-Nov-2018 20:52:18 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
===========================================================================================
==========
```

Published with MATLAB® R2018a

Tutorial 30: Creating graphical drawings using point, lines, surfaces, frames etc.

2017-02-10: 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

Motivation for this tutorial: (Originally SolidGeometry 3.4 required)

In this tutorial, features are introduced that allow you to create drawings for the descriptive geometry using Matlab. The goal is to display lines, planes, spaces and polygons, surfaces and surface-bounded volumes and to label them mathematically (Tex-style). some functions are based in individual points such as:

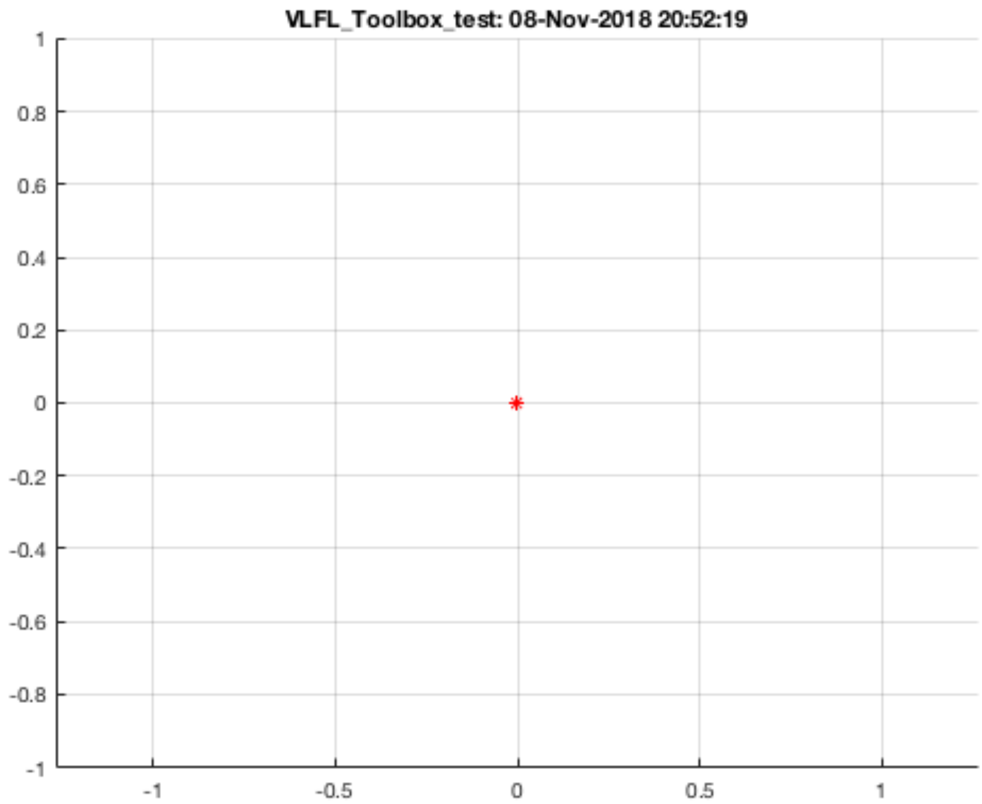
- pplot - plot a point in defined color, shape and size
- lplot - plot a line between two points with color, width, tip, start point end point
- aplot - plot an angle at a point using a line and a second line or angle
- slplot - plot a staight line using a start point and direction vector
- tplot;
- tfplot; some functions are based on point lists (PL) or vertex lists (VL), such as: PLplot, VLplot,

1. Plotting points (PL) and vertices (3D)

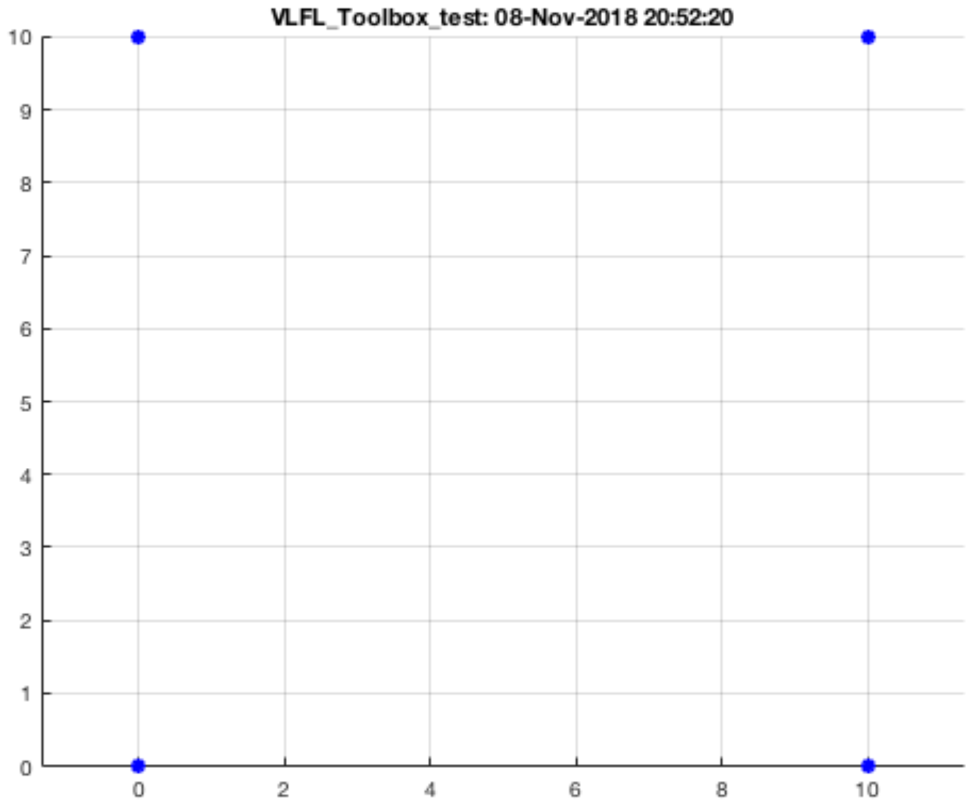
```
p=[0 0]       % row style
p=[0;0]       % column style
v=[0 0 0]   % row style
v=[0; 0; 0]   % column style


SGfigure; pplot(p);
```

```
p =

     0     0


p =
```

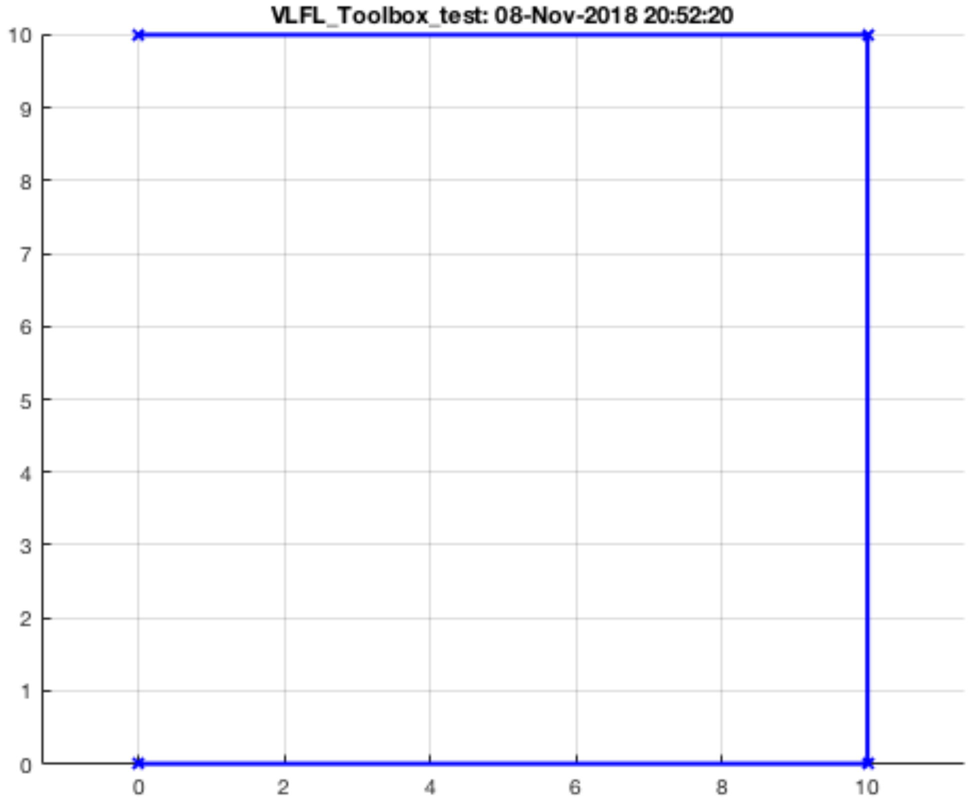
```
    0
    0


v =

    0    0    0


v =

    0
    0
    0
```



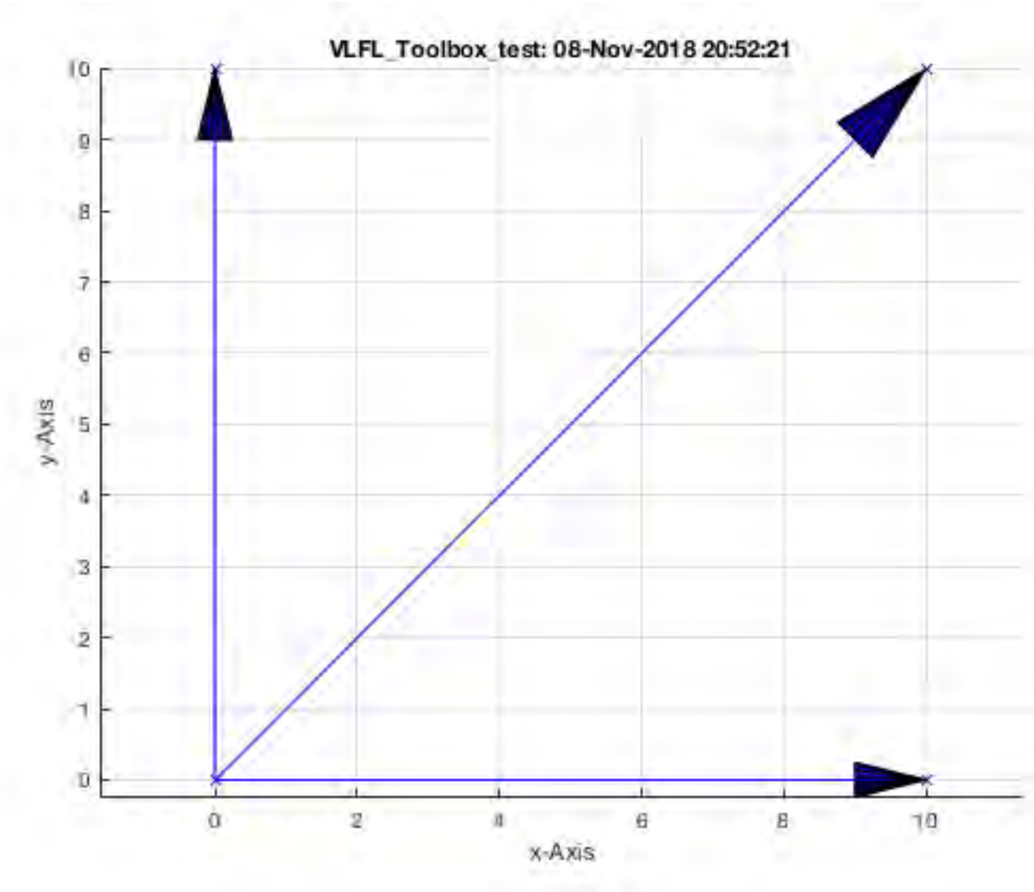

```
SGfigure; PLplot([0 0;10 0;10 10; 0 10],'b*',2);          % point plot of point list
```



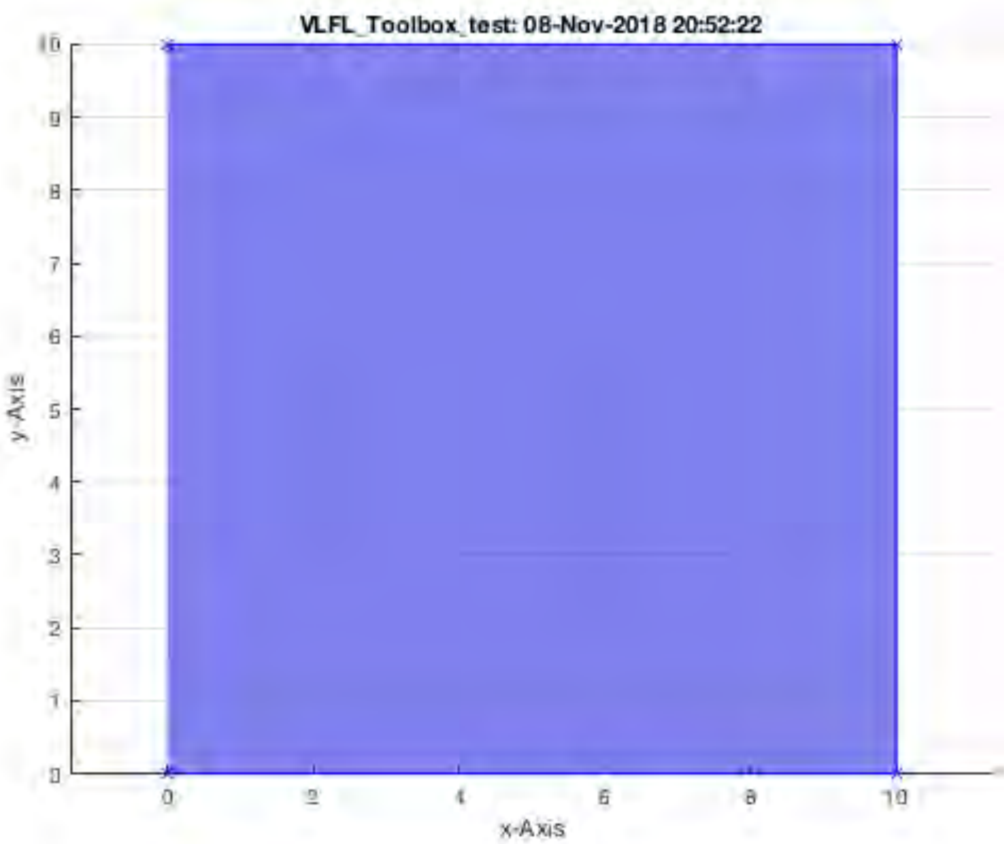

```
SGfigure; PLplot([0 0;10 0;10 10; 0 10],'bx-',2);          % Line plot of point list
```



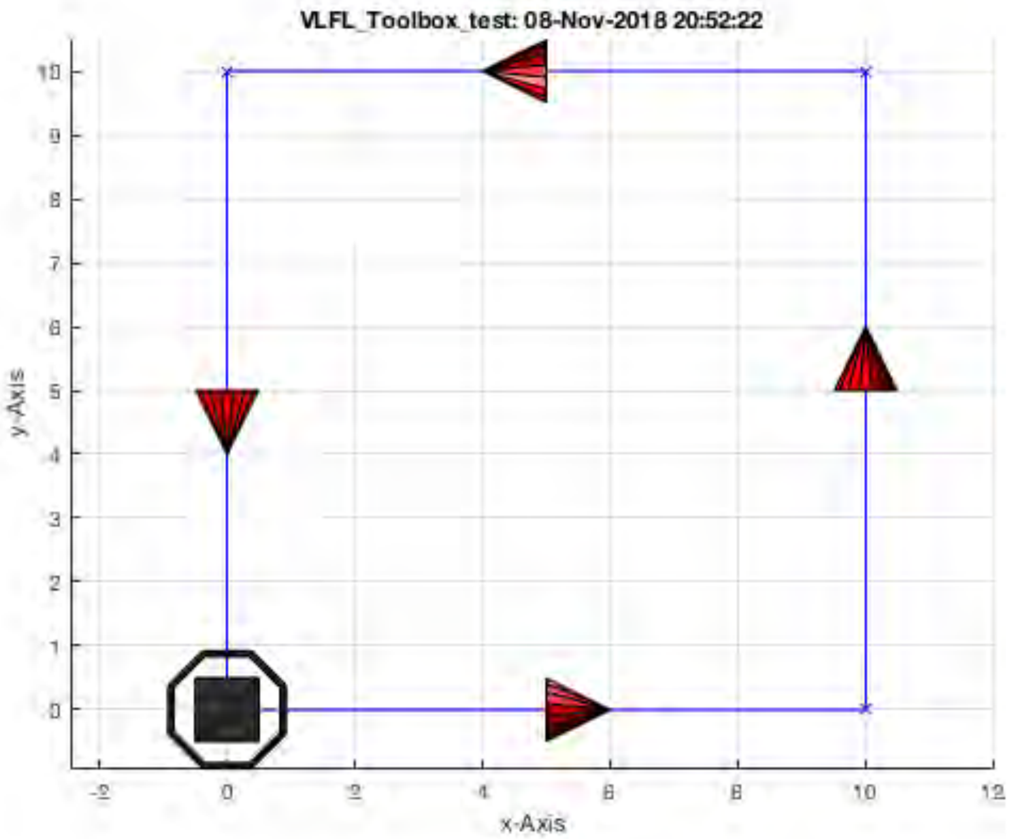

```
SGfigure; PLplot([0 0;10 0;10 10; 0 10],'bx-',1,1);         % Vector plot of point list
```



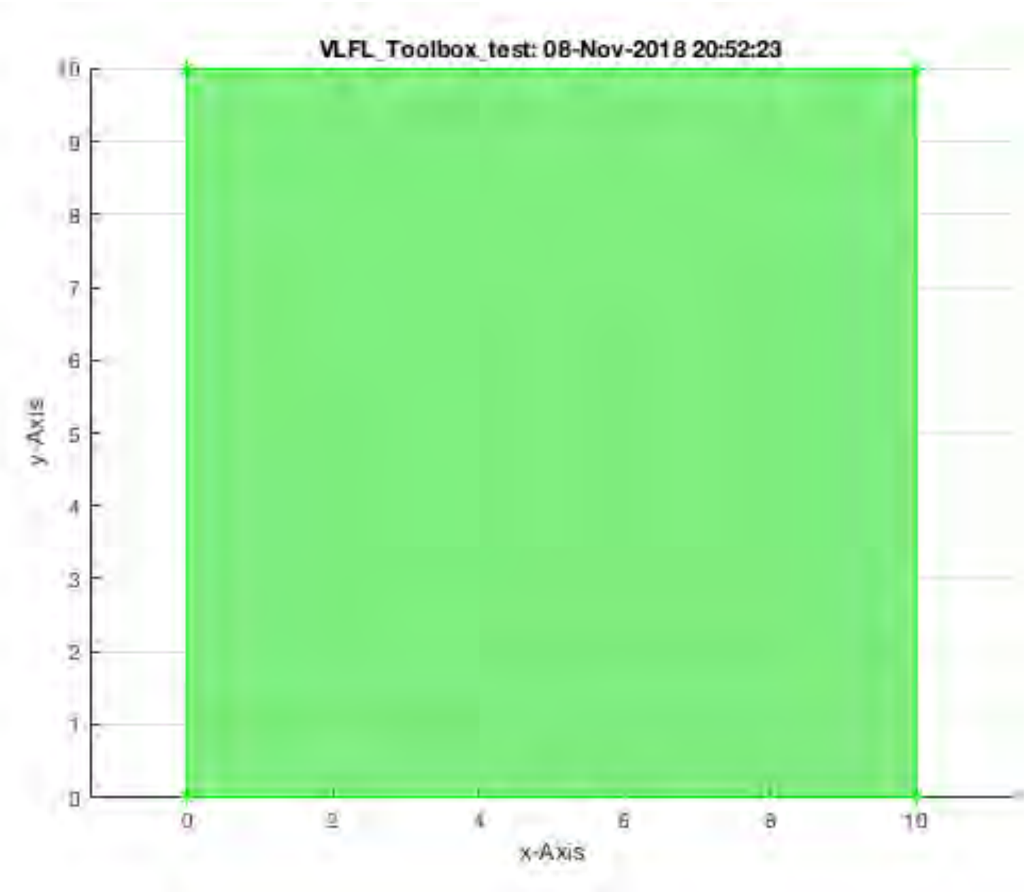

```
SGfigure; PLplot([0 0;10 0;10 10; 0 10],'bx-',1,'',0.5);     % Surface enlosed by point list
```




```
SGfigure; CPLplot([0 0;10 0;10 10; 0 10],'bx-',1,1,1,1);    % Plotting closed polygon
```

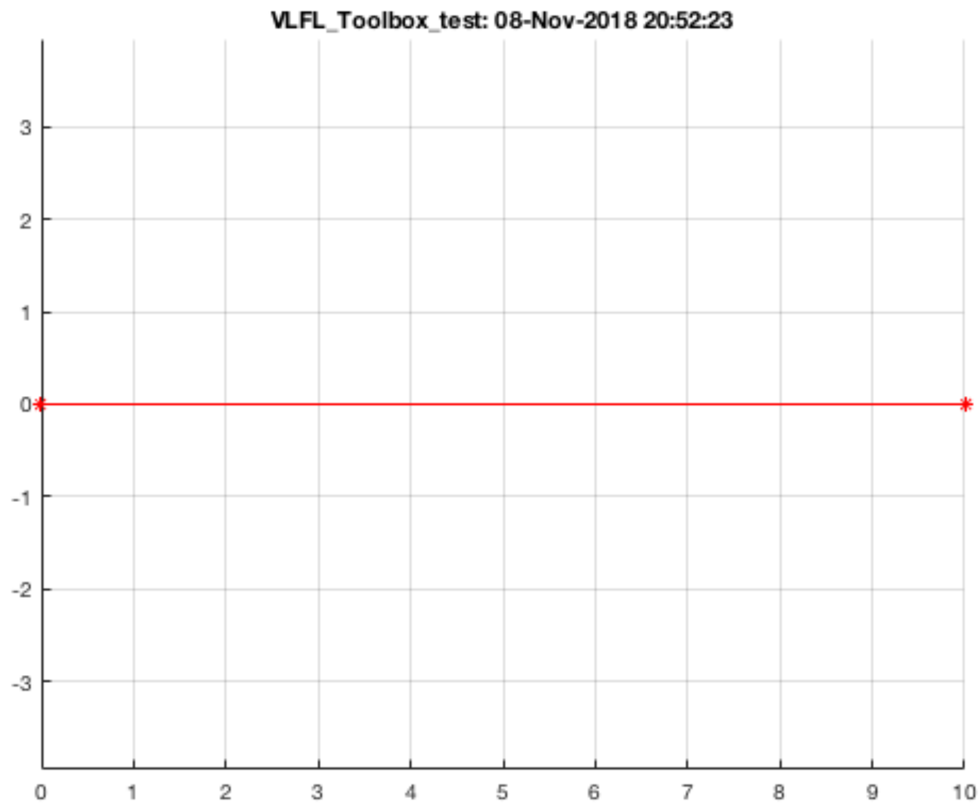



```
SGfigure; CPLfaceplot([0 0;10 0;10 10; 0 10],'g*-',1,0.5);  % Plotting closed polygon surfa
ces
```

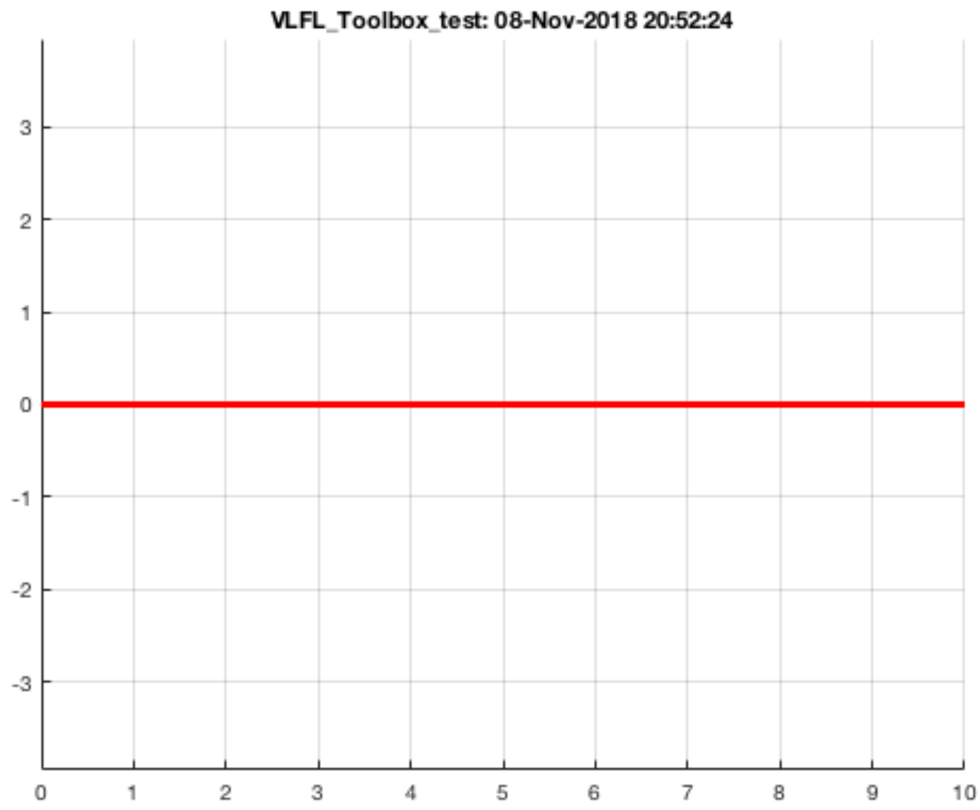


2. Plotting lines

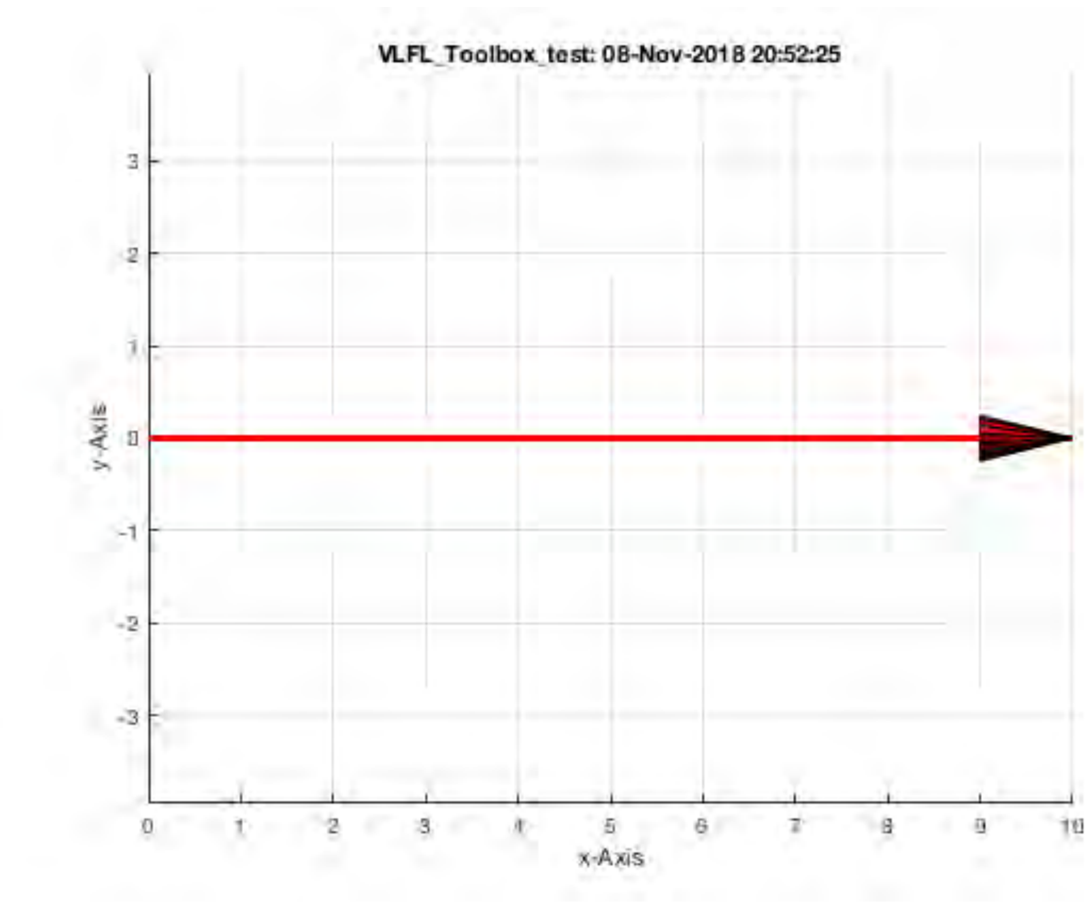
```
SGfigure; lplot([0 0],[10 0],'r*-');
```



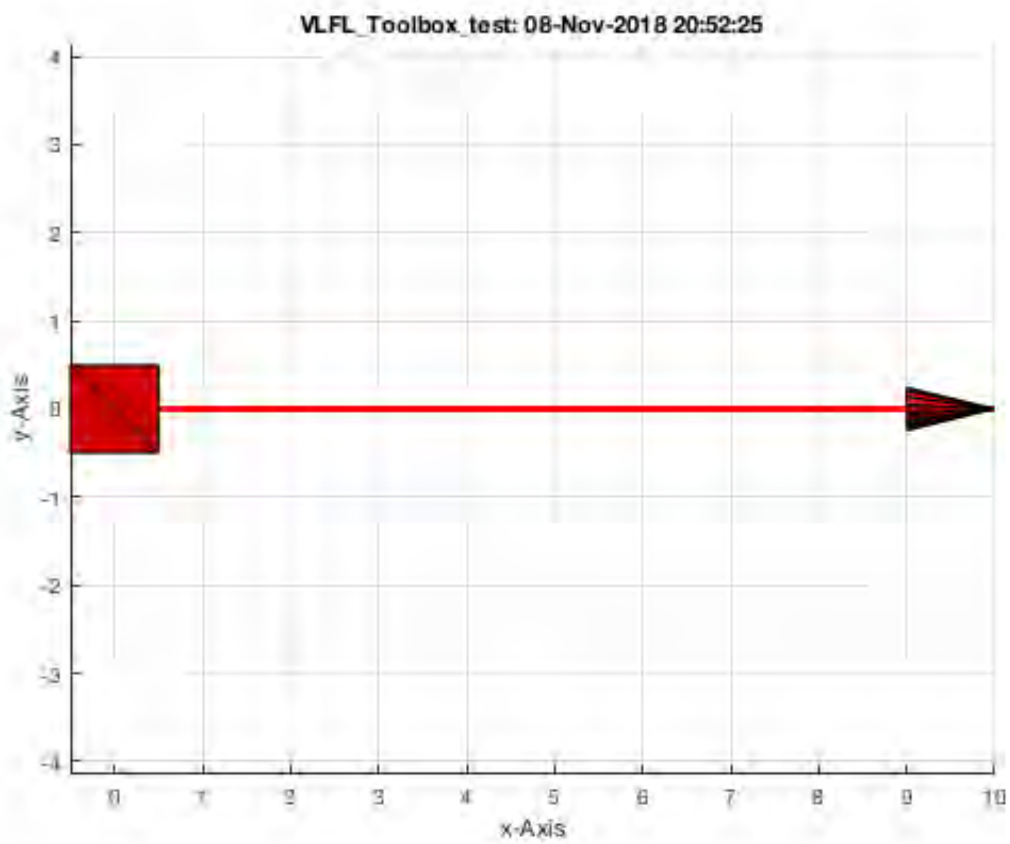

```
SGfigure; lplot([0 0],[10 0],'r-',3);
```




```
SGfigure; lplot([0 0],[10 0],'r-',3,1);
```

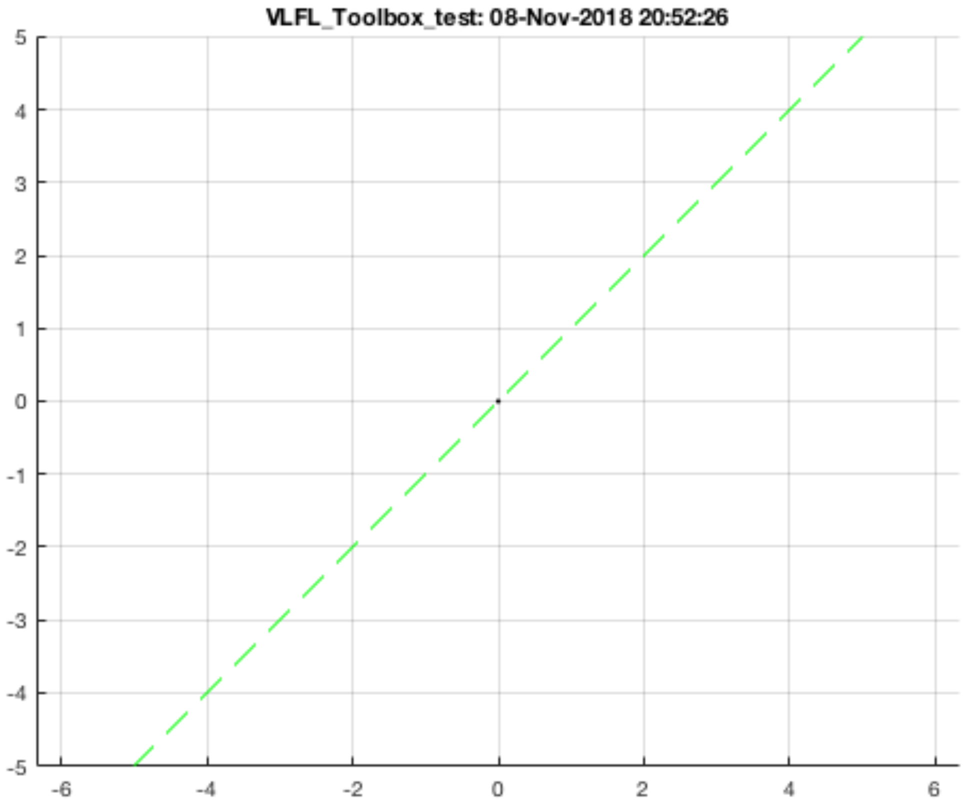



```
SGfigure; lplot([0 0],[10 0],'r-',3,1,1);
```



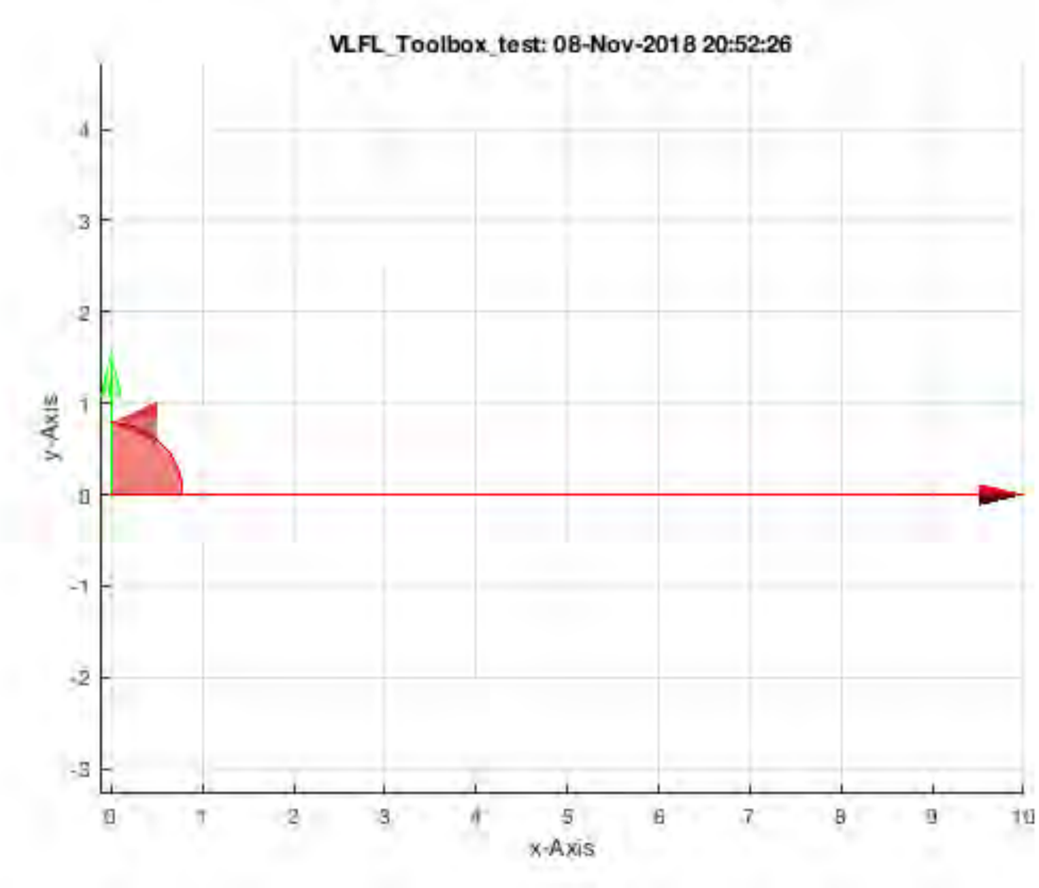
slplot([0 0],[10 0],'r-',3,1,1);

```
SGfigure; slplot([0 0 0],[1 1 1],'color','g--')
```



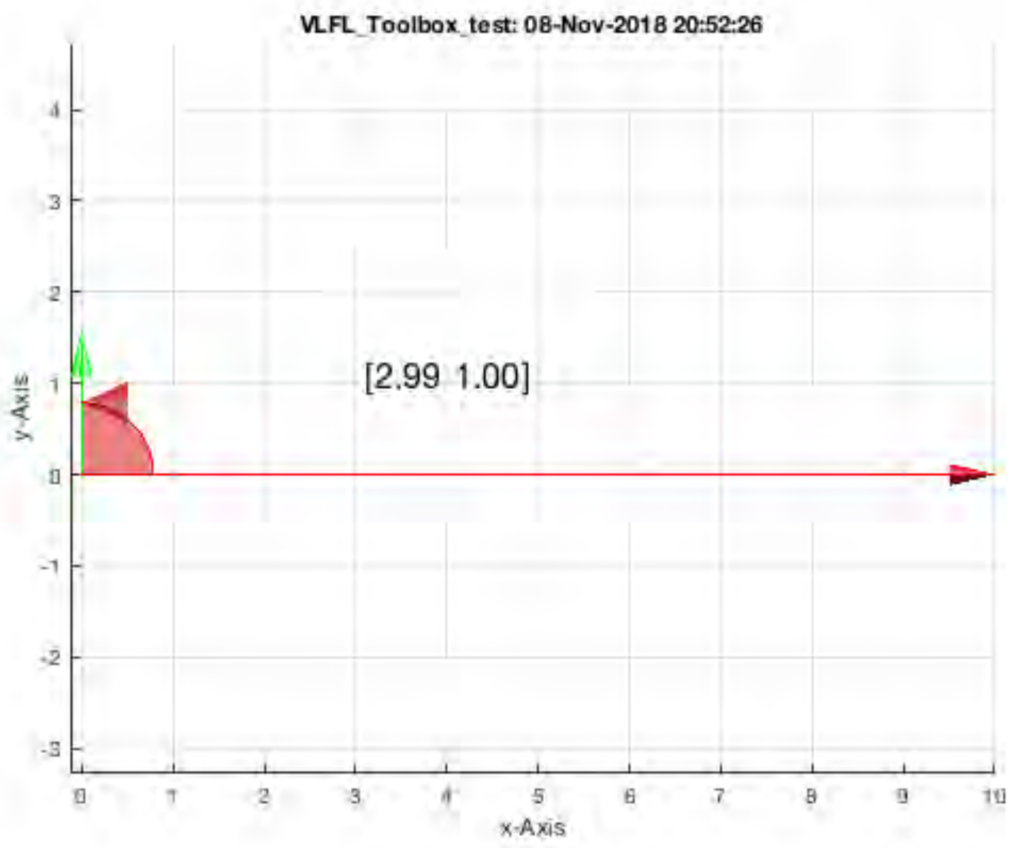
3. Plotting angles

```
SGfigure; aplot([0 0],[10 0],pi/2);
```



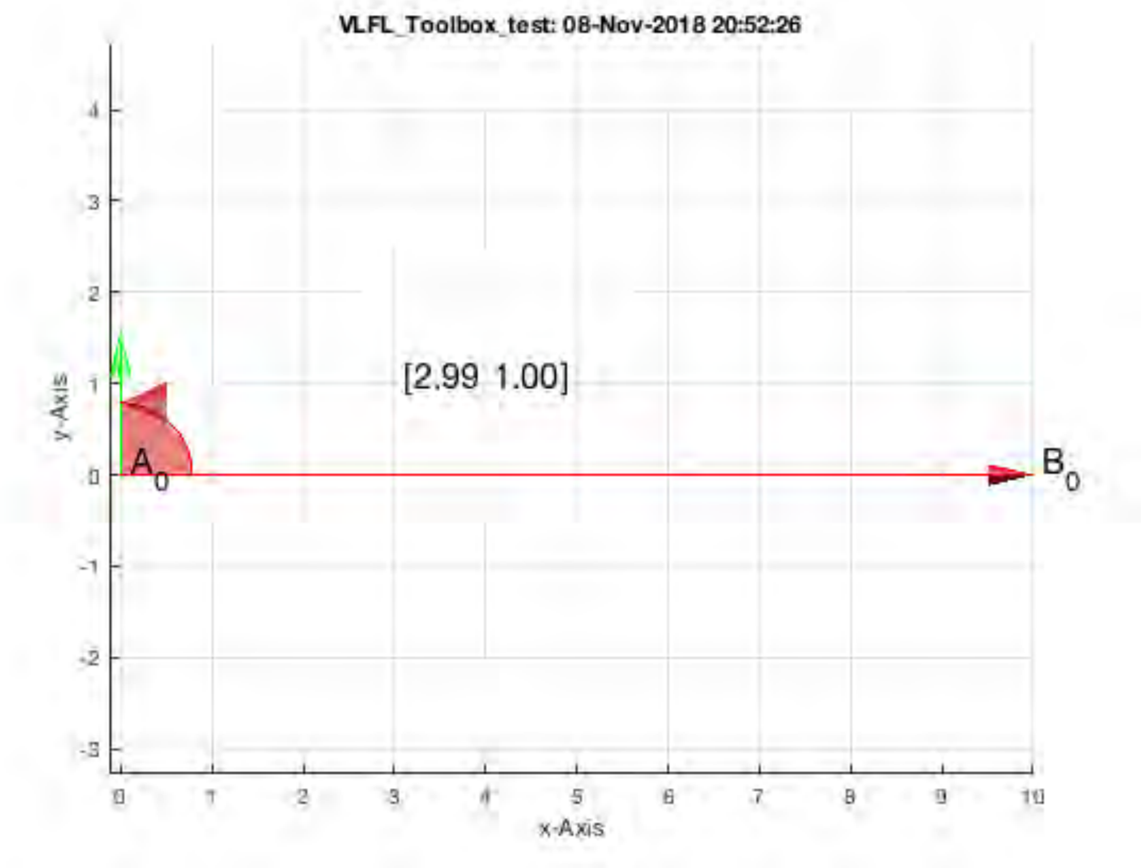
4. Plotting coordinate

```
p=ginput(1); textP (p,sprintf('[%.2f %.2f]',p));
```



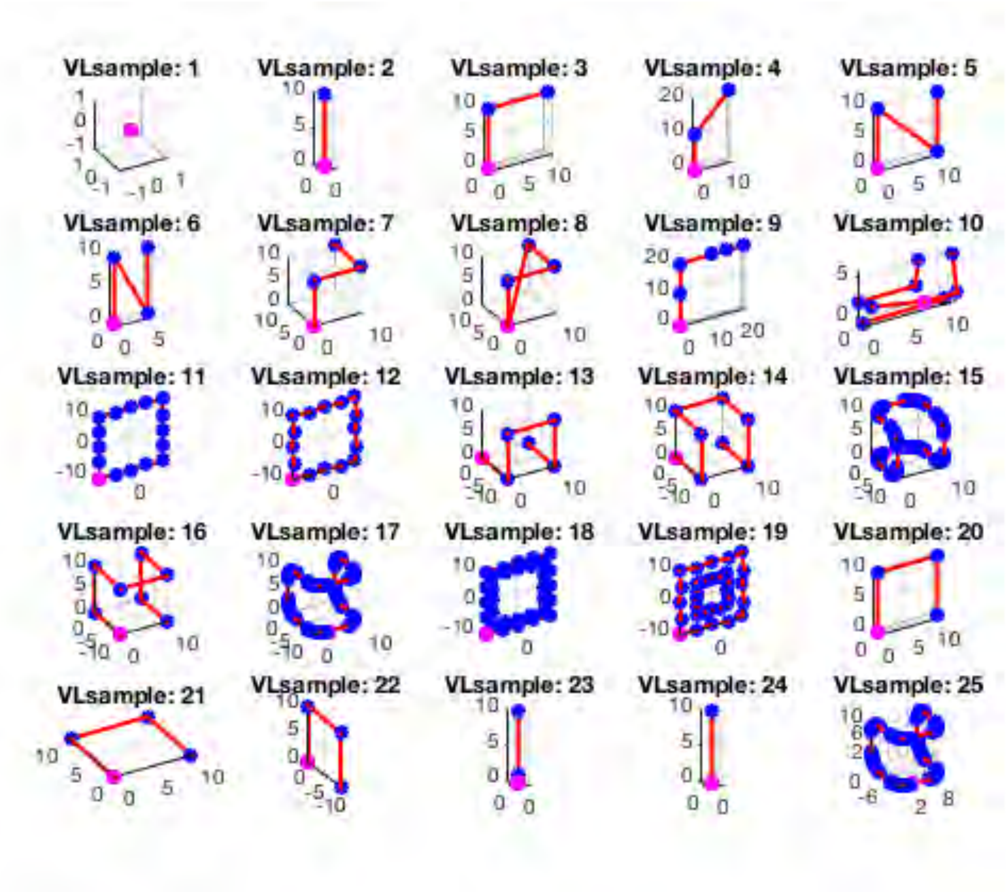
8. Adding text to the drawings

```
textP ([0 0],'A0'); textP ([10 0],'B0');
```

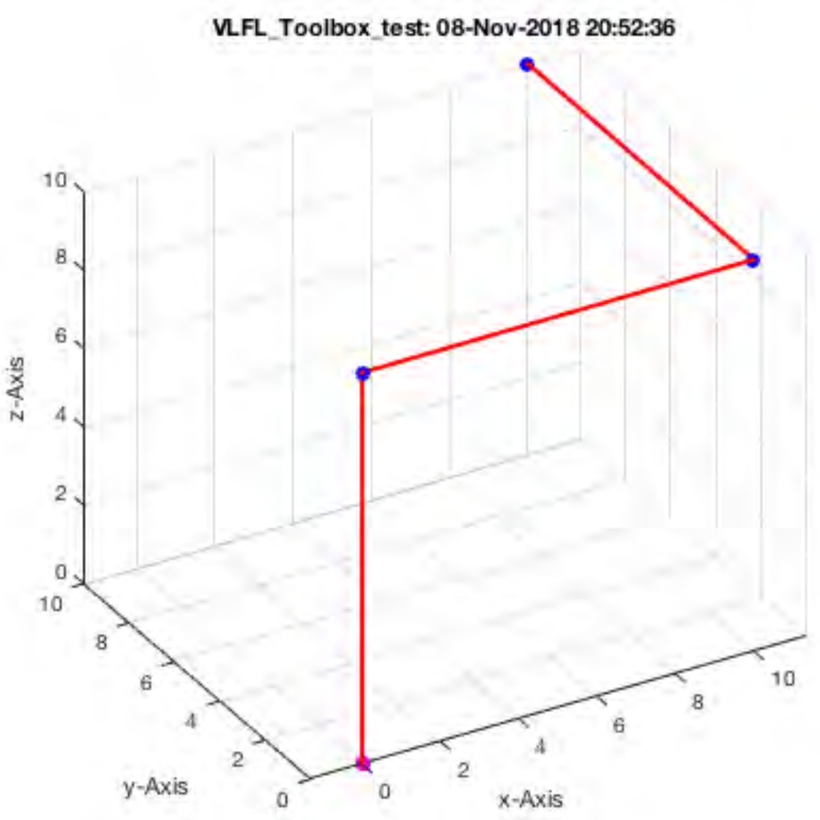


9. Helpful generic polygons for

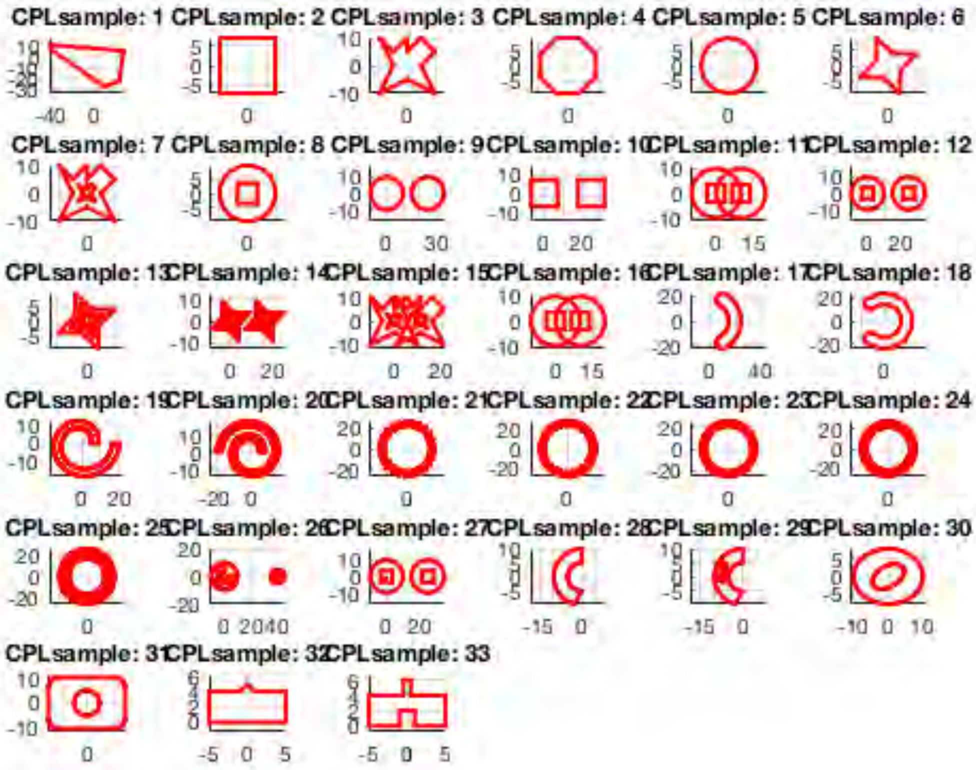
```
SGfigure; VLsample;
```



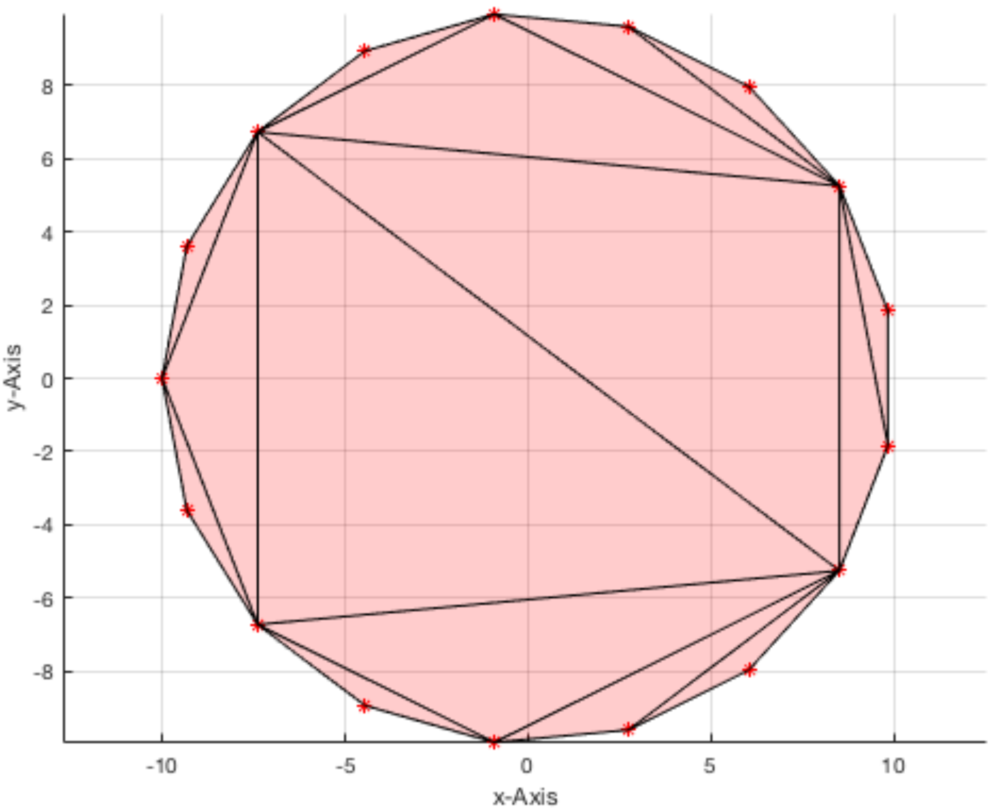

```
SGfigure; VLsample(7);
```


VLFL_Toolbox_test: 08-Nov-2018 20:52:36
z-Axis
y-Axis
x-Axis
0
2
4
6
8
10

```
SGfigure; CPLsample;
```




```
SGfigure; CPLsample(5);
```



Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:52:42!
Executed 08-Nov-2018 20:52:44 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
===========================================================================================
==========
```

Published with MATLAB® R2018a

Tutorial 31: Importing 3D Medical DICOM Image Data and converting into 3D Solids

2017-02-19: 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

Motivation for this tutorial: (Originally SolidGeometry 3.4 required)

- VMreaddicomdir - reads in a voxel model
- VMresize - resizes of a voxel model
- SGofVMdelaunay - creates a surface model using delaunay (slow)
- SGofVMmarchcube - creates a surface model using marching cube (fast)
- SGcut - cuts surface models
- CPLofSGslice - creates a slice contour at a specific height/direction
- PLFLofCPLdelauny - tesselates the facets of a CPL using delaunay
- PLFLofCPLpoly - tesselates the facets of a CPL using mapping toolbox
- VMplot - plots a voxel mode
- VMplotslide - plots a voxel mode for slider navigation
- VMimage - plots a voxel image
- VMmontage - montage of voxel
- VMpseudo3D - creates a pesudo 3D image
- VMuidicom - select and read a voxel model
- VMreaddicom - read a dicom file

1. Reading DICOM models as voxel model from disk and resize voxel models (VM)

```
% load AIM_Patientmodel.mat % Does work world-wide=
```

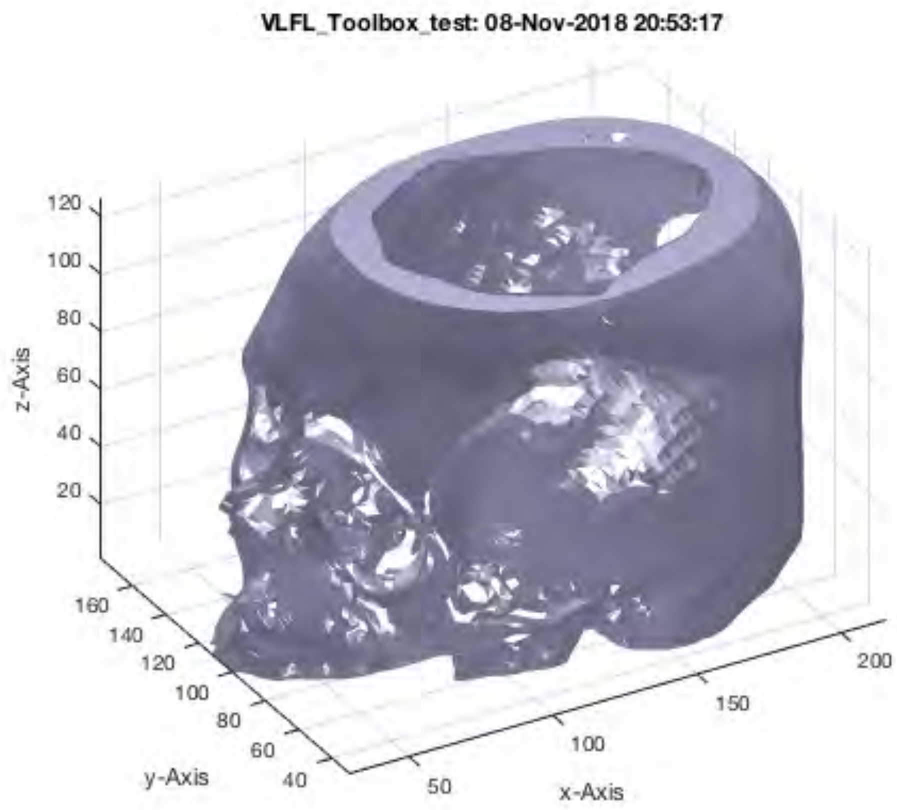
```
[V,vs]=VMreaddicomdir('/Volumes/LUETH-WIN/WIN AIM Matlab Libraries/VLFL-Lib/AIM_DICOMFILES'
);
vs
[a,as]=VMresize(V,[0.5 0.5 0.5],vs);
as
[a,as]=VMresize(V,vs,vs);
as
```

```
VMreaddicomdir: Analyzing 129 entries in path: /Volumes/LUETH-WIN/WIN AIM Matlab Libraries/
VLFL-Lib/AIM_DICOMFILES
............................................................................................
......................................
Stack of 126 DICOM images read from disk.

vs =

    0.4219    0.4219    1.0000

VMresize: Resize voxel image [512 512 126] to [256 256 63] with voxel size [0.84mm 0.84mm 2
.00mm]

as =

    0.8438    0.8438    2.0000

VMresize: Resize voxel image [512 512 126] to [216 216 126] with voxel size [1.00mm 1.00mm
1.00mm]

as =

     1     1     1
```

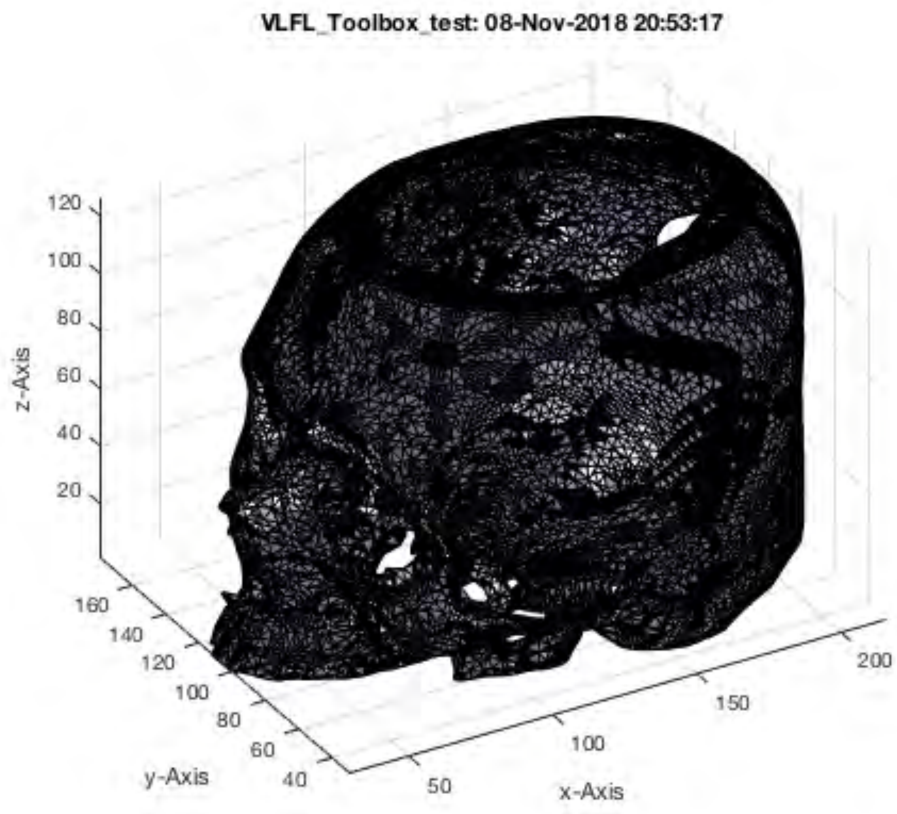
2. Solid skull bone reconstruction using SGofVMdelaunay

```
SG1=SGofVMdelaunay(a>1400,as);       % Takes about 30 seconds
SGfigure; VLFLplotlight (1,1); view(-30,30);
SGplot(SG1,'w');VLFLplotlight(1,1)
```

```
Elapsed time is 6.453011 seconds.
Elapsed time is 10.253265 seconds.
```

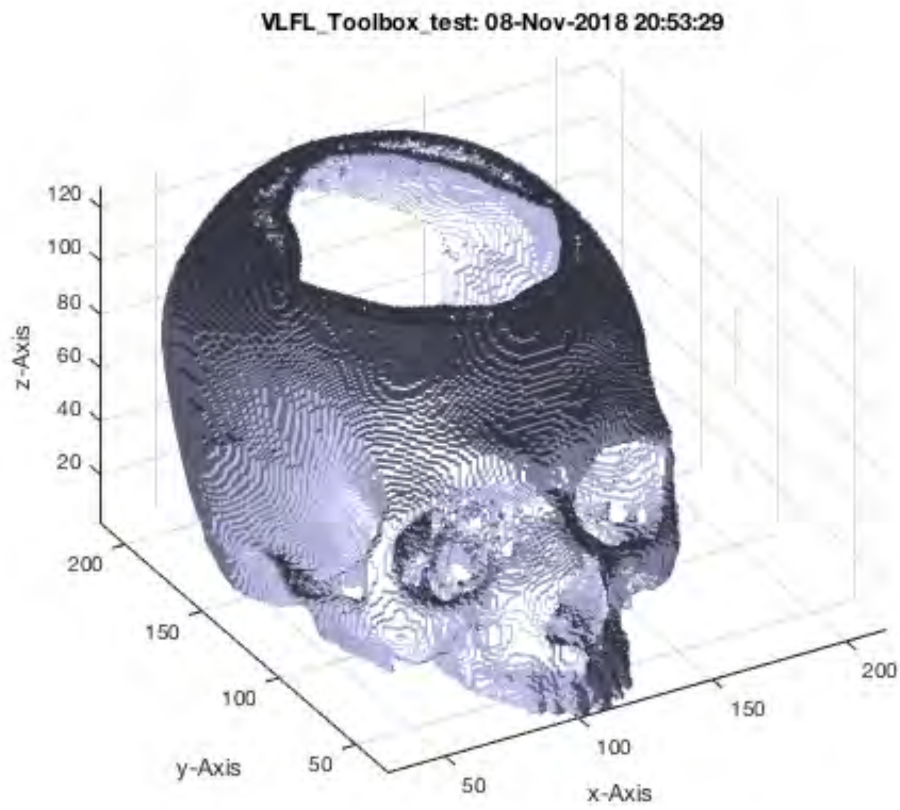



```
VLFLplotlight(0,1);
```

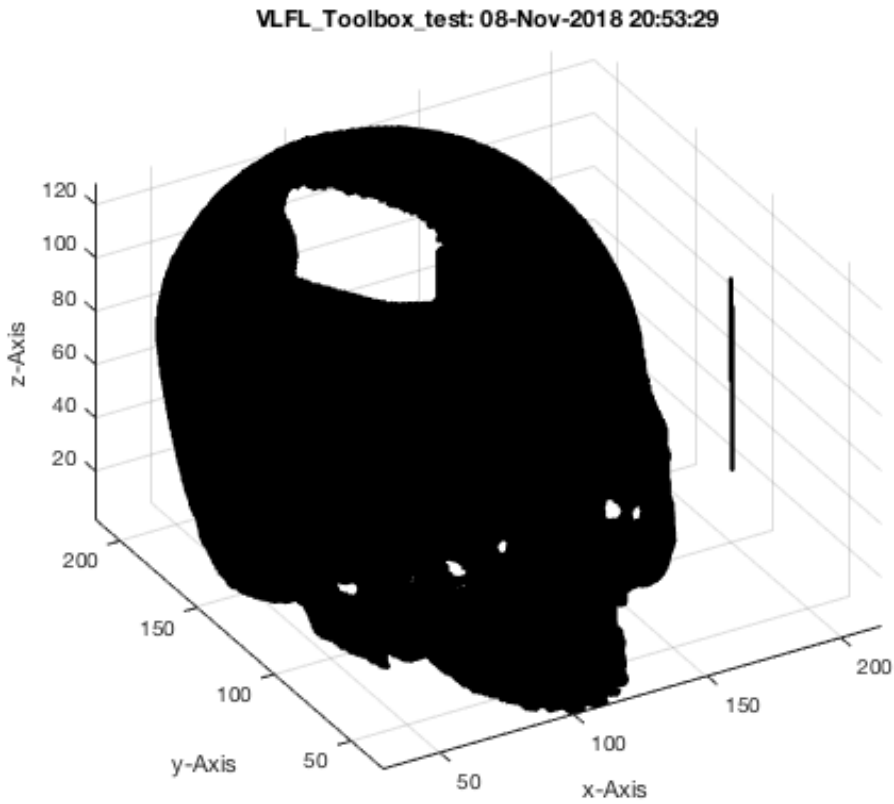


3. Solid skull bone reconstruction using SGofVMisosurface

```
SG2=SGofVMisosurface(a>1400,as);        % Takes about 7 seconds
SGfigure; VLFLplotlight (1,1); view(-30,30);
SGplot(SG2,'w');VLFLplotlight(1,1)
```

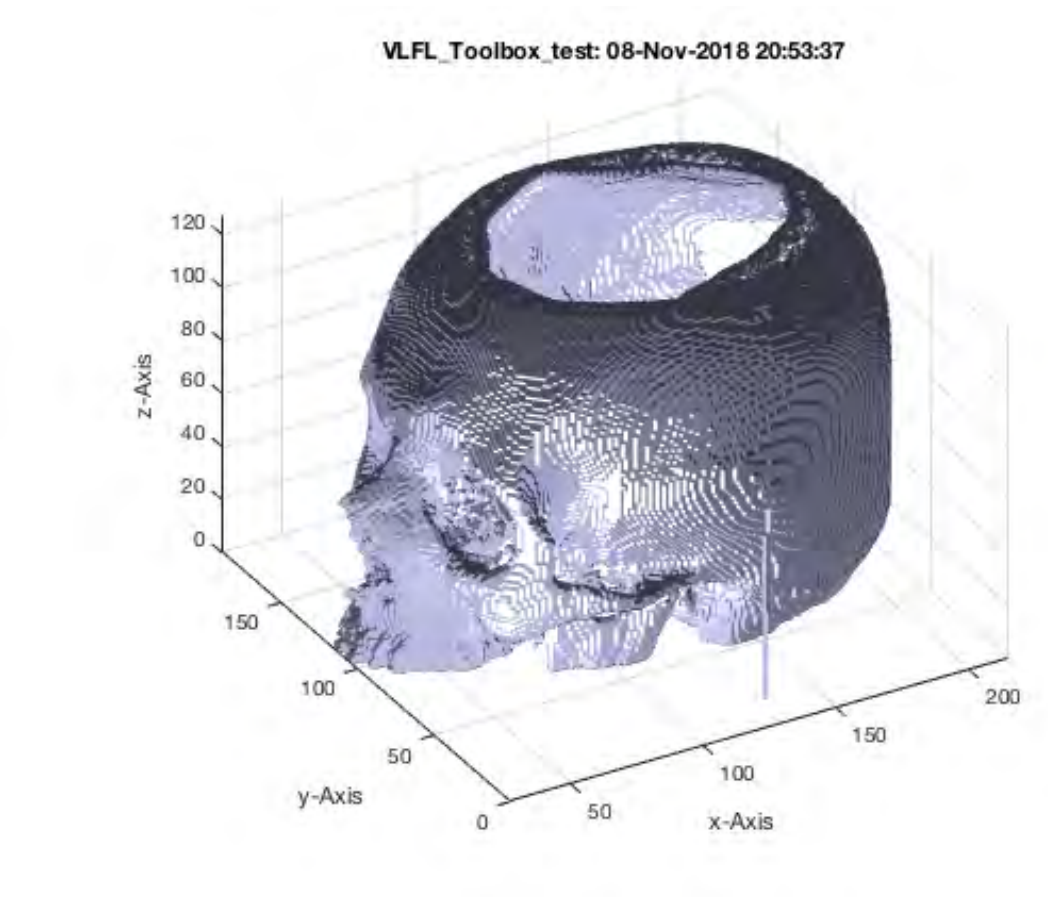



```
VLFLplotlight(0,1);
```

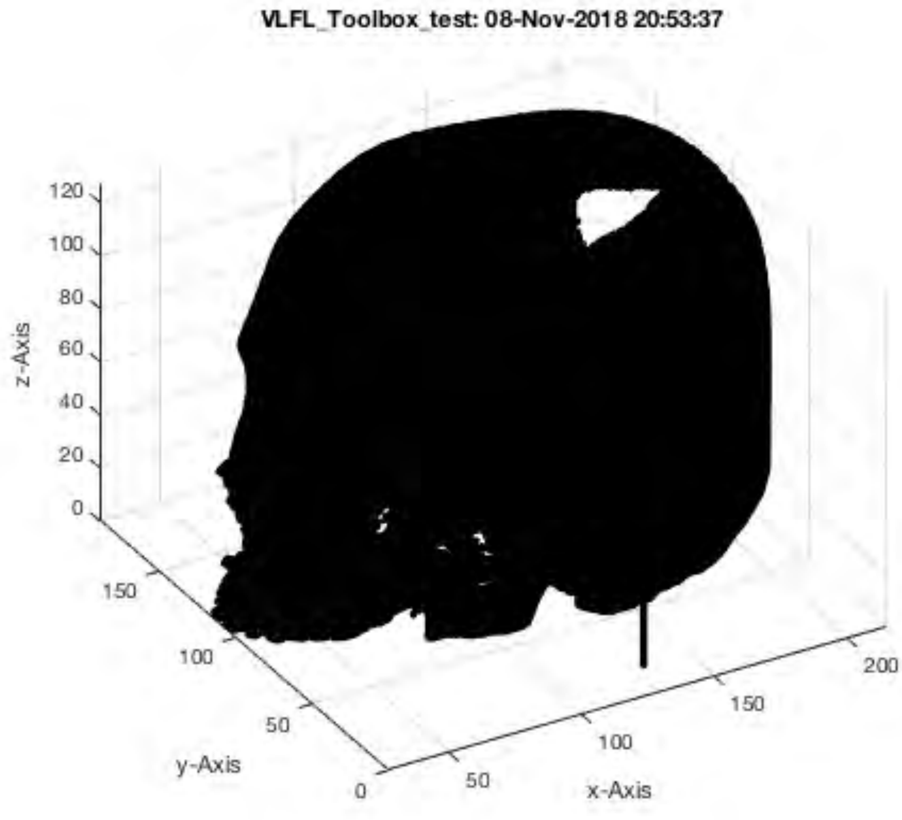


4. Solid skull bone reconstruction using SGofVMmarchcub

```
SG3=SGofVMmarchcube(a>1400,as);        % Takes about 2 seconds
SGfigure; VLFLplotlight (1,1); view(-30,30);
SGplot(SG3,'w');VLFLplotlight(1,1)
```

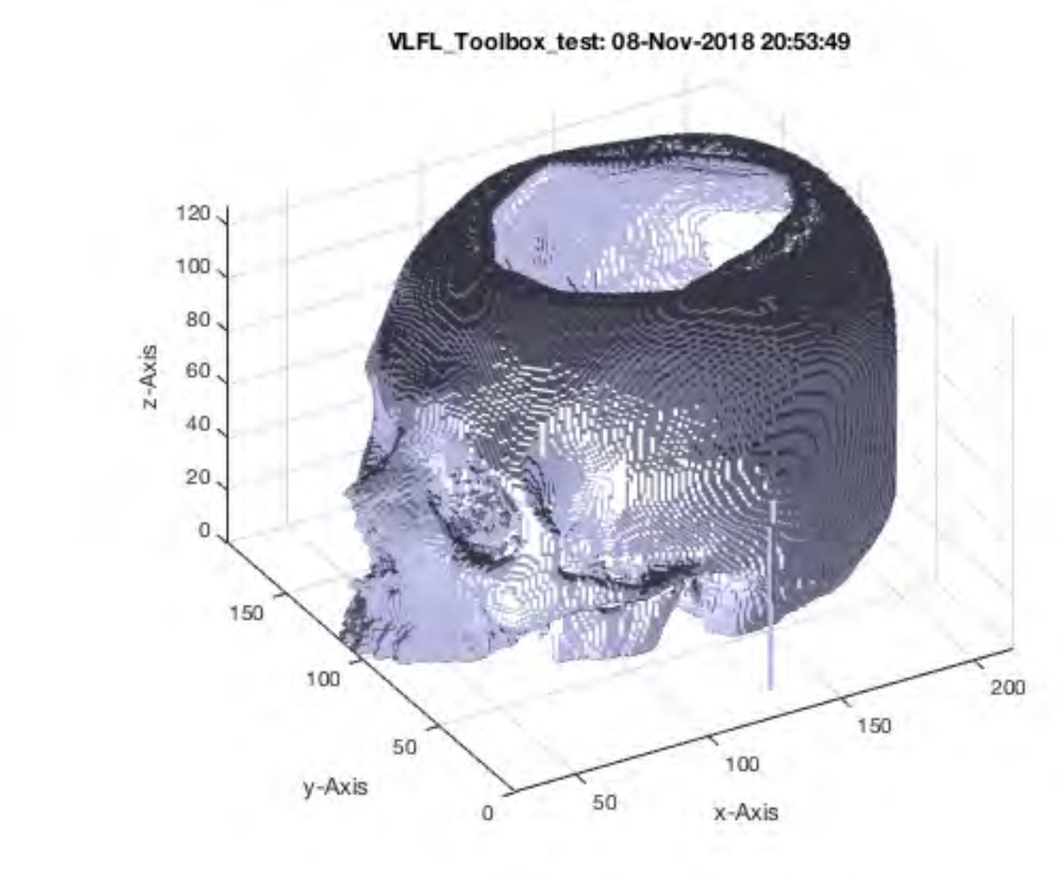



```
VLFLplotlight(0,1);
```

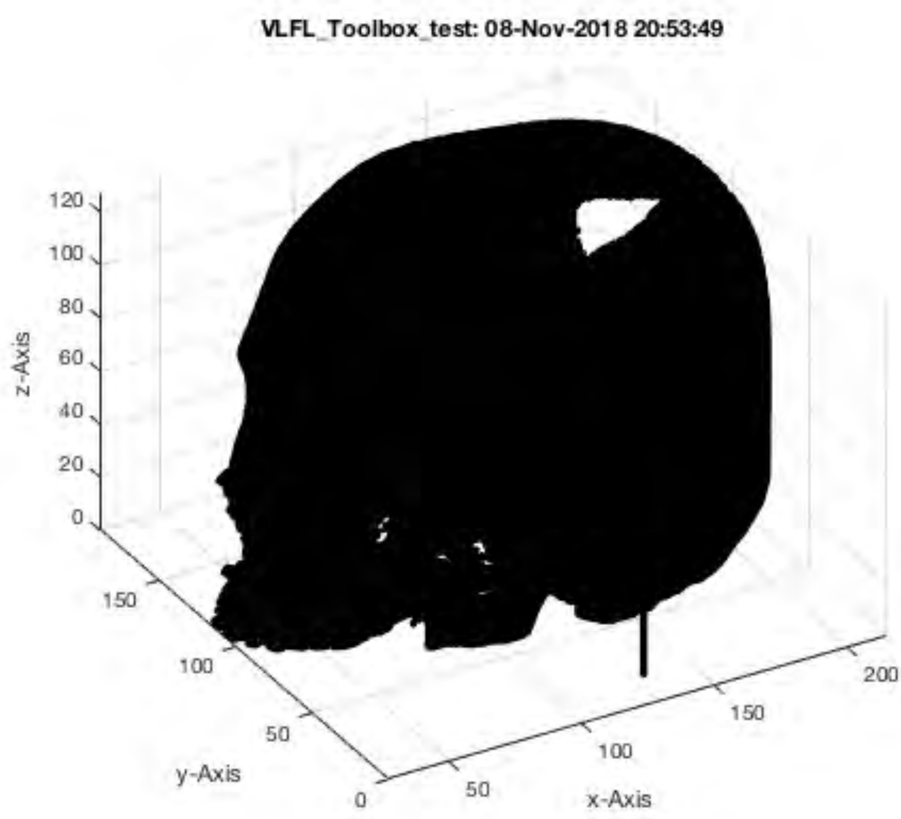


5. Reduce the numbers Facets to 300.000 facets

```
SG4=SGreduceVLFL(SG3,300000); % Takes about 2 seconds
SGfigure; VLFLplotlight (1,1); view(-30,30);
SGplot(SG3,'w');VLFLplotlight(1,1)
```

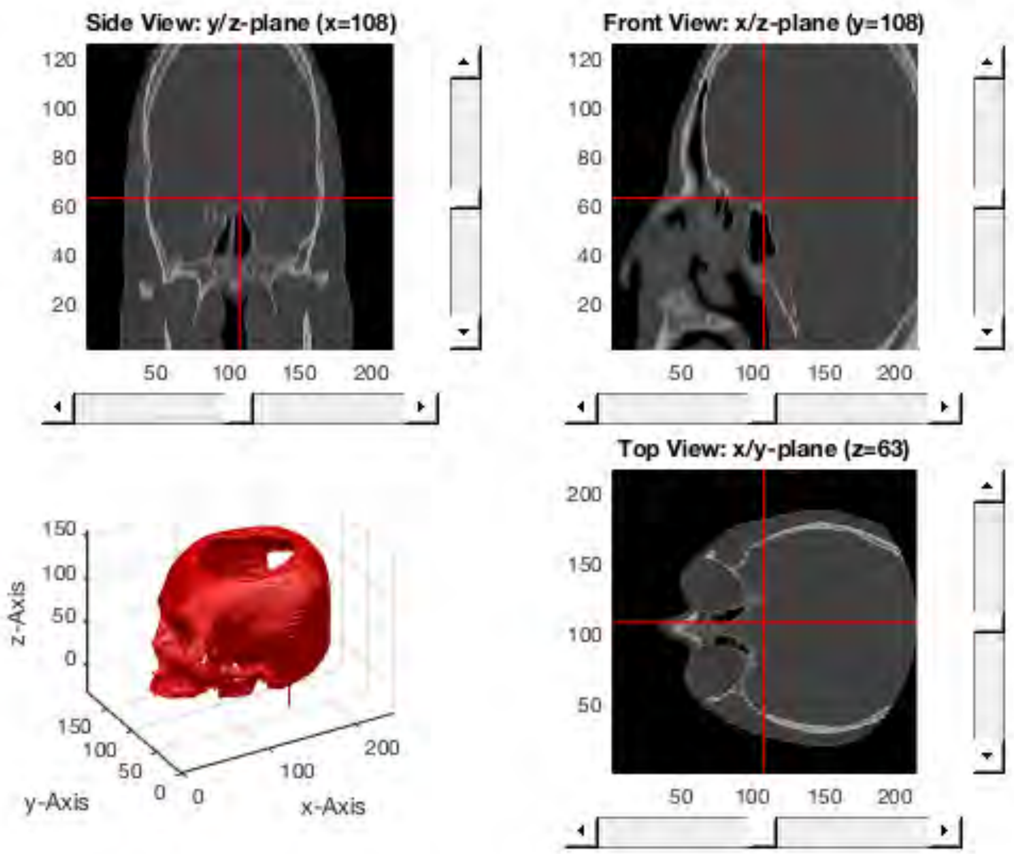



```
VLFLplotlight(0,1);
```



6 Show the Voxel model in quadrant 1-2-4 and surface model in quadrant 3

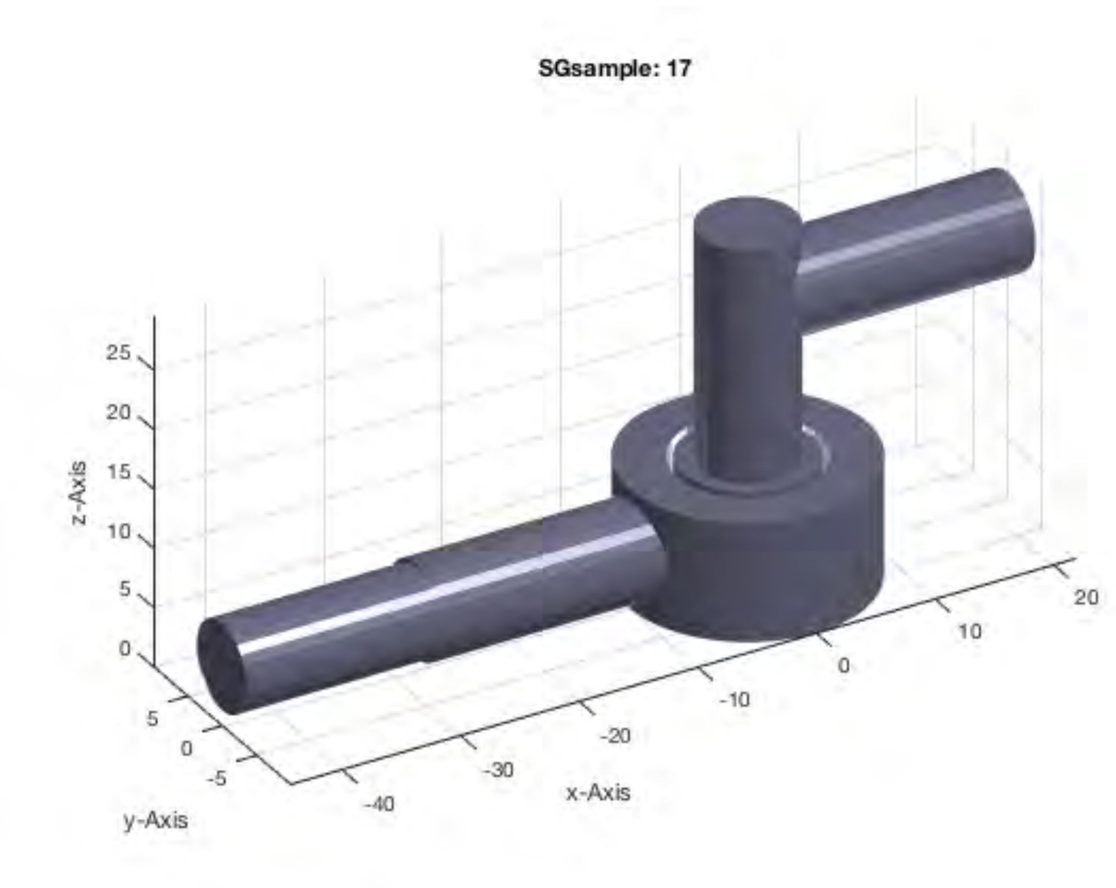
```
VMplot(a,'',SG4)
```



7. Create a surface model and convert it into a Voxel model

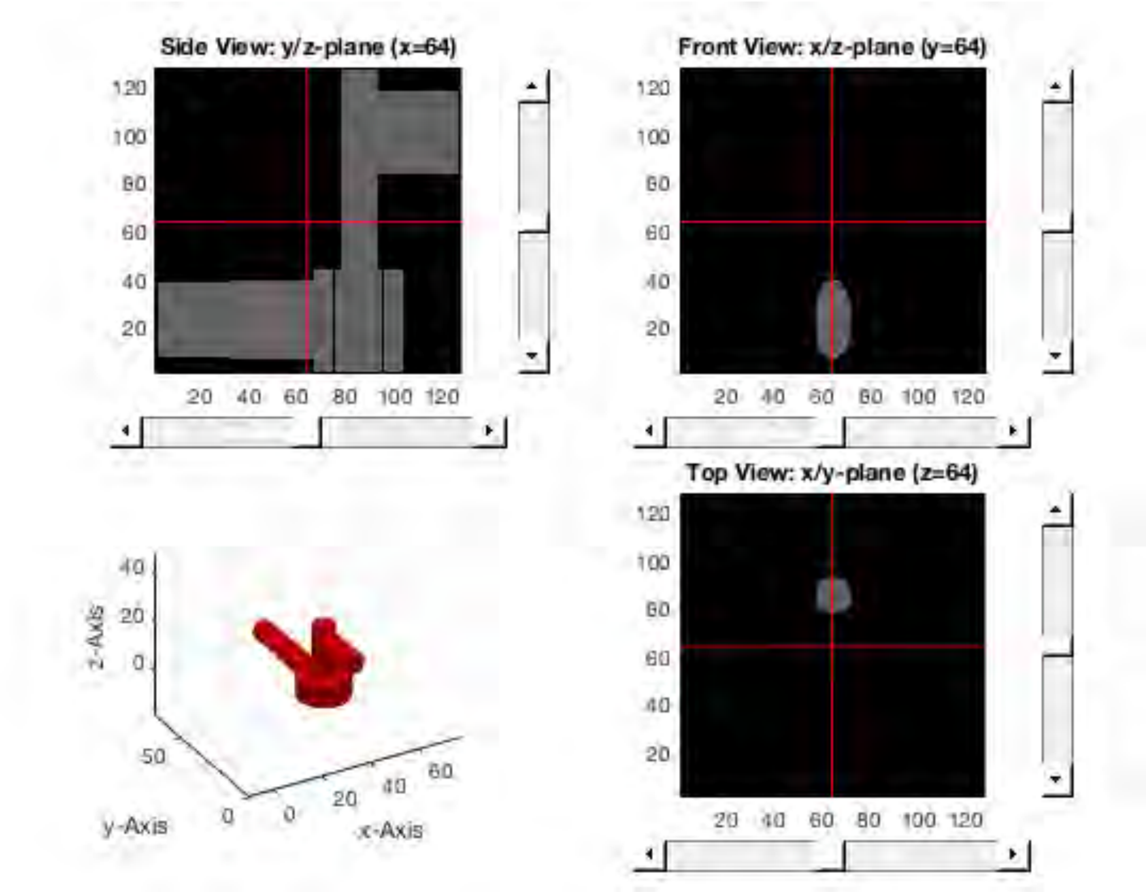
```
SGsample(17); VLFLplotlight(1,1);

[VM,ms,SG3]=VMofSG(SGsample(17),[128 128 128],true);
whos VM
ms
SG3
```

```
VMofSG: 5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100%
  Name         Size                        Bytes  Class     Attributes

  VM         128x128x128            16777216  double


ms =

    0.5309    0.5309    0.2358


SG3 =

  struct with fields:

    VL: [20614×3 double]
    FL: [60580×3 double]
```



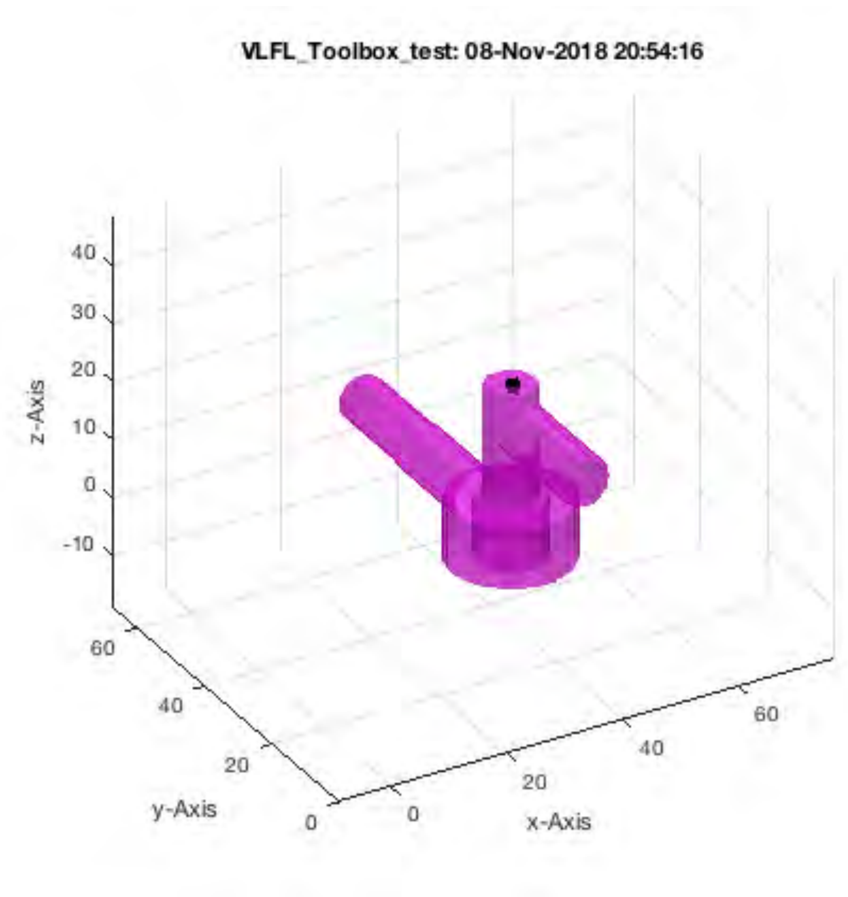
8. Plot the surface model in 4 quadrant plot

```
VMplot(VM,'',SG3)
```



9. Select Point in 3D

```
SGfigure(SG3); view(-30,30); VLFLplotlight (1,.5);
ginput(1); p=select3d; pplot(p,'k*',4); rotate3d on
```



Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:54:23!
Executed 08-Nov-2018 20:54:25 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
image_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
===========================================================================================
==========
```

Published with MATLAB® R2018a

Tutorial 32: Exchanging Data with a FileMaker Database

2017-03-01: 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)

Motivation for this tutorial: (Originally SolidGeometry 3.5 required)

List of Supported functions

- FMhelp - Returning some help text on the function set
- FMinitJDBC - called automatically by FMopen; expecting "fmjdbc.jar" in the search path
- FMopen - to open a database with user name and password
- FMgetFieldTabs - to get informations on the Database
- FMsqlQuery - to send requests or data to the Databse

1. Getting some help information on the interface

```
FMhelp
FMinitJDBC('fmjdbc.jar') % installs the Filemaker JDBC Driver
```

```
ans =

    '  FMhelp - returns a help text for the FileMaker-Matlab interface
        (by Tim Lueth, FileMaker, 2017-FEB-28)

        Tobias Lüddemann did start the connection of FileMaker and Matlab using
        2012b and FileMaker Pro 11. There is a document dated 2014-11-27 at TUM
        MIMED.
        Tim Lueth capsulated the JDBC FileMaker interface using Matlab 2016b
        and FileMaker 13 starting February 2017. The solution described here
        works with Filemaker 13 and later.

        The Matlab Database Toolbox is required. You need a license for that.
        The xDBC Drivers for Filemaker can be downloaded from the Filemaker
        WWW-Site for your Filemaker Version
        The JDBC Driver "fmjdbc.jar" is part of this package.
```

```
        This driver file has to be added to the javaclasspath (which is done by
        the fnctn FMinitJDBC)

        For connecting to the Filemaker App you have to:
        SWITCH ON FILESHARING for ALL Users (Filemaker & Database)
        SWITCH ON ODBC-JDBC-Sharing: for ALL Users (Filemaker & Database)

        Lueth's fnctns to support the connection to Filemaker are:
        FMhelp - This fnctn
        FMinitJDBC - Opens the Driver "fmjdbc.jar"
        FMopen - to open a database with user name and password
        FMgetFieldTabs - to get informations on the Database
        FMsqlQuery - to send requests or data to the Databse
        ...there are some fnctns all starting with capital letters "FM"

         (Status of: 2017-03-01)

        See also: FMhelp, FMinitJDBC, FMopen, FMgetFieldTabs, FMsqlQuery

        LITERATURE:
        Filemaker (2013): "SQL-Referenzhandbuch FM 13",
        https://fmhelp.filemaker.com/docs/13/de/fm13_sql_reference.pdf

        FMhelp

        EXAMPLE: How to use the library after copying "fmjdbc.jar" in a search
        path directory:
        FMinitJDBC('fmjdbc.jar')
        conn=FMopen('Basename.fmp12','user','passw')
        FMgetFieldTabs(conn)
        FMsqlQuery(conn,'SELECT * FROM FileMaker_Tables')

    '

Java class installed: '/Volumes/LUETH-WIN/WIN AIM Matlab Libraries/VLFL-Lib/fmjdbc.jar'
1. Make sure that Filemaker Application's Sharing is ON for ALL USER.
2. Make sure that Filemaker Database's Sharing is ON for ALL USER.
3. Make sure that Filemaker ODBC/JDBC's Sharing is ON for ALL USER.
```

2. Open a Database and establishes a connection

```
conn=FMopen('FileMakerTestBase.fmp12')
```

```
conn =

  connection with properties:

                DataSource: ''
                  UserName: ''
                    Driver: ''
                       URL: ''
                   Message: '[FileMaker][FileMaker JDB ...'
                      Type: 'JDBC Connection Object'
  Database Properties:
```

```
                  AutoCommit: ''
                    ReadOnly: ''
                LoginTimeout: 0
      MaxDatabaseConnections: -1

    Catalog and Schema Information:

              DefaultCatalog: ''
                    Catalogs: {}
                     Schemas: {}

    Database and Driver Information:

         DatabaseProductName: ''
      DatabaseProductVersion: ''
                  DriverName: ''
               DriverVersion: ''
```

3. Getting Informations on FileMaker Database Fields

```
FMgetFieldTabs(conn)
```

```
Undefined function 'fetch' for input arguments of type 'struct'.

Error in FMsqlQuery (line 24)
ans=fetch(curs); ans.Data;

Error in FMgetFieldTabs (line 21)
FMsqlQuery(conn,'SELECT TableName FROM FileMaker_Tables')

Error in VLFL_EXP32 (line 66)
FMgetFieldTabs(conn)
```

4. Retrieving Information

```
FMsqlQuery(conn ,'Select "Fragen" from "Prüfungsfragen"')
```

Closing the Database Connection

```
FMclose(conn);
```

Final Remarks

```
close all
VLFLlicense
```

Published with MATLAB® R2018a

Tutorial 33: Using a Round-Robin realtime multi-tasking system

2017-03-05: 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)

Motivation for this tutorial: (Originally SolidGeometry 3.5 required)

Matlab can be converted relatively simply to a realtime-multi-tasking environment according to the round-robin method. The following conditions must apply: A) There is a fixed time base for all tasks B) All tasks are called up one after the other in the fixed time clock

The environment presented in this tutorial has the following properties:

- A) Shell character = While the real-time environment is running, Matlab commands can still be typed as before.
- B) All variables of the real-time environment can be modified directly.

List of supported functions

The input-interpreter routine of the real-time environment has additional commands that superimpose comparable commands on the Matlab command line.

- ***HELP*** - shows a help text
- ***EXIT*** or ***QUIT*** - ends the environment
- ***SHOWLIST*** or ***TASKS*** - shows all tasks
- ***KILLTASKS*** or ***KILLALL*** - removes all tasks
- ***ADD*** - appends a task at the task list
- ***KILLLAST*** - removes the last task
- ***BREAK*** or ***STOP*** - stops the realtime execution
- ***STEP*** - runs the realtime loop only ones
- ***GO*** or ***START*** or ***CONT*** - starts the realtime loop
- ***KILL*** Tasknumber - removes a task with that number

- ***SAVE*** - saves the current task list on disk
- ***LOAD*** - loads the current task list on disk
- ***EXE*** or ***EXECUTE*** filename - executes a command line file
- ***edit*** filename edits a command line file
- ***whos*** shows the variables

Intended use of RRrun or RRshell (both are the same)

- ***RRrun*** - starts the Shell
- ***RRrun*** commandstring (lines are spearated by \r)

1. Starting the shell

The following commands could be typed in absolute in the same way as part of the command string. So please try to type them also manually instead of using them als input parameter of RRrun There is no other chance to create a publishable document as to describe them as input parameter.

```
RRrun 'quit'    % Quit immediately
```

```
RRkeyboardLine =

    'quit'

====LOOP STARTS 08-Nov-2018 20:54:33 for 600 SECONDS with CYCLETIME 0.100 SECONDS==========>
> END =======
RRrun>>
TERMINATED by User
================LOOP ENDS 08-Nov-2018 20:54:34 USING 0.11 SECONDS =========================
```

```
RRrun 'RRstop=RRcputime+1;'  % Quit after 1 second
```

```
RRkeyboardLine =

    'RRstop=RRcputime+1;'

====LOOP STARTS 08-Nov-2018 20:54:34 for 600 SECONDS with CYCLETIME 0.100 SECONDS==========>
> END =======
RRrun>>
================LOOP ENDS 08-Nov-2018 20:54:35 USING 1.20 SECONDS =========================
```

```
RRrun 'LIST \r QUIT' % show the tasks and quit
```

```
RRkeyboardLine =

    'LIST        QUIT'

====LOOP STARTS 08-Nov-2018 20:54:35 for 600 SECONDS with CYCLETIME 0.100 SECONDS==========>
```

```
> END =======
RRrun>>
RRtasklist =

  struct with fields:

       t0: 0.1000
    twarn: 0.1000
    tstop: 1
      cnt: 0
    tlist: []


TERMINATED by User
================LOOP ENDS 08-Nov-2018 20:54:35 USING 0.20 SECONDS =========================
```

```
RRrun 'whos \r QUIT' % show the variables and quit
```

```
RRkeyboardLine =

    'whos       QUIT'

====LOOP STARTS 08-Nov-2018 20:54:35 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>  Name                 Size             Bytes  Class              Attributes

  RRbreak             1x1                  1  logical
  RRcurs              0x0                  0  double             persistent
  RRdelay             1x1                  8  double             global
  RRkeyboardLine      1x6                 12  char               global
  RRlastcommand       1x4                  8  char               global
  RRlastexectime      1x1                  8  double
  RRlasttime          1x1                  8  double             global
  RRmaxtime           1x1                  8  double             global
  RRpause             1x1                  8  double
  RRprompt            1x5                 10  char               global
  RRstart             1x1                  8  double             global
  RRstop              1x1                  8  double             global
  RRtasklist          1x1                912  struct             global
  RRwindow            1x1                  8  matlab.ui.Figure   global
  RRwindowNr          1x1                  8  double             global
  cmd                 1x4                  8  char
  remain              0x0                  0  char
  token               1x4                  8  char
  varargin            1x1                136  cell



TERMINATED by User
================LOOP ENDS 08-Nov-2018 20:54:36 USING 0.20 SECONDS =========================
```

```
RRrun ('fprintf(''%.2f\n'',RRcputime) \rQUIT') % show the cputime and quit
```

```
RRkeyboardLine =

    'fprintf('%.2f\n',RRcputime)      QUIT'

====LOOP STARTS 08-Nov-2018 20:54:36 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>2.51


TERMINATED by User
================LOOP ENDS 08-Nov-2018 20:54:36 USING 0.20 SECONDS =========================
```

2. Adding realtime tasks and define the stop time

```
RRrun 'ADD fprintf(''%.2f\n'',RRcputime) \r RRstop=RRcputime+1;' % show the cputime every cyc
le and quit after 1 second
```

```
RRkeyboardLine =

    'ADD fprintf('%.2f\n',RRcputime)       RRstop=RRcputime+1;'

====LOOP STARTS 08-Nov-2018 20:54:36 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>2.94

3.03
3.13
3.23
3.34
3.43
3.53
3.63
3.73
3.83
3.93
4.03
================LOOP ENDS 08-Nov-2018 20:54:37 USING 1.30 SECONDS =========================
```

3. Adding realtime tasks and change the cycle time

```
RRrun 'ADD fprintf(''%.2f\n'',RRcputime) \r RRtasklist.t0=0.05 \r RRstop=RRcputime+1;'
```

```
RRkeyboardLine =

    'ADD fprintf('%.2f\n',RRcputime)       RRtasklist.t0=0.05       RRstop=RRcputime+1;'

====LOOP STARTS 08-Nov-2018 20:54:38 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>4.36

RRtasklist =
```

```
  struct with fields:

        t0: 0.0500
     twarn: 0.1000
     tstop: 1
       cnt: 2
     tlist: {' fprintf('%.2f\n',RRcputime)'  [Inf]  [1]}


4.41

4.46
4.51
4.56
4.61
4.66
4.71
4.76
4.81
4.86
4.91
4.96
5.01
5.06
5.11
5.16
5.21
5.26
5.31
5.36
5.41
5.46
================LOOP ENDS 08-Nov-2018 20:54:39 USING 1.30 SECONDS =========================
```

4. Adding realtime tasks and change the cycle time and kill the task

```
RRrun 'ADD fprintf(''%.2f\n'',RRcputime) \r RRtasklist.t0=0.05 \r KILLLAST \r RRstop=RRcputim
e+1;'
```

```
RRkeyboardLine =

    'ADD fprintf('%.2f\n',RRcputime)       RRtasklist.t0=0.05       KILLLAST       RRstop=RRc
putime+1;'

====LOOP STARTS 08-Nov-2018 20:54:39 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>5.80

RRtasklist =

  struct with fields:

        t0: 0.0500
     twarn: 0.1000
```

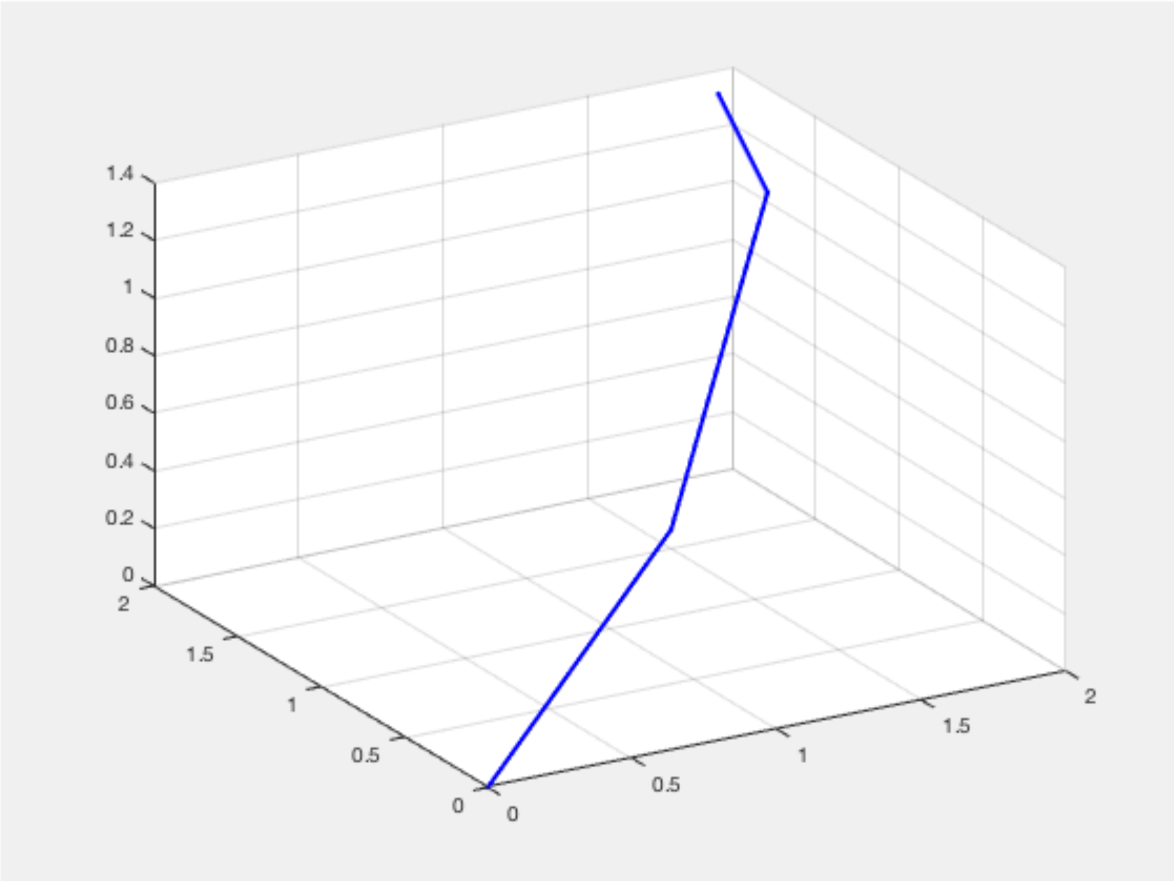
```
    tstop: 1
      cnt: 2
    tlist: {' fprintf('%.2f\n',RRcputime)'  [Inf]  [1]}


5.85

RRtasklist =

  struct with fields:

       t0: 0.0500
    twarn: 0.1000
    tstop: 1
      cnt: 2
    tlist: {0×3 cell}


================LOOP ENDS 08-Nov-2018 20:54:40 USING 1.35 SECONDS =========================
```

5. Run a plotting task and save the task list by using "save"

```
RRrun 'KILLALL \r global PL; PL=[0 0 0]; \r  ADD global PL; PL=[PL; PL(end,:)+rand(1,3)]; \r
ADD global PL; delete(gca); VLplot(PL,''b.-'',2); view(-30,30); grid on; \r save \r copyplot
\r RRstop=RRcputime+3;'
```

```
RRkeyboardLine =

    'KILLALL       global PL; PL=[0 0 0];        ADD global PL; PL=[PL; PL(end,:)+rand(1,3)];
       ADD global PL; delete(gca); VLplot(PL,'b.-',2); view(-30,30); grid on;       save
  copyplot       RRstop=RRcputime+3;'

====LOOP STARTS 08-Nov-2018 20:54:40 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>
RRtasklist =

  struct with fields:

       t0: 0.1000
    twarn: 0.1000
    tstop: 1
      cnt: 0
    tlist: []



RRtasklist saved in RRtasklist.mat
  Name              Size              Bytes  Class     Attributes

  RRtasklist        1x1                1832  struct    global


RRrun: realtime condition broken by 133 milliseconds
```

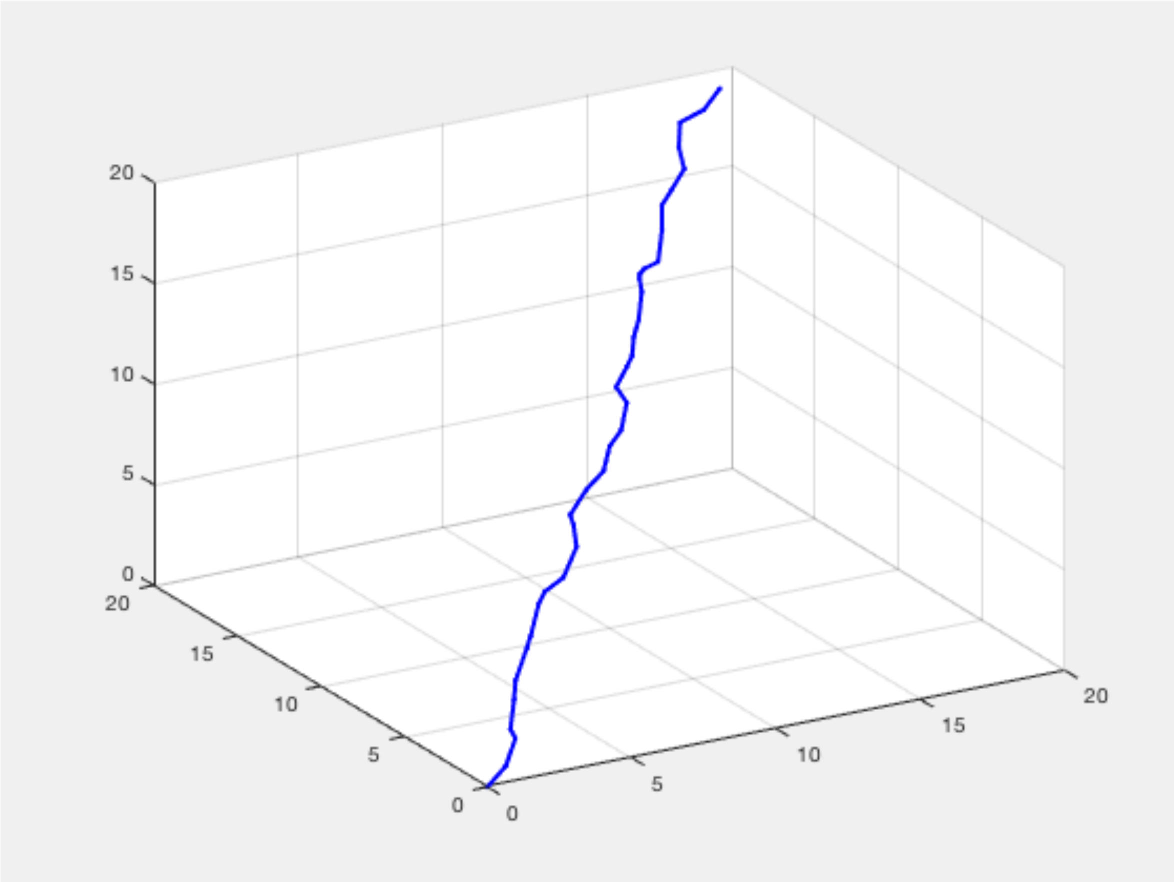
```
RRrun: realtime condition broken by 243 milliseconds

================LOOP ENDS 08-Nov-2018 20:54:45 USING 4.32 SECONDS ========================
```



6. Run the saved task a second time by using "load"

```
RRrun 'load \r start \r copyplot \r RRstop=RRcputime+3;'
```

```
RRkeyboardLine =

    'load       start       copyplot       RRstop=RRcputime+3;'

====LOOP STARTS 08-Nov-2018 20:54:45 for 600 SECONDS with CYCLETIME 0.100 SECONDS==========>
> END =======
RRrun>>
New RRtasklist loaded from RRtasklist.mat


RRrun: realtime condition broken by 132 milliseconds

================LOOP ENDS 08-Nov-2018 20:54:49 USING 3.67 SECONDS ========================
```

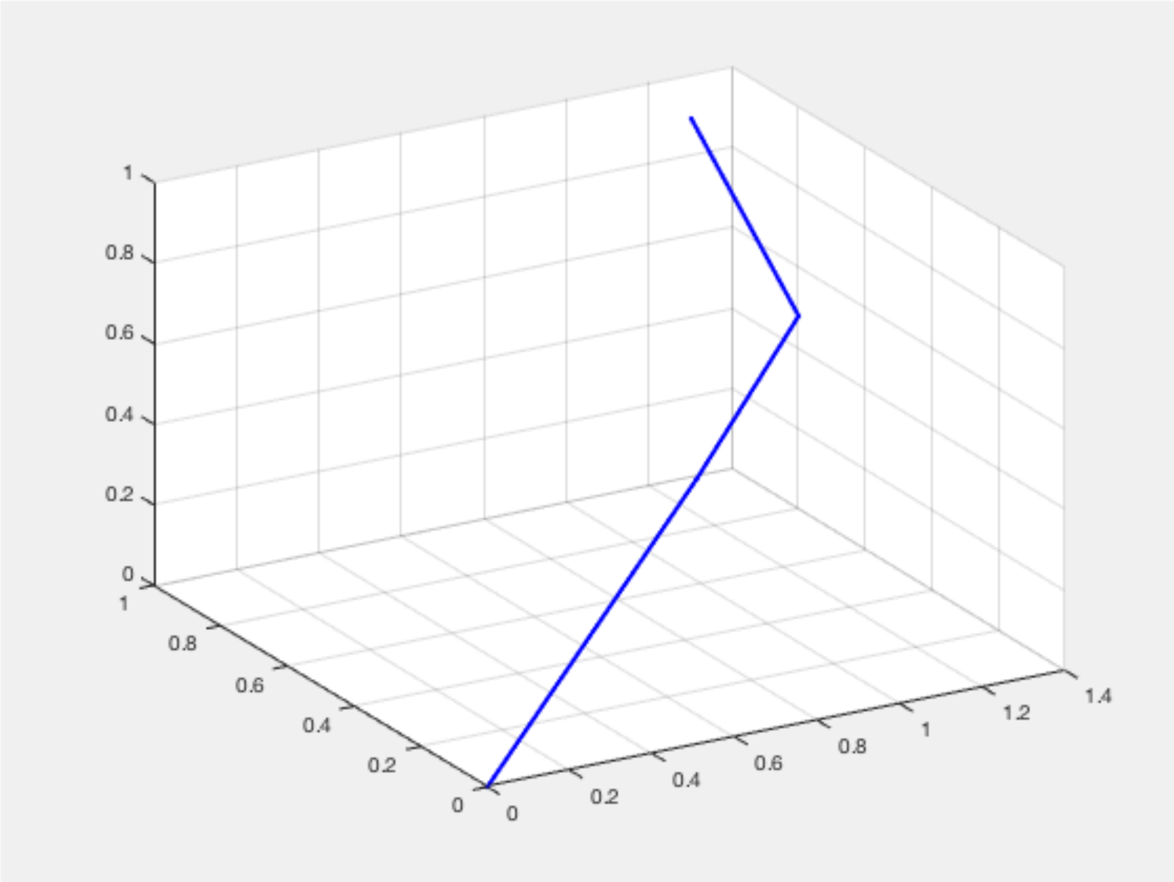


7. Execute a command file

```
RRrun 'execute RRrun_testcommands.txt'
```

```
RRkeyboardLine =

    'execute RRrun_testcommands.txt'

====LOOP STARTS 08-Nov-2018 20:54:49 for 600 SECONDS with CYCLETIME 0.100 SECONDS===========>
> END =======
RRrun>>
fname =

    'RRrun_testcommands.txt'

Warning: Inputs must be character vectors, cell arrays of character vectors, or
string arrays.

RRtasklist =

  struct with fields:

       t0: 0.1000
    twarn: 0.1000
```

```
    tstop: 1
      cnt: 0
    tlist: []



RRtasklist =

  struct with fields:

       t0: 0.1000
    twarn: 0.1000
    tstop: 1
      cnt: 4
    tlist: {2×3 cell}


ans =

  2×3 cell array

    {' global PL; PL=[…'}    {[Inf]}    {[1]}
    {' global PL; dele…'}    {[Inf]}    {[3]}

RRtasklist currently stopped. Use "START" or "STEP" to start task execution.

Elapsed time is 0.011606 seconds.



RRrun: realtime condition broken by 129 milliseconds

RRrun>>================LOOP ENDS 08-Nov-2018 20:54:53 USING 3.97 SECONDS ====================
=====
```



8. Edit a command file

```
edit 'RRrun_testcommands.txt'
```

Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:54:54!
Executed 08-Nov-2018 20:54:57 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a MAC
I64
======================================== Used Matlab products: ==============================
========
antenna_toolbox
database_toolbox
image_toolbox
map_toolbox
matlab
```

```
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
==============================================================================================
========
```

Published with MATLAB® R2018a

Tutorial 34: 2D Projection Images and Camera Coordinate System Reconstruction

2017-05-15: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de) - Last Change: 2017-07-25

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

Motivation for this tutorial: (Originally SolidGeometry 3.8 required)

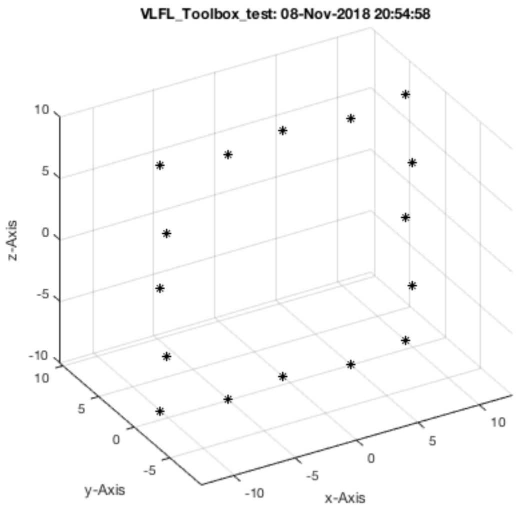
Many surgical procedures in orthopedics are not based on three-dimensional CT or MRI image data but on C-arm images. These C-arm images are 2D projection images of a spatial region of the patient. In this, the most important strategies for the conversion of volume images to projection images are presented. It is also explained how the position of the X-ray camera can be calculated from projection images, if one knows the exact location of objects in space and the 2D image. The research area is also called Camera Calibration.

ATTENTION >>> The Publishermode changes the aspect ratio of figures, therefor it is strongly recommended to copy lines from this tutorial instead of just executing the publishabe example

1. Create a number of random points around the center

The following commands could be typed in absolute in the same way as part

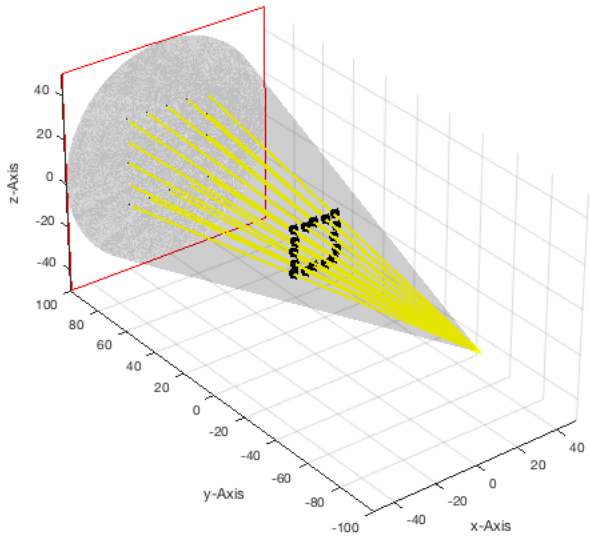
```
SGfigure; view(-30,30); xlabel 'x-Axis', ylabel 'y-Axis', zlabel 'z-Axis';
VL=50*rand(10,3)-25; VL(:,2)=1*rand(10,1)';
% VL=VLsample(9)
VL=VLsample(11); VL=VLtransT(VL,TofR(rot(-pi/20,0,pi/20)));
VL=VLsample(12);
VLplot(VL,'k*');
```



2. Create an X-ray image by using the camera parameter of Matlab

The x-ray source is at position [0 100 0]; The target is at [0 +100 0] The screen has a size of 100x100 The scaling factor is 4, i.e. the pixel size if 025 x 0.25 mm

```
imageofVLprojection(VL,[100 100],[0 -100 0],[0 100 0],4);
set(gca,'Projection','perspective');    % this line is only required because of publishing function
```

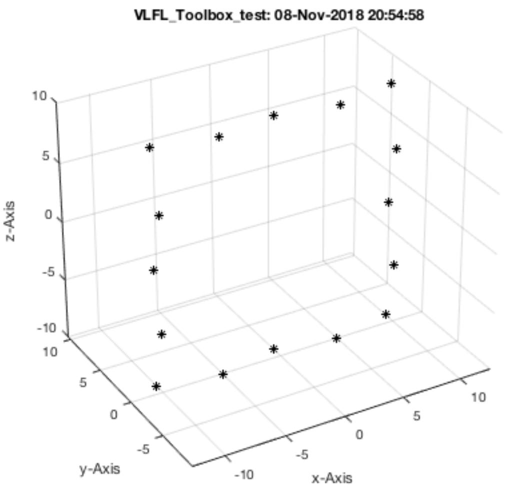


show the image

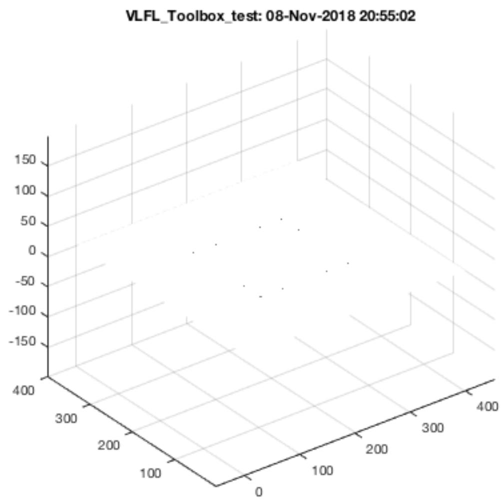
```
I=imageofVLprojection(VL,[100 100],[0 -100 0],[0 100 0],4);
set(gca,'Projection','perspective');    % this line is only required because of publishing function
whos I                                  % this line is only required because of publishing function
```

```
  Name        Size               Bytes  Class     Attributes
```

```
  I          400x400            1280000  double
```




```
SGfigure
imwarpT(I);
```

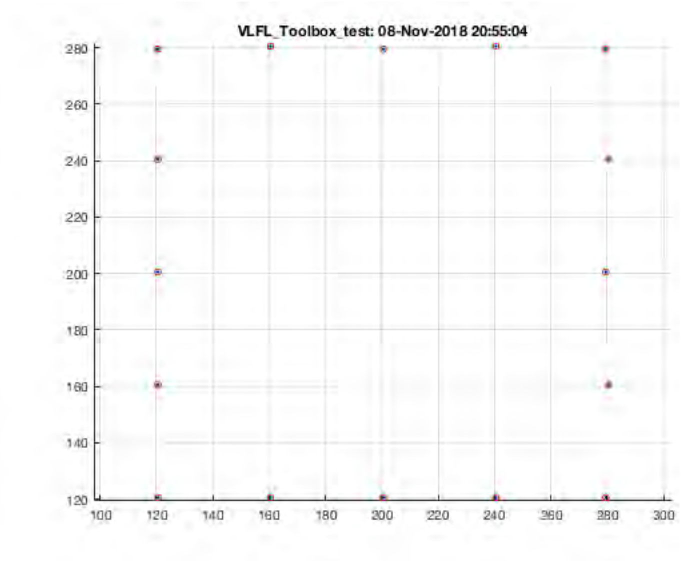


3. Find the marker points in the image

```
SGfigure;
CPL=CPLcontourc(I,1); % Contour segmentation on image base
CPLplot(CPL,'r-');
PL=centerCPL(CPL)
PLplot(PL,'b.',4);
size(I,2)                          % this line is only required because of publishing function
```

```
PL =

  120.5000  279.5000
  120.5000  240.5000
  120.5000  200.5000
  120.5000  160.5000
  120.5000  120.5000
  280.5000  240.5000
  280.5000  160.5000
  279.5000  279.5000
  279.5000  200.5000
  279.5000  120.5000
  160.5000  280.5000
  160.5000  120.5000
  240.5000  280.5000
  240.5000  120.5000
  200.5000  279.5000
```

```
  200.5000  120.5000


ans =

   400
```

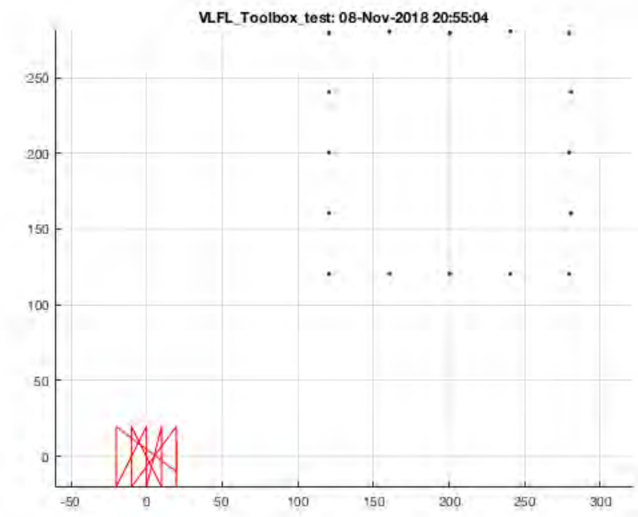


turn the coordinate

```
PL(:,2)=-PL(:,2)+size(I,2)   % flip up and down (y-axis)
PL=(PL-1)-size(I)/2          % Move coordinate into center
PL=PL/4                      % Scale using pixle size
CPLplot(PL,'r-');
```

```
PL =

  120.5000  120.5000
  120.5000  159.5000
  120.5000  199.5000
  120.5000  239.5000
  120.5000  279.5000
  280.5000  159.5000
  280.5000  239.5000
  279.5000  120.5000
  279.5000  199.5000
  279.5000  279.5000
  160.5000  119.5000
  160.5000  279.5000
  240.5000  119.5000
  240.5000  279.5000
  200.5000  120.5000
  200.5000  279.5000


PL =

  -80.5000  -80.5000
  -80.5000  -41.5000
  -80.5000   -1.5000
  -80.5000   38.5000
  -80.5000   78.5000
   79.5000  -41.5000
   79.5000   38.5000
   78.5000  -80.5000
   78.5000   -1.5000
   78.5000   78.5000
  -40.5000  -81.5000
  -40.5000   78.5000
   39.5000  -81.5000
   39.5000   78.5000
   -0.5000  -80.5000
   -0.5000   78.5000


PL =

  -20.1250  -20.1250
  -20.1250  -10.3750
  -20.1250   -0.3750
  -20.1250    9.6250
```

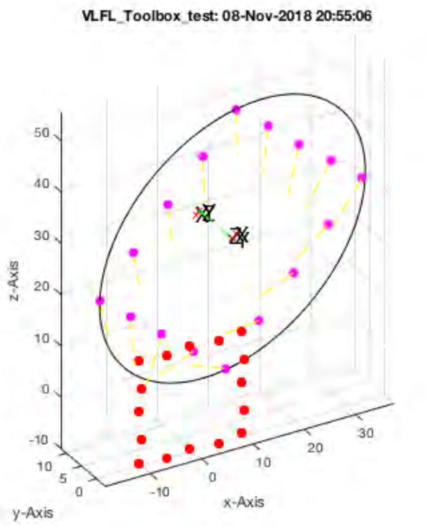
```
  -20.1250   19.6250
   19.8750  -10.3750
   19.8750    9.6250
   19.6250  -20.1250
   19.6250   -0.3750
   19.6250   19.6250
  -10.1250  -20.3750
  -10.1250   19.6250
    9.8750  -20.3750
    9.8750   19.6250
   -0.1250  -20.1250
   -0.1250   19.6250
```



Some knowledge on corresponding axis

```
TofcamVLPL(sortrows(VL,[1 3]),sortrows(PL,[1 2]))
```

```
K =

    0.9305   -1.6421   11.9064
   -0.0000   10.5902  -53.3133
    0.0000    0.0000    0.1091


s =

    9.1668


ans =

   -0.6635    0.7339    0.1454    7.9960
    0.0140    0.2065   -0.9784   11.6671
   -0.7481   -0.6471   -0.1473   28.6879
         0         0         0    1.0000
```

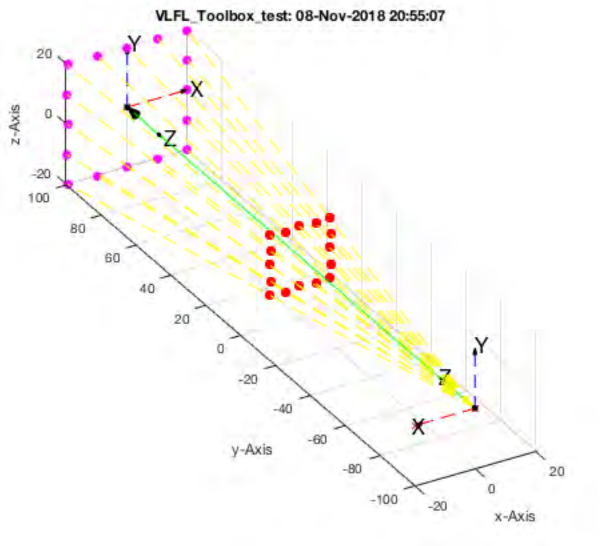


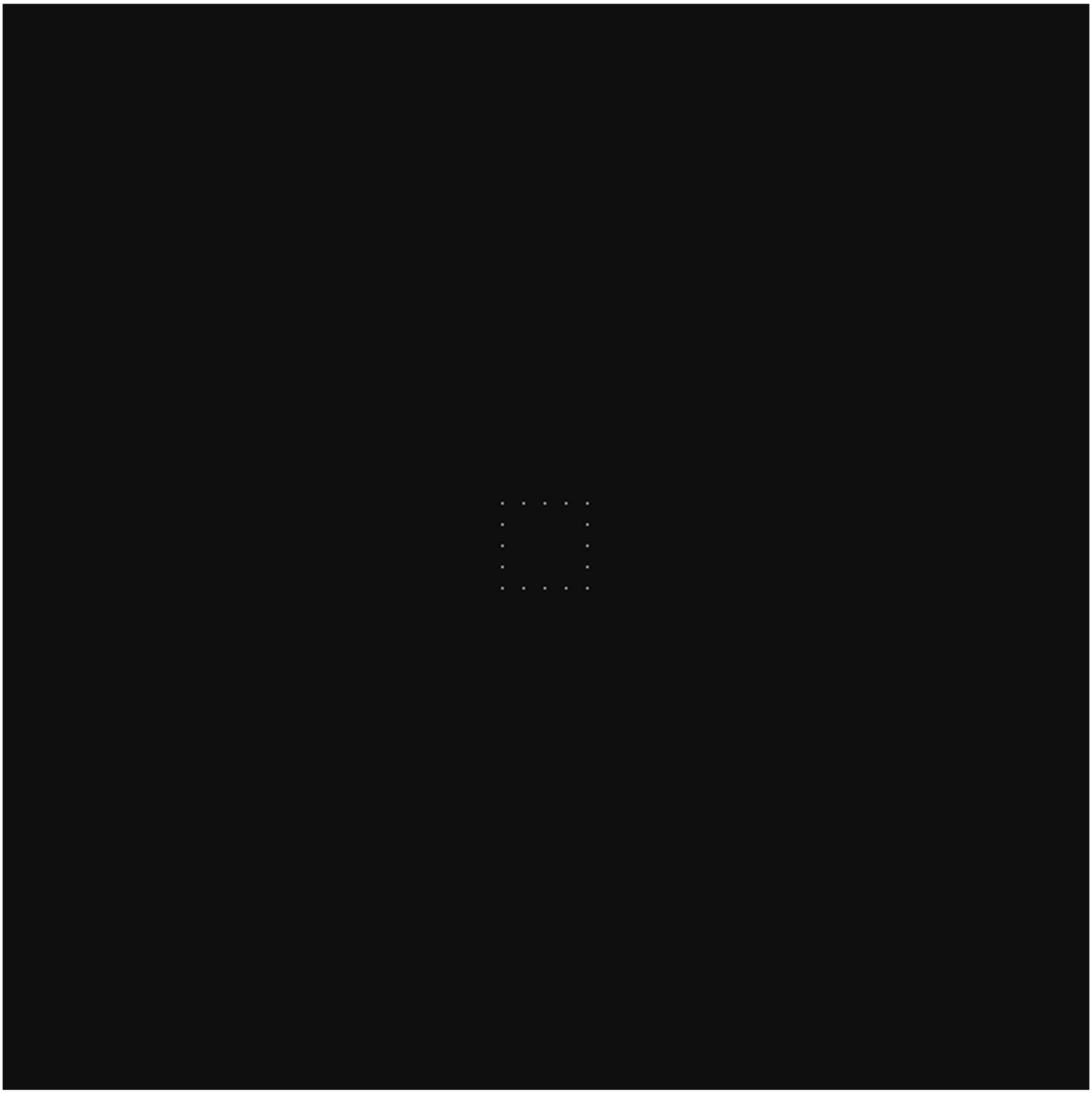
3. Calculate the Point Position of a X-Ray Camera

The x-ray source is at position [0 100 0]; The target is at [0 +100 0]

```
PLofVLprojection(VL,[0 -100 0],[0 100 0]);
PL=PLofVLprojection(VL,[0 -100 0],[0 100 0])
```

```
  Name         Size                 Bytes  Class     Attributes

  I          512x512              2097152  double              

  Name         Size                 Bytes  Class     Attributes

  I          512x512              2097152  double              


PL =

  -19.8020  -19.8020
  -10.0000  -20.0000
         0  -19.8020
   10.0000  -20.0000
   19.8020  -19.8020
   20.0000  -10.0000
   19.8020         0
   20.0000   10.0000
   19.8020   19.8020
   10.0000   20.0000
         0   19.8020
  -10.0000   20.0000
  -19.8020   19.8020
  -20.0000   10.0000
  -19.8020         0
  -20.0000  -10.0000
```


VLFL_Toolbox_test: 08-Nov-2018 20:55:07
z-Axis
y-Axis
x-Axis
X
Y
Z
20
0
-20
100
80
60
40
-40
-60
-80
-100

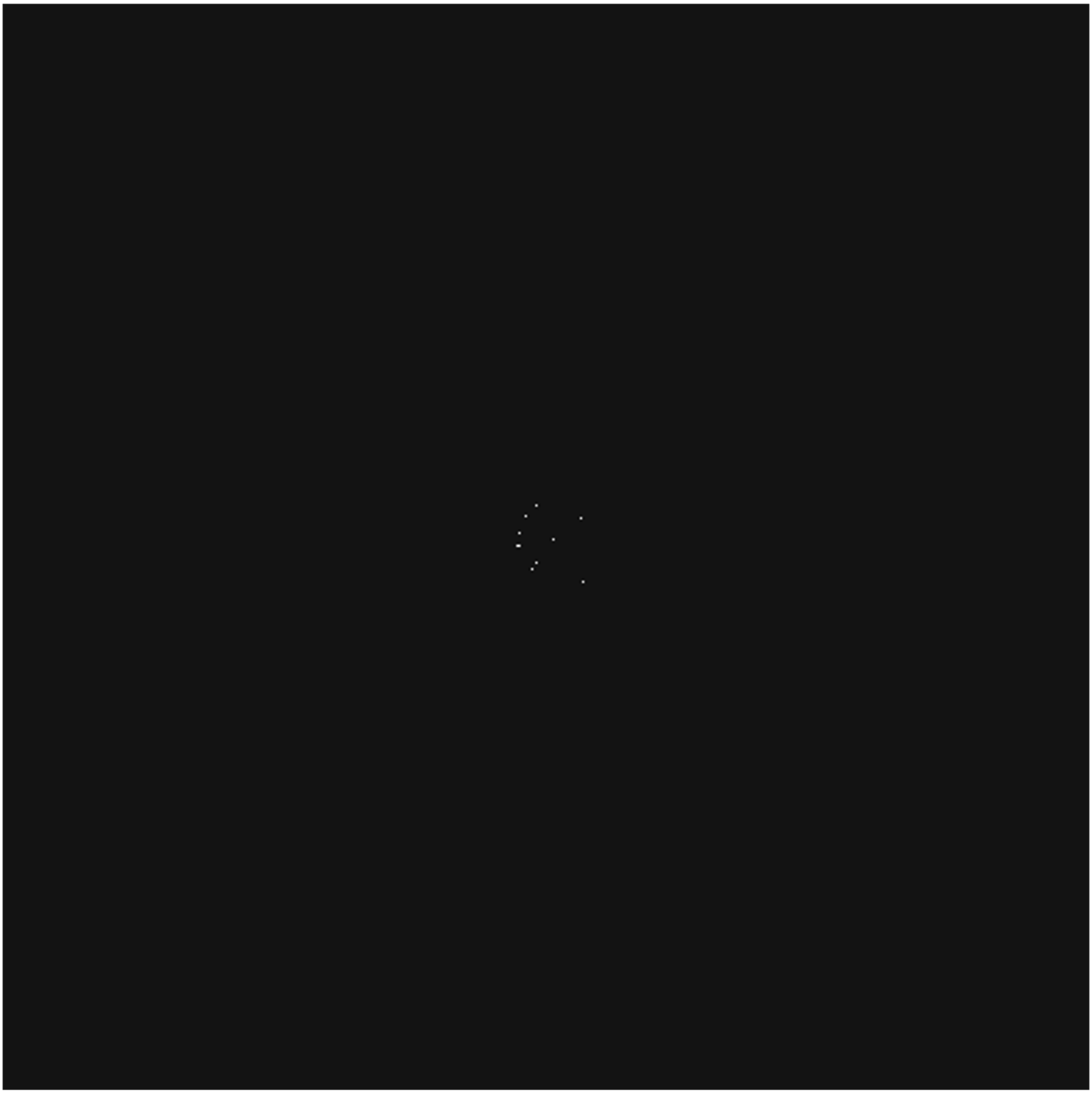


5. Comparision of point lists created by numerical projection or projection image reconstruction

```
VL=20*rand(10,3)-10; VL(:,2)=5*rand(10,1)';
I=imageofVLprojection(VL,[100 100],[0 -100 0],[0 100 0],4);
[sortrows(PLofimcontourc(I,true,1/4)) sortrows(PLofVLprojection(VL,[0 -100 0],[0 100 0]))]
```

```
  Name        Size              Bytes  Class     Attributes

  I         512x512            2097152  double


ans =

  -13.6250    0.1250  -13.3491    0.2301
  -12.1250   -0.1250  -11.9947   -0.1417
  -12.1250    5.6250  -11.9091    5.5214
   -9.1250   13.6250   -8.9570   13.6803
   -6.3750  -10.8750   -6.1497  -10.8245
   -4.6250   18.8750   -4.3536   18.7583
   -4.1250   -8.3750   -3.9363   -8.4596
    3.3750    2.6250    3.6623    2.6745
```

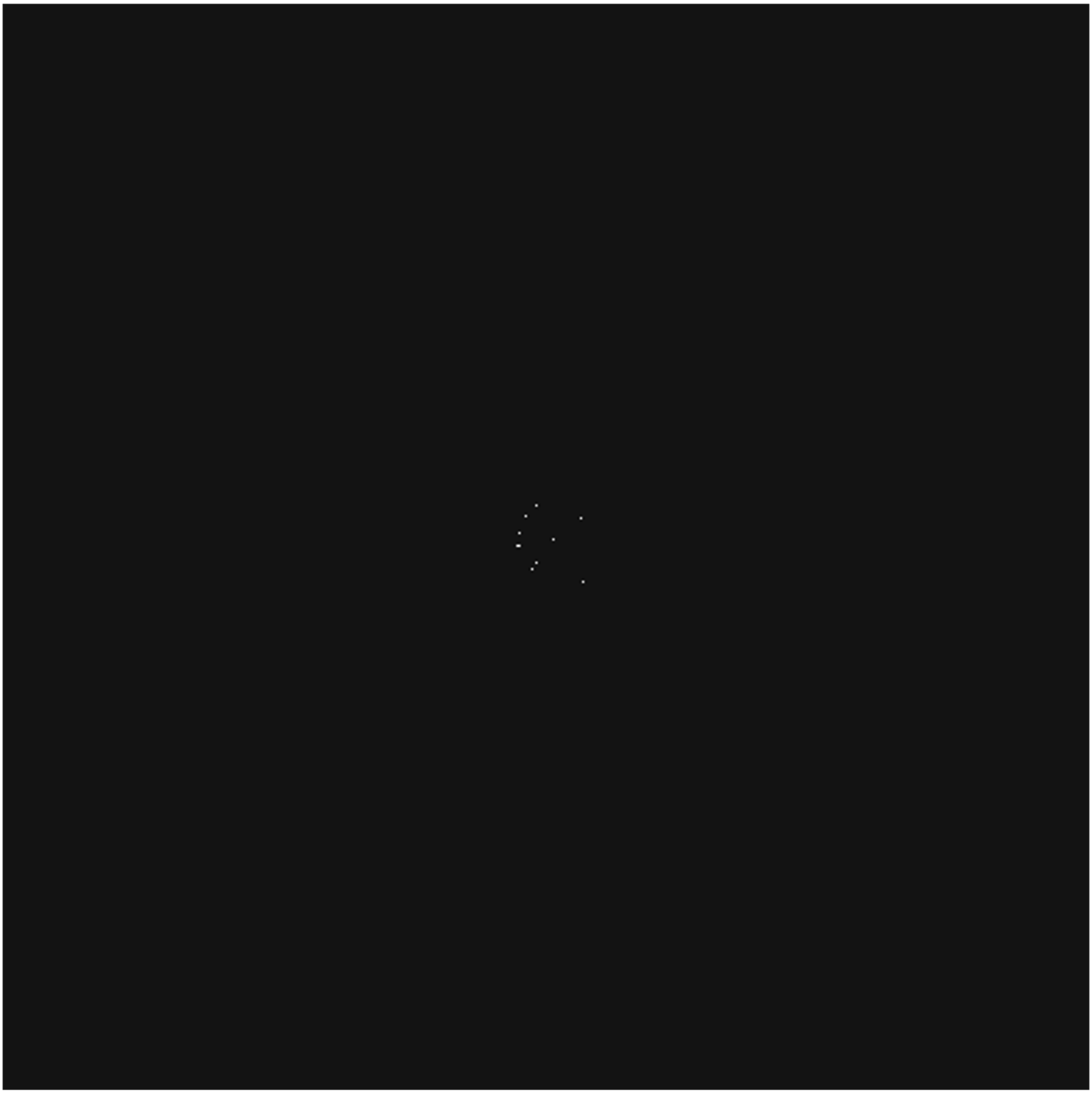
```
   16.8750   13.3750   17.1678   13.3669
   18.1250  -17.3750   18.2700  -17.3385
```

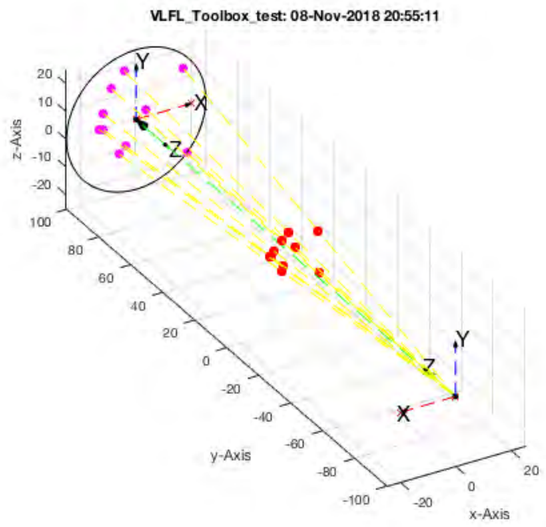


```
TofcamVLPL(sortrows(VL,[1 3]),sortrows(PLofVLprojection(VL,[0 -100 0],[0 100 0]),[1 2]))
```

```
  Name        Size              Bytes  Class     Attributes

  I         512x512            2097152  double


K =

    1.0000    0.0000   -0.0000
         0    1.0000    0.0000
    0.0000   -0.0000    0.0050


s =
```

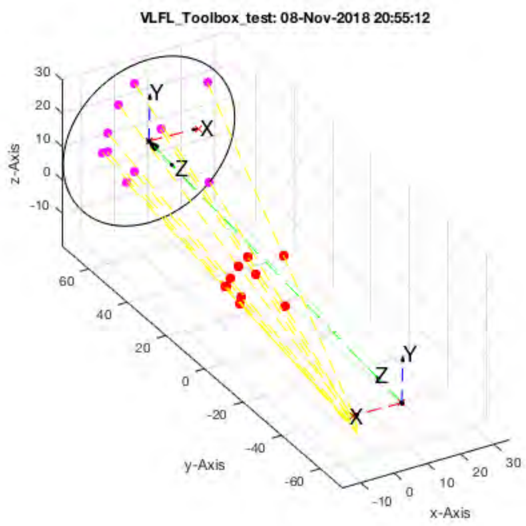
```
   200


ans =

  -1.0000         0         0   -0.0000
        0         0    1.0000 -100.0000
        0    1.0000         0    0.0000
        0         0         0    1.0000
```






```
TofcamVLPL(sortrows(VL,[1 3]),sortrows(PLofimcontourc(I,true,1/4),[1 2]))
```

```
K =

   -0.9971   -0.0011    0.0365
   -0.0000    0.9992    0.0390
   -0.0000   -0.0000    0.0070


s =

  143.3620


ans =

   -0.9989    0.0002    0.0467    2.6871
    0.0467   -0.0358    0.9983  -71.0152
    0.0019    0.9994    0.0357    0.1063
         0         0         0    1.0000
```



Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:55:13!
```

```
Executed 08-Nov-2018 20:55:15 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a MACI64
======================================== Used Matlab products: ======================================
antenna_toolbox
database_toolbox
image_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
=====================================================================================================
```

Published with MATLAB® R2018a

Tutorial 35: Creation of Kinematic Chains and Robot Structures

2017-07-04: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de) - Last Change: 2017-07-25

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

Motivation for this tutorial: (Originally SolidGeometry 4.0 required)

Already in the tutorials 11 and 12 kinematic chains were presented. This tutorial is about creating tree-like structures for robotic systems. The example uses the structures of the robot JACO. function VLFL_EXP35

1. Loading STL Files or Surface Data

The Elements of the JACO were prepared by reading STL data in and save the variables using the save command. Now the surface data is available but also those surfaces have already defined frames "B" for base and "F" for follower. clear all

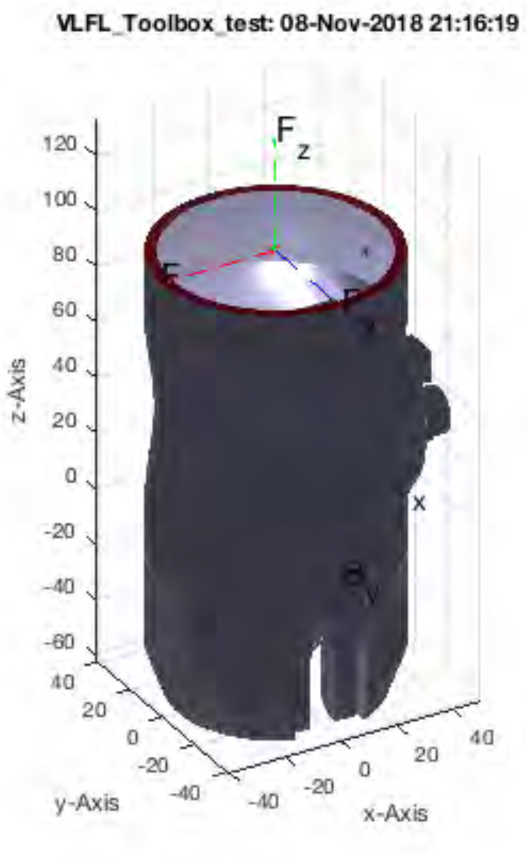
```
loadweb JACO_robot.mat
whos
```

```
Downloading "http://www.mimed.mw.tum.de/fileadmin/w00bhh/www/Matlab_Toolboxes/JACO_robot.ma
t" into: /Users/lueth/Desktop/Toolbox_test/2018-11-08_TL_PCODE!
  Name       Size                Bytes  Class     Attributes

  JACO       1x8               6371408  cell
  JC0        1x1               1100646  struct
  JC00       1x1               1465958  struct
  JC01       1x1                369662  struct
  JC1        1x1                843878  struct
  JC2        1x1                757118  struct
  JC3        1x1                695158  struct
  JC4        1x1                477846  struct
  JC5        1x1                477846  struct
  JC6        1x1               3731758  struct
```

```
JC61        1x1              1431998  struct
JCF         1x1               220710  struct
```

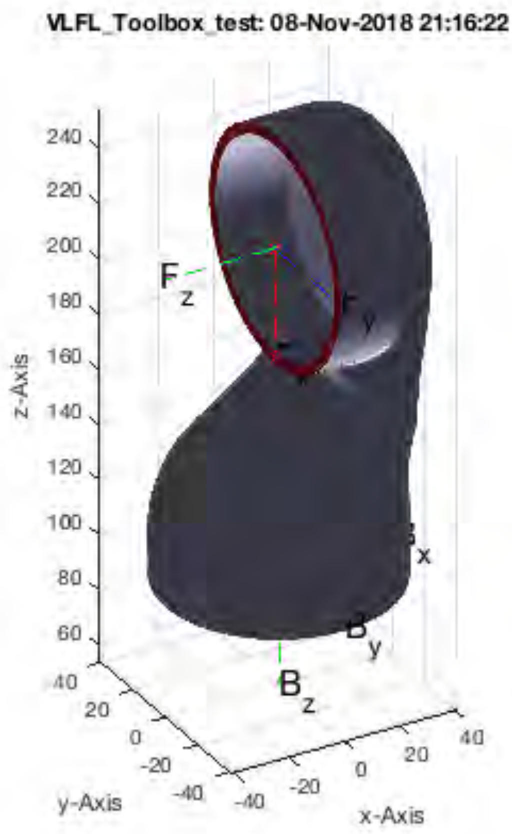
Plot the controller module/base of the Jaco robot

```
SGfigure; view(-30,30); SGTplot(JC0);
```



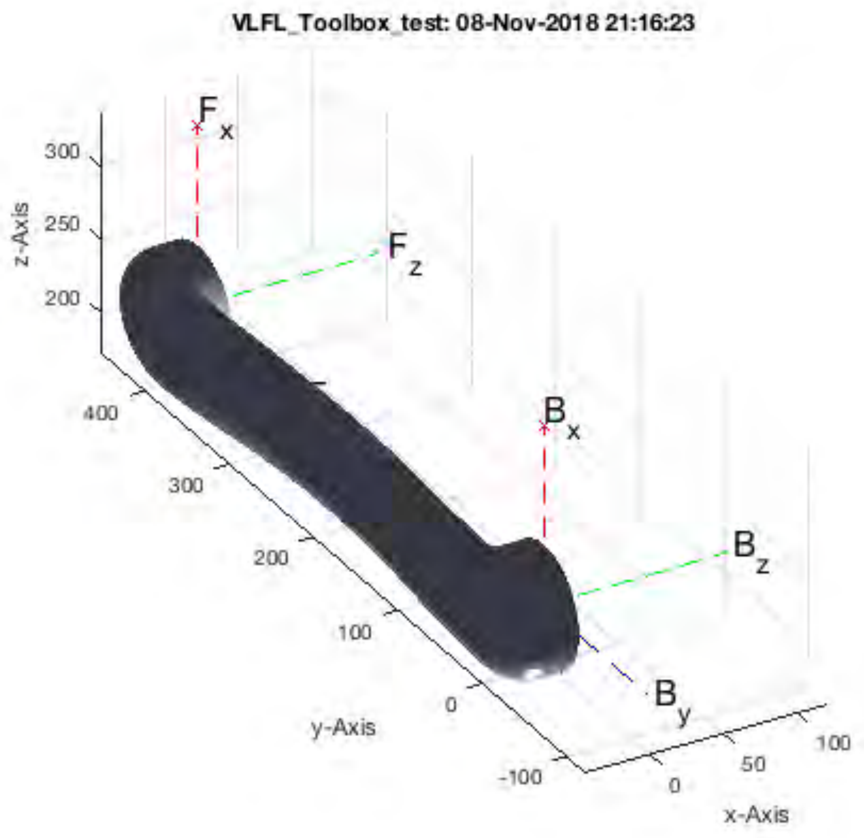
Plot the arm segment 1 of the Jaco robot

```
SGfigure; view(-30,30); SGTplot(JC1);
```



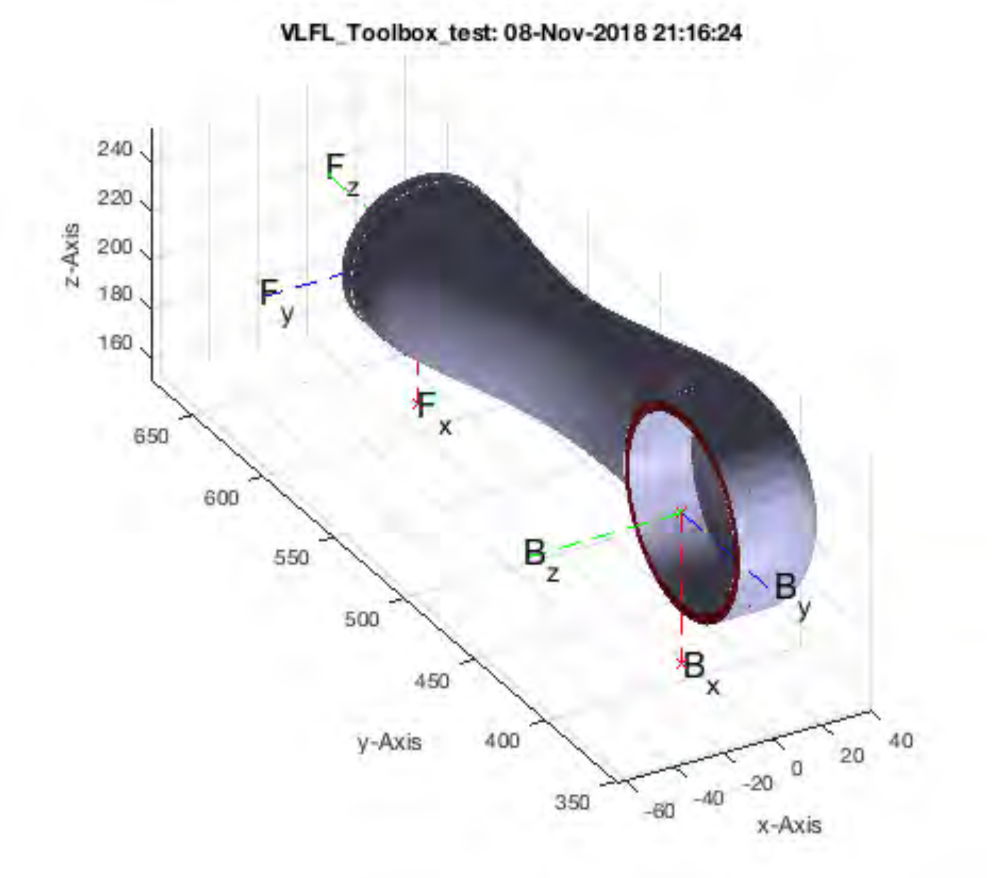
Plot the arm segment 2 of the Jaco robot

```
SGfigure; view(-30,30); SGTplot(JC2);
```



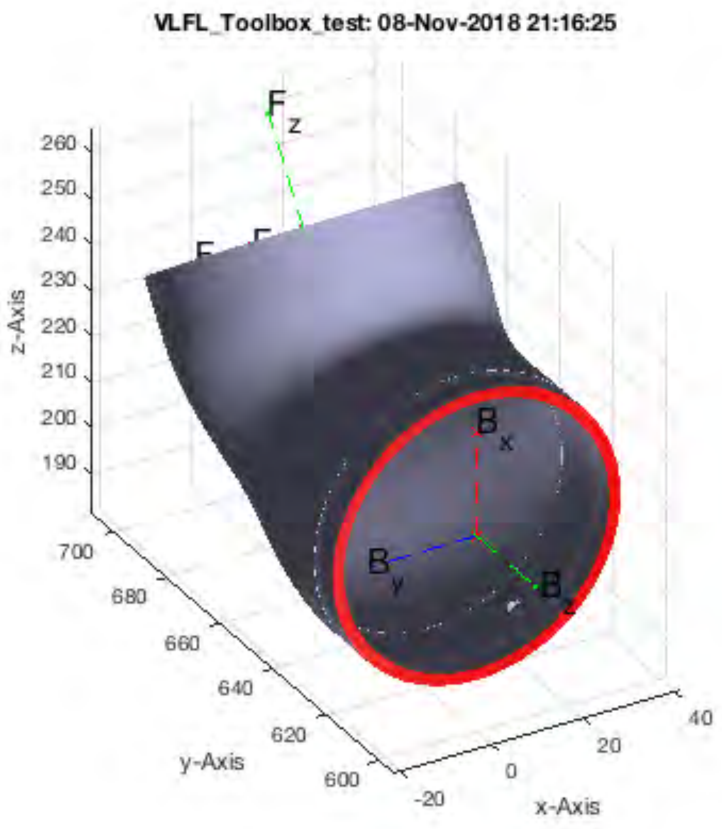
Plot the arm segment 3 of the Jaco robot

```
SGfigure; view(-30,30); SGTplot(JC3);
```



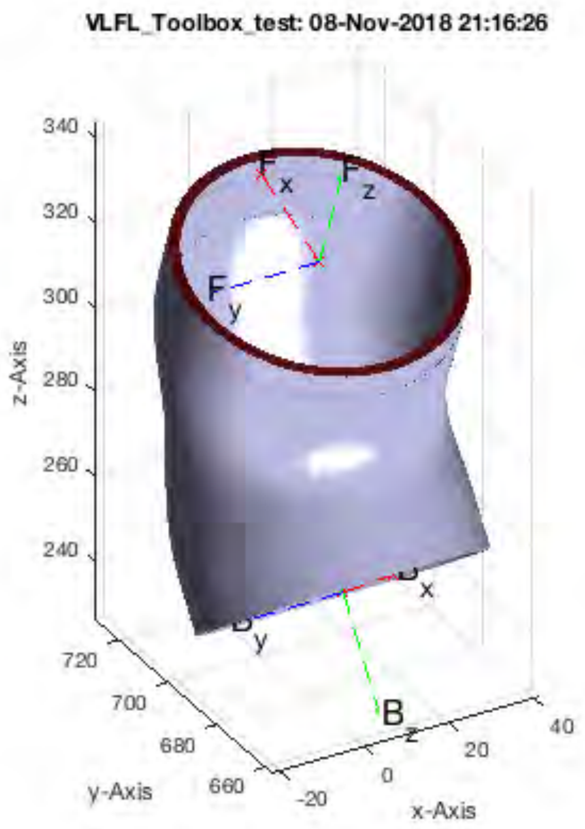
Plot the arm segment 4 of the Jaco robot

```
SGfigure; view(-30,30); SGTplot(JC4);
```



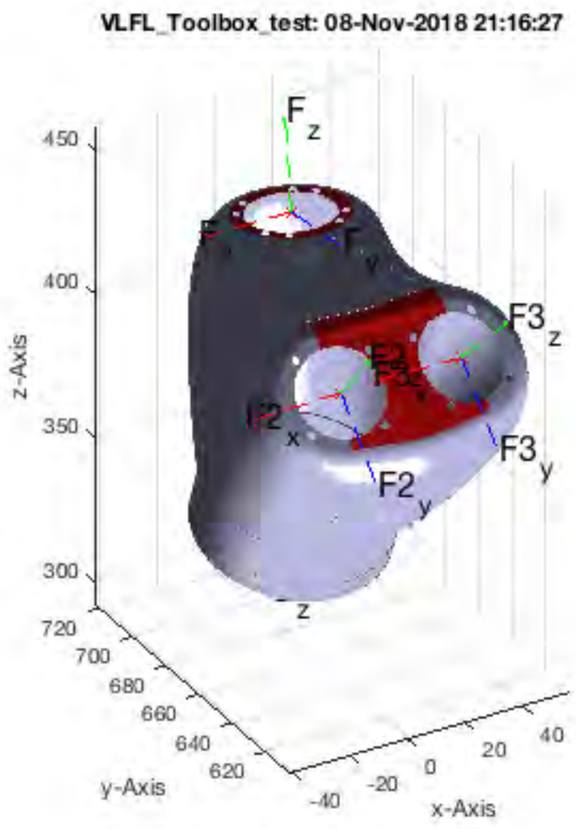
Plot the arm segment 5 of the Jaco robot

```
SGfigure; view(-30,30); SGTplot(JC5);
```



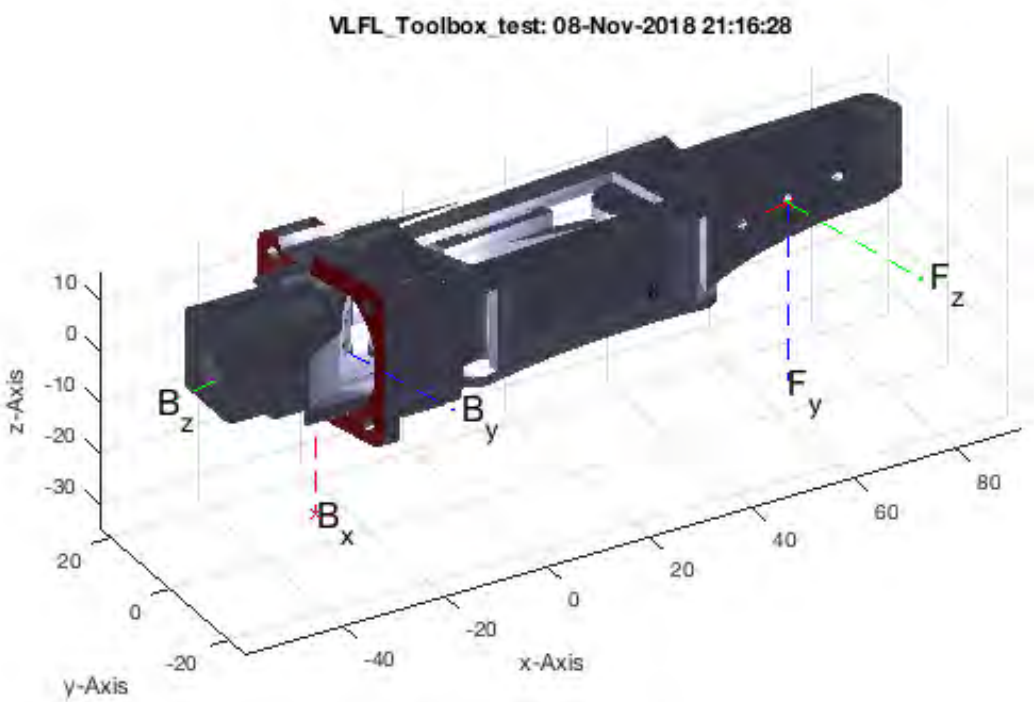
Plot the arm segment 6/the hand of the Jaco robot

```
SGfigure; view(-30,30); SGTplot(JC61);
```



Plot one finger segment 3 of the Jaco robot's hand

```
SGfigure; view(-30,30); SGTplot(JCF);
```



2. Attaching Frames to a Surface Model

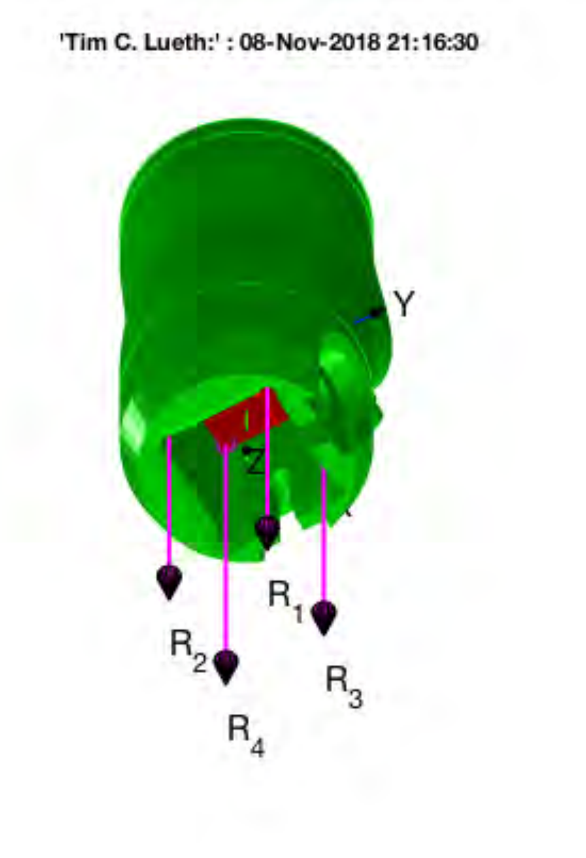
To learn how to attach frames, we make a copy of only the surface of jaco's base.

```
SG.VL=JC0.VL; SG.FL=JC0.FL; SG.col='w'; SG.alpha=0.9;

SGfigure; view(-30,30); SGplot(SG);
```

Now use SGTui to specify a planar or freeform surface by klicking on the surface. Turn the object before the klick into the desired orientation Now try to create a base frame by clicking on the lower surface. If you touch a freeform surface it may take while until the surfaces are autoamtically selected

```
SGfigure; SG=SGTui(SG,'B'), view(-60,-60);
```

```
SG =

  struct with fields:

       VL: [15230×3 double]
       FL: [30472×3 double]
      col: 'w'
    alpha: 0.9000
    Tname: {'B'}
        T: {[4×4 double]}
     TFiL: {[86×1 double]}
     TFoL: {[]}
```




```
SGfigure; SG=SGTui(SG,'F'), view(-60,+60);
```

```
SG =

  struct with fields:

       VL: [15230×3 double]
       FL: [30472×3 double]
      col: 'w'
    alpha: 0.9000
    Tname: {'B'  'F'}
        T: {[4×4 double]  [4×4 double]}
     TFiL: {[86×1 double]  [42×1 double]}
     TFoL: {[]}
```

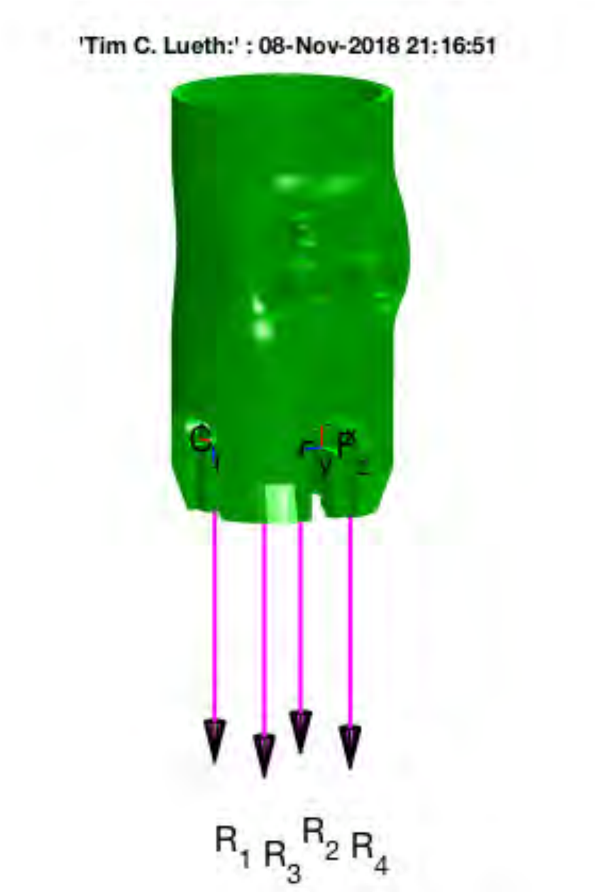
You may have notices that not only a surface but also the center of circular contours were detected and those can also be used for selection

```
SGfigure; SG=SGTui(SG,'C'), view(70,+10);
```

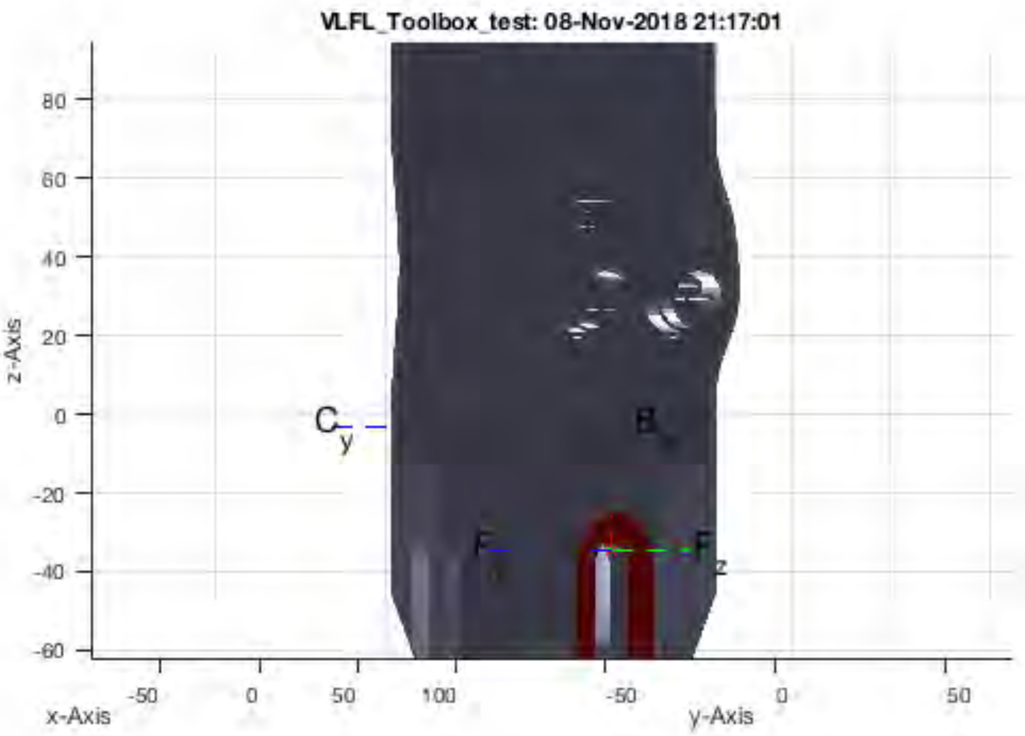
There is a slight difference between the center of the faces and the circle R1. By using 'R1' as parameter, the R1 coordinate system is used for the frame "C"

```
SGfigure; SG=SGTui(SG,'C','','R1'), view(60,+10);
```

```
SG =

  struct with fields:

       VL: [15230×3 double]
       FL: [30472×3 double]
      col: 'w'
    alpha: 0.9000
    Tname: {'B'  'F'  'C'}
        T: {[4×4 double]  [4×4 double]  [4×4 double]}
     TFiL: {[86×1 double]  [42×1 double]  [86×1 double]}
     TFoL: {[]}
```

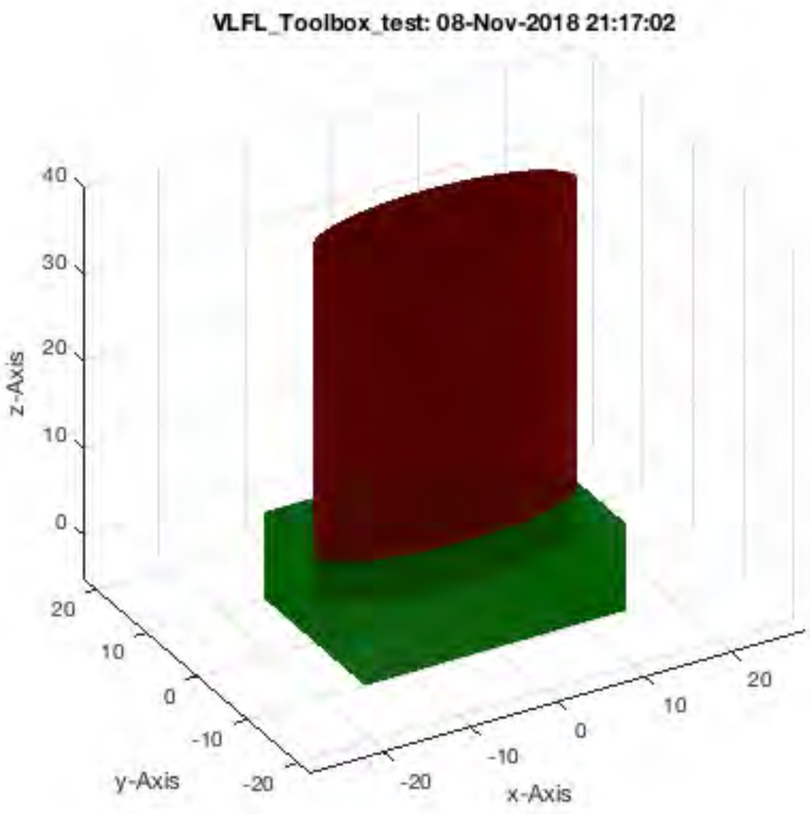



```
SGfigure;  SGTplot(SG,'C'); view(60,+0);
```



3. Spatial Arrangment of Solids relative to Frames

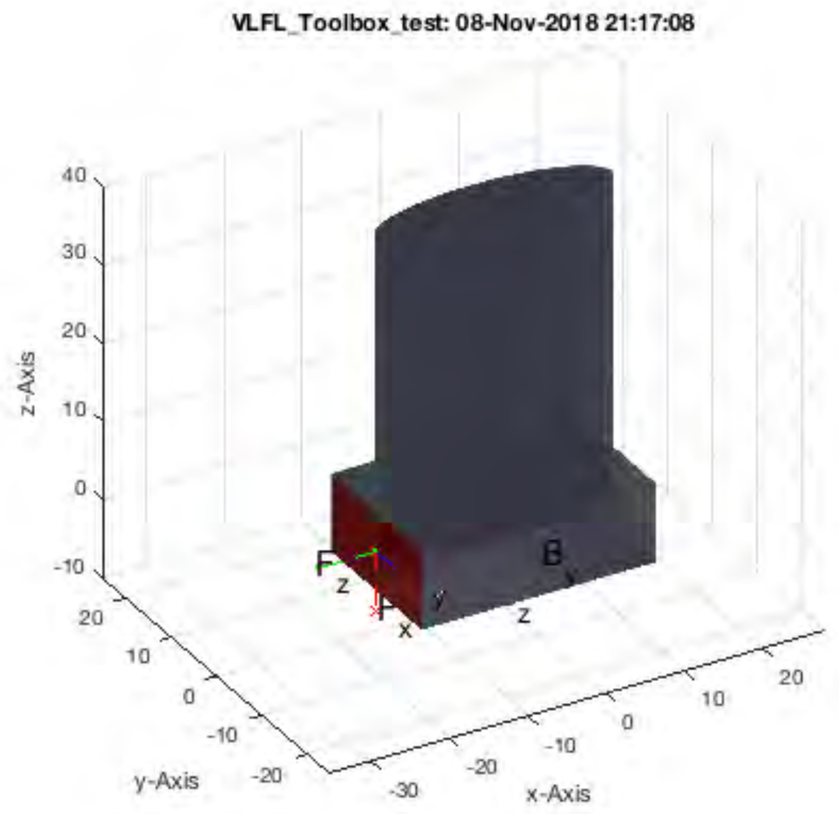
```
A=SGbox([30,20,10]); A.col='g'; A.alpha=0.9;
B=SGofCPLz(PLcircle(15,'','',5),40); B.col='r'; B.alpha=0.9;
SGfigure; SGplot({A,B}); view(-30,30);
```



Now attach frame to both solids

```
A=SGTui(A,'F');  % Follower Frame
B=SGTui(B,'B');  % Base Frame
```

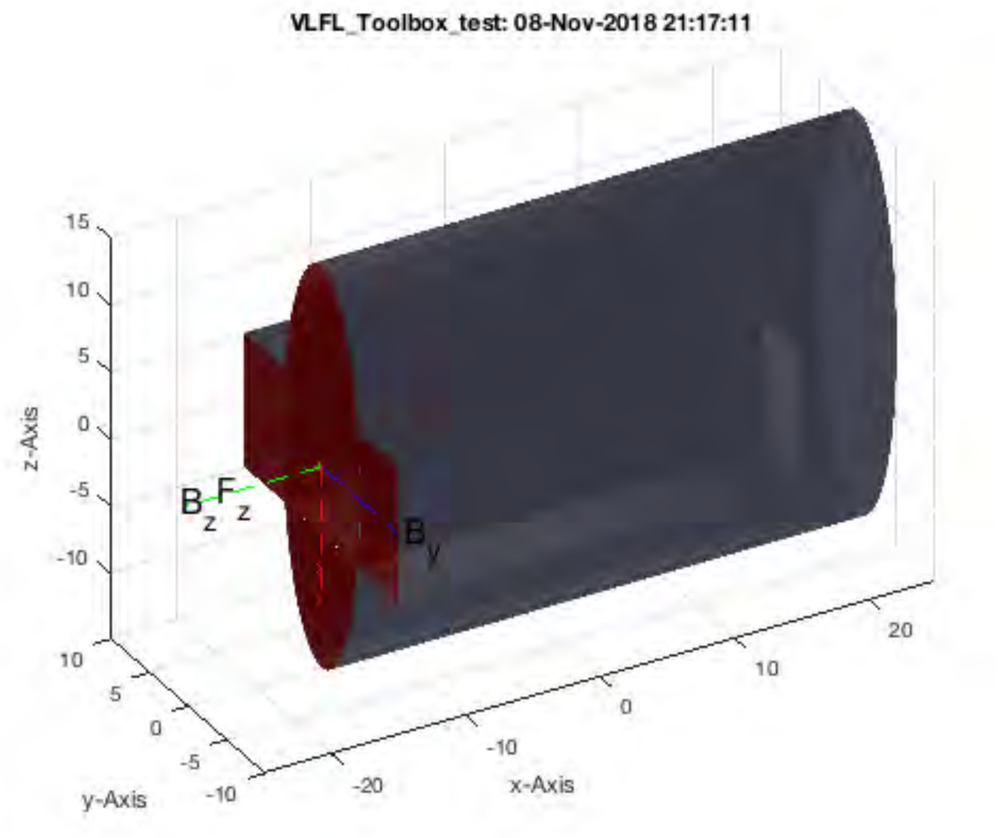

```
SGfigure; SGTplot(A); SGTplot(B); view(-30,30);
```



Now position solid B that its Frame 'B' matches with Frame 'F' of Solid A Afterwards, both Frames overlap completely.

```
SGtransrelSG(B,A,'matchT',{'B','F'})
```

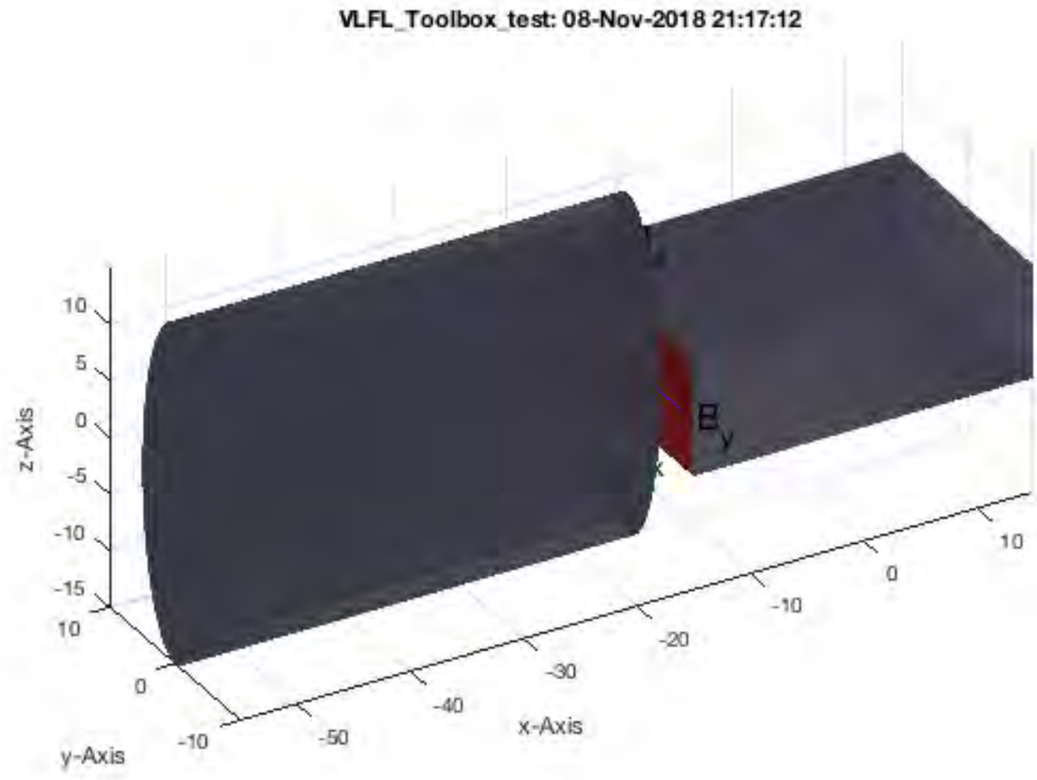
```
ans = 

  struct with fields:

      CPL: [55×2 double]
       VL: [110×3 double]
       FL: [216×3 double]
       PL: [55×2 double]
       EL: [55×2 double]
      col: 'r'
    alpha: 0.9000
    Tname: {'B'}
        T: {[4×4 double]}
     TFiL: {[53×1 double]}
     TFoL: {[]}
```



Now position solid B that its Frame 'B' aligns with Frame 'F' of Solid A Afterwards, both only axis Y overlap completely. Z and X have opposite orienations.

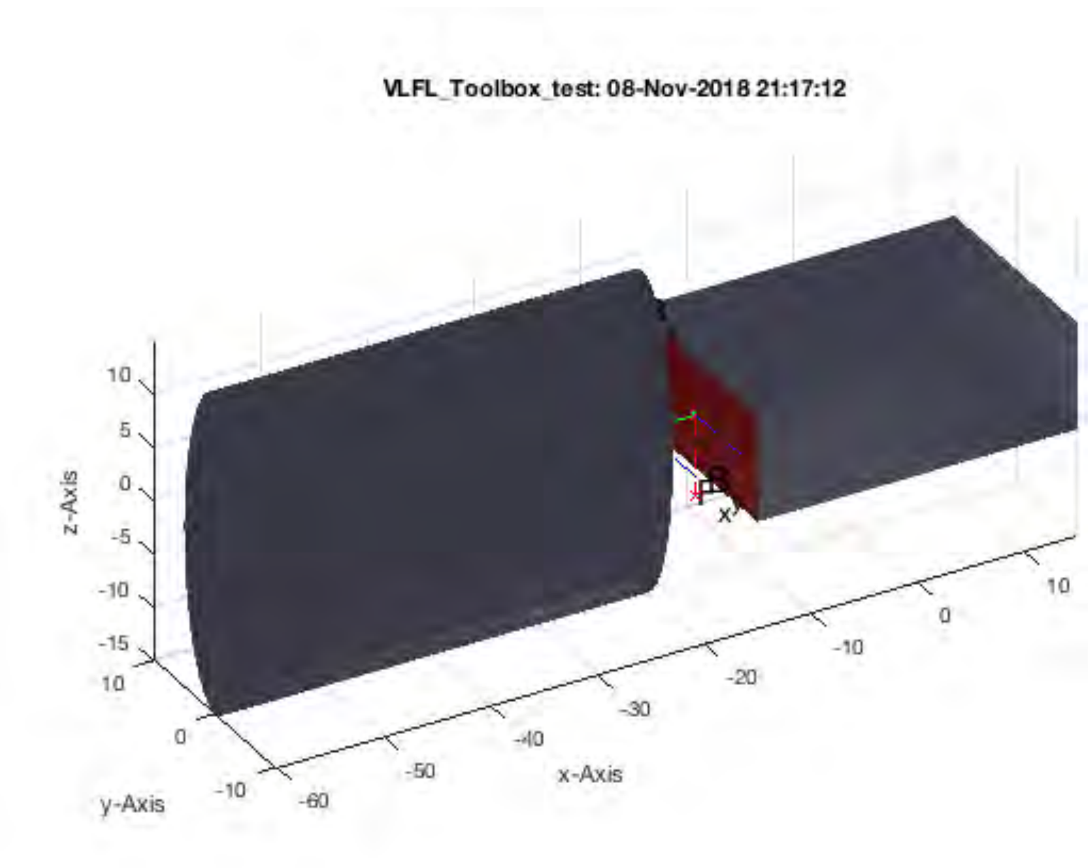
```
SGtransrelSG(B,A,'alignT',{'B','F'})
```

```
ans =

  struct with fields:

      CPL: [55×2 double]
       VL: [110×3 double]
       FL: [216×3 double]
       PL: [55×2 double]
       EL: [55×2 double]
      col: 'r'
    alpha: 0.9000
    Tname: {'B'}
        T: {[4×4 double]}
     TFiL: {[53×1 double]}
     TFoL: {[]}
```



Now position solid B that its Frame 'B' aligns with Frame 'F' of Solid A Afterwards, both only axis Y overlap completely. Z and X have opposite orienations. IN ADDITION create a distance of 5 mm

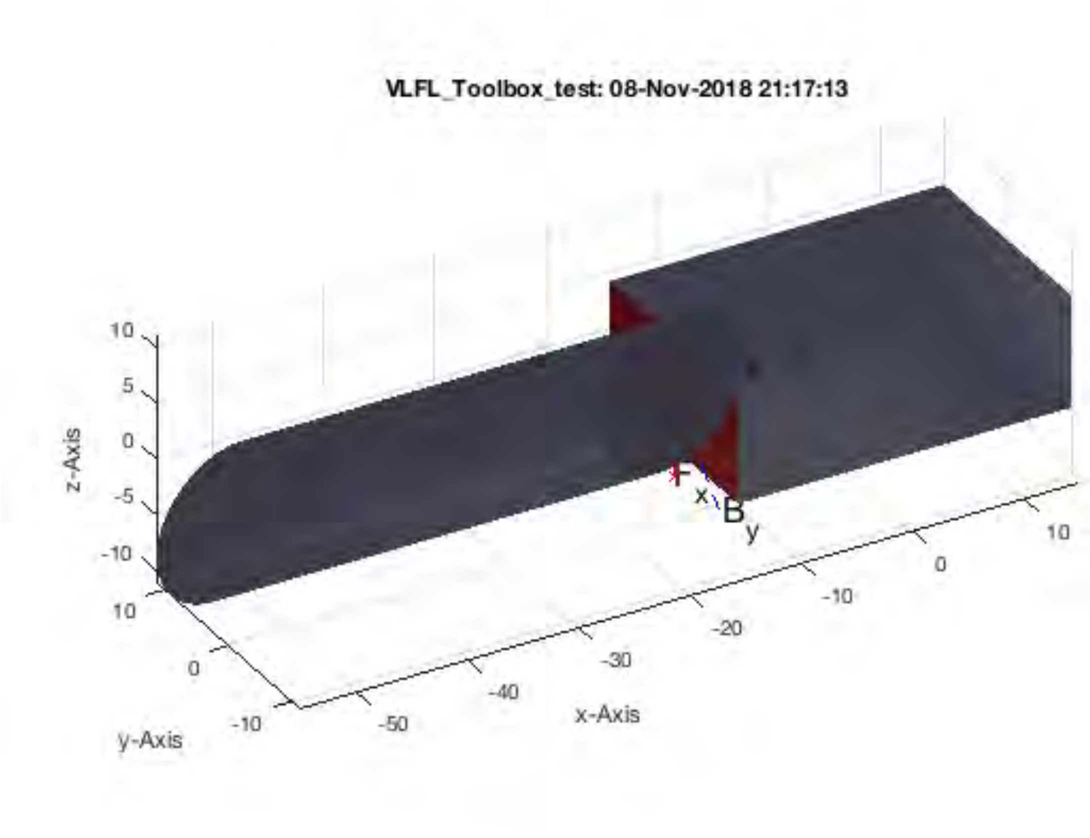
```
SGtransrelSG(B,A,'alignT',{'B','F',TofP([0 0 -5])});
```



Now position solid B that its Frame 'B' aligns with Frame 'F' of Solid A Afterwards, both only axis Y overlap completely. Z and X have opposite orienations. IN ADDITION TURN 45 degrees

```
SGtransrelSG(B,A,'alignT',{'B','F',TofR(rot(0,0,pi/4))})
```

```
ans =

  struct with fields:

      CPL: [55×2 double]
       VL: [110×3 double]
       FL: [216×3 double]
       PL: [55×2 double]
       EL: [55×2 double]
      col: 'r'
    alpha: 0.9000
    Tname: {'B'}
        T: {[4×4 double]}
     TFiL: {[53×1 double]}
     TFoL: {[]}
```



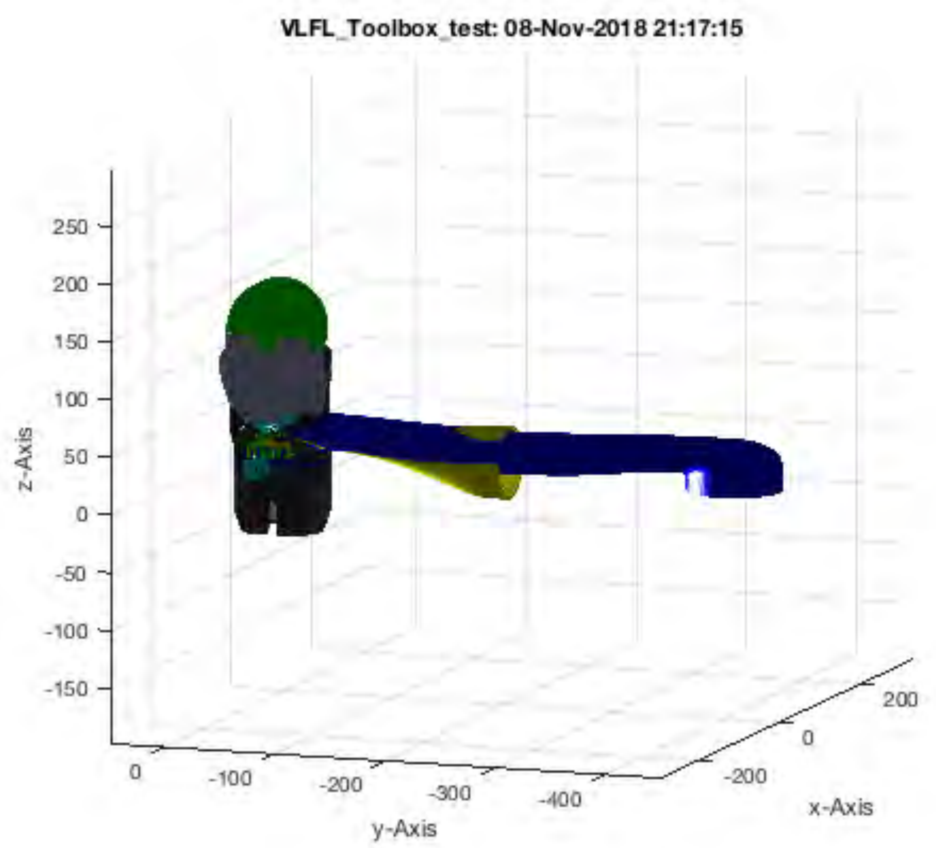
4. Simple Sequential Kinematic Chains

As soon as all solids have a base frame and a follower frame, it is possible to consider them als kinematic chain with some degrees of freedom between the frame. Such as rotation around the z-axis of the follower frame. The easist case is to define a cell list of all involved solids. To explain this feature, the origins of all solids are changed to their base frames. This is done just to avoid misunderstandings.

```
JC0=SGTsetorigin(JC0,'B'); % change the origin of Solid to Frame 'B'
JC1=SGTsetorigin(JC1,'B'); % change the origin of Solid to Frame 'B'
JC2=SGTsetorigin(JC2,'B'); % change the origin of Solid to Frame 'B'
JC3=SGTsetorigin(JC3,'B'); % change the origin of Solid to Frame 'B'
JC4=SGTsetorigin(JC4,'B'); % change the origin of Solid to Frame 'B'
JC5=SGTsetorigin(JC5,'B'); % change the origin of Solid to Frame 'B'
JC6=SGTsetorigin(JC6,'B'); % change the origin of Solid to Frame 'B'
JACO={JC0,JC1,JC2,JC3,JC4,JC5,JC6,JCF}
SGfigure; SGplot(JACO);   view(-70,10);
```

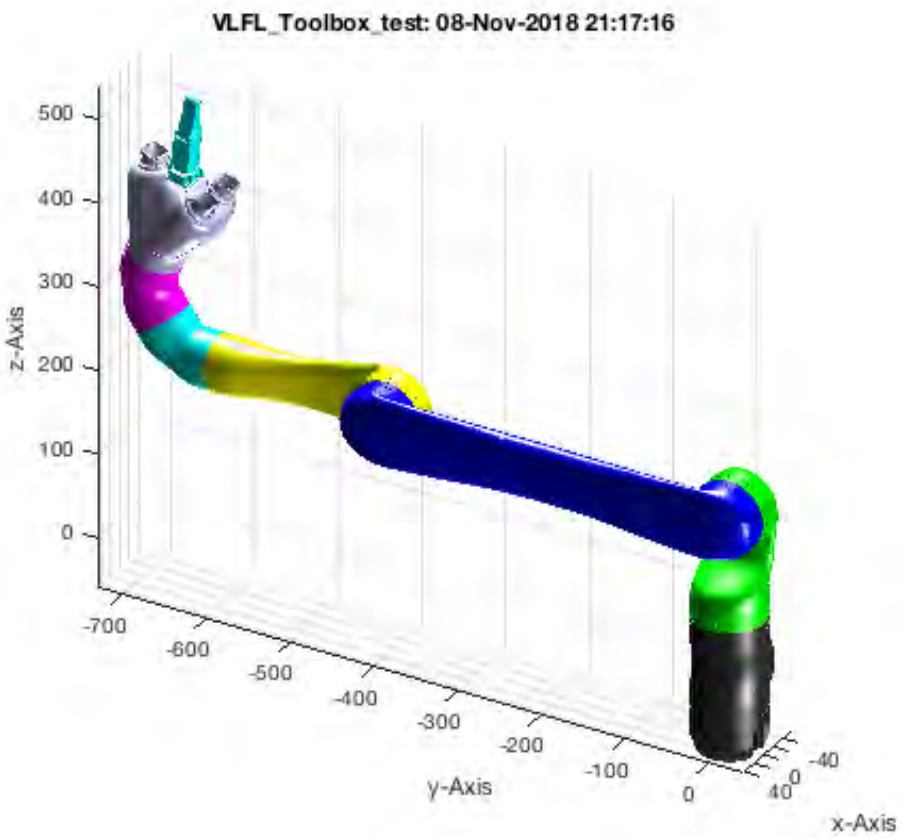
```
JACO =

  1×8 cell array

  Columns 1 through 4

    {1×1 struct}    {1×1 struct}    {1×1 struct}    {1×1 struct}

  Columns 5 through 8
```

```
    {1×1 struct}    {1×1 struct}    {1×1 struct}    {1×1 struct}
```



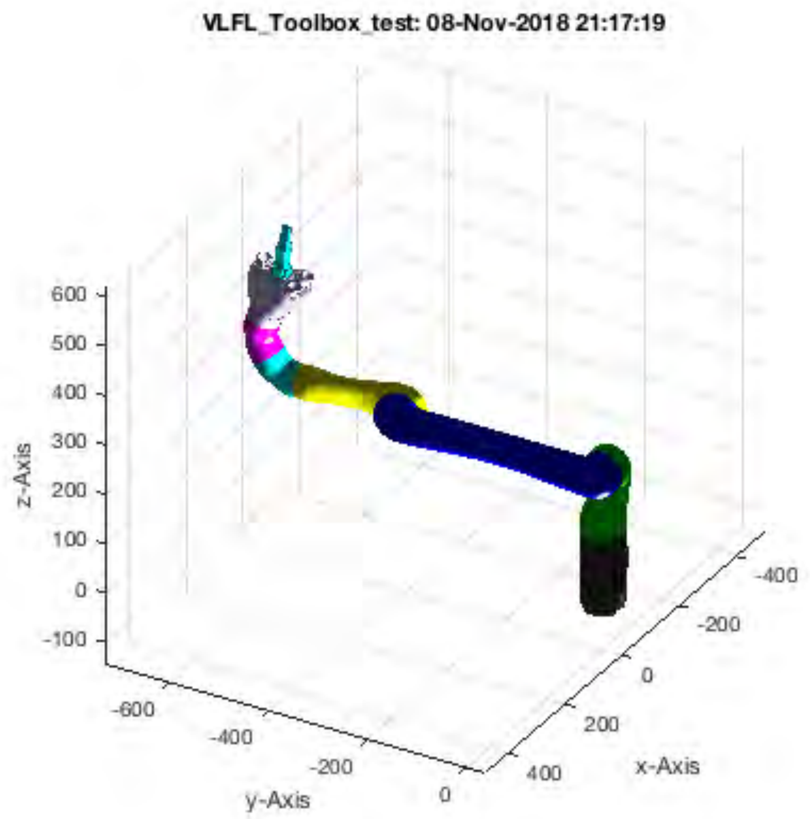
There is a function that aligns automatically base and follower frame AND modifies the vertex list for all of the solids but the first one.

```
SGfigure; SGTchain(JACO); view(120,30);
```



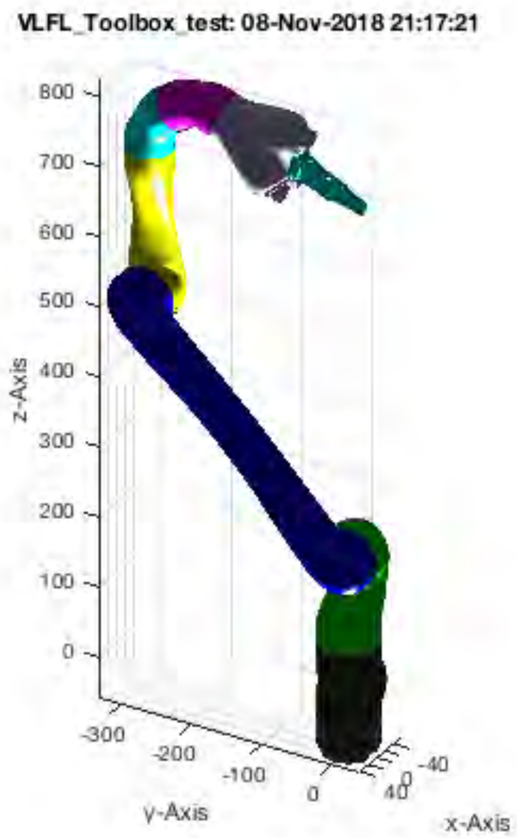
The function SGTchain changes the all vertex coordinates, therefor afterwards the parts seem to stay is space as the kinematic chain. In this example X is a pose of the robot if all frames are aligned. If X is plotted as a solid it looks like a robot in a specific pose.

```
X=SGTchain(JACO);
SGfigure; SGplot(X); view(120,30);
```



SGTchain also allow to deliver additional rotatorial parameters. For each joint a rotating angle can be specified. NEvertehelss, currently the first value is ignored, since there is no base frame. The nth rotation is relative to the base frame of the nth element.

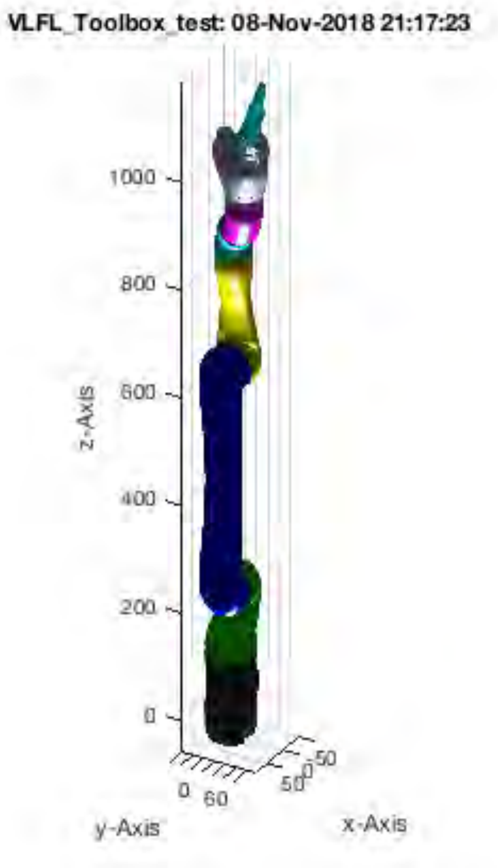
```
SGTchain(JACO,[nan 0 +pi/4 -pi/4]); view(120,30);

% again, the output value is the same surface cell list but describing
% exactly this position.
```



5. Calibration of a Sequential Kinematic Chain

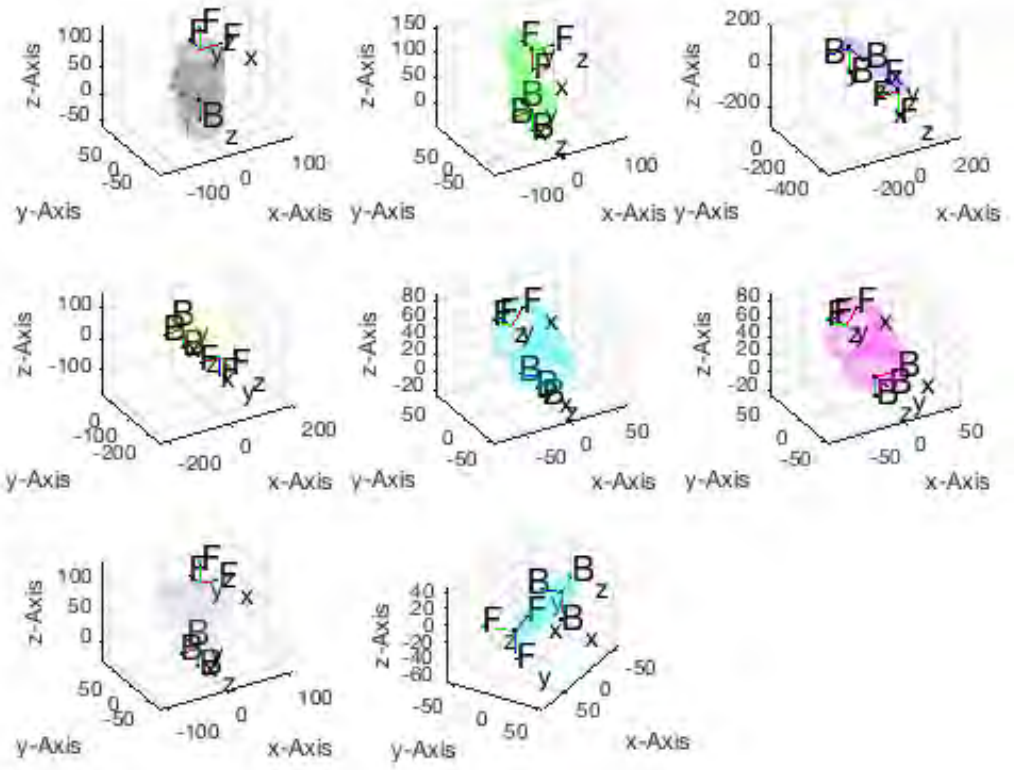
Often it is not possible to specify the frames using SGTui exactly as the real motor configuration is. Therefor it is necessary to calibrate the zero position. In case try to bring the robot by a set of rotating angles into the desired zero position or use an additional angle vector as offest. As soon as the offset is know call SGTcalibchain using the offset values. For example

```
SGTchain(JACO,[nan 0 pi/2 0 pi/4 -pi 0]); view(120,30);
```



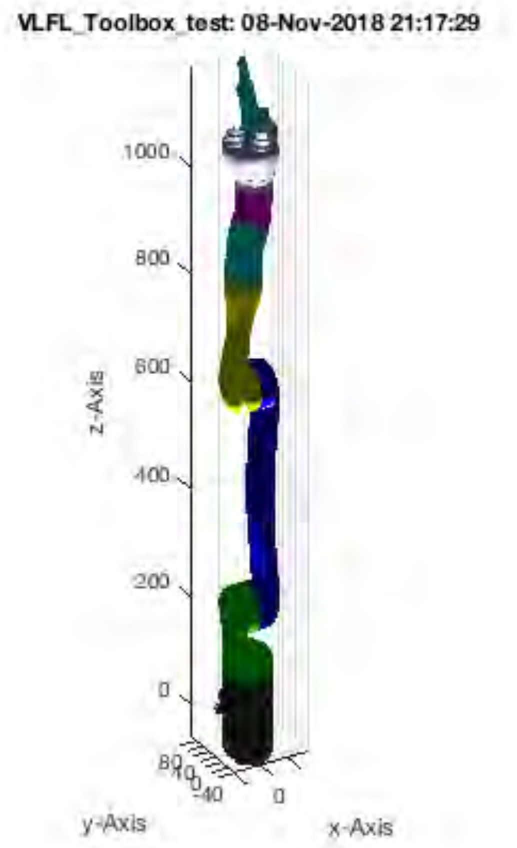
now change all frames of the chain to create a new zero position. In this case ALL elements need a value. Even the finger element.

```
SGTcalibchain(JACO,[nan 0 pi/2 0 pi/4 -pi 0 0]); view(120,30);
JACO_cal=ans;
```



Now the robot has a new zero position The position shown here has nothing to do with the real zero position of KINOVA's JACO robot.

```
SGTchain(JACO_cal);
```

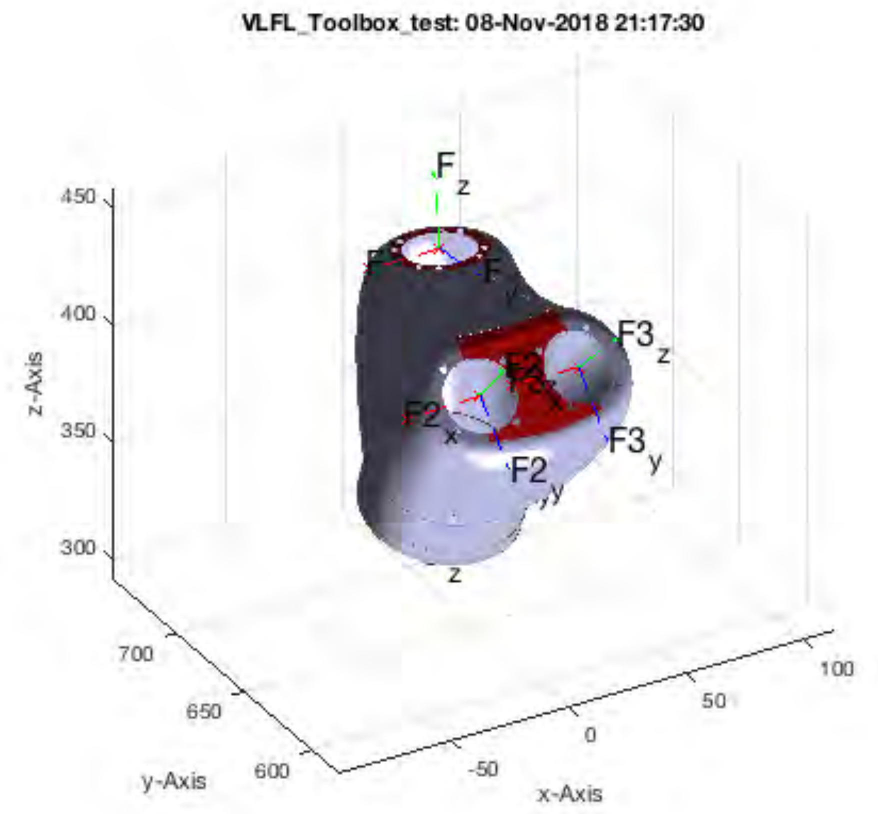


6. Creating Kinematic Trees

It is easy to see that the real JACO has three fingers and a simple chain is not enough. Therefor there is an additional format for SGTchain to explain the kinematic structure and the order of motors/angles. At first we need three follower frames. THis is part of solid JC61. Beside "F" tehre is also "F1" and "F2"

```
SGfigure; SGTplot(JC61); view(-30,30)
JACO={JC0,JC1,JC2,JC3,JC4,JC5,JC61,JCF}
```

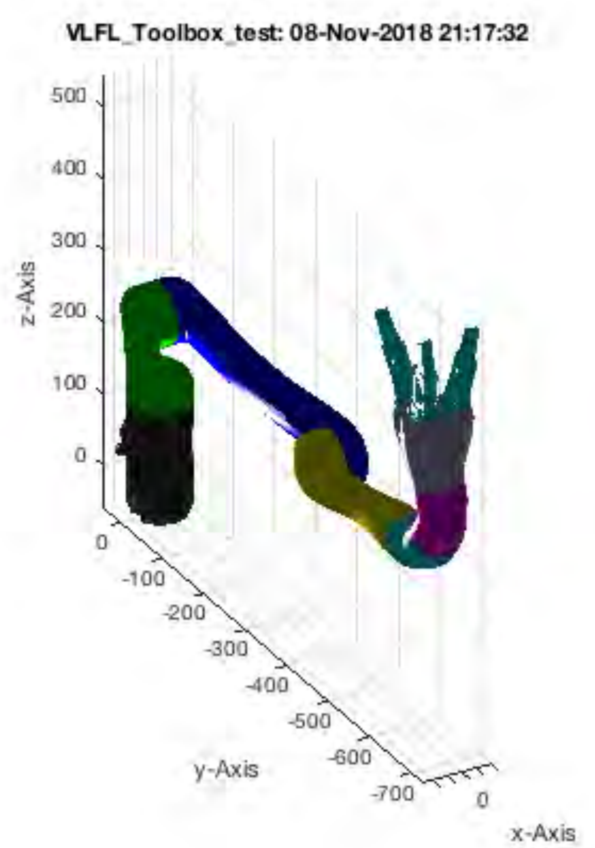
```
JACO =

  1×8 cell array

  Columns 1 through 4

    {1×1 struct}    {1×1 struct}    {1×1 struct}    {1×1 struct}

  Columns 5 through 8

    {1×1 struct}    {1×1 struct}    {1×1 struct}    {1×1 struct}
```



Next is to specify two additional degrees of freedom between Part 7 and Frame "F2" and Part 8 Frame "B" and Part 7 and Frame "F3" and Part 8 Frame "B". Automatically, there are two additional rotations or motors introduced. In case of the real JACO robot, the joints 7, 8, 9 are not rotational but linear for the fingers.

```
SGTchain(JACO,'','',1:8,[7 'F2' 8 'B', 7 'F3' 8 'B'])
```

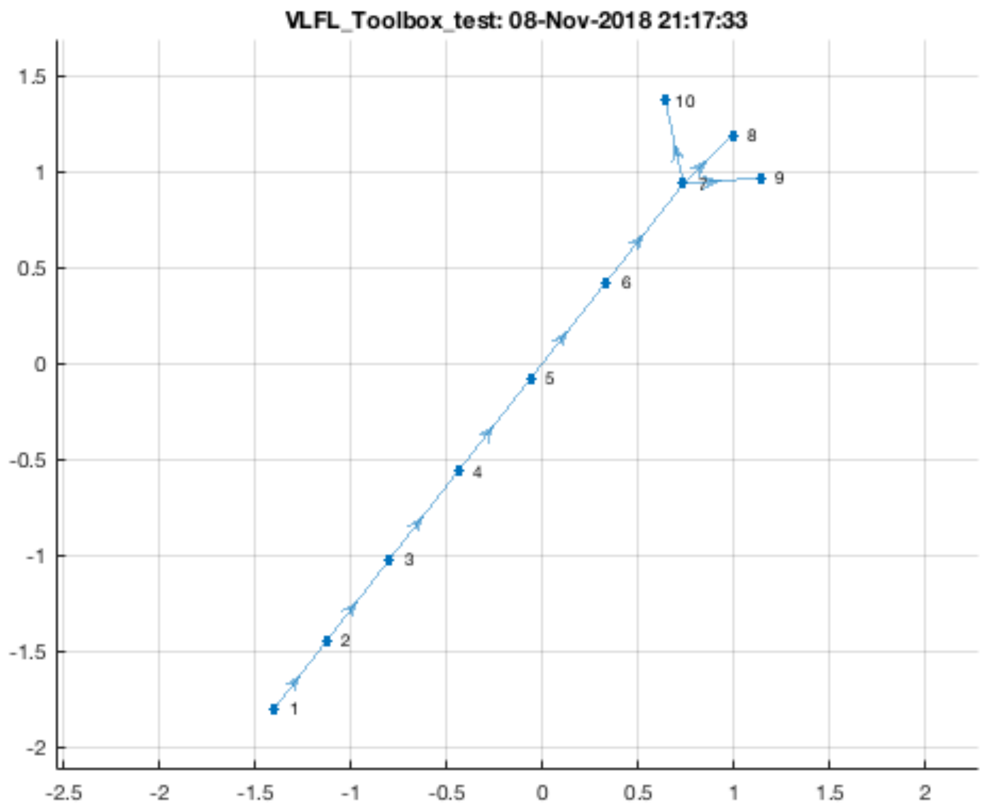
```
ans =

  1×10 cell array

  Columns 1 through 4

    {1×1 struct}    {1×1 struct}    {1×1 struct}    {1×1 struct}

  Columns 5 through 8

    {1×1 struct}    {1×1 struct}    {1×1 struct}    {1×1 struct}

  Columns 9 through 10

    {1×1 struct}    {1×1 struct}
```



To understand better the kinematic chains, it is also possible to call a auxiliary function to create a kinematic chain table. This function returns the number/order of the DoF and which frames are connected and which solid was used for the connection. In Future also the type of DoF will be added to this list

```
SGTframeChain(1:8,[7 'F2' 8 'B', 7 'F3' 8 'B'])
```

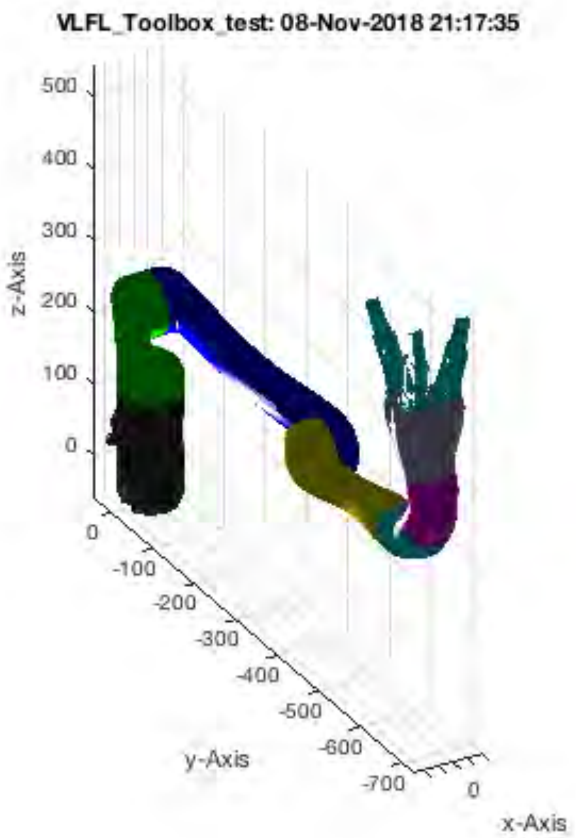
```
ans =

  10×5 cell array

    {[ 1]}    {'_' }    {'B'}    {[0]}    {[1]}
    {[ 2]}    {'F' }    {'B'}    {[1]}    {[2]}
    {[ 3]}    {'F' }    {'B'}    {[2]}    {[3]}
    {[ 4]}    {'F' }    {'B'}    {[3]}    {[4]}
    {[ 5]}    {'F' }    {'B'}    {[4]}    {[5]}
    {[ 6]}    {'F' }    {'B'}    {[5]}    {[6]}
    {[ 7]}    {'F' }    {'B'}    {[6]}    {[7]}
    {[ 8]}    {'F' }    {'B'}    {[7]}    {[8]}
    {[ 9]}    {'F2'}    {'B'}    {[7]}    {[8]}
    {[10]}    {'F3'}    {'B'}    {[7]}    {[8]}
```



It is also possible to call SGT directly using this table:

```
FC=SGTframeChain(1:8,[7 'F2' 8 'B', 7 'F3' 8 'B'])
SGTchain(JACO,'','',FC);
```

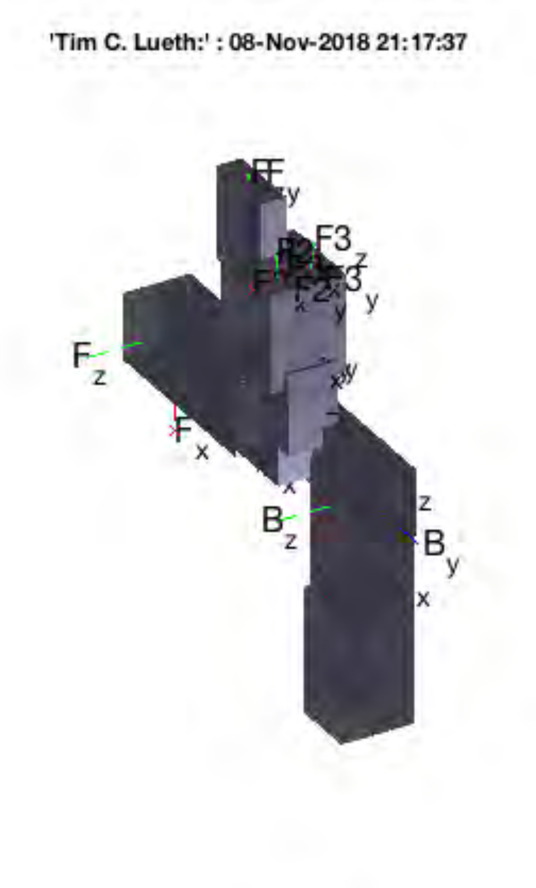
```
FC =

  10×5 cell array

    {[ 1]}    {'_' }    {'B'}    {[0]}    {[1]}
    {[ 2]}    {'F' }    {'B'}    {[1]}    {[2]}
    {[ 3]}    {'F' }    {'B'}    {[2]}    {[3]}
    {[ 4]}    {'F' }    {'B'}    {[3]}    {[4]}
    {[ 5]}    {'F' }    {'B'}    {[4]}    {[5]}
    {[ 6]}    {'F' }    {'B'}    {[5]}    {[6]}
    {[ 7]}    {'F' }    {'B'}    {[6]}    {[7]}
    {[ 8]}    {'F' }    {'B'}    {[7]}    {[8]}
    {[ 9]}    {'F2'}    {'B'}    {[7]}    {[8]}
    {[10]}    {'F3'}    {'B'}    {[7]}    {[8]}
```



7. Calculating Boxes for Quick Collision Checks

The algorithms for collision check are very time consuming since there is a need for testing all traingles for collision/penetration. This makes sensor for bollean operations but is not suitable for gast follision checks during a movement of a kinematic chain. Therefor there is a wish to perform these steps with a simplified kineamtic model, consisting of bounding boxes

```
J=SGTchain(JACO,[nan,0 pi pi]);

SGTBB(J); JB=ans; view(-30,30);
```



8. Collision Check

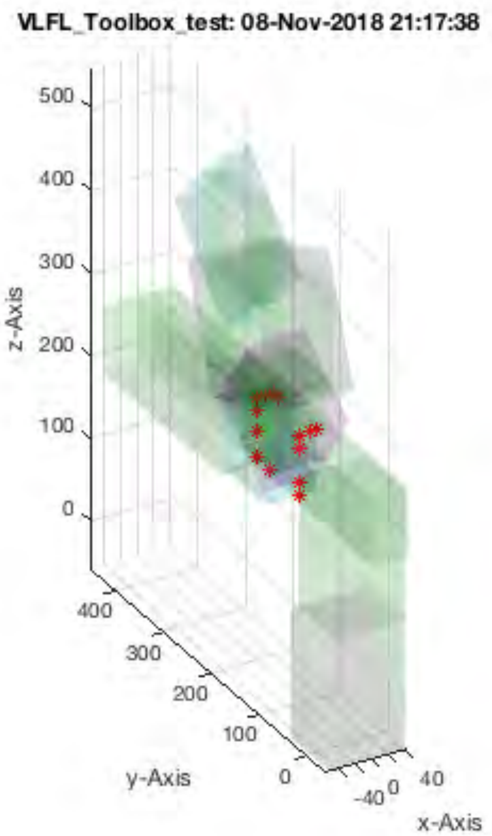
There are two functions:

- **iscollofVLBB** for testing of Vertices are inside of a bounding box
- **iscollofSG** for face testing of two solids or selftest of one solid Please read the documentation for both functions to see what is possible

```
% Self collision test in a safe configuration
iscollofSG(SGTchain(JB,[nan 0 pi pi]))
```

```
ans =

    1.4999  111.7080  257.2501
   14.7772  111.7080  257.2501
   13.5351  111.7080  257.2501
   13.5351  111.7080  257.2501
   21.7502  111.7080  257.2501
   21.7502  111.7080  257.2501
    1.4999  111.7080  241.2684
    1.4999  111.7080  241.2684
    1.4999  111.7080  257.2501
   14.7772  111.7080  257.2501
    1.4999  111.7080  184.4999
    1.4999  111.7080  184.4999
    1.4999  111.7080  200.3724
    1.4999  111.7080  200.3724
```

```
    1.4999  173.6560  184.4999
    1.4999  173.6560  184.4999
   21.7502  191.0959  257.2501
   21.7502  191.0959  257.2501
   21.7502  192.0389  257.2501
   21.7502  192.0389  257.2501
    1.4999  203.0001  241.2684
   13.5351  203.0001  257.2501
    1.4999  203.0001  241.2684
   13.5351  203.0001  257.2501
   21.7502  203.0001  257.2501
   21.7502  203.0001  257.2501
    1.4999  203.0001  215.6650
    1.4999  203.0001  257.2501
    1.4999  203.0001  184.4999
    1.4999  203.0001  184.4999
    1.4999  203.0001  257.2501
    1.4999  203.0001  215.6650
```

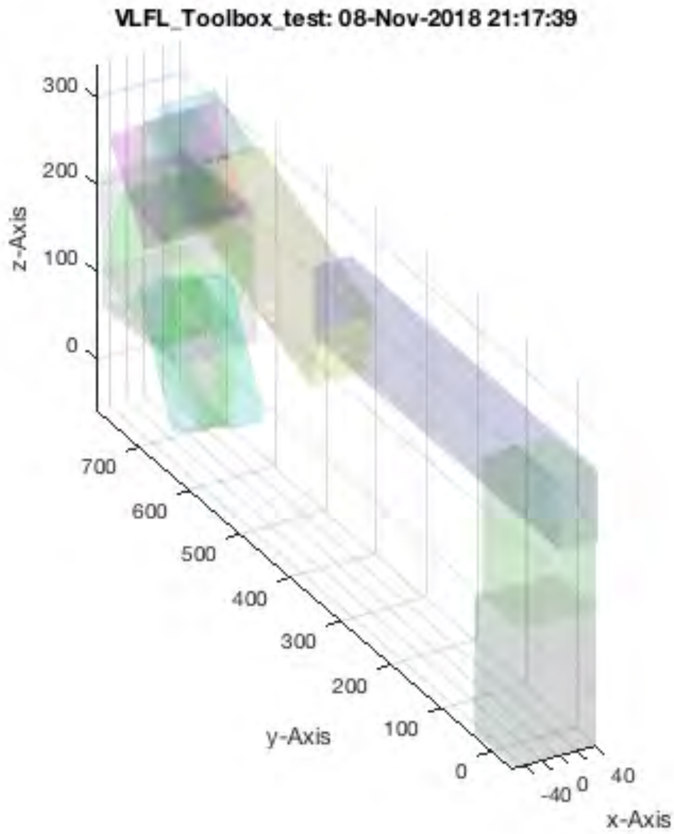


Self collision test in a problematic configuration

```
iscollofSG(SGTchain(JB,[nan 0 pi pi/10]))
```

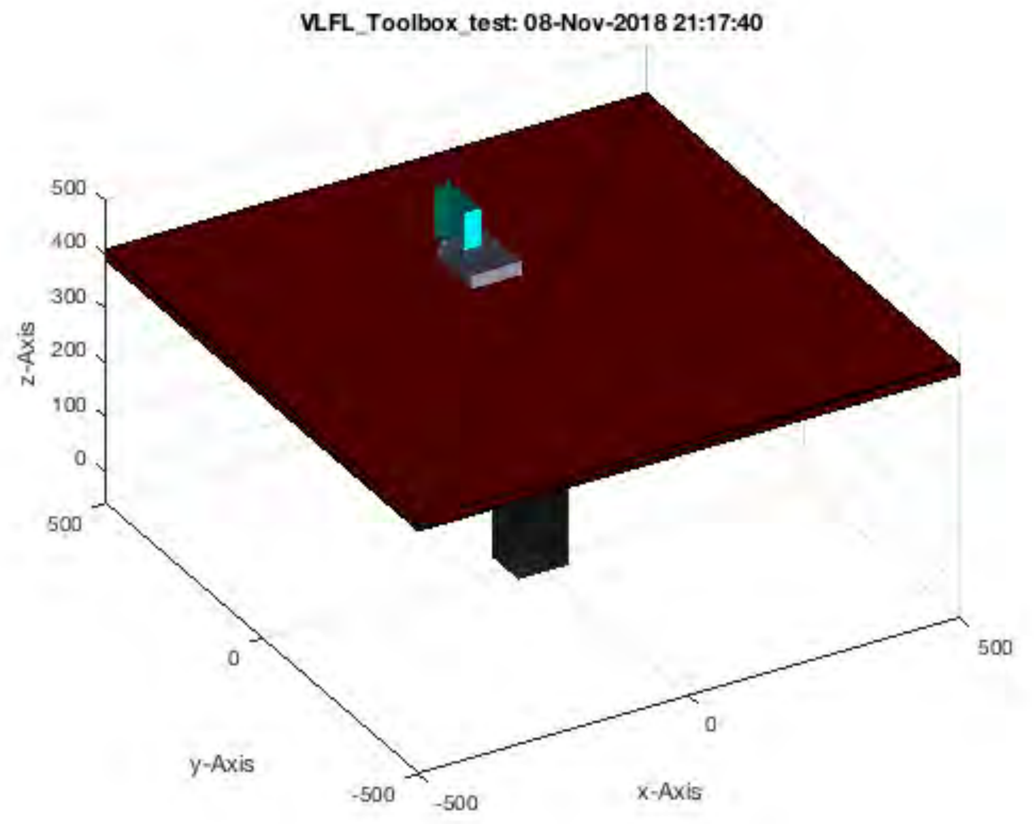
```
ans =
```

```
0×3 empty double matrix
```



Collsion collision test in a problematic configuration

```
A=SGbox([1000,1000,20]); A=SGtransP(A,[0 0 400]);
SGTchain(JB,[nan 0 pi pi]); SGplot(A);
```

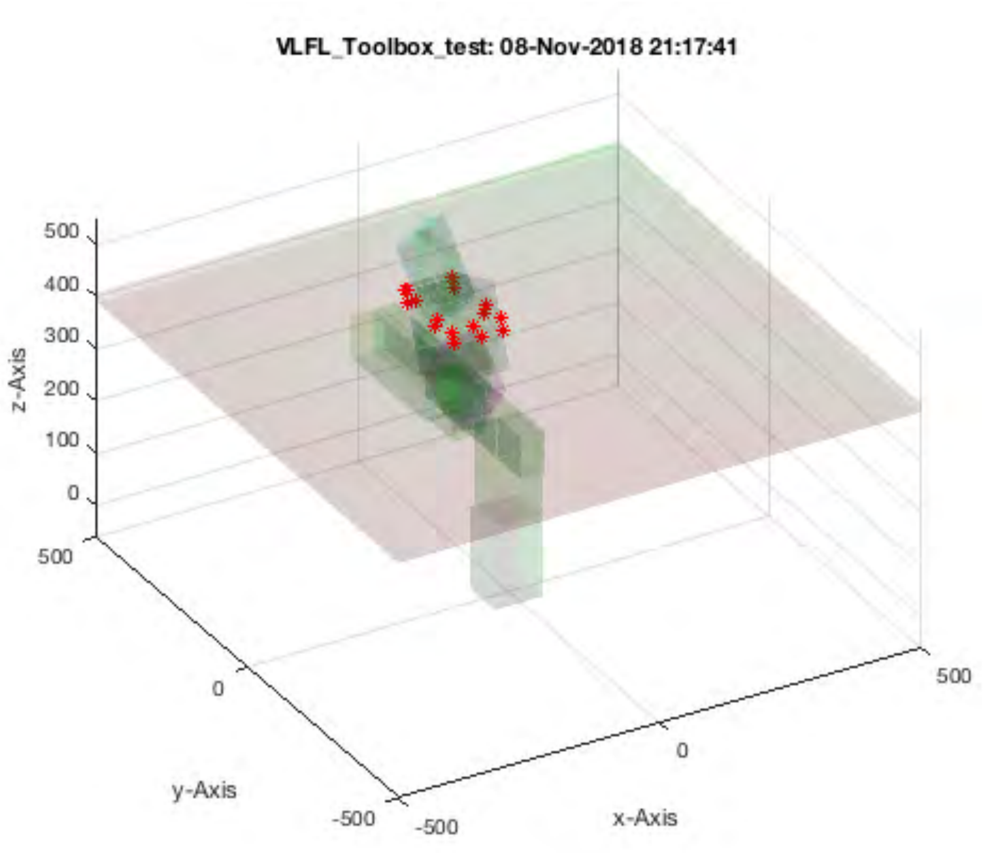


Now make a test for crossing robot and solid

```
iscollofSG(SGTchain(JB,[nan 0 pi pi]),A)
```

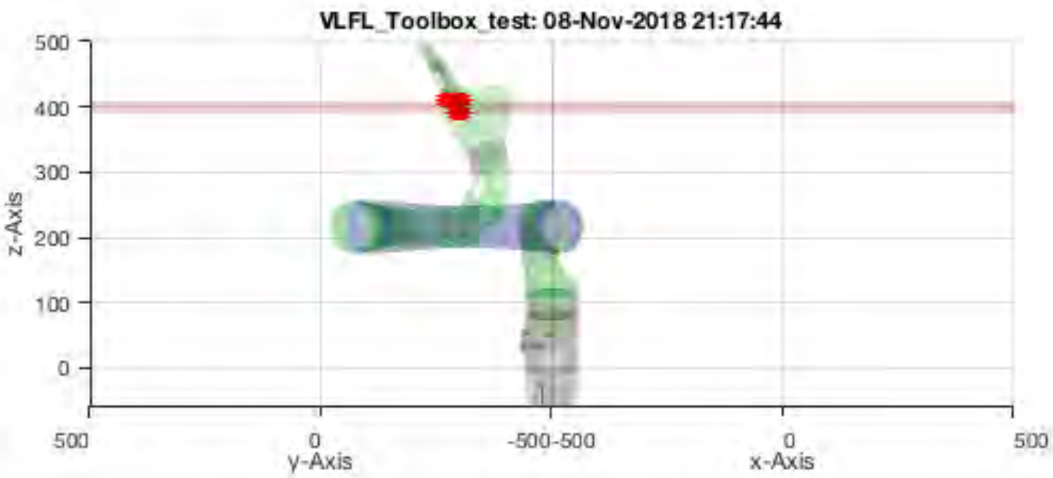
```
ans =

   36.9130   82.2649  389.9999
   36.9130   82.2649  389.9999
   -3.9110   82.2649  389.9999
   -3.9110   82.2649  389.9999
  -55.9248   82.2650  389.9999
  -55.9248   82.2650  389.9999
   36.9130   89.5443  410.0001
   36.9130   89.5443  410.0001
  -16.8148   89.5444  410.0001
  -16.8148   89.5444  410.0001
  -55.9248   89.5444  410.0001
  -55.9248   89.5444  410.0001
   36.9130  138.7881  410.0001
   36.9130  138.7881  410.0001
  -55.9248  138.7881  410.0001
  -55.9248  138.7881  410.0001
   36.9130  147.7560  389.9999
   36.9130  147.7560  389.9999
  -55.9248  147.7560  389.9999
  -55.9248  147.7560  389.9999
   36.9131  240.4606  389.9999
```

```
  36.9131   240.4606   389.9999
 -36.7140   240.4606   389.9999
 -36.7140   240.4606   389.9999
 -55.9247   240.4607   389.9999
 -55.9247   240.4607   389.9999
  36.9131   247.7400   410.0001
  36.9131   247.7400   410.0001
 -49.6179   247.7401   410.0001
 -49.6179   247.7401   410.0001
 -55.9247   247.7401   410.0001
 -55.9247   247.7401   410.0001
```



*The full test with the original geometry is much slower if the collision objects have more facets than those 12 of the simple box!

```
iscollofSG(SGTchain(JACO,[nan 0 pi pi]),A,true); view(-45,0)
```



Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 21:17:45!
Executed 08-Nov-2018 21:17:48 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
map_toolbox
matlab
robotics_system_toolbox
===========================================================================================
==========
```

Published with MATLAB® R2018a

Tutorial 36: Creating a Patient-Individual Arm-Skin Protector-Shell

2017-07-07: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de) - Last Change: 2017-07-25

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)

Motivation for this tutorial: (Originally SolidGeometry 4.0 required)

I'd like to thank Gweni-Viani Alonso-Aruffo for her student internship in Spring 2017 in my lab at TU of Munich. She recorded the surface data and wrote Matlab fnctns for also creating SVG data for laser cutting of cloth to protect the skin from getting direct in contact to the 3D printed polymers.

```
% function VLFL_EXP36
```

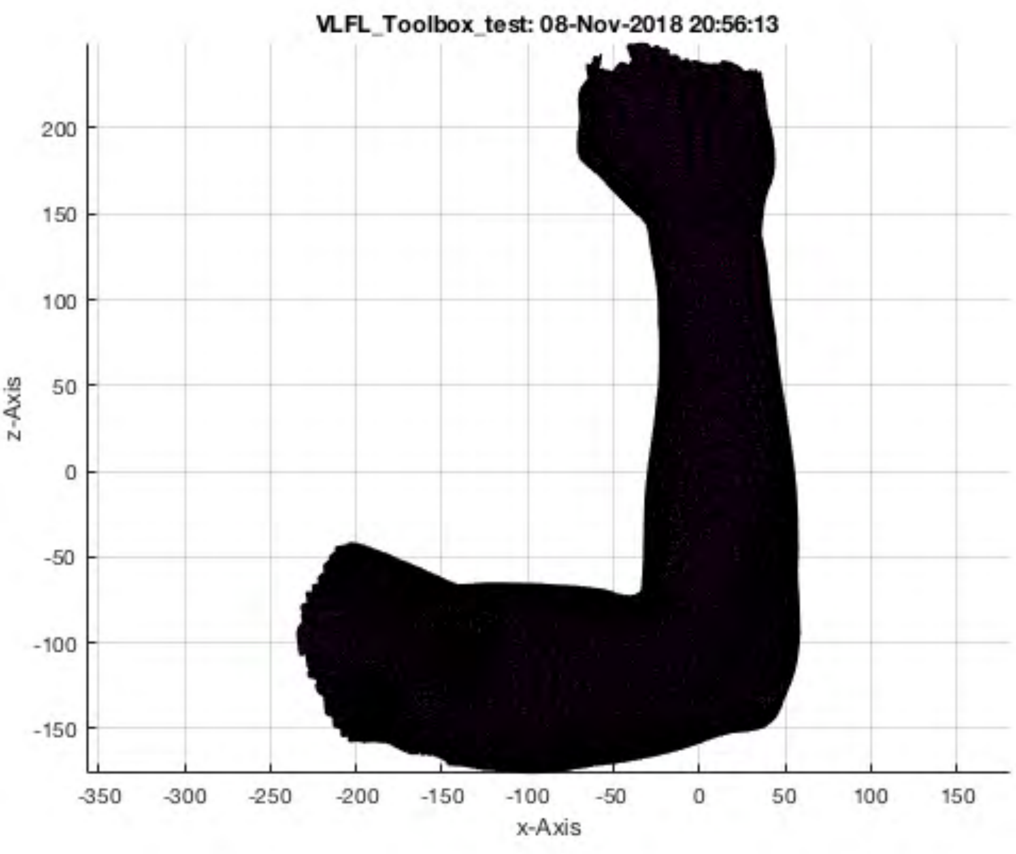
1. Loading Surface Data

2. Interactive Selection of the Area

```
load AAruffo_surf.mat  % use loadweb('AAruffo_surf.mat'); the first time

SGfigure(SG1); view(19,15); camlight;
beep; pause;
ginput(1); p1=select3d
ginput(1); p2=select3d
T=T2P(p1,p2-p1)
Ti=eye(4)/T
pn=VLtransT(p2',Ti); z=pn(3)

SGX=SGtransT(SG1,Ti); SGfigure(SGX); view(0,0)
```

```
p1 =

  -14.1774
  569.0837
```

```
  -21.2273


p2 =

  -24.5089
  583.5873
   76.9487


T =

   -0.9946         0   -0.1035  -14.1774
   -0.0151   -0.9893    0.1454  569.0837
   -0.1024    0.1461    0.9839  -21.2273
         0         0         0    1.0000


Ti =

   -0.9946   -0.0151   -0.1024   -7.6640
    0.0000   -0.9893    0.1461  566.0758
   -0.1035    0.1454    0.9839  -63.3028
         0         0         0    1.0000


z =

   99.7779
```



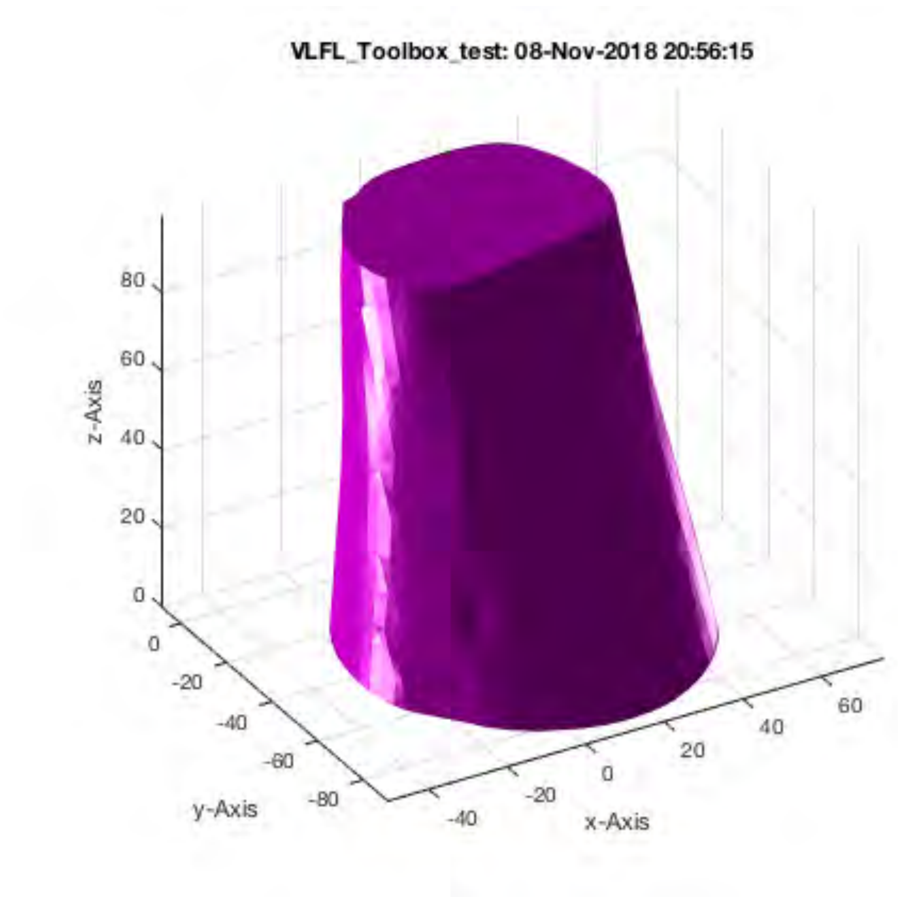
Cut the selected arm area

```
pn=VLtransT(p2',Ti); z=pn(3)
SGcut(SGX,[0 z]);
[SGout,SGin]=SGcut(SGX,[0 z]);
```

```
z =

   99.7779
```

Show only the selected

```
SGfigure(SGin); view(-30,30); VLFLplotlight(1,1);
```

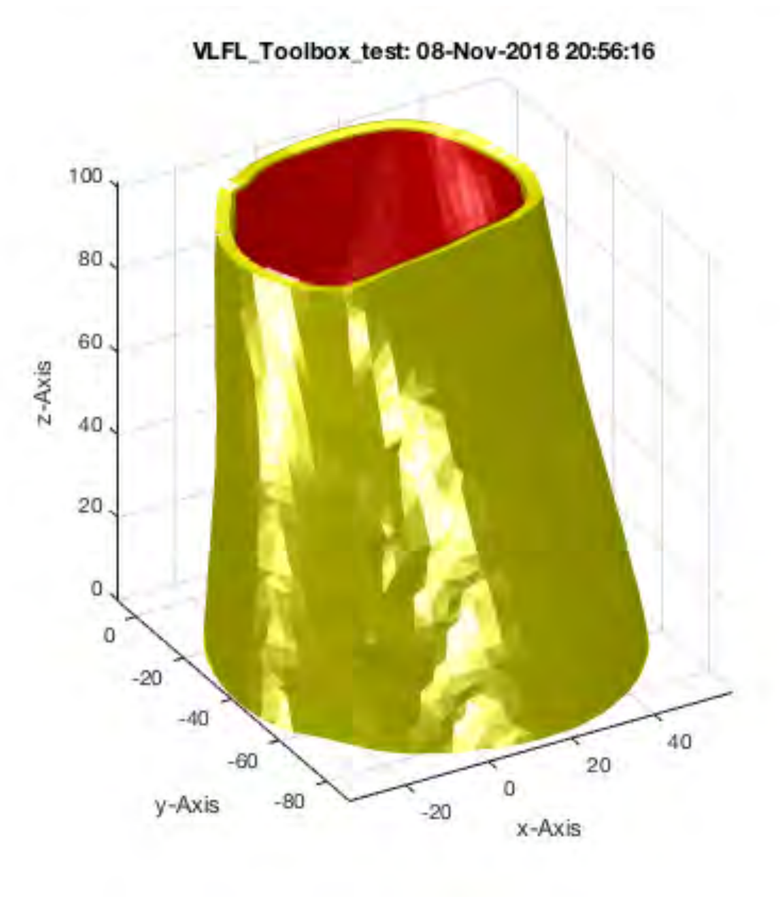


Create the surface of vectors along x axis

```
[FIL,k,FNL]=surfacesofSG(SGin);
[~,d]=VLnorm(FNL-[1 0 0]); dmin=min(d)
fi=find(d(d==dmin))
s=FIL(fi)
[S.VL,~,S.FL]=VLFLselect(SGin.VL,SGin.FL(FIL==s,:))
```

```
dmin =

    0.0382


fi =

     1


s =

     1


S =

  struct with fields:

    VL: [2385×3 double]
    FL: [4462×3 double]


S =

  struct with fields:

    VL: [2385×3 double]
    FL: [4462×3 double]
```

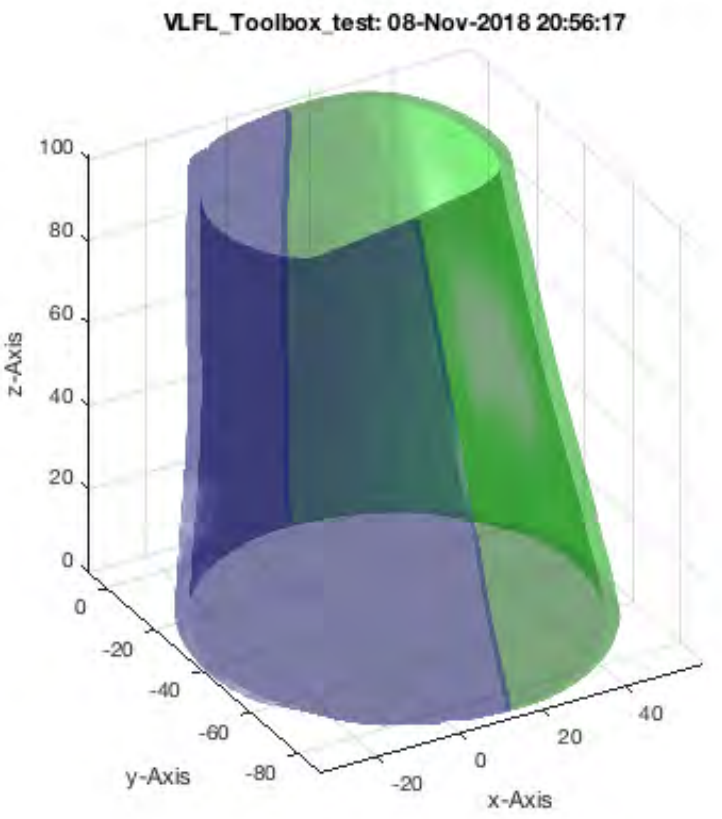
```
SGshell=SGofSurface(S.VL,S.FL,3,1);
SGofSurface(S.VL,S.FL,3,1); view(-30,30); VLFLplotlight(1,1);
```


VLFL_Toolbox_test: 08-Nov-2018 20:56:16
z-Axis
y-Axis
x-Axis
100
80
60
40
20
0
0
-20
-40
-60
-80
-20
0
20
40

Cut the

```
SGpuzzlecut3D(SGshell,[0.5 1 1]); VLFLplotlight(1,0.3);
SGshell=SGpuzzlecut3D(SGshell,[0.5 1 1]); VLFLplotlight(1,0.3);
```

```
50% 100%
50% 100%
```




```
SGwriteMultipleSTL(SGshell);
```

```
SGwritemultipleSTL: Writing 2 STL files in /Users/lueth/Desktop/Toolbox_test/EXP-2018-11-08
/
```

Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:56:18!
Executed 08-Nov-2018 20:56:20 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
database_toolbox
image_toolbox
map_toolbox
matlab
```

```
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
==============================================================================================
==========
```

Published with MATLAB® R2018a

Tutorial 37: Dimensioning of STL Files and Surface Data

2017-07-24: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de) - Last Change: 2017-07-25

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

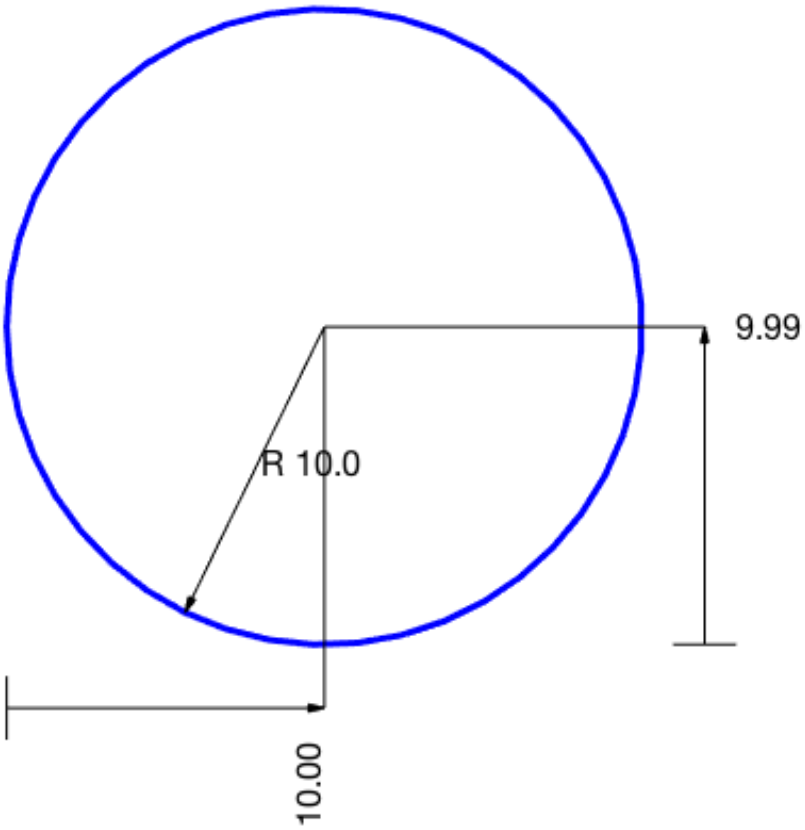
Motivation for this tutorial: (Originally SolidGeometry 4.0 required)

If you use an STL file from the third page, then certain dimensions have been used which have an influence on a constructions. In this tutorial you will find the function to analyze STL files and to draw technical drawings for these surfaces.

1. Basic functions for dimensioning

PLdimensioning creates complete drawings for a single Point list. it should be used as a drawing function

```
PL=PLcircle(10);
PLdimensioning(PL);
```

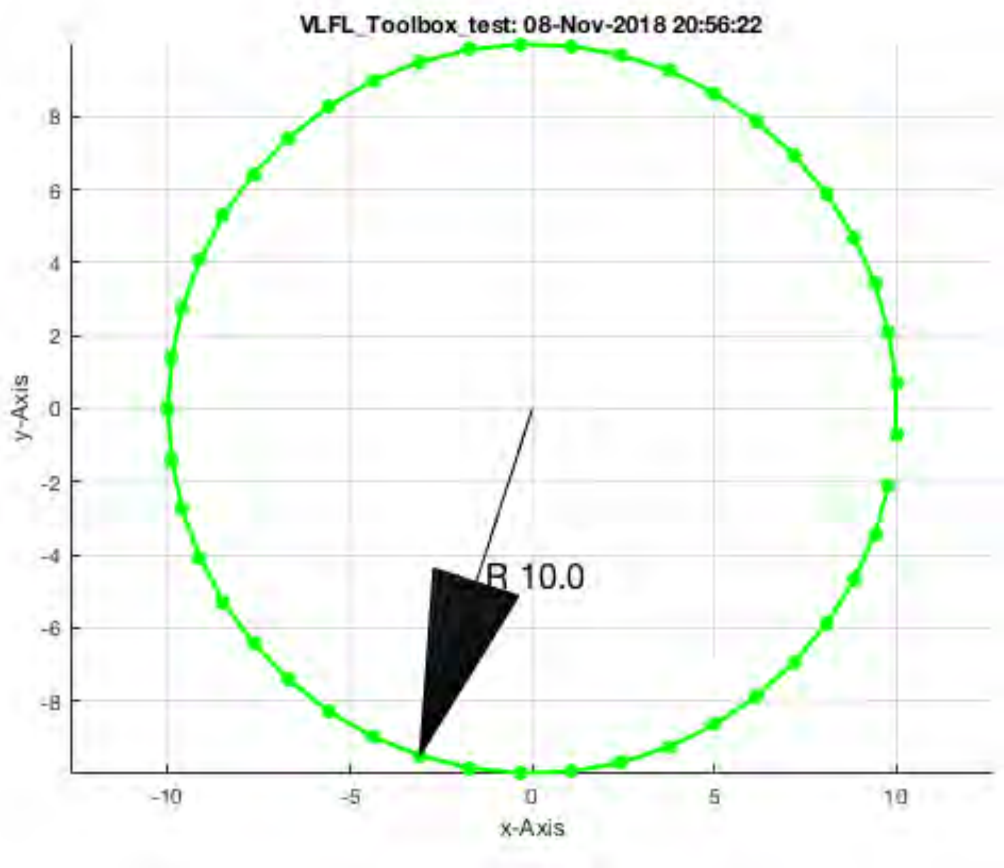


CVLdimclassifier is an auxiliary function that creates results for PLdimensioning It should be used for calculating features and supports CVL in 3D too. It returns points that are not part of a circe/ellipse and a list of circles and a list which vertices belong to circles and the normal vectors for the circles

```
CVLdimclassifier(PL)
[DVL,RL,RIL,Rnv]=CVLdimclassifier(PL)
```

```
ans =

  0×3 empty double matrix


DVL =

  0×3 empty double matrix


RL =

  Columns 1 through 7

         0         0         0   45.0000  360.0000   10.0000    2.4192

  Columns 8 through 9

    9.7030         0
```

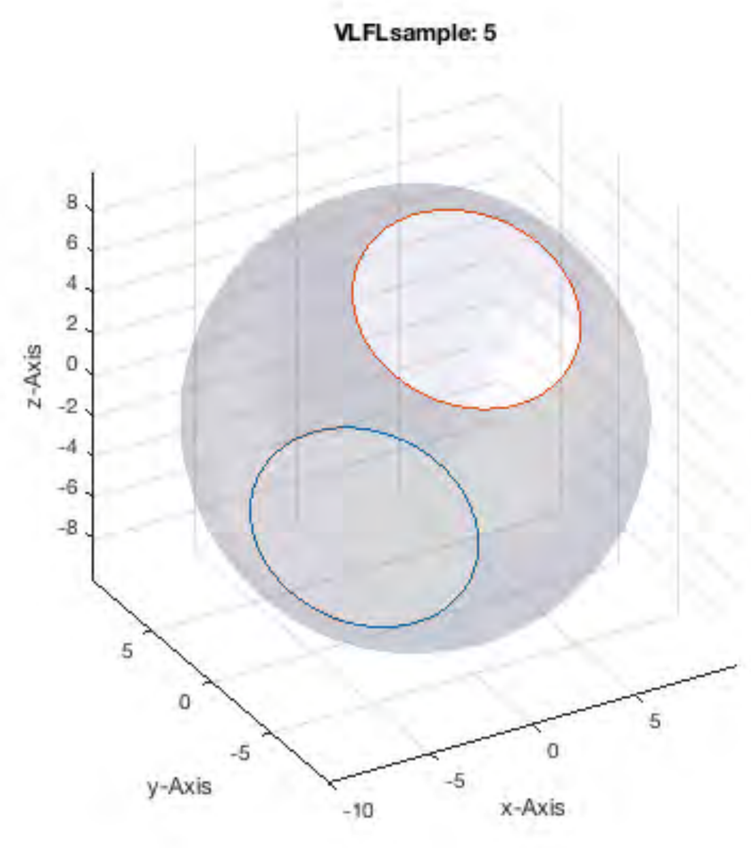
```
RIL =

     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1


Rnv =

     0     0    -1
```



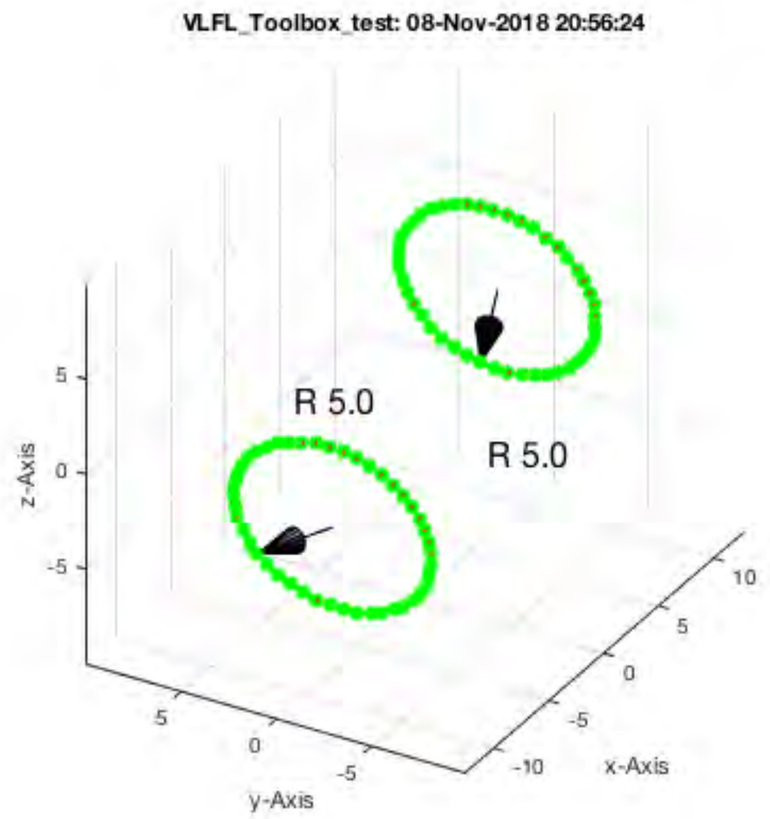
2. Classifying 3D Contours

```
% *CVLdimclassifier* is ablt to classify several indepenent contours in
3D space as CVL.
```

```
VLFLsample(5); VLFLplotlight (1,0.2);
[~,~,~,CVL]=VLFLsample(5);
CVLplot(CVL,'-');
```




```
CVLdimclassifier(CVL); view(-60,30);
```

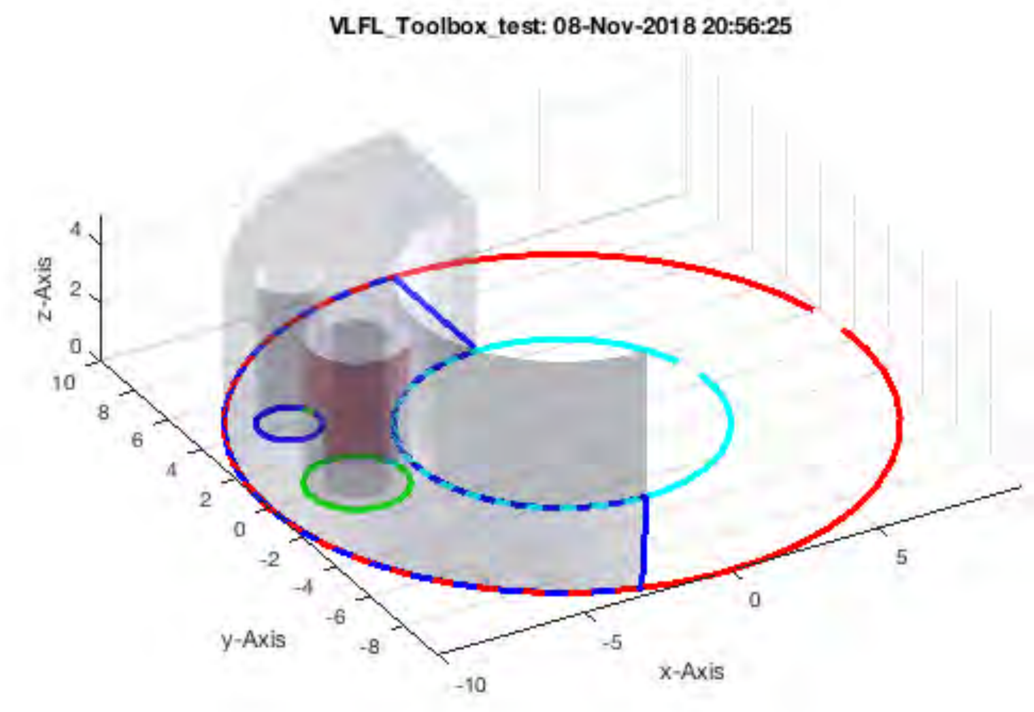


3. Finding Surfaces and Contours for Dimensioning

There are not several functions that help to define surfaces for dimensioning similar to featureedges we see feature surfaces to separate surfaces if a solid

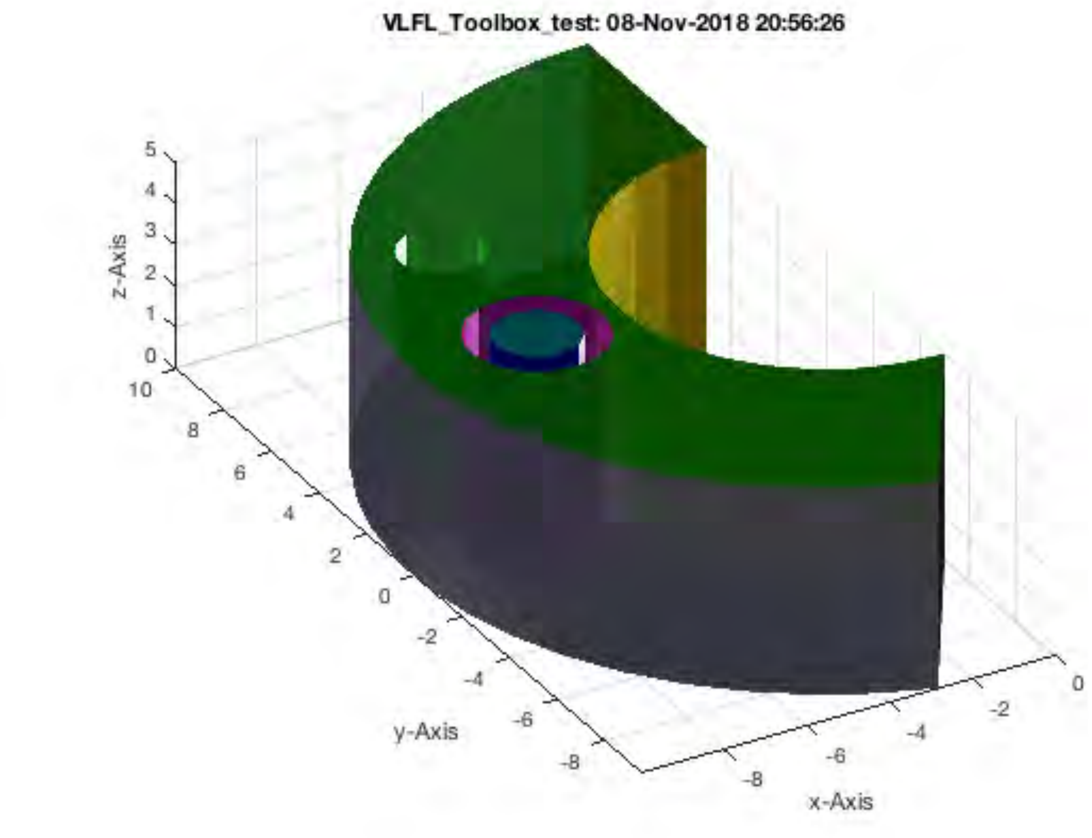
- **TR3mountingfaces** - finds connected surfaces starting from one or more faces
- **MLofSG** is a simliar function for a complete solid
- **surfacesofSG** - generates features surfaces of ONE closed solid
- **TR3neighborsAngle** and **neighborsAngleSurface** - find feature surfaces
- **FSofSG** supports also cells of solids

```
TR3mountingfaces(SGsample(25),1); % facets, normals, neigbors, radial list, CVL of ONE sur
face
```

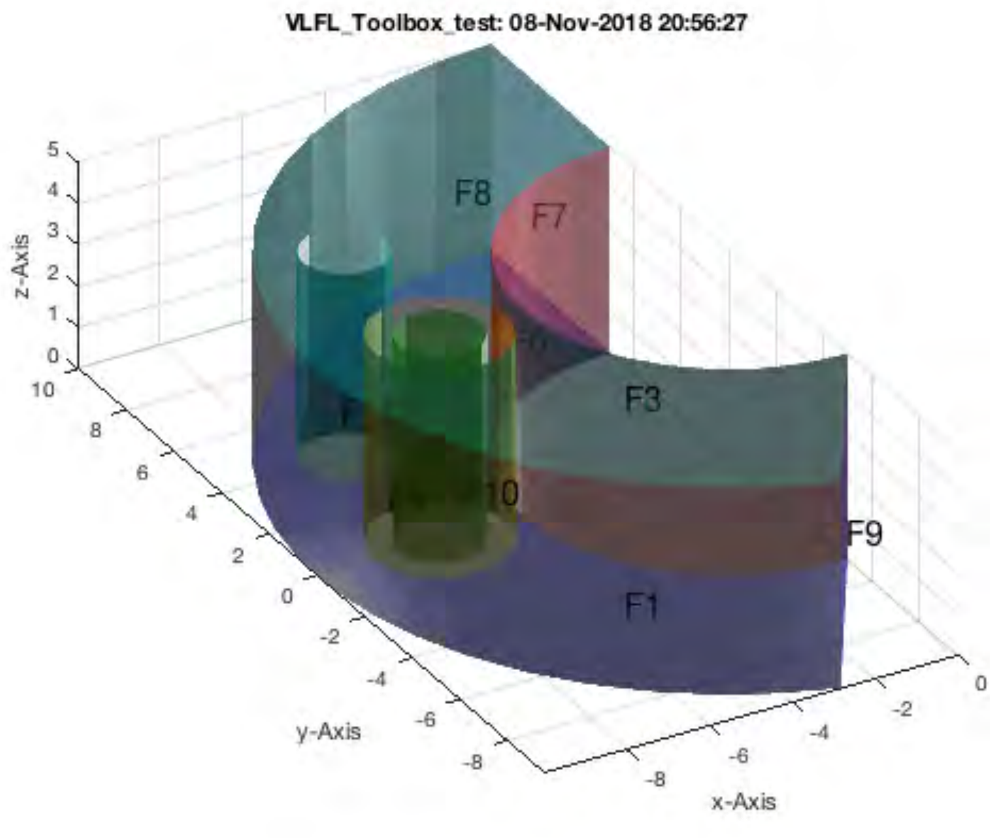



```
surfacesofSG(SGsample(25)); % facet index, normals, angles, neigbors, area
```

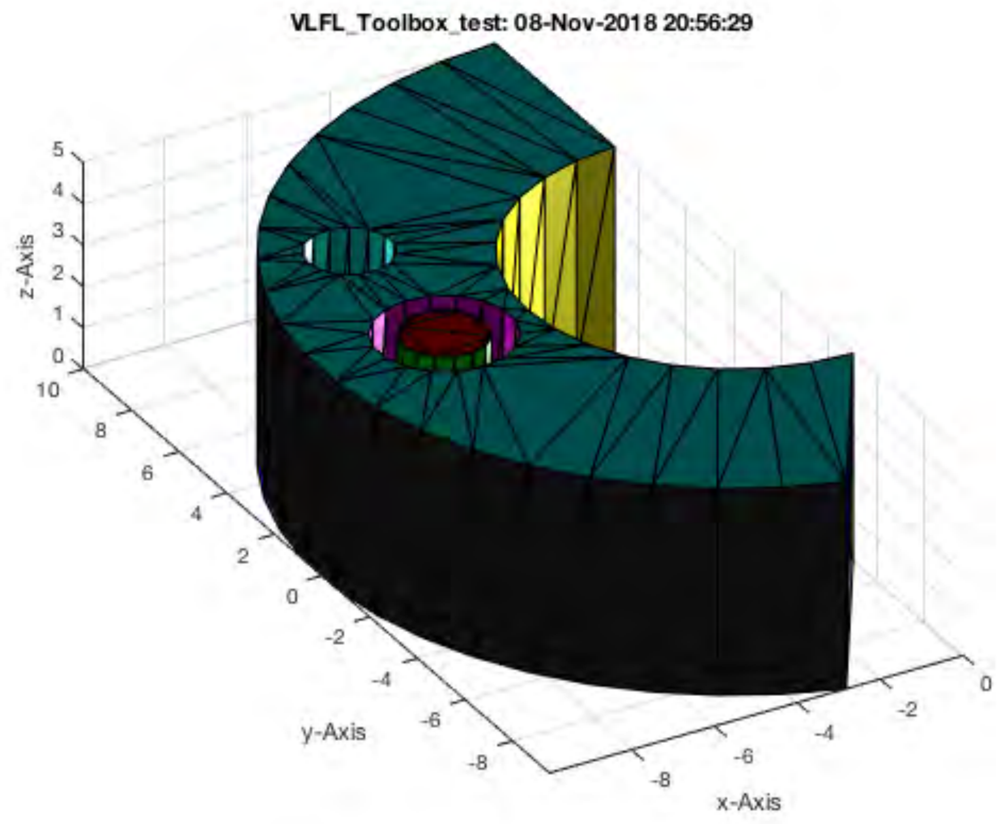
```
11 Feature Surfaces found! Only the largest 99% (0..143mm^2), i.e. 11 of 11 are shown.
```




```
FSofSG(SGsample(25)); % surface index list
```




```
MLofSG(SGsample(25));
```

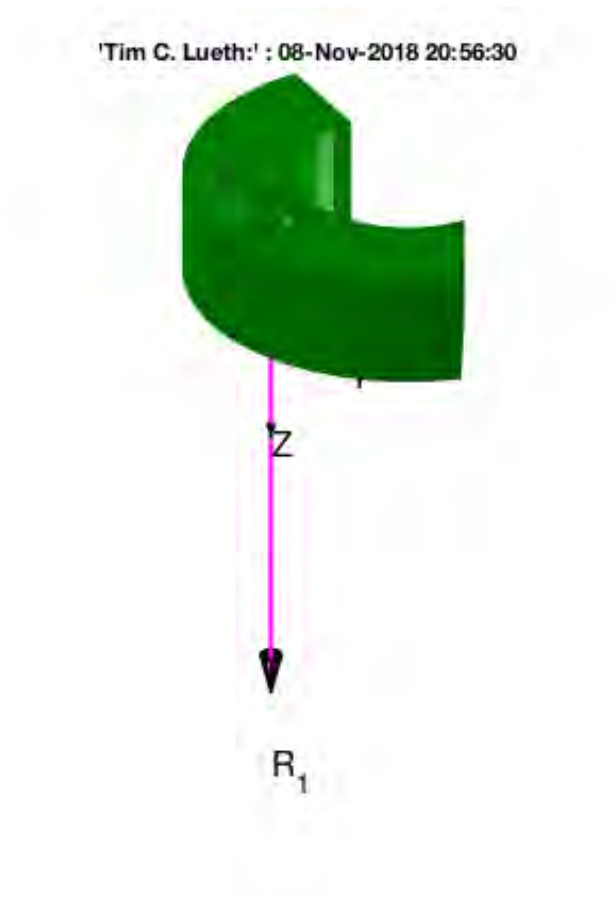


4. Interactive specifiying faces and coordinate systems

We already used ina nearlier tutorial(VLFL_EXP11) the function SGTui to specify coordinate system for planar surfaces or edges. By using the third parameter if SGTui, it is possible to make a feature surface search. A common value is 1 rad ~ 60 degree. The function is able to detect radial structures using CVLdimclassifier and allow to address other coordinate systems too.

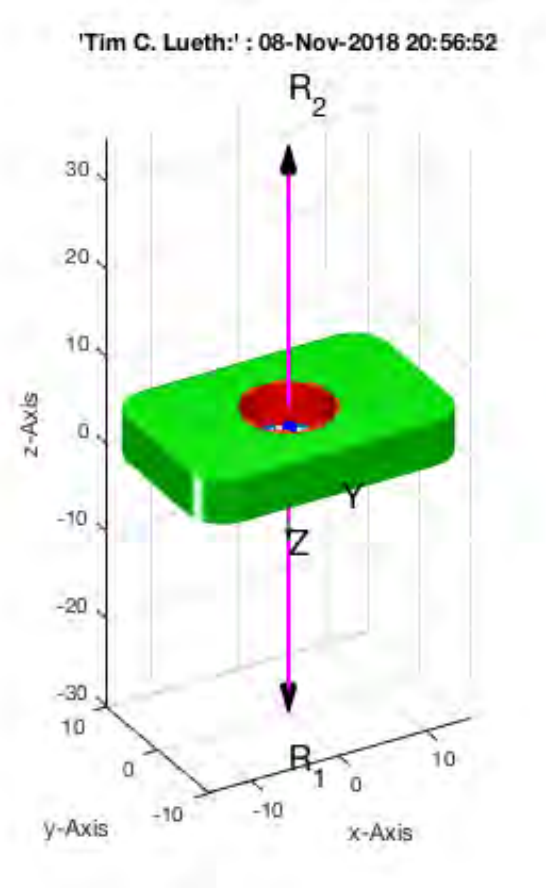
```
SGTui(SGsample(25),'Frame',1)
```

```
ans =

  struct with fields:

       VL: [174×3 double]
       FL: [348×3 double]
    Tname: {'Frame'}
        T: {[4×4 double]}
     TFiL: {[13×1 double]}
     TFoL: {[]}
```

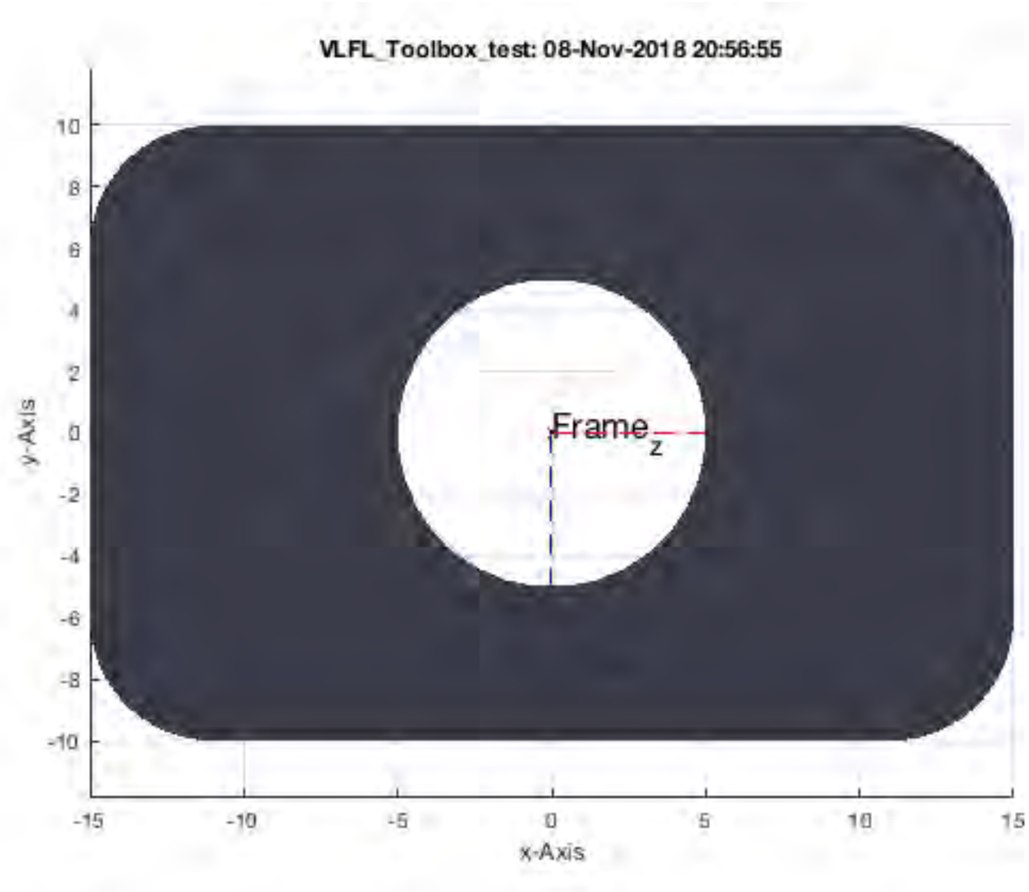



```
SG=SGTui(SGsample(27),'Frame',1,'R1')
```

```
SG =

  struct with fields:

       VL: [200×3 double]
       FL: [400×3 double]
    Tname: {'Frame'}
        T: {[4×4 double]}
     TFiL: {[64×1 double]}
     TFoL: {[]}
```

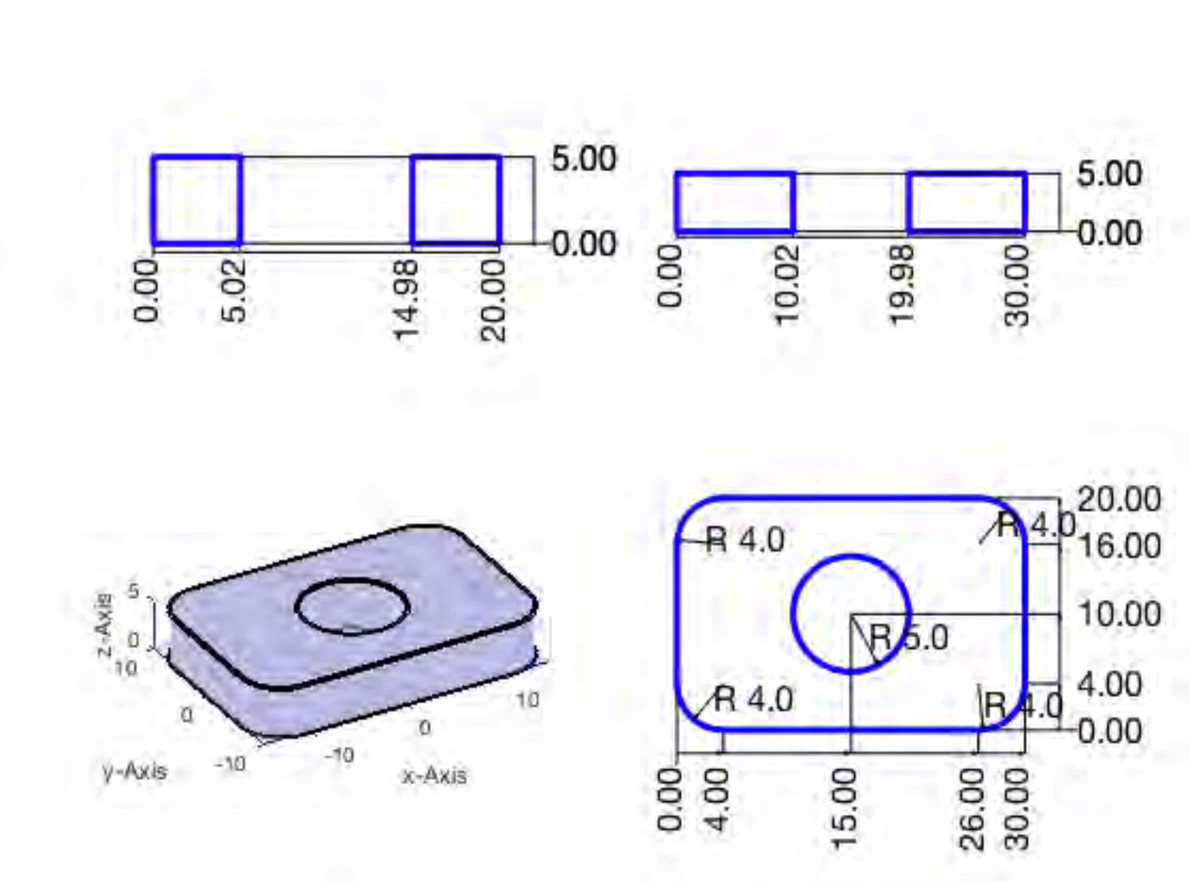



```
SGfigure; SGTplot(SG);
```



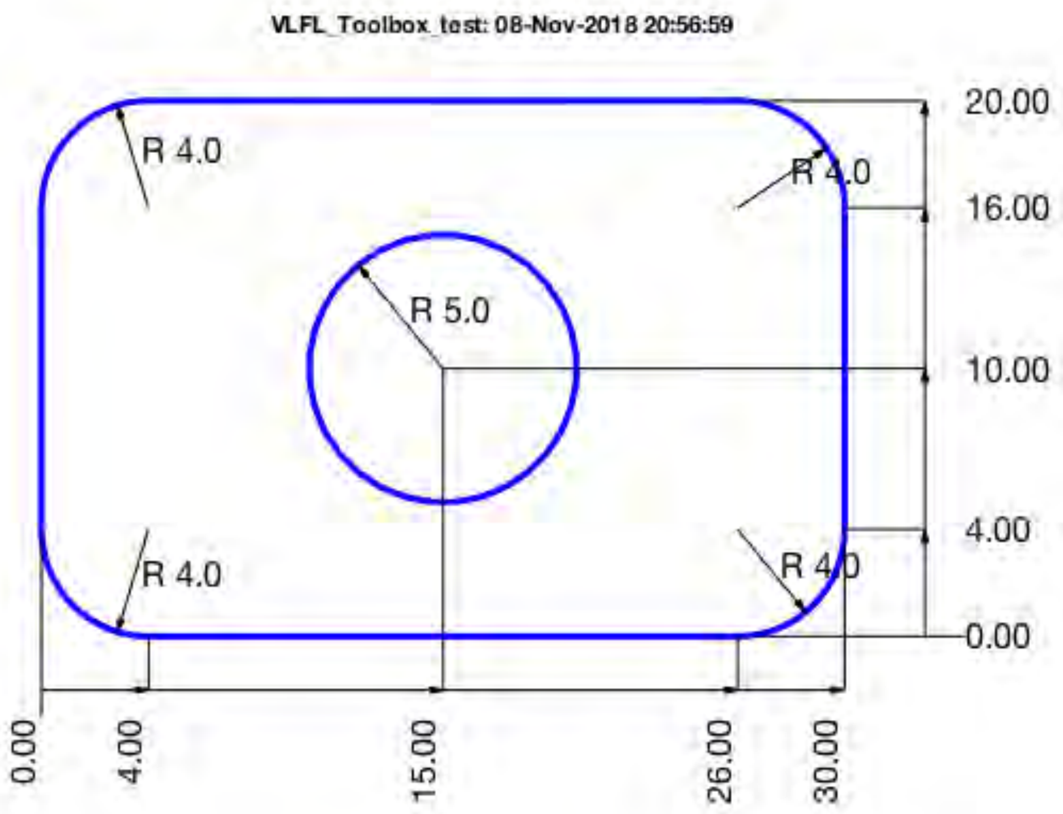
5. Dimensioning of border of surfaces: SGdimensioning

```
SGTdimensioning(SG,'Frame');
```

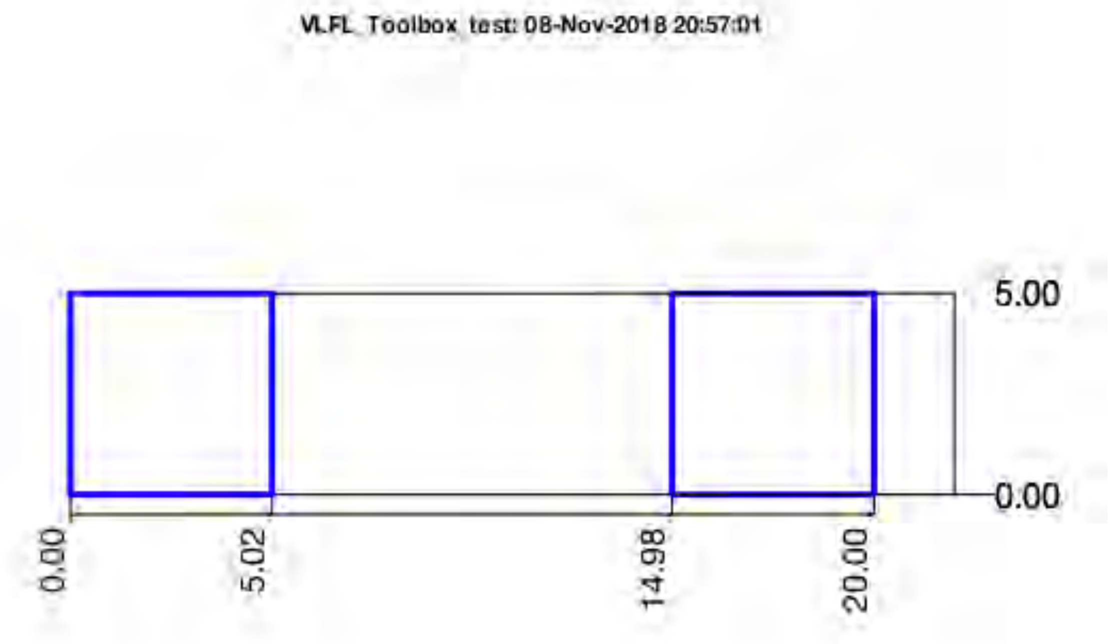


6. Creating of standard dimensioning using view angles: SGdimensioning

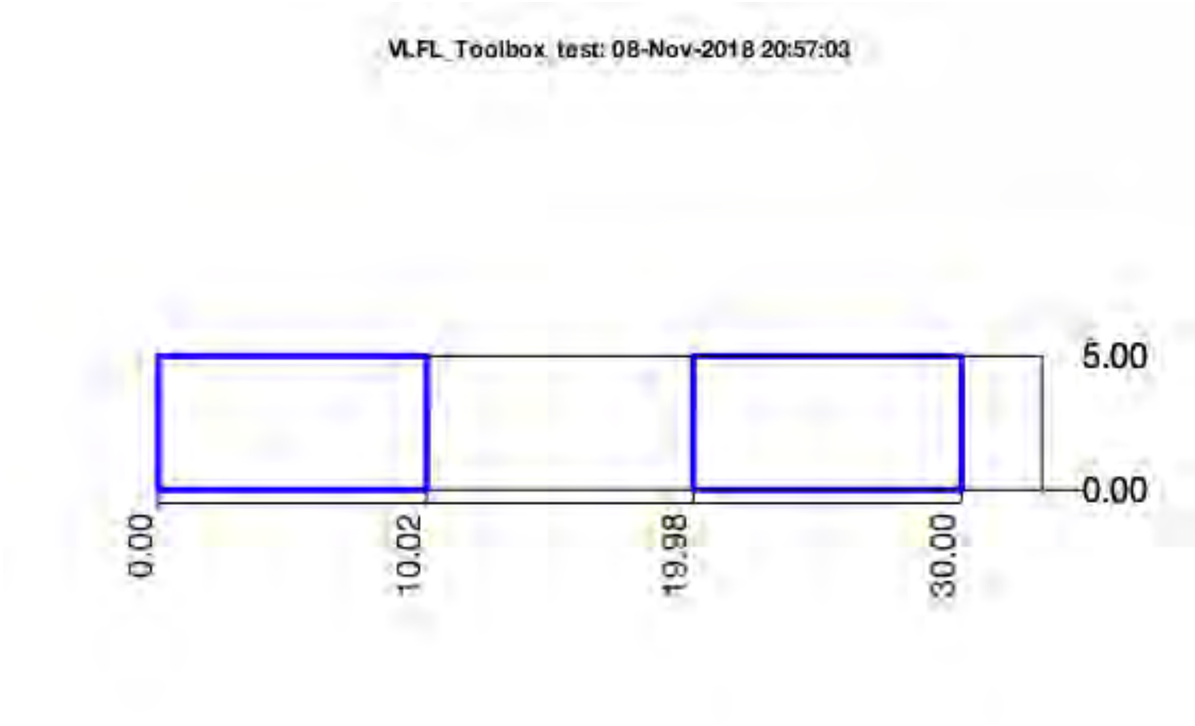
```
SGdimensioning(SG,0,90);
```




```
SGdimensioning(SG,90,0);
```

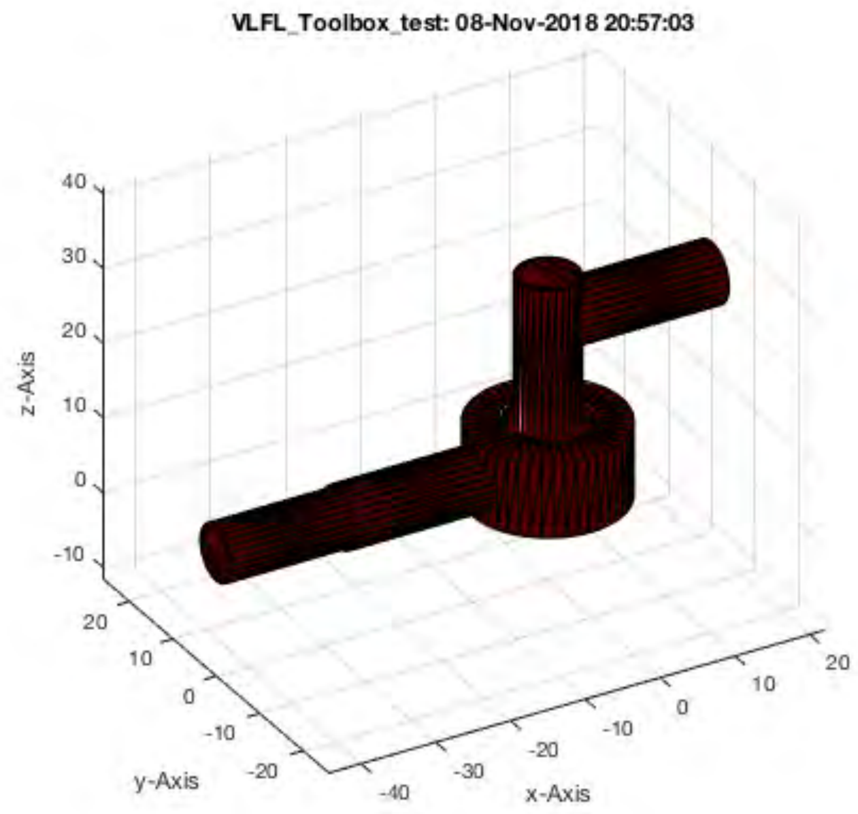



```
SGdimensioning(SG,0,0);
```

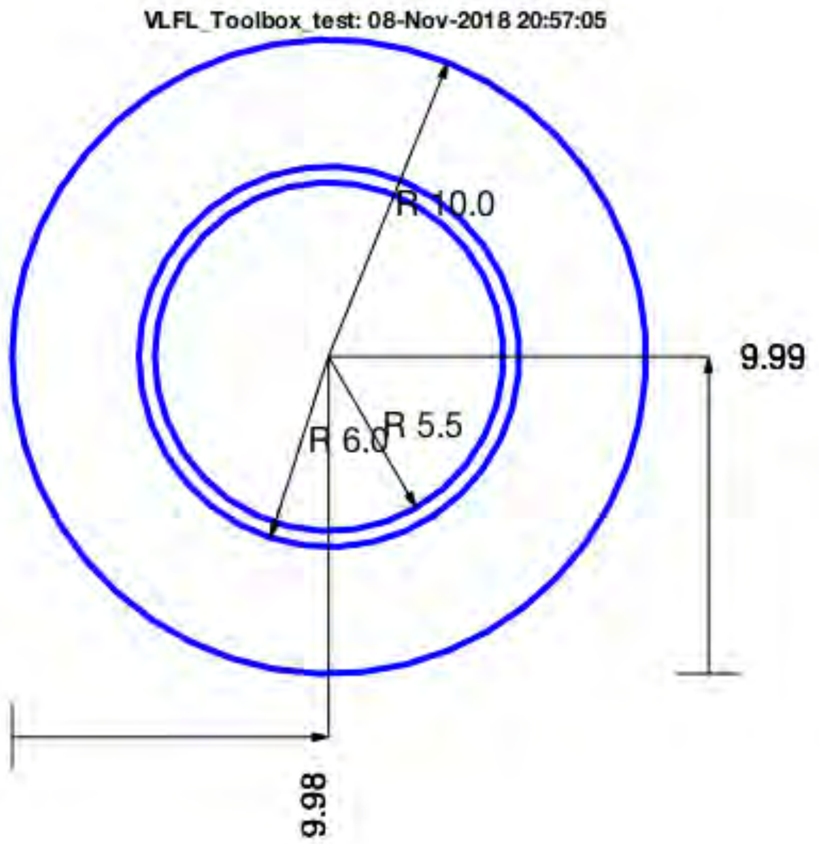


7. Creating of standard dimensioning using view angles and cross cuts

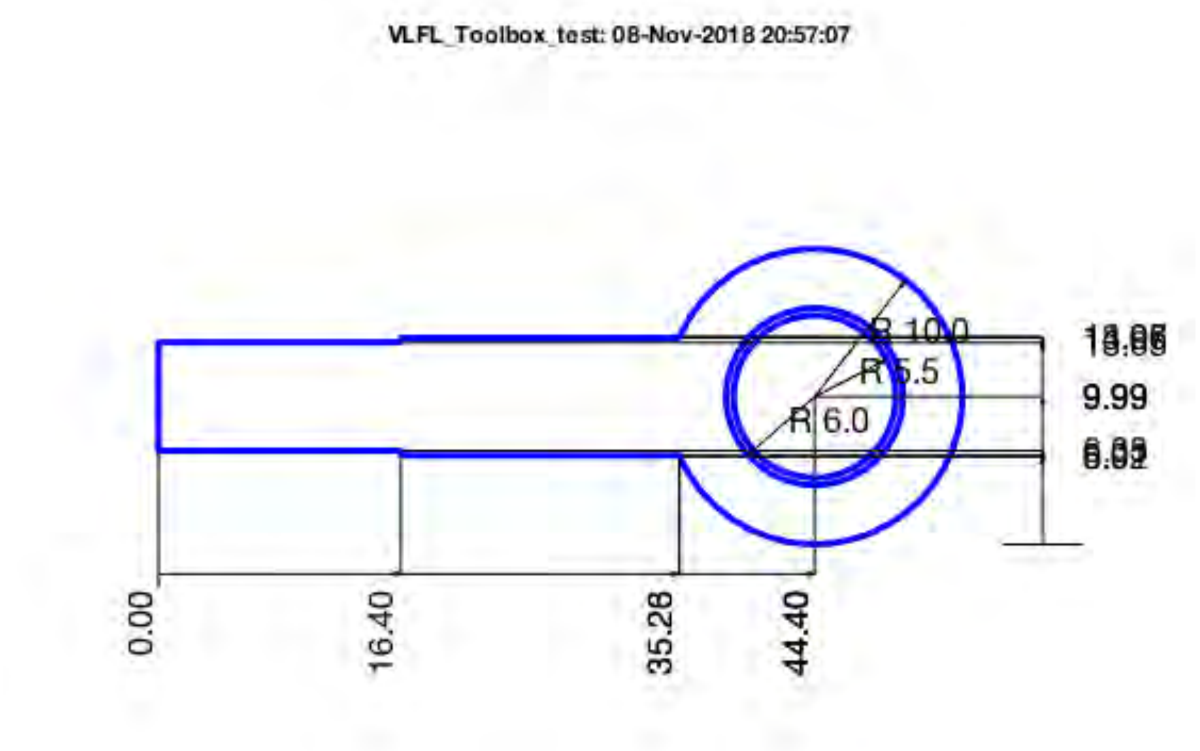
```
SG=SGsample(17); SGfigure; SGplot(SG); view(-30,30);
```



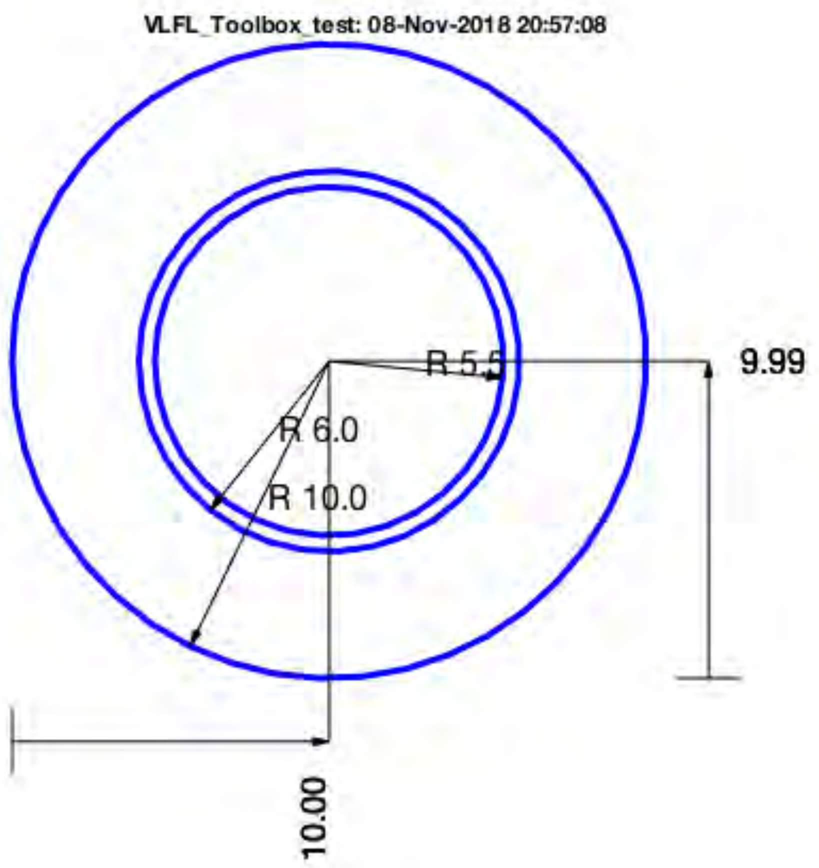

```
SGdimensioning(SG,0,90);
```



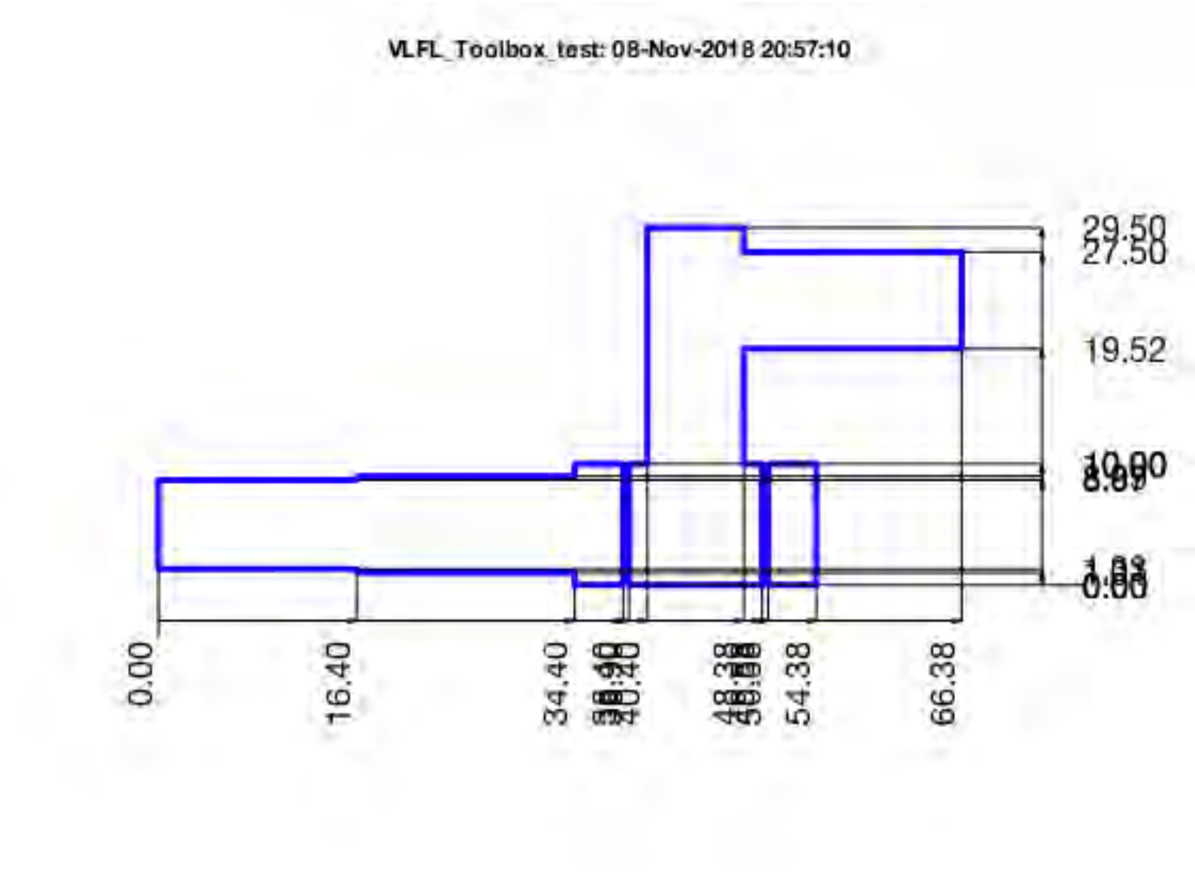

```
SGdimensioning(SG,0,90,[0 0 5]);
```




```
SGdimensioning(SG,0,90,[0 0 +10]);
```



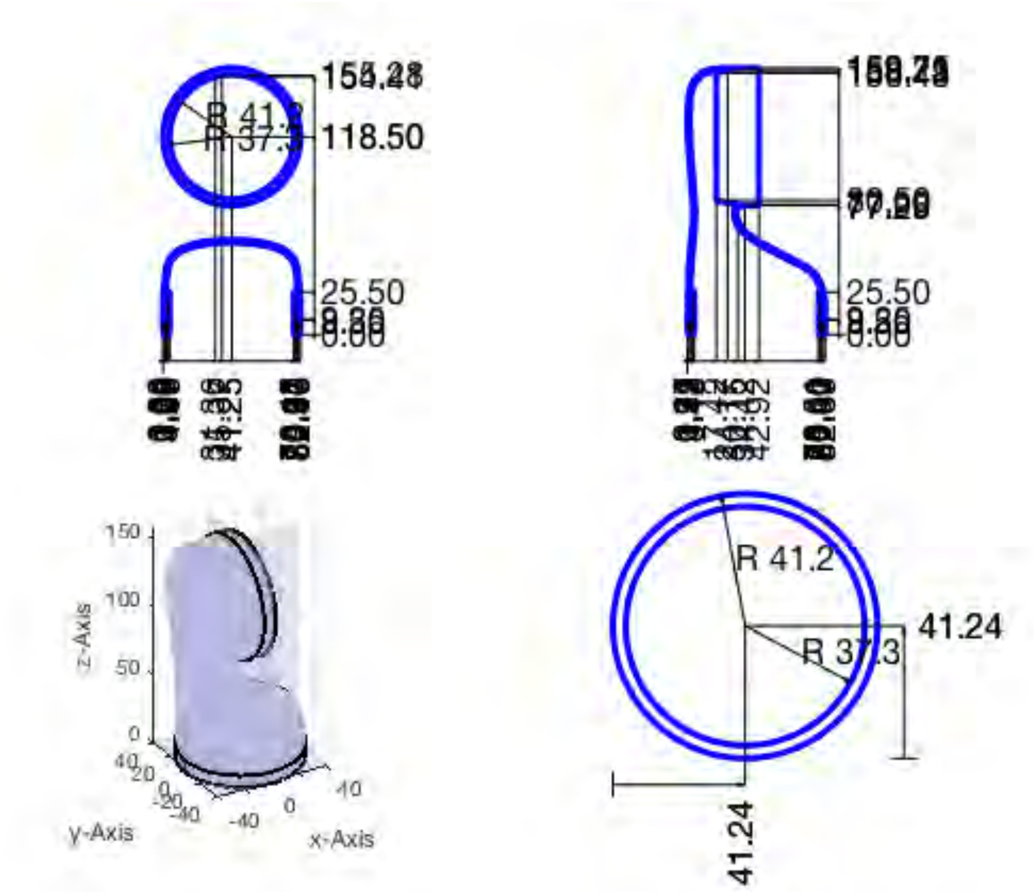

```
SGdimensioning(SG,0,0,[0 0 0]);
```



8. Using frames for dimensioning

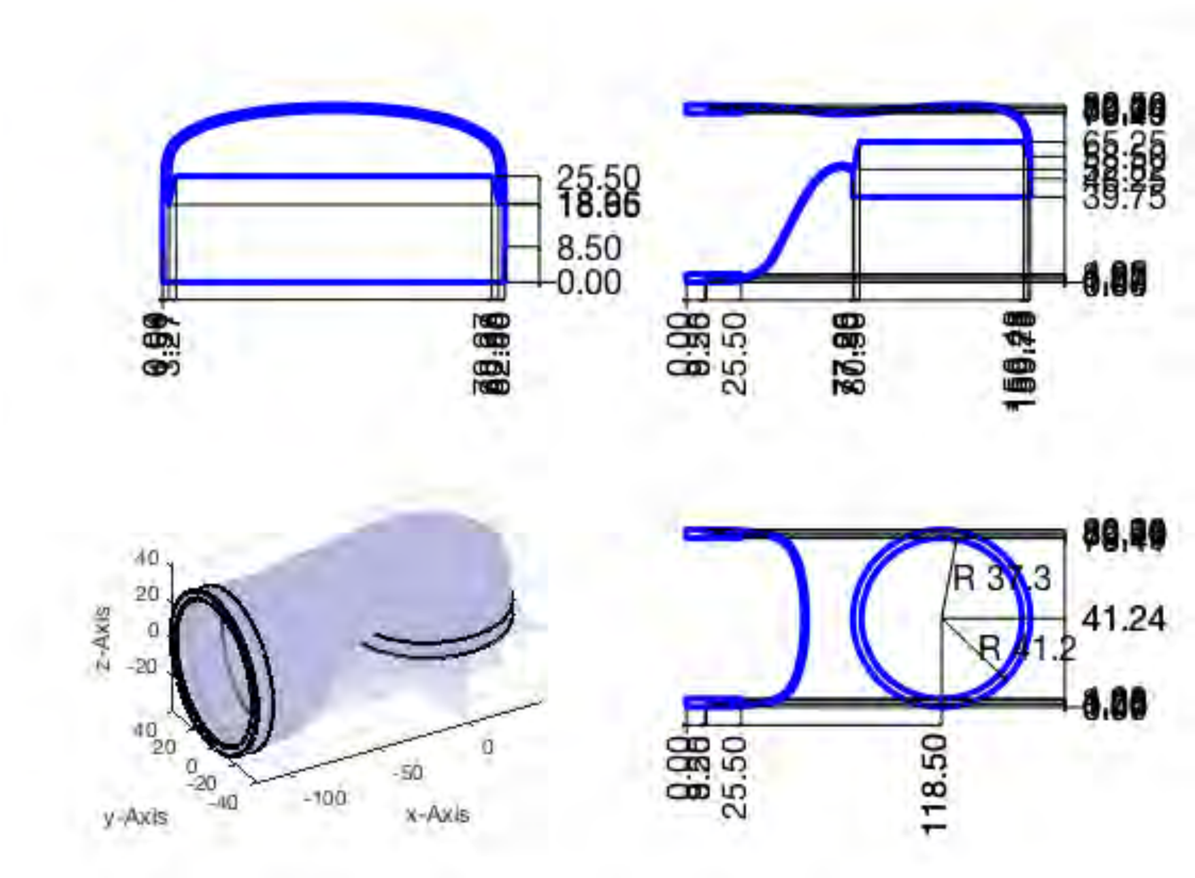
```
load JACO_robot.mat
SGTdimensioning(JC1,'B');
```

```
Warning: The triangulation is empty - the points may be collinear.
Warning: The triangulation is empty - the points may be collinear.
Warning: The triangulation is empty - the points may be collinear.
```




```
SGTdimensioning(JC1,'F');
```

```
Warning: The triangulation is empty - the points may be collinear.
Warning: The triangulation is empty - the points may be collinear.
Warning: The triangulation is empty - the points may be collinear.
Warning: The triangulation is empty - the points may be collinear.
Warning: The triangulation is empty - the points may be collinear.
Warning: The triangulation is empty - the points may be collinear.
```



Published with MATLAB® R2018a

Tutorial 38: Some more solid geometry modelling function

2017-07-24: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de) - Last Change: 2017-07-25

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

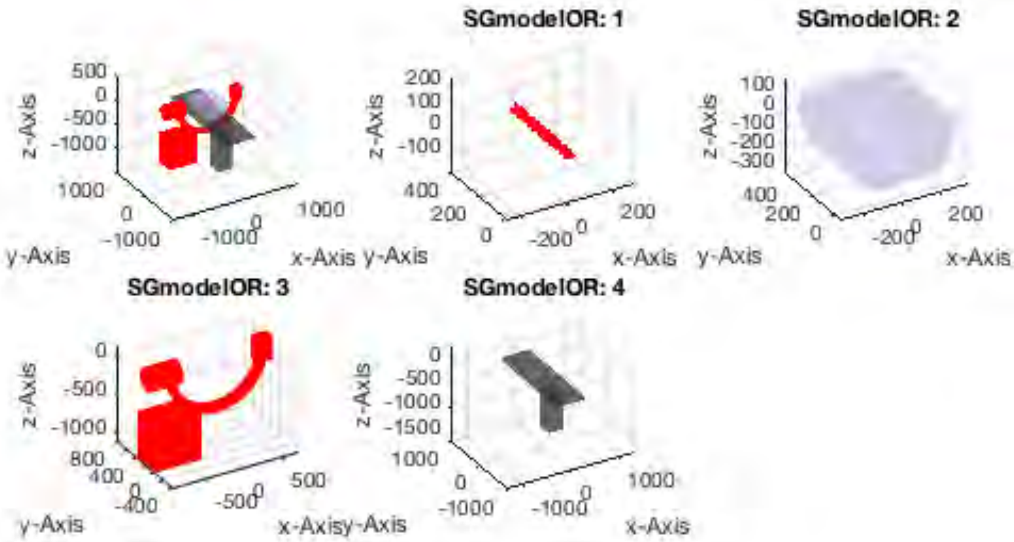
- Tutorial 18: Analyzing and modifying STL files from CSG modeler (Catia)
- Tutorial 19: Creating drawing templates and dimensioning from polygon lines
- Tutorial 20: Programmatically Interface to SimMechanics Multi-Body Toolbox
- Tutorial 21: Programmatically Convert Joints into Drives (SimMechanics)
- 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

Motivation for this tutorial: (Originally SolidGeometry 4.0 required)

function VLFL_EXP38

1. Some elements of medical equipemnt in the operating room

```
SGmodelOR;
```

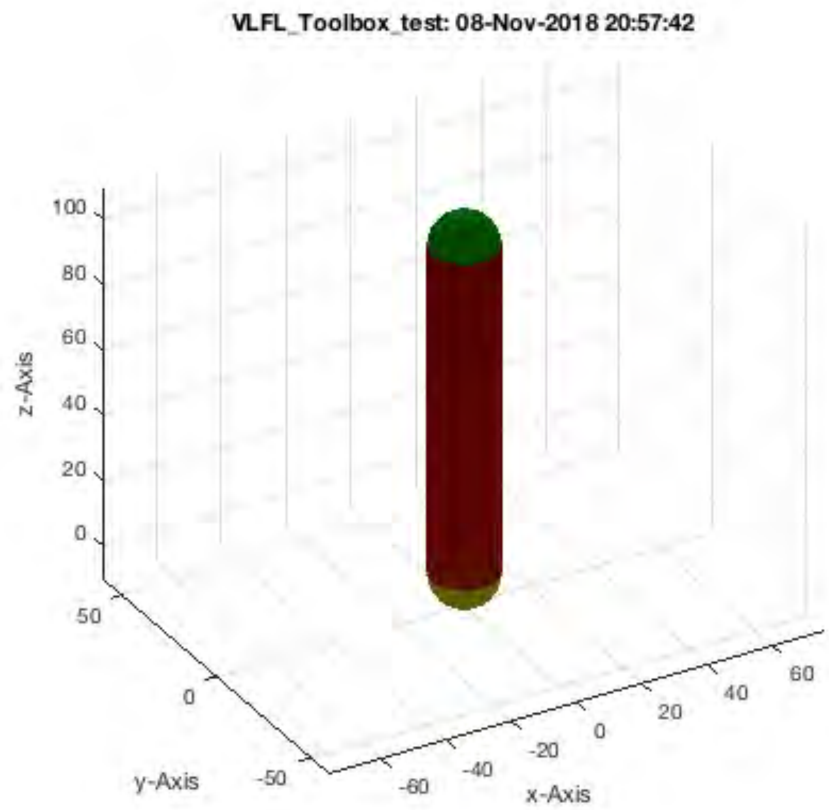


or select a c-arm device

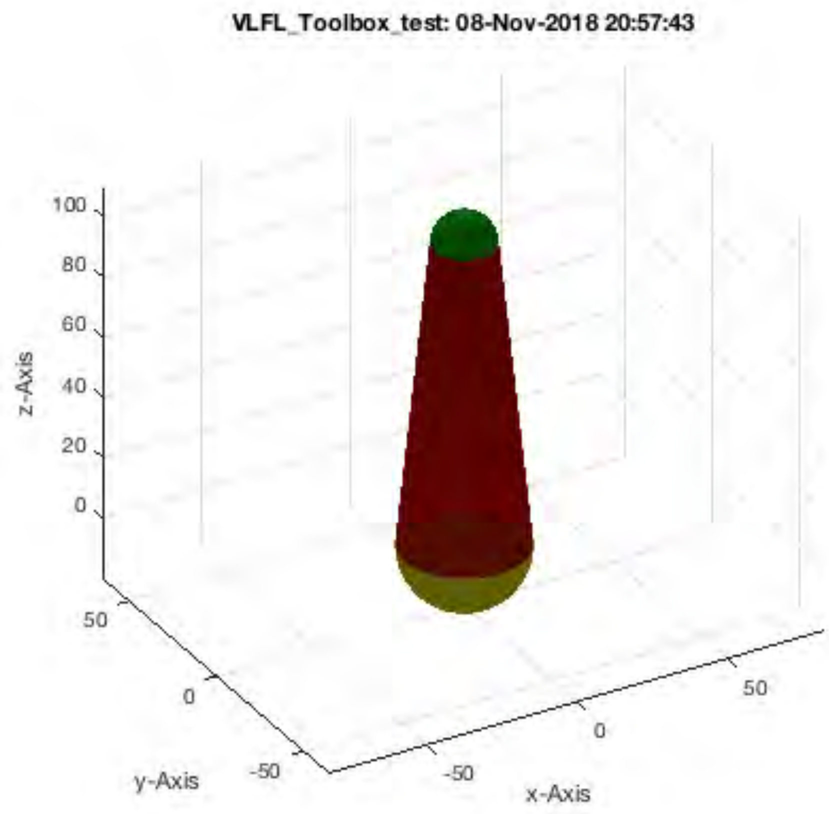
SGmodelOR(3)

2. Creating solids as links with spheres at the end

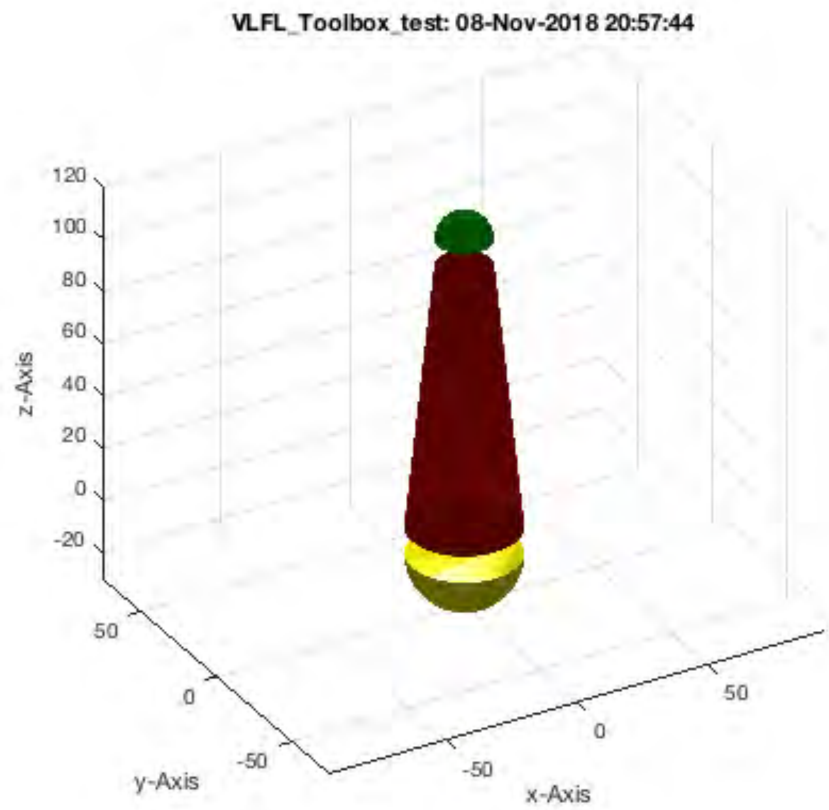
```
SGspherelink (100,10);
```



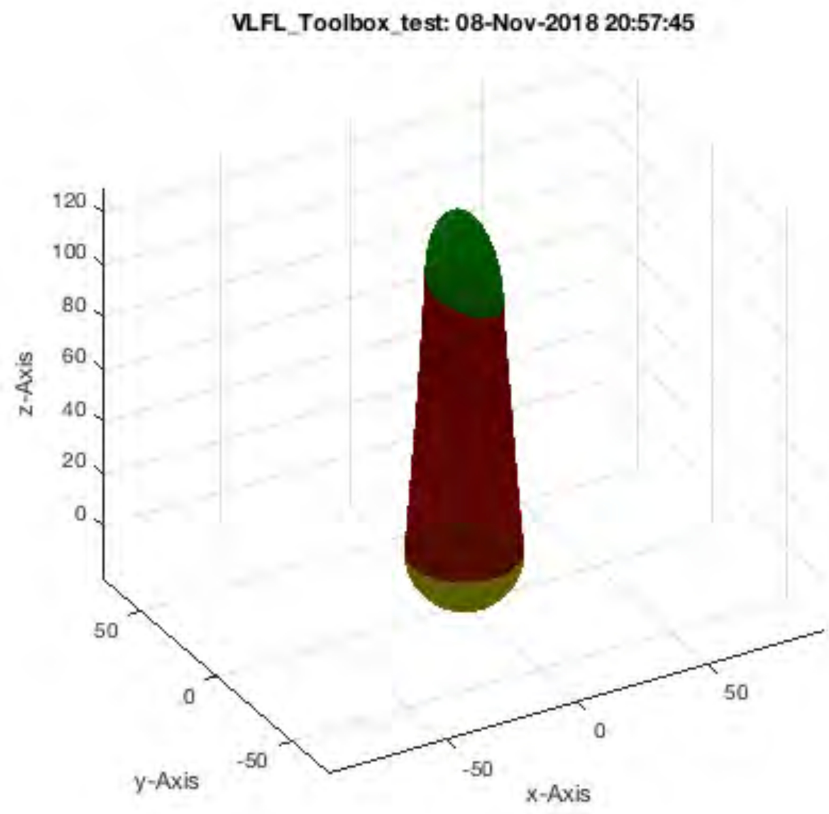

```
SGspherelink (100,10,20);
```



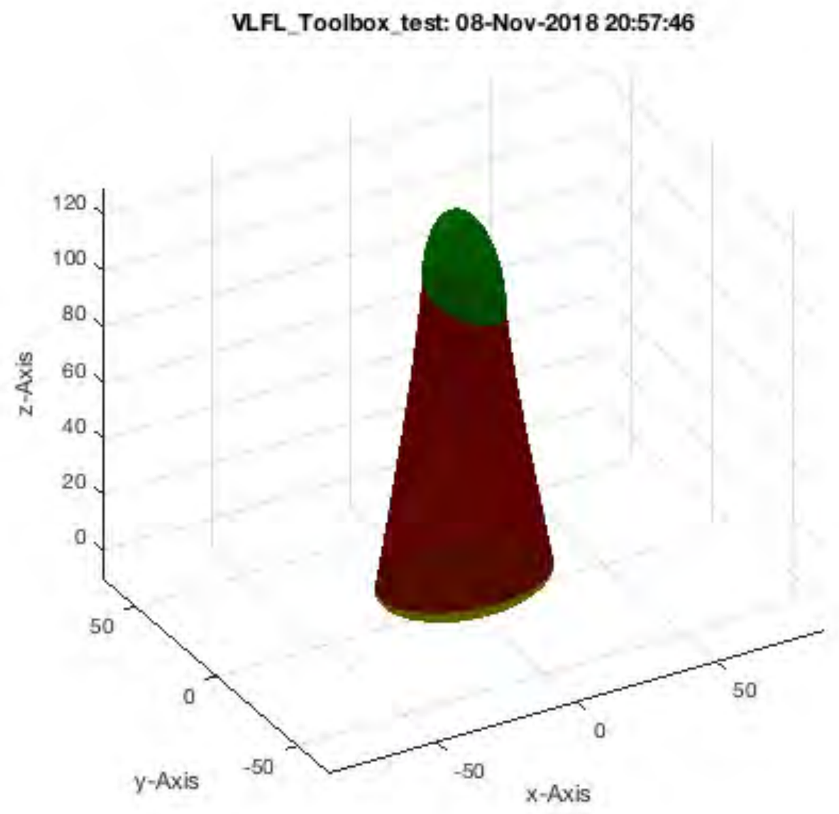

```
SGspherelink (100,10,20,-10);
```




```
SGspherelink (100,[10,20,30],20);
```

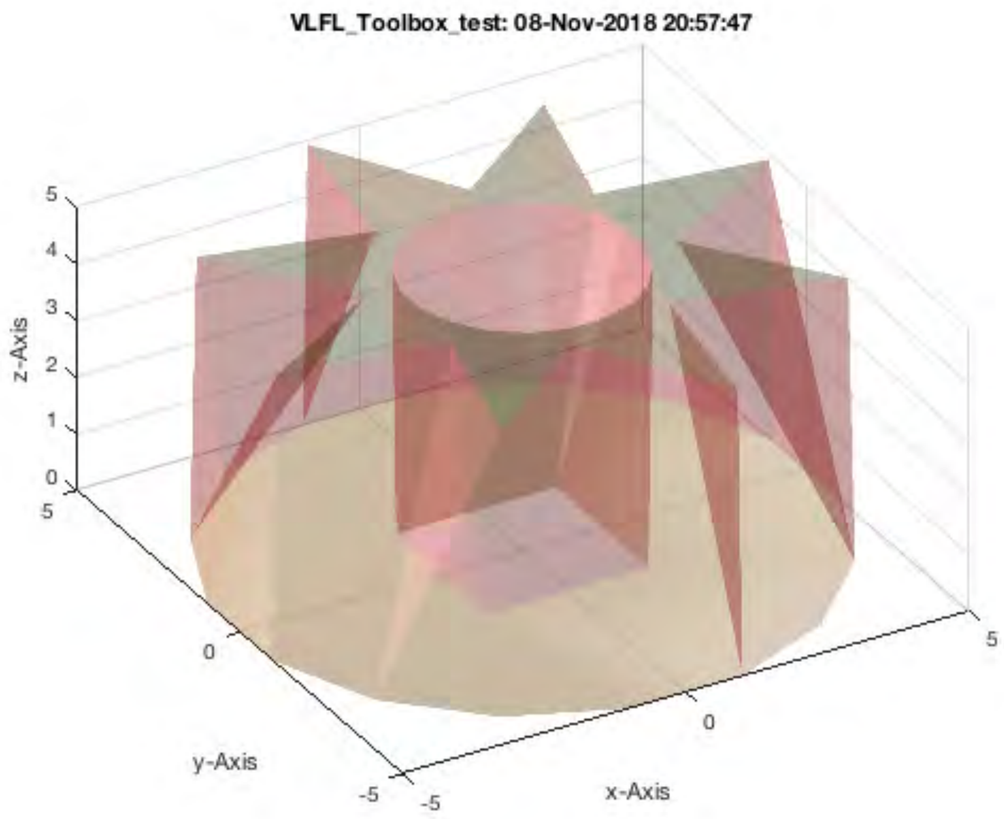



```
SGspherelink (100,[10,20,30],[30,20,10]);
```



3. Creating Solids by connecting two CPLs with enclosed contours

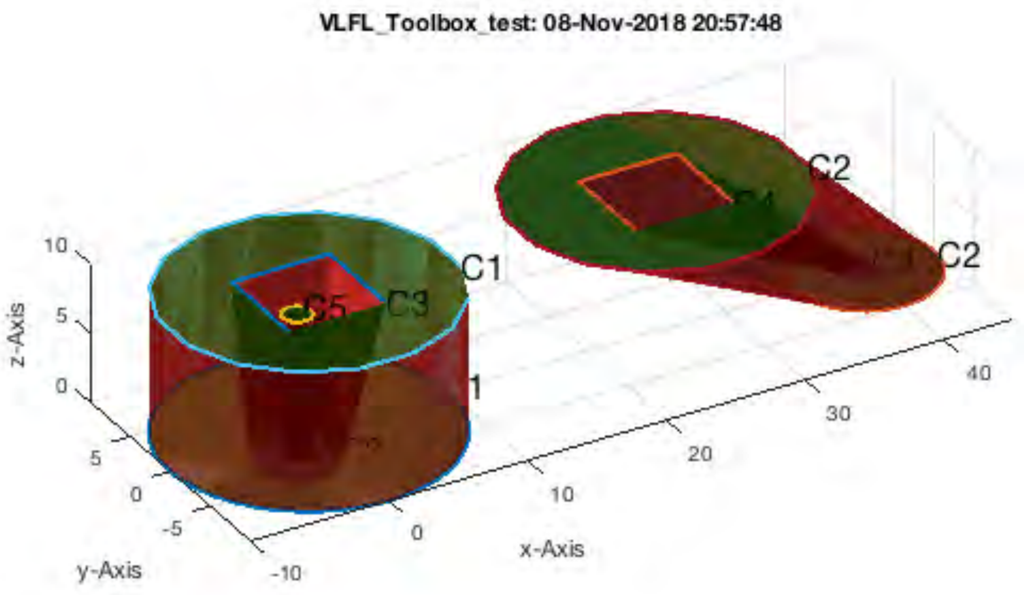
```
CPA=[PLcircle(5.1,16);NaN NaN; PLcircle(2,4)];
CPB=[PLstar(5,16,[],[],[],0.5);NaN NaN; PLcircle(2)];
SGof2CPLsz(CPA,CPB,5); VLFLplotlight(1,0.2);
```



4. Creating Solids by connecting two planar CPLS of different strucure

```
SGof2CPLzheurist(CPLsample(26),CPLsample(27),10)
```

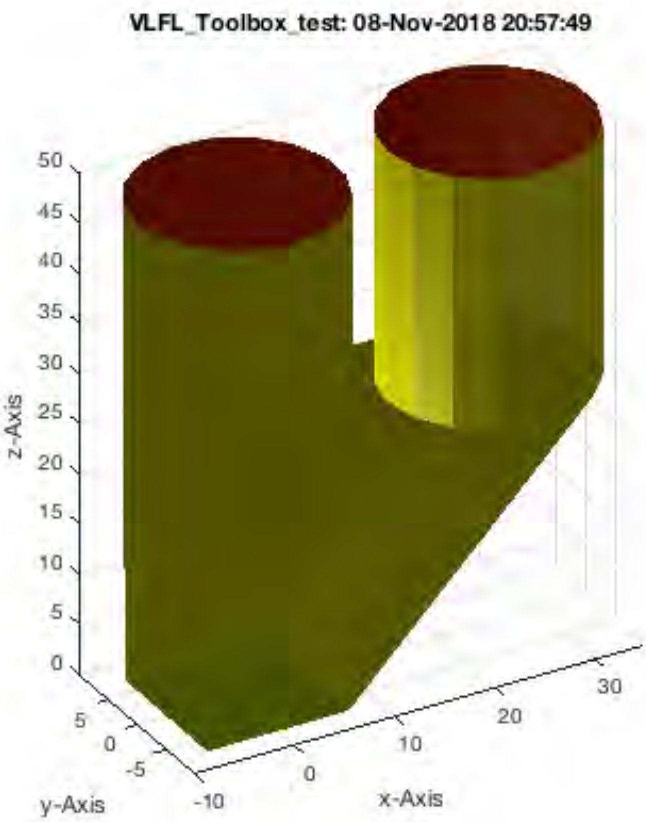
```
ans =

  struct with fields:

        VL: [187×3 double]
        FL: [370×3 double]
       col: 'w'
     alpha: 0.9000
```



5. Creating branches between two contour

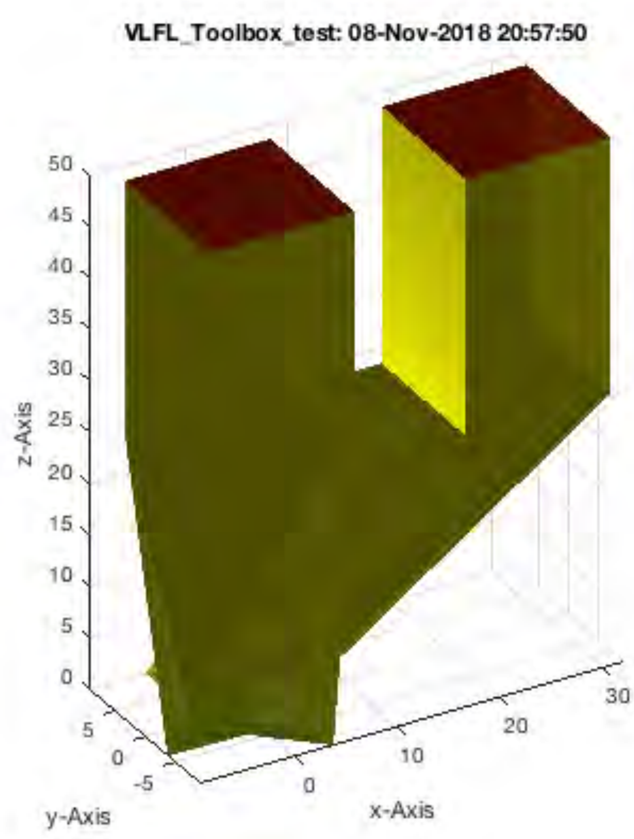
```
SGof2CPLzbranch(CPLsample(2), CPLsample(9),50)
```

```
ans =

  struct with fields:

    VL: [72×3 double]
    FL: [140×3 double]
```




```
SGof2CPLzbranch(CPLsample(6), CPLsample(10),50)
```

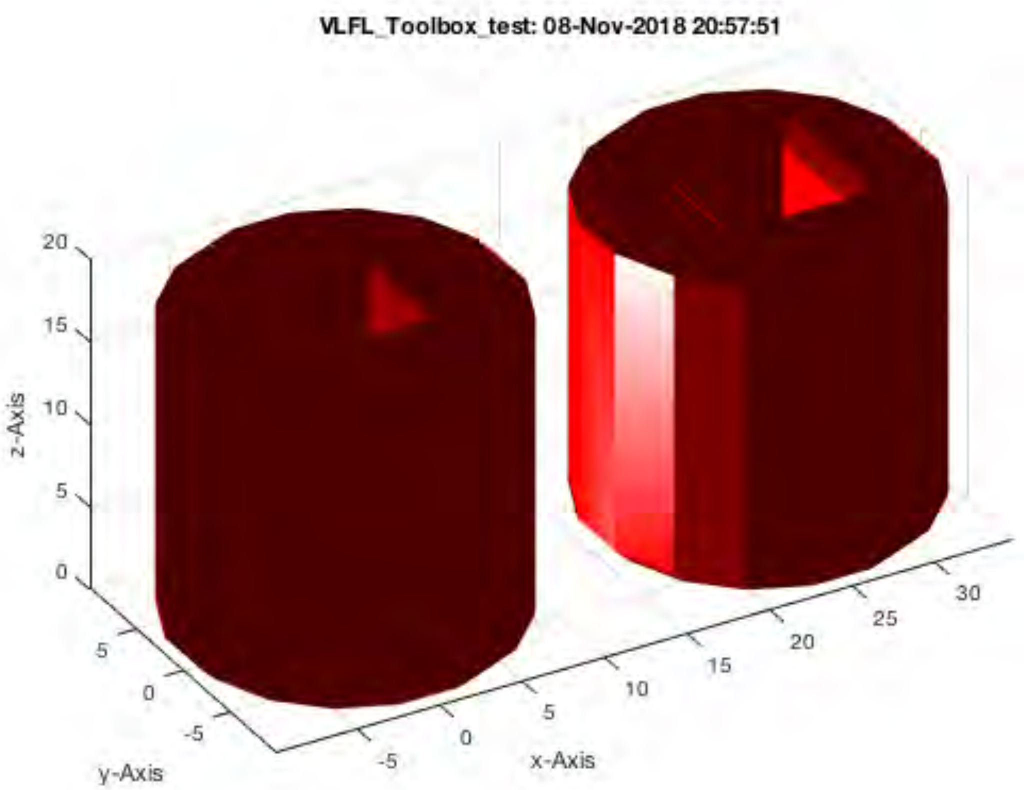
```
Warning: Edge constraints have been split by a coincident point.

ans =

  struct with fields:

    VL: [24×3 double]
    FL: [44×3 double]
```



6. Chamfer the edges of a solid

```
SGofCPLzchamfer(CPLsample(12),20,1)
```

```
ans = 

  struct with fields:

    VL: [192×3 double]
    FL: [384×3 double]
```



7. Creating a drawing temmplate

```
SGdrawingtemplateofCPL(CPLoftext('test'),'','','','','',true)
```

```
CPL =

   -2.3500   -2.3501
   65.2437   -2.3501
   65.2437   25.2609
   -2.3500   25.2609
   -2.3500   -2.3501
       NaN       NaN
    6.3467   -0.3501
    5.2322   -0.2284
    4.6284    0.2221
    4.1309    0.6132
    3.4867    1.2712
    2.6707    2.3528
    2.4506    3.4438
    2.1988    5.4782
    2.1876   15.4157
    2.0961   15.5949
    1.2304   16.1539
    1.2040   16.1593
    0.1199   16.4539
   -0.3500   17.4193
   -0.3500   18.6354
    0.1901   19.2289
```

```
    1.2793   19.3915
    2.1206   19.6932
    2.1875   20.5142
    2.1875   21.5818
    2.7607   22.7415
    3.2450   23.0549
    4.3330   23.1706
    5.4910   23.2609
    6.1298   22.6667
    6.2484   21.5180
    6.2484   20.5096
    6.2955   19.6528
    6.5155   19.4328
    7.3717   19.4005
   10.5142   19.4005
   11.2437   18.6408
   11.2437   17.4157
   10.6666   16.2868
   10.5040   16.1336
    7.3788   16.1336
    6.5868   16.0715
    6.2484   15.4151
    6.2484    6.5053
    6.2779    5.5239
    6.3759    4.6083
    6.6667    4.0878
    6.8961    3.7185
    7.5785    3.0318
    7.9812    2.8078
    8.4654    2.5528
    9.3642    2.4763
   10.5390    2.5823
   11.2001    1.6075
   11.2001    0.3550
   10.5106   -0.3344
       NaN       NaN
    6.3835    0.3501
    5.4979    0.4467
    5.0541    0.7779
    4.5993    1.1354
    4.0184    1.7288
    3.3254    2.6472
    3.1421    3.5562
    2.8987    5.5218
    2.8874   15.5843
    2.6340   16.0808
    1.4997   16.8133
    1.3662   16.8407
    0.6102   17.0461
    0.3500   17.5807
    0.3500   18.3646
    0.5400   18.5734
    1.4508   18.7094
    2.5571   19.1061
    2.7944   19.3434
    2.8875   20.4858
    2.8875   21.4182
```

```
    3.3028   22.2585
    3.4851   22.3765
    4.3972   22.4735
    5.2392   22.5391
    5.4605   22.3333
    5.5484   21.4820
    5.5484   20.4904
    5.6112   19.3472
    6.2147   18.7437
    7.3585   18.7005
   10.2159   18.7005
   10.5437   18.3592
   10.5437   17.5843
   10.1599   16.8336
    7.3514   16.8336
    6.1433   16.7389
    5.5484   15.5849
    5.5484    6.4947
    5.5790    5.4761
    5.6951    4.3917
    6.0635    3.7322
    6.3434    3.2815
    7.1517    2.4682
    7.6479    2.1922
    8.2647    1.8673
    9.3660    1.7737
   10.1911    1.8481
   10.5001    1.3925
   10.5001    0.6450
   10.2196    0.3645
       NaN       NaN
   26.3785   -0.3500
   21.3549   -0.3500
   20.3069   -0.2891
   19.1908    0.0246
   18.9731    0.2225
   18.1627    0.7546
   17.5433    1.2269
   17.1453    1.5627
   16.2676    2.2394
   16.0893    2.4272
   15.5173    3.3049
   15.0852    3.9296
   14.7671    4.3187
   14.3314    5.3767
   14.0393    6.2330
   13.9689    6.3885
   13.7345    7.4384
   13.5967    8.4624
   13.5123    9.4936
   13.5589   10.5263
   13.6665   11.5446
   13.8227   12.5769
   14.0314   13.2723
   14.1483   13.6230
   14.5807   14.6574
   15.0787   15.4983
```

```
   15.2374   15.6869
   15.6818   16.6966
   16.1247   17.1641
   16.8378   17.7629
   17.1483   18.0489
   18.1507   18.7153
   18.2365   18.7916
   19.2620   19.3113
   20.3110   19.4371
   21.3041   19.6290
   22.3785   19.8015
   23.4332   19.5495
   24.4267   19.3925
   25.5247   19.1728
   25.9928   18.7658
   27.1875   17.7580
   27.6274   17.3180
   28.1332   16.6734
   28.5078   15.6720
   28.6592   15.4755
   29.1282   14.6135
   29.2924   13.5424
   29.3835   12.5374
   29.5057   11.5541
   29.6961   10.5464
   29.7892    9.3666
   29.5659    9.1433
   28.4383    8.6500
   18.4626    8.6500
   17.9355    8.3316
   18.1127    7.5629
   18.2431    6.6010
   18.6613    5.7206
   18.6776    5.7016
   19.2418    4.7036
   19.6108    4.2828
   20.5489    3.4924
   21.4870    3.1054
   22.3253    2.8277
   22.4716    2.7653
   23.4119    2.5622
   24.3529    2.5105
   25.3064    2.6297
   26.2698    2.8375
   27.2743    3.1904
   28.4296    3.4070
   29.6295    2.7002
   29.6312    1.3993
   28.9094    0.2515
   28.5071   -0.0212
   27.4178   -0.2701
       NaN       NaN
   26.3516    0.3500
   21.3752    0.3500
   20.4233    0.4053
   19.5393    0.6537
   19.4032    0.7775
```

```
   18.5675    1.3262
   17.9815    1.7731
   17.5849    2.1077
   16.7379    2.7606
   16.6409    2.8629
   16.0985    3.6951
   15.6449    4.3509
   15.3748    4.6813
   14.9870    5.6233
   14.6909    6.4911
   14.6363    6.6115
   14.4242    7.5616
   14.2929    8.5376
   14.2136    9.5064
   14.2572   10.4737
   14.3610   11.4554
   14.5074   12.4231
   14.6988   13.0610
   14.8041   13.3770
   15.2077   14.3426
   15.6515   15.0919
   15.8376   15.3131
   16.2735   16.3034
   16.6054   16.6536
   17.3003   17.2371
   17.5819   17.4966
   18.5794   18.1597
   18.6340   18.2084
   19.4681   18.6310
   20.4192   18.7451
   21.4260   18.9396
   22.3516   19.0882
   23.2970   18.8623
   24.3034   18.7033
   25.2054   18.5228
   25.5374   18.2342
   26.7134   17.2420
   27.1027   16.8528
   27.5156   16.3266
   27.8890   15.3280
   28.0709   15.0921
   28.4549   14.3865
   28.5972   13.4576
   28.6874   12.4626
   28.8138   11.4459
   29.0013   10.4536
   29.0618    9.6868
   28.2919    9.3500
   18.2676    9.3500
   17.1395    8.6684
   17.4234    7.4371
   17.5640    6.3990
   18.0688    5.3365
   18.1014    5.2984
   18.6679    4.2964
   19.1194    3.7815
   20.1812    2.8869
```

```
   21.2431    2.4488
   22.0776    2.1723
   22.2585    2.0952
   23.3182    1.8663
   24.3773    1.8081
   25.4238    1.9389
   26.4604    2.1625
   27.4559    2.5123
   28.3005    2.6706
   28.9301    2.2998
   28.9310    1.6007
   28.3950    0.7485
   28.2231    0.6319
   27.3124    0.4239
       NaN       NaN
   18.1888   11.3222
   17.3892   12.4031
   17.5258   13.5748
   17.8803   14.6715
   18.0809   14.9139
   18.5883   15.7215
   19.1470   16.2660
   19.9627   16.7820
   20.2031   16.9890
   21.3232   17.2402
   22.3929   17.2594
   23.5143   17.0597
   23.8588   16.7836
   24.5878   16.3066
   25.1768   15.7114
   25.6581   14.8351
   25.7915   14.6652
   26.1321   13.5635
   26.2039   12.4146
   25.5650   11.3328
       NaN       NaN
   18.5413   12.0227
   18.1165   12.5969
   18.2131   13.4252
   18.5051   14.3285
   18.6493   14.5028
   19.1366   15.2785
   19.5831   15.7136
   20.3804   16.2180
   20.5271   16.3443
   21.4069   16.5416
   22.3373   16.5583
   23.2159   16.4018
   23.4473   16.2164
   24.1424   15.7615
   24.6104   15.2886
   25.0721   14.4479
   25.1609   14.3348
   25.4387   13.4365
   25.4918   12.5854
   25.1651   12.0322
       NaN       NaN
```

```
   43.3742   -0.3500
   38.3427   -0.3458
   37.2901   -0.2117
   36.1941    0.1317
   36.0283    0.2701
   35.3593    1.4045
   35.3593    2.5836
   35.9457    3.7416
   36.3738    4.0217
   36.6796    3.8032
   37.4836    3.4251
   38.4647    3.1765
   39.4626    2.8365
   40.4289    2.6041
   41.3758    2.4799
   42.2754    2.5423
   42.8488    2.8123
   43.1820    2.9935
   44.1394    3.6502
   44.1830    3.6944
   44.5200    4.5561
   44.4705    5.4006
   44.0747    6.1207
   43.9713    6.2446
   43.1520    6.8602
   42.6909    7.2215
   42.2039    7.5819
   41.2187    7.8935
   40.7188    8.2000
   40.2187    8.4943
   39.2045    8.8370
   38.1565    9.5795
   37.3106   10.2358
   37.1072   10.4351
   36.4031   11.2848
   36.0778   11.7295
   35.6507   12.3736
   35.5025   13.4730
   35.4814   14.5152
   35.5951   15.5730
   35.9518   16.6829
   36.0927   16.8317
   36.7107   17.7221
   37.1416   18.1727
   38.1003   18.7808
   38.1718   18.8438
   39.2620   19.3498
   40.3119   19.5163
   42.3624   19.8212
   44.4054   19.5495
   45.3953   19.4517
   46.5111   19.3687
   47.2967   18.6558
   47.3035   17.4762
   47.1383   16.3165
   46.4451   15.7437
   45.2475   16.0928
```

```
   45.0595   16.1723
   44.2900   16.4288
   43.3365   16.5325
   42.3662   16.5860
   41.4389   16.5289
   40.5539   16.1970
   39.6269   15.3237
   39.4332   14.5403
   39.6751   14.0508
   39.8635   13.7118
   40.6056   12.9599
   40.8102   12.7796
   41.5337   12.3102
   42.5483   11.8488
   42.6115   11.7991
   43.5060   11.3713
   44.5450   10.9538
   44.7451   10.7906
   45.5511   10.3407
   46.3647    9.7720
   46.6012    9.5479
   47.5049    8.7512
   47.6451    8.6009
   48.2323    7.6426
   48.5370    6.5889
   48.7048    5.8786
   48.8019    5.5071
   48.5635    4.4223
   48.3227    3.3995
   47.9253    2.3122
   47.6276    1.9967
   47.0218    1.2625
   46.5670    0.8198
   45.5547    0.3452
   45.3914    0.2087
   44.4620   -0.2924
       NaN       NaN
   43.3560    0.3500
   38.3874    0.3542
   37.4401    0.4749
   36.5570    0.7515
   36.0593    1.5955
   36.0593    2.4164
   36.4294    3.1473
   37.2465    2.7630
   38.2655    2.5048
   39.2675    2.1635
   40.3012    1.9149
   41.3543    1.7767
   42.4548    1.8531
   43.1653    2.1877
   43.5481    2.3958
   44.5908    3.1110
   44.7825    3.3056
   45.2277    4.4439
   45.1601    5.5994
   44.6555    6.5172
```

```
    44.4567    6.7554
    43.5782    7.4155
    43.1151    7.7785
    42.5263    8.2141
    41.5115    8.5351
    41.0793    8.8000
    40.5114    9.1343
    39.5256    9.4674
    38.5736   10.1418
    37.7716   10.7642
    37.6230   10.9097
    36.9556   11.7152
    36.6523   12.1298
    36.3230   12.6264
    36.2016   13.5270
    36.1821   14.4848
    36.2834   15.4270
    36.5694   16.3171
    36.6374   16.3888
    37.2545   17.2779
    37.5886   17.6273
    38.5217   18.2192
    38.5584   18.2515
    39.4681   18.6737
    40.4182   18.8244
    42.3678   19.1143
    44.3248   18.8540
    45.3349   18.7543
    46.2190   18.6885
    46.5985   18.3442
    46.6032   17.5238
    46.4835   16.6835
    46.2850   16.5195
    45.4826   16.7534
    45.3070   16.8277
    44.4402   17.1166
    43.3937   17.2304
    42.3640   17.2872
    41.2913   17.2211
    40.1763   16.8030
    39.1182   15.8061
    38.9925   15.6742
    38.6923   14.4597
    39.0550   13.7256
    39.2980   13.2882
    40.1246   12.4507
    40.3861   12.2204
    41.1965   11.6946
    42.1819   11.2464
    42.2398   11.2009
    43.2242   10.7301
    44.1852   10.3440
    44.3503   10.2094
    45.1790    9.7467
    45.9210    9.2280
    46.1289    9.0311
    47.0163    8.2488
```

```
   47.0851    8.1751
   47.5861    7.3574
   47.8598    6.4111
   48.0254    5.7096
   48.0821    5.4929
   47.8809    4.5777
   47.6509    3.6005
   47.3173    2.6878
   47.1025    2.4602
   46.5062    1.7375
   46.1632    1.4036
   45.1754    0.9405
   44.9969    0.7913
   44.2681    0.3983
       NaN       NaN
   58.3467   -0.3501
   57.2322   -0.2284
   56.6284    0.2221
   56.1309    0.6132
   55.4867    1.2712
   54.6707    2.3528
   54.4506    3.4438
   54.1988    5.4782
   54.1876   15.4157
   54.0961   15.5949
   53.2304   16.1539
   53.2040   16.1593
   52.1199   16.4539
   51.6500   17.4193
   51.6500   18.6354
   52.1901   19.2289
   53.2793   19.3915
   54.1206   19.6932
   54.1875   20.5142
   54.1875   21.5818
   54.7607   22.7415
   55.2450   23.0549
   56.3330   23.1706
   57.4910   23.2609
   58.1298   22.6667
   58.2484   21.5180
   58.2484   20.5096
   58.2955   19.6528
   58.5155   19.4328
   59.3717   19.4005
   62.5142   19.4005
   63.2437   18.6408
   63.2437   17.4157
   62.6666   16.2868
   62.5040   16.1336
   59.3788   16.1336
   58.5868   16.0715
   58.2484   15.4151
   58.2484    6.5053
   58.2779    5.5239
   58.3759    4.6083
   58.6667    4.0878
```

```
   58.8961    3.7185
   59.5785    3.0318
   59.9812    2.8078
   60.4654    2.5528
   61.3642    2.4763
   62.5390    2.5823
   63.2001    1.6075
   63.2001    0.3550
   62.5106   -0.3344
       NaN       NaN
   58.3835    0.3501
   57.4979    0.4467
   57.0541    0.7779
   56.5993    1.1354
   56.0184    1.7288
   55.3254    2.6472
   55.1421    3.5562
   54.8987    5.5218
   54.8874   15.5843
   54.6340   16.0808
   53.4997   16.8133
   53.3662   16.8407
   52.6102   17.0461
   52.3500   17.5807
   52.3500   18.3646
   52.5400   18.5734
   53.4508   18.7094
   54.5571   19.1061
   54.7944   19.3434
   54.8875   20.4858
   54.8875   21.4182
   55.3028   22.2585
   55.4851   22.3765
   56.3972   22.4735
   57.2392   22.5391
   57.4605   22.3333
   57.5484   21.4820
   57.5484   20.4904
   57.6112   19.3472
   58.2147   18.7437
   59.3585   18.7005
   62.2159   18.7005
   62.5437   18.3592
   62.5437   17.5843
   62.1599   16.8336
   59.3514   16.8336
   58.1433   16.7389
   57.5484   15.5849
   57.5484    6.4947
   57.5790    5.4761
   57.6951    4.3917
   58.0635    3.7322
   58.3434    3.2815
   59.1517    2.4682
   59.6479    2.1922
   60.2647    1.8673
   61.3660    1.7737
```

```
    62.1911    1.8481
    62.5001    1.3925
    62.5001    0.6450
    62.2196    0.3645


CPL =

   -2.7500   -2.7502
   65.6437   -2.7502
   65.6437   25.6733
   -2.7500   25.6733
   -2.7500   -2.7502
       NaN       NaN
    6.3257   -0.7502
    5.0804   -0.6143
    4.3852   -0.0955
    3.8633    0.3148
    3.1829    1.0098
    2.2966    2.1845
    2.0555    3.3796
    1.7988    5.4533
    1.7877   15.3179
    1.0703   15.7812
   -0.1602   16.1155
   -0.7500   17.3272
   -0.7500   18.7902
   -0.0098   19.6035
    1.1814   19.7813
    1.7428   19.9827
    1.7875   20.5305
    1.7875   21.6752
    2.4509   23.0175
    3.1079   23.4426
    4.2963   23.5689
    5.6349   23.6733
    6.5123   22.8573
    6.6484   21.5386
    6.6484   20.5206
    6.6865   19.8275
    6.6874   19.8266
    7.3792   19.8005
   10.6847   19.8005
   11.6437   18.8018
   11.6437   17.3194
   10.9912   16.0432
   10.6628   15.7336
    7.3944   15.7336
    6.8402   15.6901
    6.6484   15.3180
    6.6484    6.5113
    6.6773    5.5512
    6.7649    4.7321
    7.0114    4.2910
    7.2118    3.9682
    7.8223    3.3539
    8.1716    3.1596
```

```
    8.5801    2.9445
    9.3632    2.8779
   10.7378    3.0019
   11.6001    1.7303
   11.6001    0.1893
   10.6769   -0.7338
       NaN       NaN
    6.4045    0.7502
    5.6497    0.8325
    5.2974    1.0955
    4.8669    1.4338
    4.3222    1.9902
    3.6995    2.8155
    3.5372    3.6204
    3.2987    5.5467
    3.2873   15.6807
    2.9414   16.3584
    1.6536   17.1901
    1.4589   17.2300
    0.8904   17.3845
    0.7500   17.6728
    0.7500   18.2003
    1.5488   18.3196
    2.7765   18.7599
    3.1811   19.1645
    3.2875   20.4695
    3.2875   21.3248
    3.6126   21.9825
    3.6223   21.9888
    4.4339   22.0751
    5.0799   22.1255
    5.1484   21.4614
    5.1484   20.4794
    5.2202   19.1725
    6.0428   18.3499
    7.3509   18.3005
   10.0455   18.3005
   10.1437   18.1982
   10.1437   17.6806
    9.9152   17.2336
    7.3357   17.2336
    5.8899   17.1202
    5.1484   15.6820
    5.1484    6.4887
    5.1797    5.4488
    5.3060    4.2679
    5.7188    3.5290
    6.0276    3.0318
    6.9078    2.1461
    7.4574    1.8404
    8.1501    1.4756
    9.3670    1.3721
    9.9923    1.4285
   10.1001    1.2697
   10.1001    0.8107
   10.0533    0.7639
       NaN       NaN
```

```
   26.3939   -0.7500
   21.3433   -0.7500
   20.2404   -0.6859
   18.9916   -0.3349
   18.7273   -0.0946
   17.9313    0.4280
   17.2929    0.9148
   16.8941    1.2513
   15.9988    1.9415
   15.7741    2.1783
   15.1851    3.0819
   14.7654    3.6889
   14.4199    4.1115
   13.9569    5.2359
   13.6670    6.0856
   13.5875    6.2610
   13.3404    7.3680
   13.1989    8.4193
   13.1116    9.4863
   13.1598   10.5564
   13.2697   11.5956
   13.4315   12.6647
   13.6500   13.3931
   13.7735   13.7636
   14.2223   14.8373
   14.7513   15.7306
   14.8944   15.9006
   15.3437   16.9213
   15.8501   17.4557
   16.5735   18.0633
   16.9005   18.3645
   17.9058   19.0328
   18.0093   19.1249
   19.1442   19.7000
   20.2491   19.8326
   21.2345   20.0229
   22.3939   20.2091
   23.5110   19.9421
   24.4972   19.7864
   25.7072   19.5442
   26.2530   19.0696
   27.4583   18.0528
   27.9273   17.5838
   28.4862   16.8716
   28.8613   15.8685
   28.9954   15.6945
   29.5130   14.7433
   29.6896   13.5908
   29.7813   12.5801
   29.9011   11.6160
   30.0932   10.5994
   30.2025    9.2142
   29.7953    8.8071
   28.5220    8.2500
   18.5741    8.2500
   18.3904    8.1390
   18.5066    7.6349
```

```
   18.6311    6.7164
   18.9999    5.9401
   19.0068    5.9320
   19.5697    4.9364
   19.8916    4.5692
   20.7590    3.8384
   21.6264    3.4806
   22.4669    3.2021
   22.5934    3.1482
   23.4654    2.9599
   24.3389    2.9119
   25.2393    3.0244
   26.1609    3.2232
   27.1705    3.5779
   28.5034    3.8278
   30.0293    2.9290
   30.0314    1.2842
   29.2033   -0.0325
   28.6694   -0.3944
   27.4780   -0.6666
       NaN       NaN
   26.3363    0.7500
   21.3869    0.7500
   20.4897    0.8022
   19.7385    1.0133
   19.6491    1.0946
   18.7988    1.6528
   18.2319    2.0852
   17.8361    2.4191
   17.0067    3.0585
   16.9561    3.1118
   16.4306    3.9181
   15.9648    4.5917
   15.7221    4.8885
   15.3615    5.7641
   15.0632    6.6386
   15.0177    6.7390
   14.8184    7.6320
   14.6907    8.5807
   14.6143    9.5137
   14.6563   10.4436
   14.7578   11.4044
   14.8987   12.3353
   15.0802   12.9403
   15.1789   13.2364
   15.5661   14.1627
   15.9788   14.8596
   16.1806   15.0994
   16.6116   16.0787
   16.8801   16.3620
   17.5645   16.9367
   17.8296   17.1810
   18.8244   17.8422
   18.8612   17.8751
   19.5859   18.2422
   20.4810   18.3497
   21.4957   18.5457
```

```
   22.3363   18.6806
   23.2192   18.4696
   24.2329   18.3095
   25.0230   18.1514
   25.2772   17.9304
   26.4426   16.9472
   26.8029   16.5869
   27.1626   16.1284
   27.5355   15.1315
   27.7347   14.8730
   28.0701   14.2567
   28.1999   13.4092
   28.2897   12.4199
   28.4184   11.3840
   28.6042   10.4006
   28.6406    9.9392
   28.2082    9.7500
   18.1561    9.7500
   16.6846    8.8610
   17.0295    7.3651
   17.1760    6.2836
   17.7302    5.1171
   17.7721    5.0680
   18.3400    4.0636
   18.8385    3.4951
   19.9711    2.5409
   21.1038    2.0736
   21.9360    1.7979
   22.1368    1.7122
   23.2647    1.4687
   24.3912    1.4068
   25.4909    1.5442
   26.5693    1.7768
   27.5596    2.1248
   28.2268    2.2498
   28.5304    2.0710
   28.5308    1.7158
   28.1011    1.0325
   28.0608    1.0052
   27.2522    0.8205
       NaN       NaN
   17.9874   10.9219
   16.9735   12.2923
   17.1331   13.6603
   17.5232   14.8674
   17.7561   15.1488
   18.2750   15.9747
   18.8978   16.5817
   19.7239   17.1043
   20.0179   17.3574
   21.2754   17.6394
   22.4246   17.6601
   23.6848   17.4356
   24.0939   17.1078
   24.8423   16.6181
   25.5005   15.9530
   25.9929   15.0564
```

```
   26.1517   14.8541
   26.5283   13.6361
   26.6107   12.3169
   25.7936   10.9331
       NaN       NaN
   18.7428   12.4230
   18.5321   12.7077
   18.6059   13.3397
   18.8621   14.1326
   18.9741   14.2679
   19.4499   15.0253
   19.8323   15.3979
   20.6191   15.8957
   20.7123   15.9759
   21.4548   16.1424
   22.3055   16.1577
   23.0453   16.0259
   23.2122   15.8922
   23.8879   15.4500
   24.2867   15.0470
   24.7373   14.2266
   24.8006   14.1459
   25.0424   13.3639
   25.0849   12.6831
   24.9366   12.4319
       NaN       NaN
   43.3846   -0.7500
   38.3172   -0.7458
   37.2043   -0.6040
   35.9988   -0.2263
   35.7189    0.0073
   34.9593    1.2953
   34.9593    2.6790
   35.6372    4.0178
   36.3837    4.5061
   36.8826    4.1498
   37.6191    3.8034
   38.5785    3.5603
   39.5741    3.2211
   40.5019    2.9980
   41.3881    2.8817
   42.1728    2.9361
   42.6679    3.1693
   42.9728    3.3350
   43.8476    3.9351
   44.1155    4.6202
   44.0765    5.2871
   43.7428    5.8941
   43.6939    5.9527
   42.9084    6.5428
   42.4486    6.9033
   42.0196    7.2206
   41.0513    7.5269
   40.5128    7.8571
   40.0515    8.1286
   39.0210    8.4768
   37.9182    9.2581
```

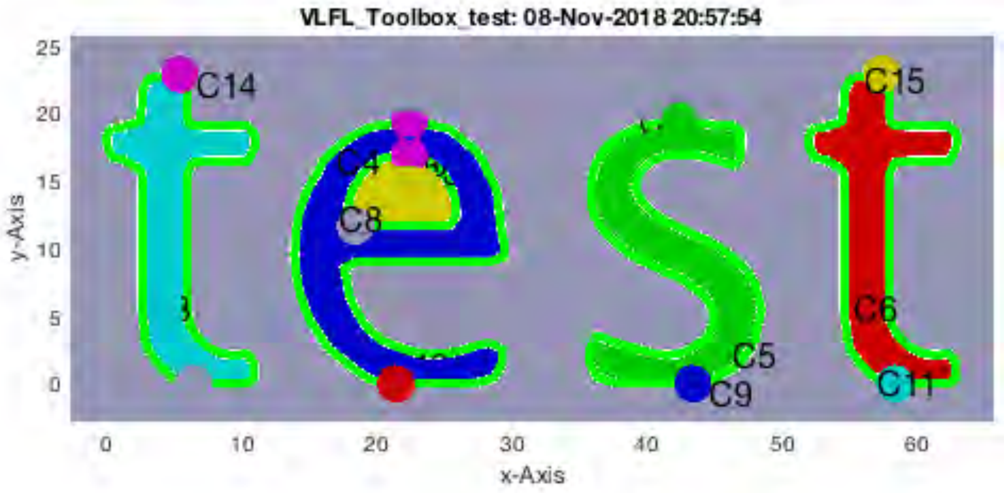
```
   37.0472    9.9339
   36.8125   10.1639
   36.0874   11.0389
   35.7496   11.5008
   35.2666   12.2291
   35.1031   13.4421
   35.0809   14.5326
   35.2017   15.6564
   35.5988   16.8920
   35.7815   17.0847
   36.3999   17.9759
   36.8862   18.4844
   37.8595   19.1018
   37.9508   19.1822
   39.1442   19.7361
   40.2512   19.9117
   42.3593   20.2251
   44.4514   19.9469
   45.4298   19.8503
   46.6780   19.7574
   47.6957   18.8339
   47.7036   17.4490
   47.5125   16.1068
   46.5366   15.3004
   45.1132   15.7153
   44.9182   15.7978
   44.2041   16.0358
   43.3038   16.1337
   42.3675   16.1853
   41.5233   16.1333
   40.7696   15.8507
   39.9869   15.1133
   39.8567   14.5863
   40.0295   14.2367
   40.1866   13.9539
   40.8805   13.2508
   41.0526   13.0992
   41.7264   12.6620
   42.7576   12.1930
   42.8240   12.1409
   43.6670   11.7377
   44.7506   11.3023
   44.9707   11.1228
   45.7637   10.6801
   46.6182   10.0828
   46.8711    9.8432
   47.7840    9.0384
   47.9651    8.8443
   48.6016    7.8056
   48.9241    6.6905
   49.0929    5.9752
   49.2132    5.5152
   48.9535    4.3336
   48.7066    3.2846
   48.2727    2.0975
   47.9277    1.7319
   47.3164    0.9911
```

```
   46.7978    0.4862
   45.7715    0.0050
   45.6169   -0.1241
   44.5729   -0.6870
       NaN       NaN
   43.3455    0.7500
   38.4130    0.7542
   37.5258    0.8672
   36.8235    1.0872
   36.4593    1.7047
   36.4593    2.3210
   36.6108    2.6200
   37.1111    2.3847
   38.1517    2.1210
   39.1560    1.7789
   40.2282    1.5210
   41.3420    1.3749
   42.5573    1.4592
   43.3462    1.8307
   43.7574    2.0542
   44.8488    2.8029
   45.1251    3.0833
   45.6322    4.3798
   45.5541    5.7129
   44.9874    6.7438
   44.7341    7.0473
   43.8217    7.7329
   43.3575    8.0967
   42.7106    8.5753
   41.6788    8.9017
   41.2853    9.1429
   40.6787    9.5000
   39.7091    9.8276
   38.8120   10.4632
   38.0350   11.0661
   37.9177   11.1810
   37.2713   11.9611
   36.9806   12.3586
   36.7071   12.7709
   36.6010   13.5579
   36.5826   14.4674
   36.6767   15.3436
   36.9224   16.1080
   36.9487   16.1357
   37.5652   17.0241
   37.8440   17.3156
   38.7625   17.8982
   38.7793   17.9130
   39.5859   18.2874
   40.4790   18.4291
   42.3709   18.7104
   44.2788   18.4566
   45.3004   18.3558
   46.0521   18.2998
   46.1995   18.1661
   46.2030   17.5510
   46.1222   16.9836
```

```
   45.6170   17.1309
   45.4484   17.2022
   44.5260   17.5096
   43.4263   17.6292
   42.3627   17.6879
   41.2069   17.6167
   39.9605   17.1493
   38.8360   16.0899
   38.6296   15.8732
   38.2688   14.4137
   38.7007   13.5398
   38.9749   13.0461
   39.8497   12.1598
   40.1438   11.9008
   41.0037   11.3429
   41.9726   10.9022
   42.0274   10.8591
   43.0632   10.3638
   43.9795    9.9956
   44.1247    9.8772
   44.9664    9.4073
   45.6675    8.9172
   45.8590    8.7358
   46.7372    7.9616
   46.7651    7.9317
   47.2168    7.1944
   47.4728    6.3095
   47.6372    5.6130
   47.6708    5.4848
   47.4909    4.6664
   47.2671    3.7154
   46.9699    2.9025
   46.8025    2.7251
   46.2115    2.0089
   45.9324    1.7372
   44.9587    1.2807
   44.7714    1.1241
   44.1573    0.7930
       NaN       NaN
   58.3257   -0.7502
   57.0804   -0.6143
   56.3852   -0.0955
   55.8633    0.3148
   55.1829    1.0098
   54.2966    2.1845
   54.0555    3.3796
   53.7988    5.4533
   53.7877   15.3179
   53.0703   15.7812
   51.8398   16.1155
   51.2500   17.3272
   51.2500   18.7902
   51.9902   19.6035
   53.1814   19.7813
   53.7428   19.9827
   53.7875   20.5305
   53.7875   21.6752
```

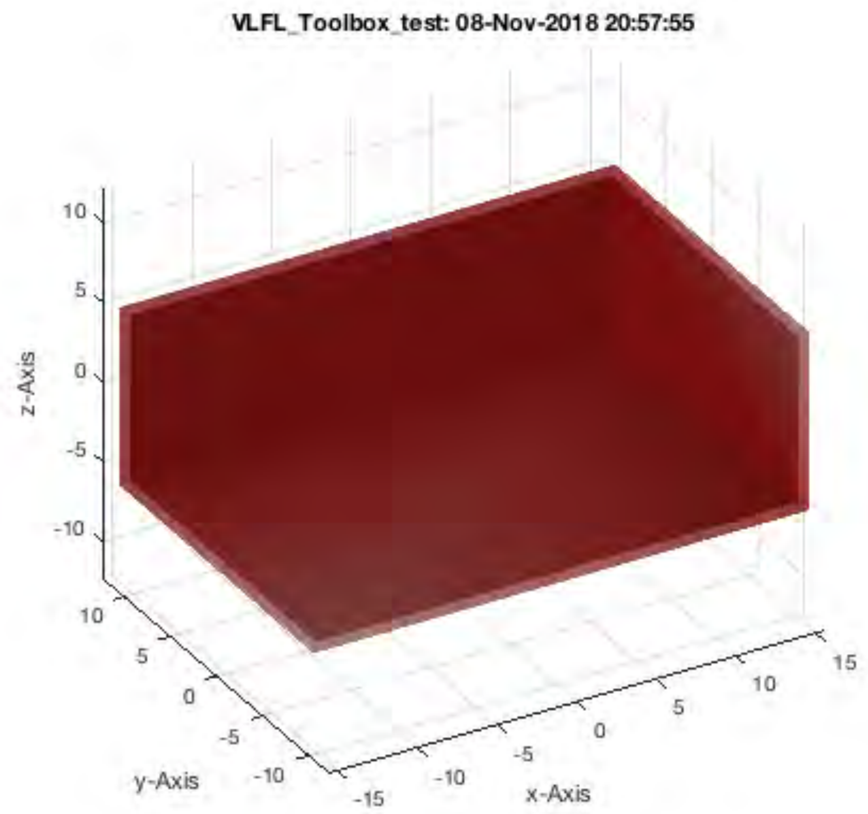
```
   54.4509   23.0175
   55.1079   23.4426
   56.2963   23.5689
   57.6349   23.6733
   58.5123   22.8573
   58.6484   21.5386
   58.6484   20.5206
   58.6865   19.8275
   58.6874   19.8266
   59.3792   19.8005
   62.6847   19.8005
   63.6437   18.8018
   63.6437   17.3194
   62.9912   16.0432
   62.6628   15.7336
   59.3944   15.7336
   58.8402   15.6901
   58.6484   15.3180
   58.6484    6.5113
   58.6773    5.5512
   58.7649    4.7321
   59.0114    4.2910
   59.2118    3.9682
   59.8223    3.3539
   60.1716    3.1596
   60.5801    2.9445
   61.3632    2.8779
   62.7378    3.0019
   63.6001    1.7303
   63.6001    0.1893
   62.6769   -0.7338
       NaN       NaN
   58.4045    0.7502
   57.6497    0.8325
   57.2974    1.0955
   56.8669    1.4338
   56.3222    1.9902
   55.6995    2.8155
   55.5372    3.6204
   55.2987    5.5467
   55.2873   15.6807
   54.9414   16.3584
   53.6536   17.1901
   53.4589   17.2300
   52.8904   17.3845
   52.7500   17.6728
   52.7500   18.2003
   53.5488   18.3196
   54.7765   18.7599
   55.1811   19.1645
   55.2875   20.4695
   55.2875   21.3248
   55.6126   21.9825
   55.6223   21.9888
   56.4339   22.0751
   57.0799   22.1255
   57.1484   21.4614
```

```
   57.1484   20.4794
   57.2202   19.1725
   58.0428   18.3499
   59.3509   18.3005
   62.0455   18.3005
   62.1437   18.1982
   62.1437   17.6806
   61.9152   17.2336
   59.3357   17.2336
   57.8899   17.1202
   57.1484   15.6820
   57.1484    6.4887
   57.1797    5.4488
   57.3060    4.2679
   57.7188    3.5290
   58.0276    3.0318
   58.9078    2.1461
   59.4574    1.8404
   60.1501    1.4756
   61.3670    1.3721
   61.9923    1.4285
   62.1001    1.2697
   62.1001    0.8107
   62.0533    0.7639

Drawing template is separated
Warning: Intersecting edge constraints have been split, this may have added new
points into the triangulation.

ans =

  struct with fields:

    VL: [1504×3 double]
    FL: [2972×3 double]
```



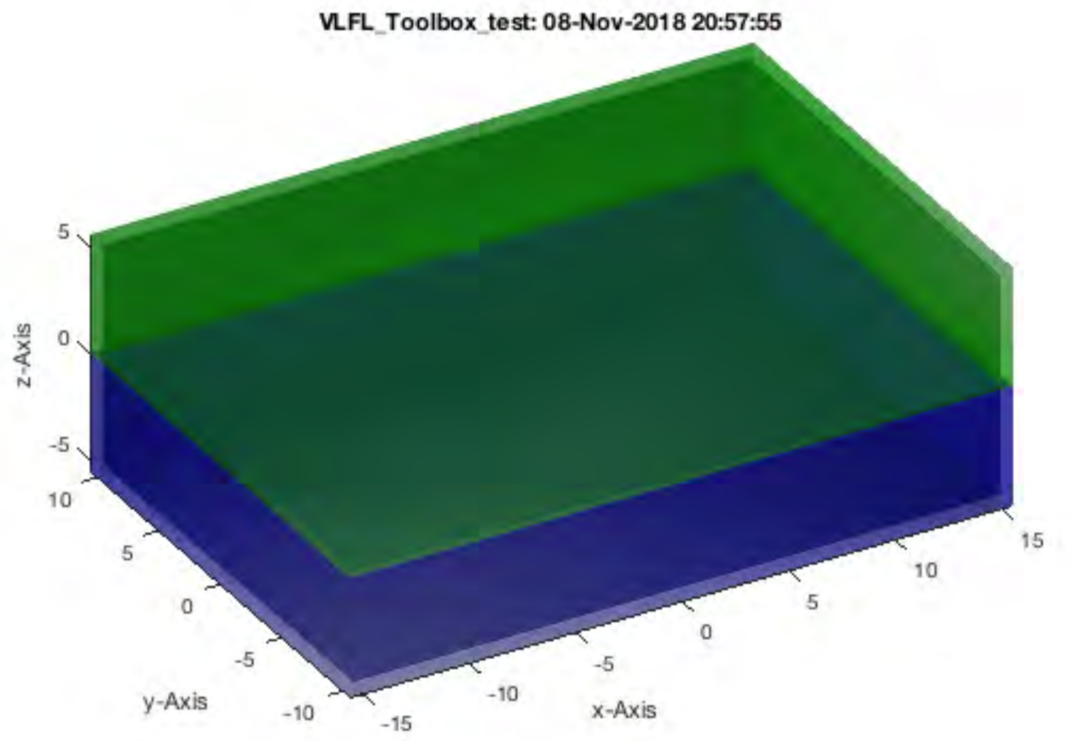
8. Separating an solid into peaces

```
SG=SGhollowsolid(SGbox([30,20,10]));
SGfigure; SGplot(SG); VLFLplotlight(1,0.5); view(-30,30);
```



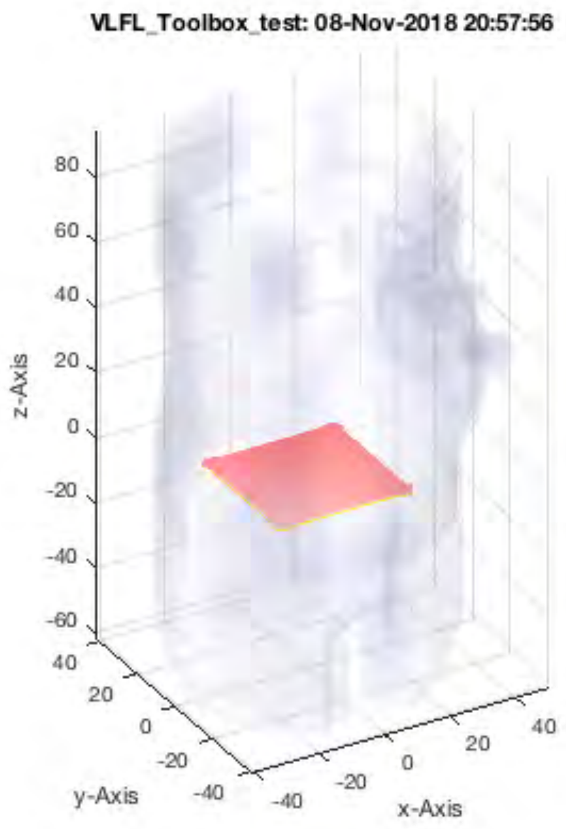

```
SGpuzzlecut3D(SG,[1 1 0.5]); VLFLplotlight(1,0.5); view(-30,30);
```

```
50% 100%
```



9. create a solid surface from an open surface

```
load JACO_robot.mat
VLFLofSGTsurface(JC0,'B'); h=SGplot(JC0); setplotlight(h,'w',0.1);
```



Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:57:57!
Executed 08-Nov-2018 20:57:59 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
database_toolbox
image_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
===========================================================================================
==========
```

Published with MATLAB® R2018a

Tutorial 39: HEBO Modules robot design

2017-07-25: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de) - Last Change: 2017-07-25

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

Motivation for this tutorial: (Originally SolidGeometry 4.0 required)

Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:58:00!
Executed 08-Nov-2018 20:58:02 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
database_toolbox
image_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
===========================================================================================
==========
```

Published with MATLAB® R2018a

Tutorial 40: JACO Robot Simulation and Control

2017-07-25: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de) - Last Change: 2017-07-25

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

Motivation for this tutorial: (Originally SolidGeometry 4.0 required)

Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:58:03!
Executed 08-Nov-2018 20:58:05 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
database_toolbox
image_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
===========================================================================================
==========
```

Published with MATLAB® R2018a

Tutorial 41: Inserting Blades, Cuts and Joints into Solid Geometries

2017-09-04: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de) - Last Change: 2017-09-04

Contents

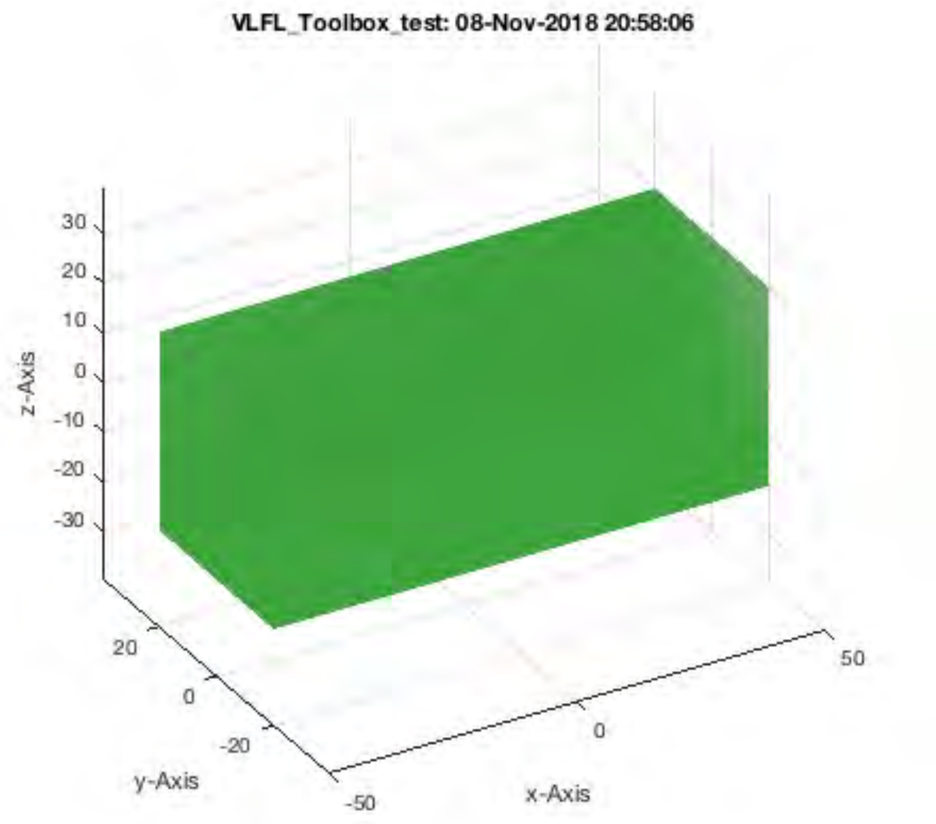
FUNCTION NOT BUGF FEREE

```
% function VLFL_EXP41
% clear all; close all;
```

Create a simple bar type link

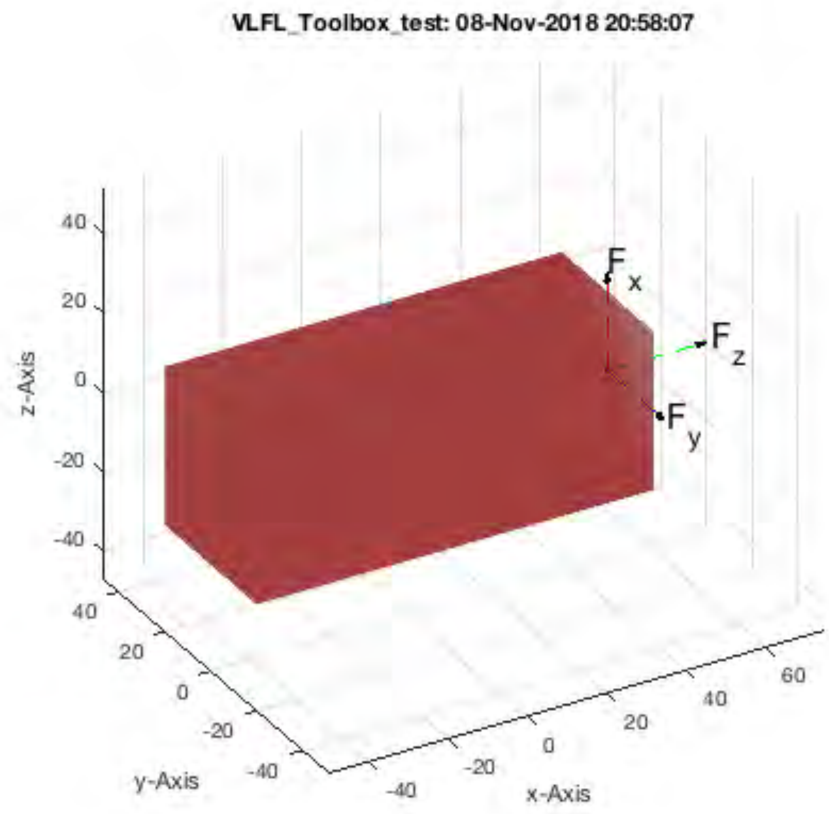
```
A=SGbox([100,40,40])
SGfigure; h=SGplot(A); view(-30,30); setplotlight(h,'g',0.5);
```

```
A =

  struct with fields:

    VL: [8×3 double]
    FL: [12×3 double]
```



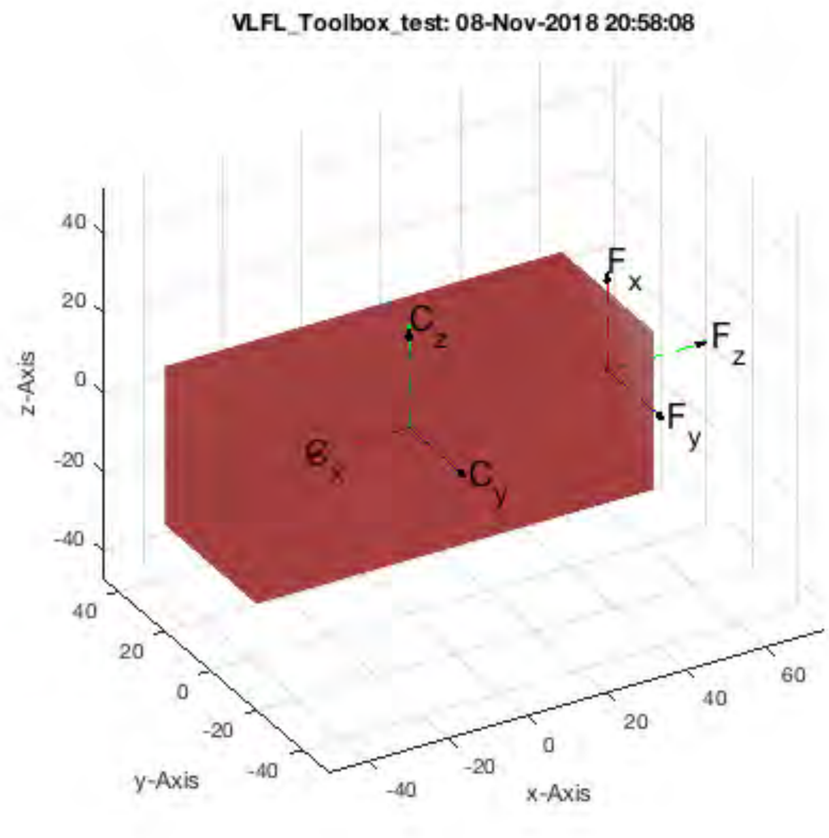
Create a Folloer Frame at the x-Side of the solid

```
A=SGTset(A,'F',TofFS(A,[1 0 0]));

SGfigure; h=SGplot(A); SGTframeplot(A);   view(-30,30); setplotlight(h,'r',0.5);
```



Create a cutting frame in the middle

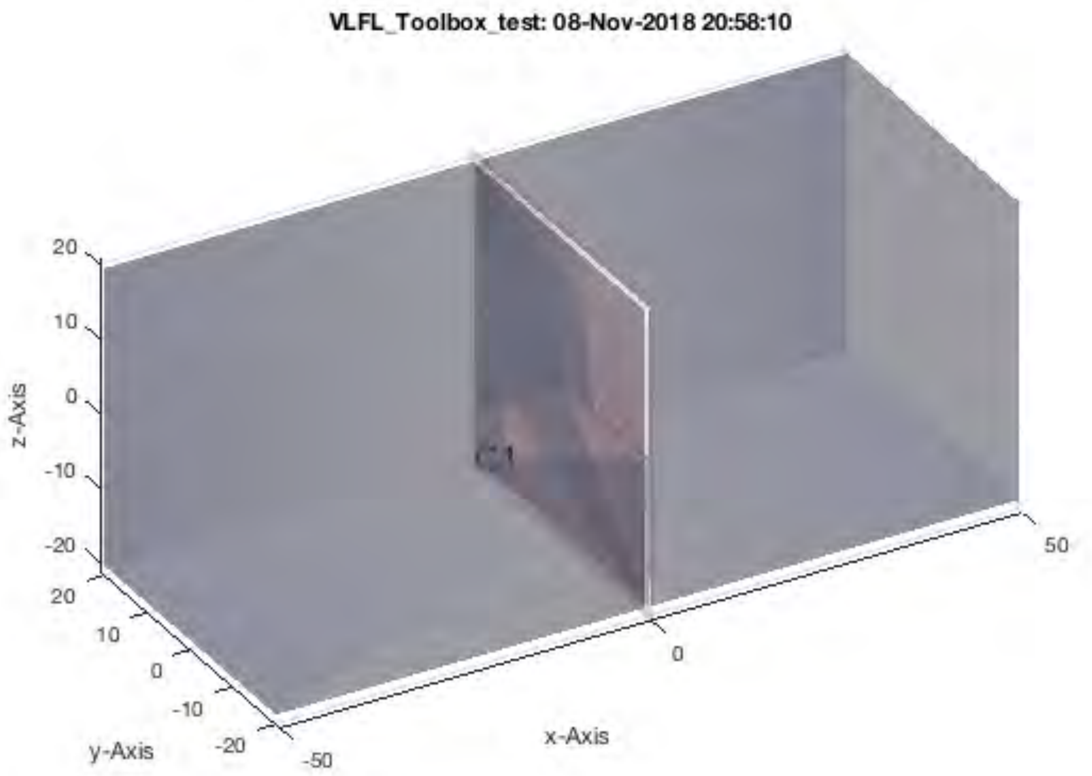
```
A=SGTset(A,'C',TofT(SGTget(A,'F'),rot(0,+pi/2,0),[0 0 -50]));

SGfigure; h=SGplot(A); SGTframeplot(A);   view(-30,30); setplotlight(h,'r',0.5);
```



Show a default cut at the cutting frame

```
TC=SGTget(A,'C');
SGinsertCut(A,TC)
```

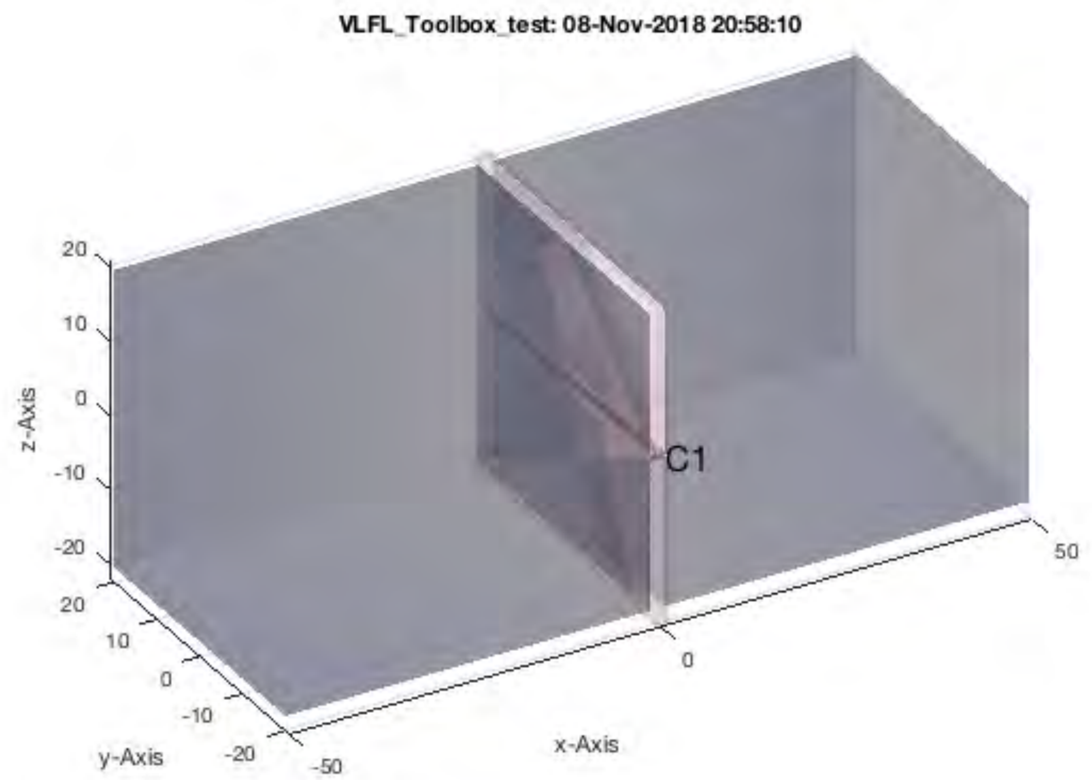
```
PL =

         0   21.0000
         0  -20.0000
         0  -21.0000
         0   -0.0000

SGchecker "A-B":
3 edges [red] are unidirected/open, not removed

ans =

  struct with fields:

    VL: [32×3 double]
    FL: [59×3 double]
```



Show a 1mm cut at the cutting frame

```
SGinsertCut(A,TC,1)
```

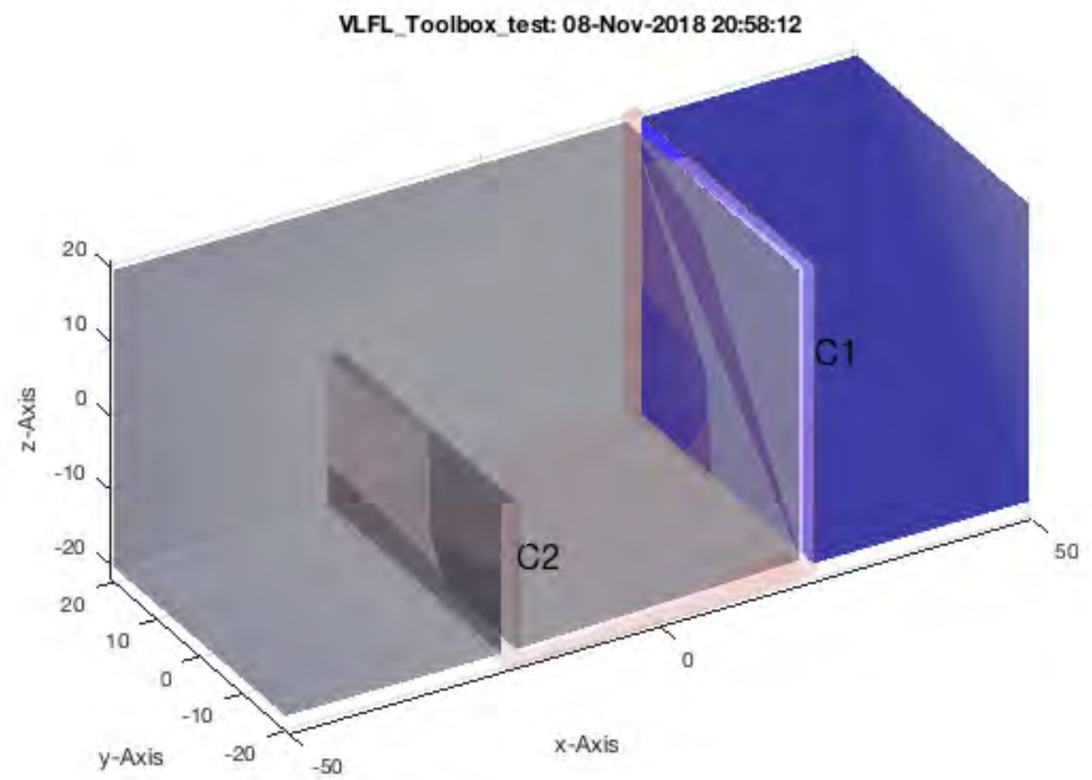
```
PL =

         0   21.0000
         0  -20.0000
         0  -21.0000
         0   -0.0000

SGchecker "A-B":

ans =

  struct with fields:

    VL: [44×3 double]
    FL: [84×3 double]
```



Show a z-cut 1mm by 40 mm at the cutting frame

```
SGinsertCut(A,TC,1,40)
```

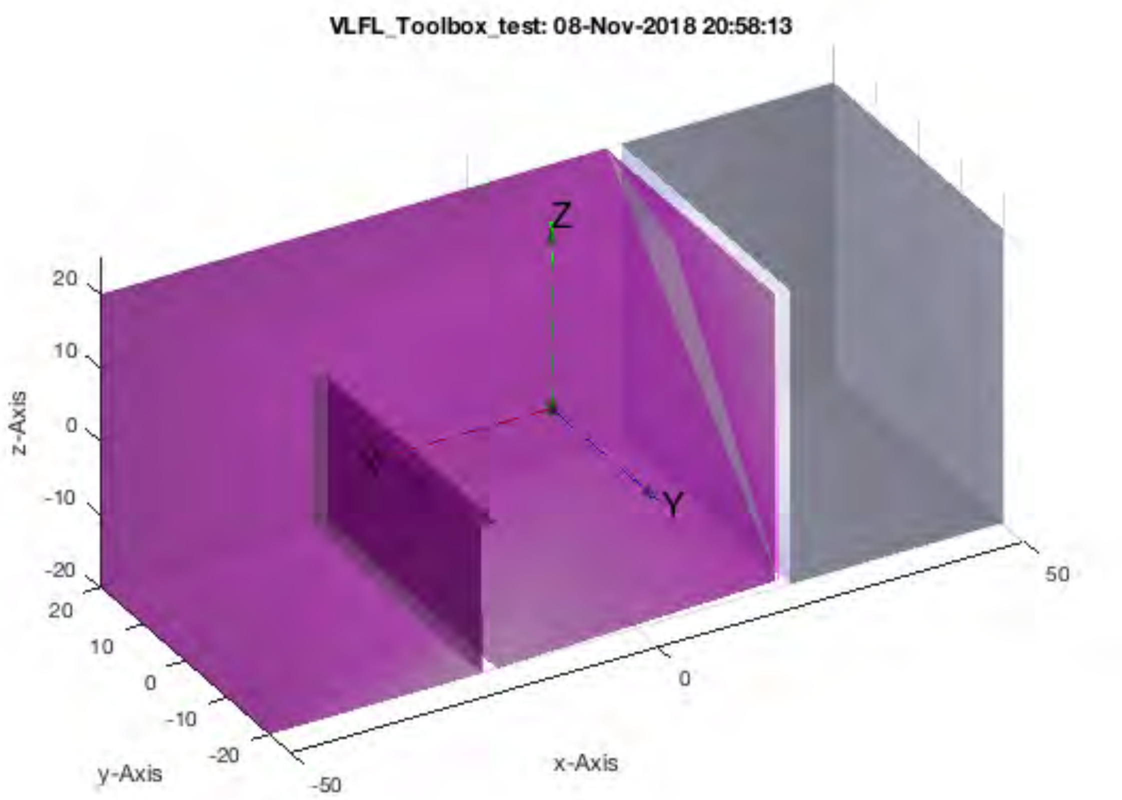
```
PL =

  -20.0000   21.0000
  -20.0000  -20.0000
   20.0000  -21.0000
   20.0000   -0.0000

SGchecker "A-B":
1 edges [blue] are doubled, not removed
10 edges [red] are unidirected/open, not removed


ans =

  struct with fields:

    VL: [50×3 double]
    FL: [88×3 double]
```



Analyze the cut and detec two separted solids

```
B=SGinsertCut(A,TC,1,40);
SGseparatebyT(B,TC)
```

```
PL =

  -20.0000   21.0000
  -20.0000  -20.0000
   20.0000  -21.0000
   20.0000   -0.0000

SGchecker "A-B":
1 edges [blue] are doubled, not removed
10 edges [red] are unidirected/open, not removed


ans =

  struct with fields:

    VL: [36×3 double]
    FL: [64×3 double]
```



Separate the solids into different solids

```
[NX,NA,NB,NC]=SGseparatebyT(B,TC)
```

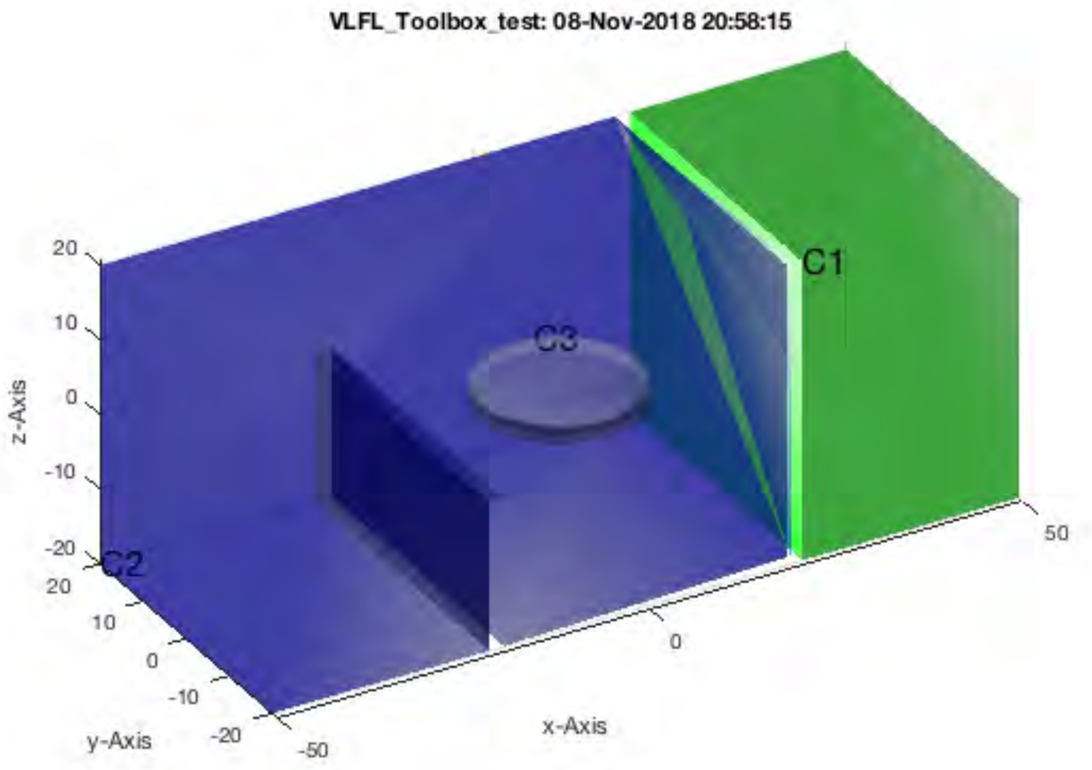
```
NX =

  struct with fields:

    VL: [36×3 double]
    FL: [64×3 double]


NA =

  struct with fields:

    VL: [0×3 double]
    FL: [0×3 double]


NB =

  struct with fields:

    VL: [0×3 double]
    FL: [0×3 double]
```

```
NC =

  struct with fields:

    VL: [14×3 double]
    FL: [24×3 double]
```

Combined Function Simplified Peg in Hole using the same parameter as the cut

```
SGinsertPeghole(B,TC,1,40)
```

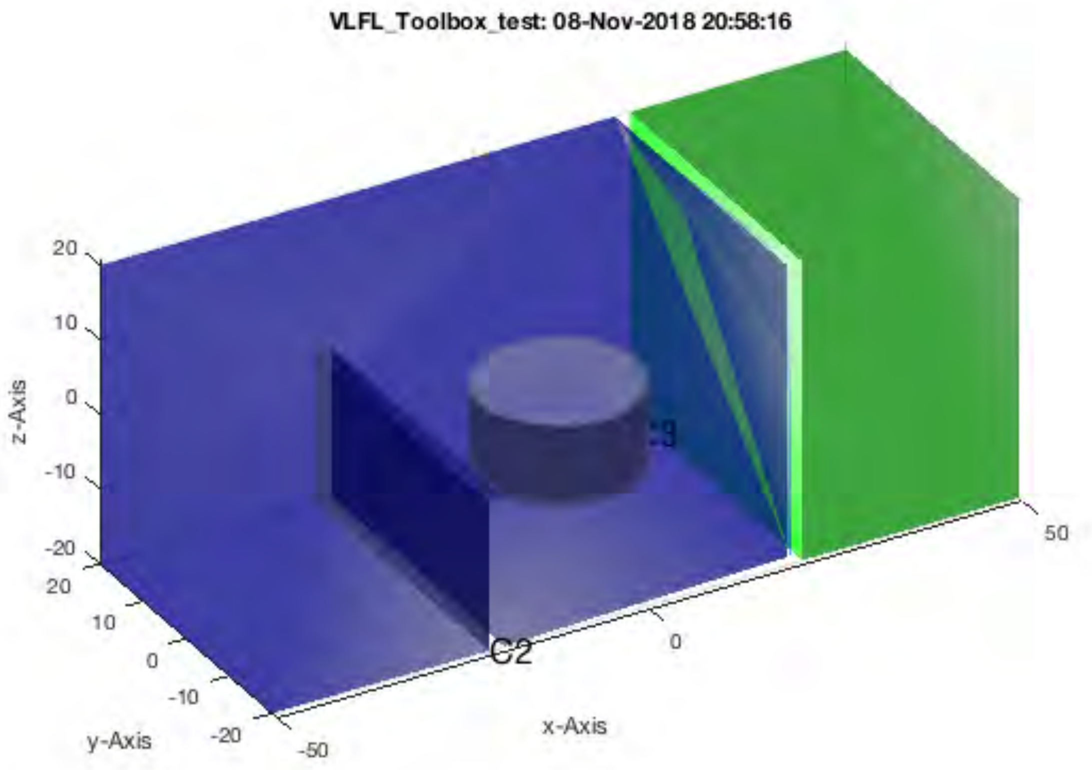
```
SGchecker "A-B":
1 edges [blue] are doubled, not removed
10 edges [red] are unidirected/open, not removed

SGchecker "A+B":
1 edges [blue] are doubled, not removed
10 edges [red] are unidirected/open, not removed


ans =

  struct with fields:

    VL: [260×3 double]
    FL: [503×3 double]
```



Simplified Peg in Hole using a longer peg

```
SGinsertPeghole(B,TC,1,40,20)
```

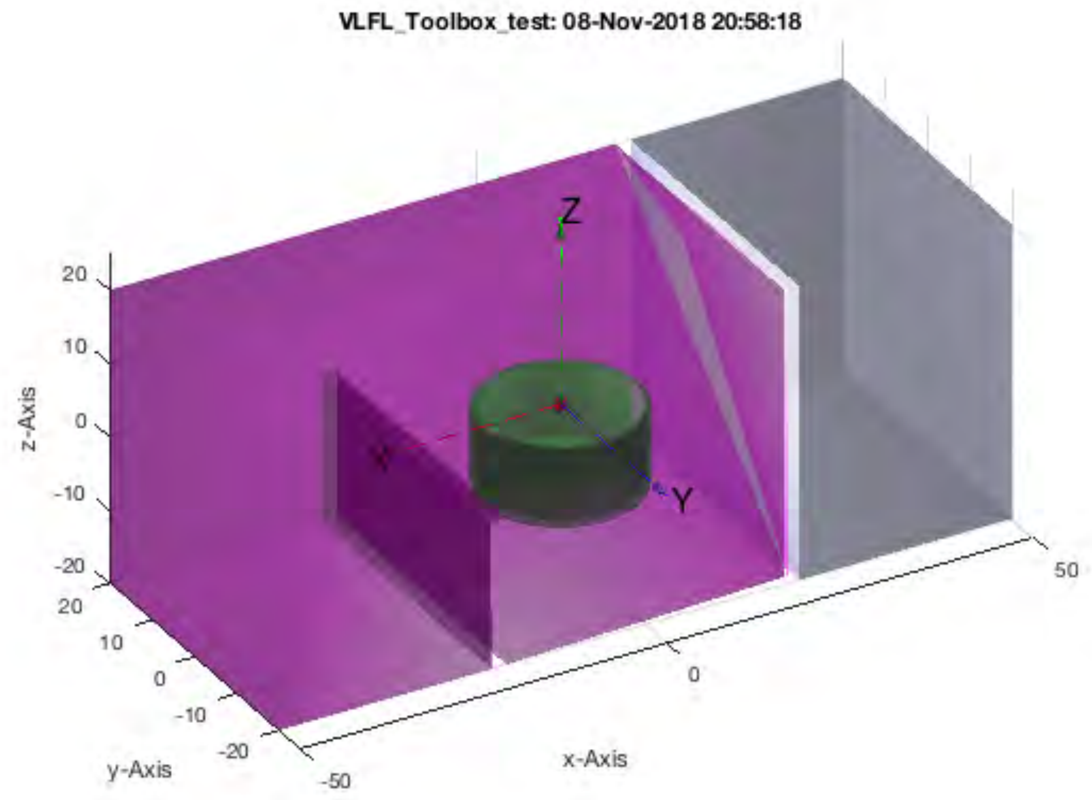
```
SGchecker "A-B":
1 edges [blue] are doubled, not removed
10 edges [red] are unidirected/open, not removed

SGchecker "A+B":
1 edges [blue] are doubled, not removed
10 edges [red] are unidirected/open, not removed


ans =

  struct with fields:

    VL: [277×3 double]
    FL: [537×3 double]
```



Now separate the parts

```
C=SGinsertPeghole(B,TC,1,40,20)
SGseparatebyT(C,TC)
```

```
SGchecker "A-B":
1 edges [blue] are doubled, not removed
10 edges [red] are unidirected/open, not removed

SGchecker "A+B":
1 edges [blue] are doubled, not removed
10 edges [red] are unidirected/open, not removed


C =

  struct with fields:

    VL: [277×3 double]
    FL: [537×3 double]


ans =

  struct with fields:

    VL: [26×3 double]
    FL: [44×3 double]
```



now start to adjust the size to the required movements

```
[X,Y]=SGseparatebyT(C,TC)

% SGboolTL(Y,'-',SGtransrelT(SGgrow(X,0.2),TC,TofR(rot(0,0,1*pi/10)))); Y=SGdelaunay(ans);
% SGboolTL(Y,'-',SGtransrelT(SGgrow(X,0.2),TC,TofR(rot(0,0,2*pi/10)))); Y=ans;
% SGboolTL(Y,'-',SGtransrelT(SGgrow(X,0.2),TC,TofR(rot(0,0,3*pi/10)))); Y=ans;
```

```
X =

  struct with fields:

    VL: [26×3 double]
    FL: [44×3 double]


Y =

  struct with fields:

    VL: [239×3 double]
    FL: [474×3 double]
```

wlim=[0 +pi/4] CVLofSGcutTrot(NB,TC,wlim,1); [~,~,~,~,XA,XB]=CVLofSGcutTrot(NB,TC,wlim,1);

%% SGboolTL(Y,'-',XA)

%% SGboolTL(Y,'-',XB)

Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 20:58:19!
Executed 08-Nov-2018 20:58:21 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
database_toolbox
image_toolbox
map_toolbox
matlab
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
===========================================================================================
==========
```

Published with MATLAB® R2018a

Tutorial 42: Performing FEM Stress and Displacement Analysis and Structural Optimization of Solids

2018-02-27: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de) - Last Change: 2018-03-08

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 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

Motivation for this tutorial: (Originally SolidGeometry 4.2 required)

Yinlun Sun of TU Munich has supplemented the SG-Library with functions that allow a structural and topological optimization of geometric bodies with surface representation.

List of function introduced in this tutorial

- pdemodelofSG - creates a pde tetrahedron mesh-model from a solid surface geometry
- pdeplot3D - Plot 3-D solution or surface mesh
- SGofpdemodel - returns a solid geometry surface model of a pde model
- SGremsurfpoints - returns a surface model without surface points that are inside of a surface - boundary/edge points are unchanged
- SGremsurfedgepoints - returns a surface model without edge points and surface points that are inside of a surface
- pdegplot - Plot PDE tetrahedron mesh geometry
- FSplot - plots the featureEdges of TR, SG or VLFL
- pdeplotfaces - simply plots the surfaces to select; similar as FSplot
- SGplotsurfaceload - plots the surface load of a solid geometry
- pdesolvesurfaceload - calculates the FEM analysis using pde for a pde mesh model
- pdestressstatic - returns the calculated static stress inside a SG based on a pde model by YINLUN SUN
- SGshapeOptiCAO - returns the optimized shape of a given structure based on biological growth

```
function VLFL_EXP42
```

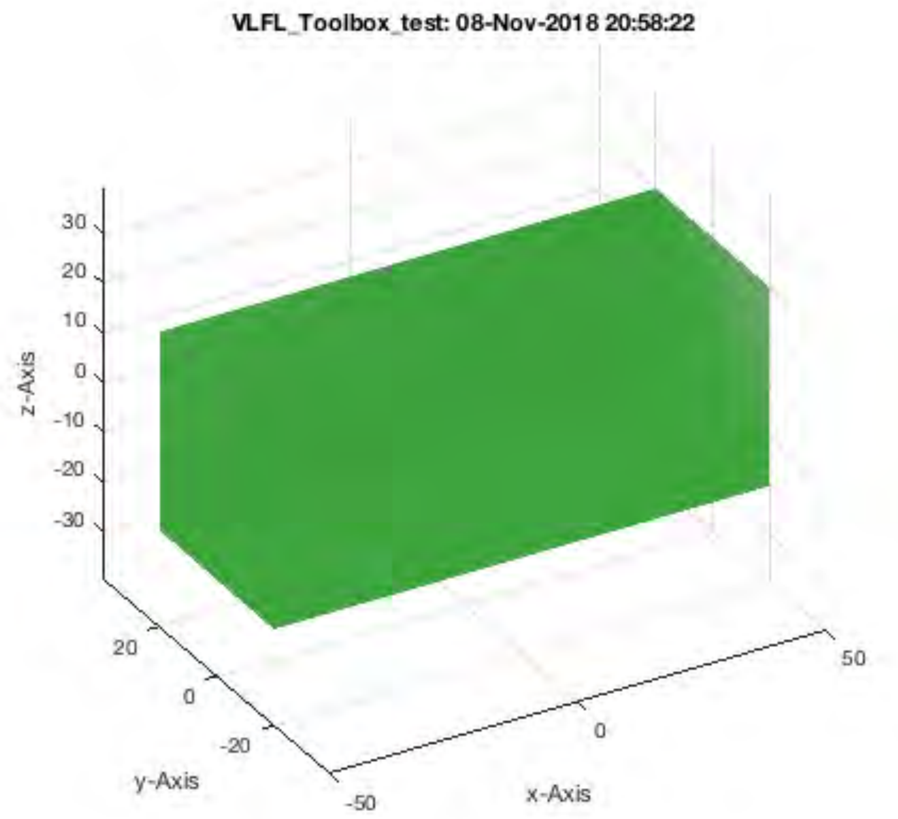
```
% clear all; close all;
```

1. Conversion between triangle surface model and tetrahedon volumen model

1.1 Create a simple bar type link

```
A=SGbox([100,40,40])
SGfigure; h=SGplot(A); view(-30,30); setplotlight(h,'g',0.5);
```

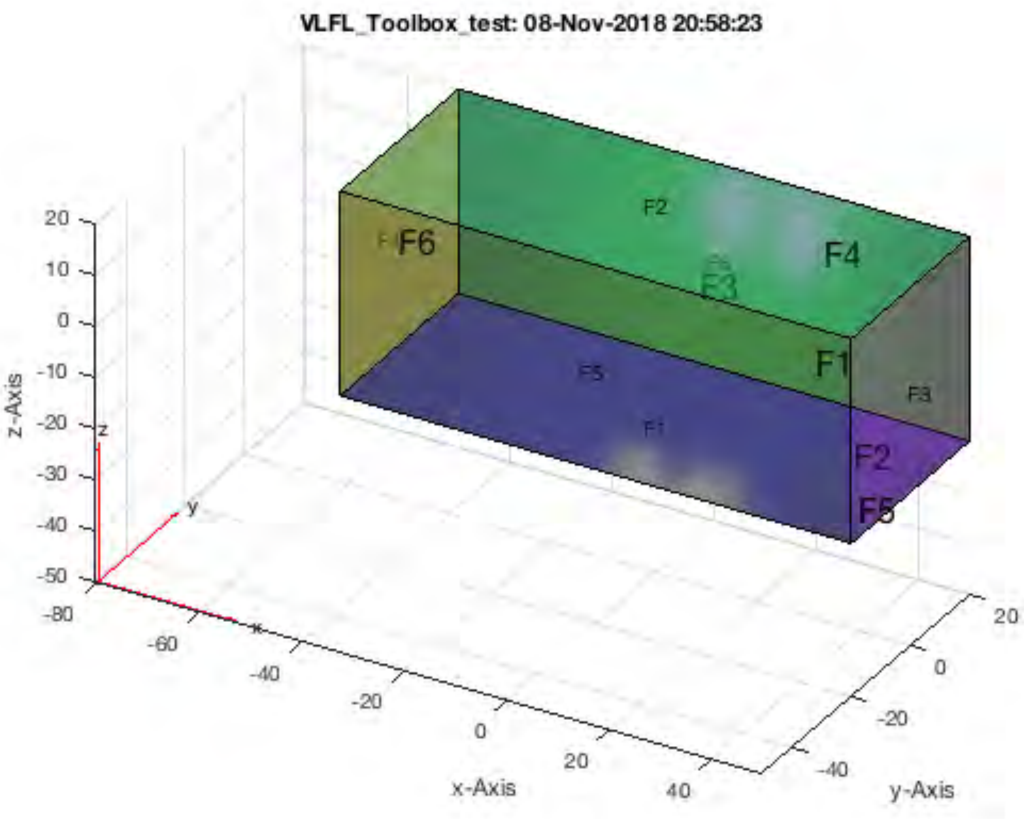
```
A =

  struct with fields:

    VL: [8×3 double]
    FL: [12×3 double]
```



1.2 Create a pde mesh model of the simple bar with voxel size 5mm

```
pdemodelofSG(A,5); model=ans
```

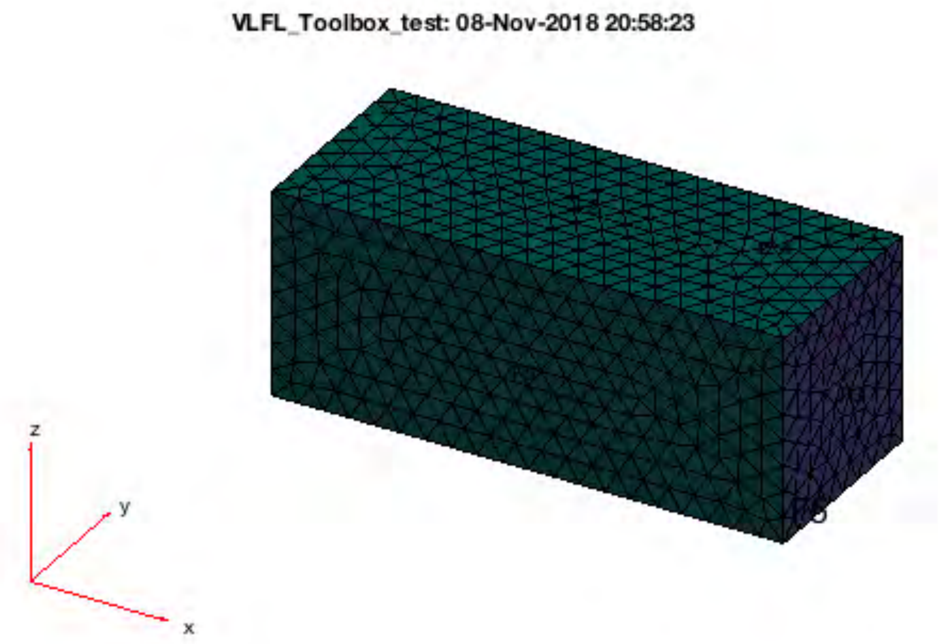
```
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
6 Feature Surfaces found! Only the largest 99.90% (4.000 .. 4000.0mm^2), i.e. 6 of 6 are sh
own.

model = 

  PDEModel with properties:

           PDESystemSize: 3
         IsTimeDependent: 0
                Geometry: [1×1 DiscreteGeometry]
    EquationCoefficients: [1×1 CoefficientAssignmentRecords]
      BoundaryConditions: []
       InitialConditions: []
                    Mesh: [1×1 FEMesh]
           SolverOptions: [1×1 PDESolverOptions]
```



1.3 Show the tetrahedron volume structure of the mesh

```
pdeplot3D(model);
```

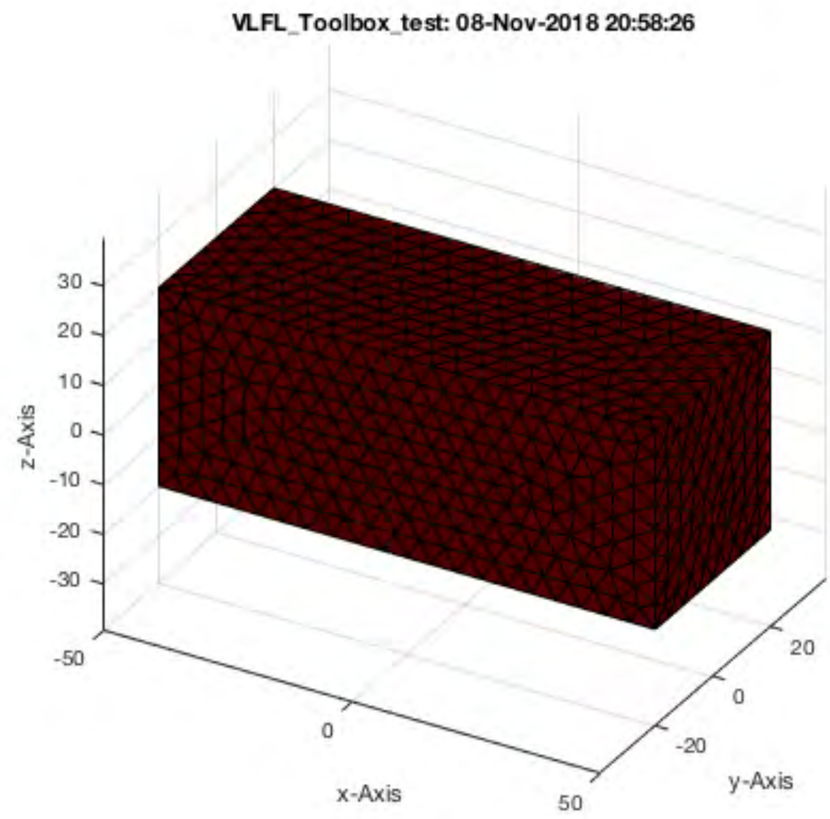
```
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```



1.4 Convert the tetrahedron volume into a surface model

```
SGofpdemodel(model); B=ans
```

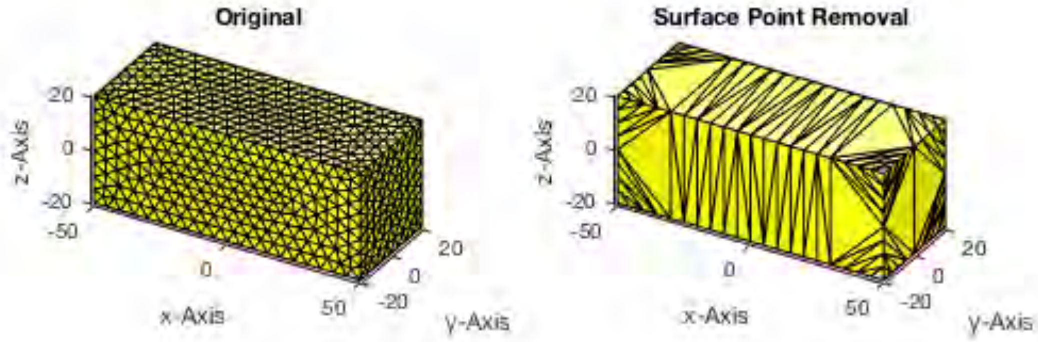
```
B =

  struct with fields:

    VL: [924×3 double]
    FL: [1844×3 double]
```



1.5 Remove surface points of the surface model but protect the edge points

```
SGremsurfpoints(B); C=ans
```

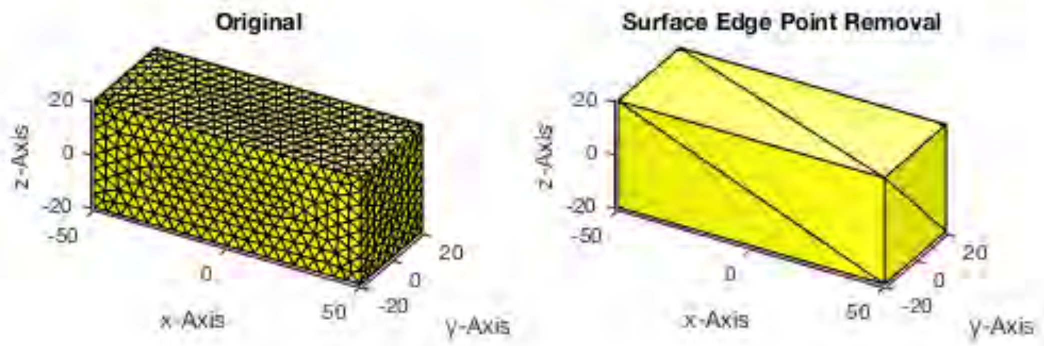
```
C =

  struct with fields:

    VL: [140×3 double]
    FL: [276×3 double]
```



1.6 Remove unused edge points and surface points of the surface model

```
SGremsurfedgepoints(B); C=ans
```

```
C =

  struct with fields:

    VL: [8×3 double]
    FL: [12×3 double]
```

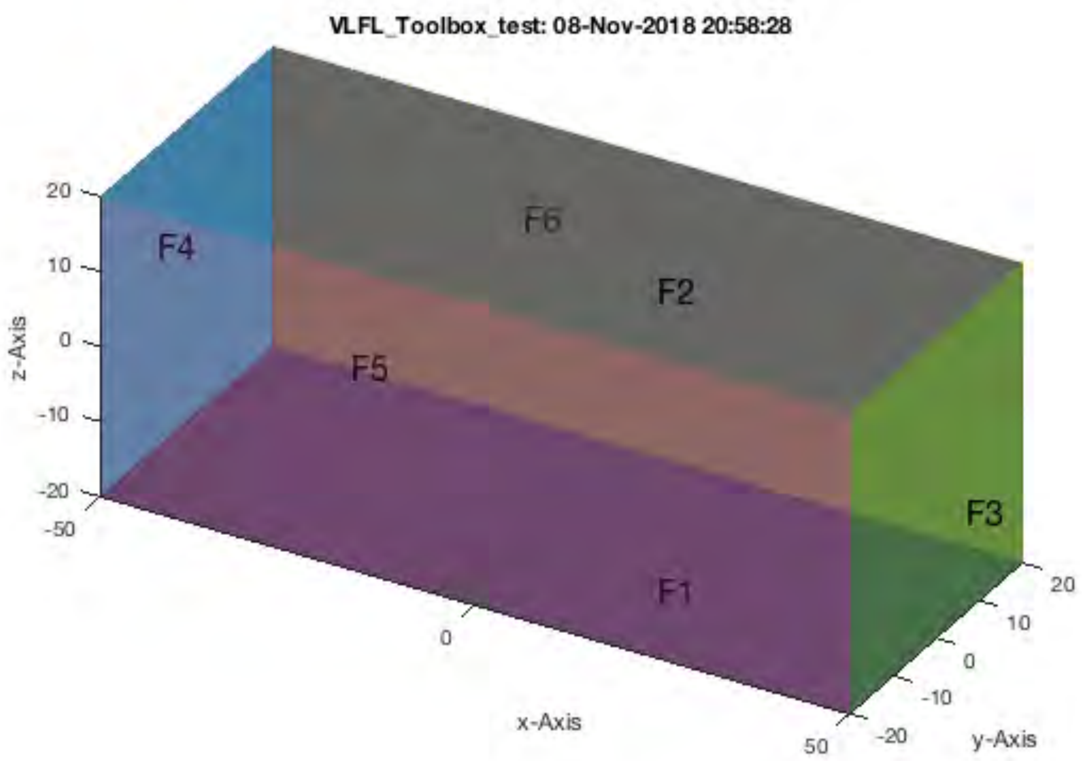


2. Selection of Feature Surfaces for load specification

2.1 Feature surface plot on surface model lebel

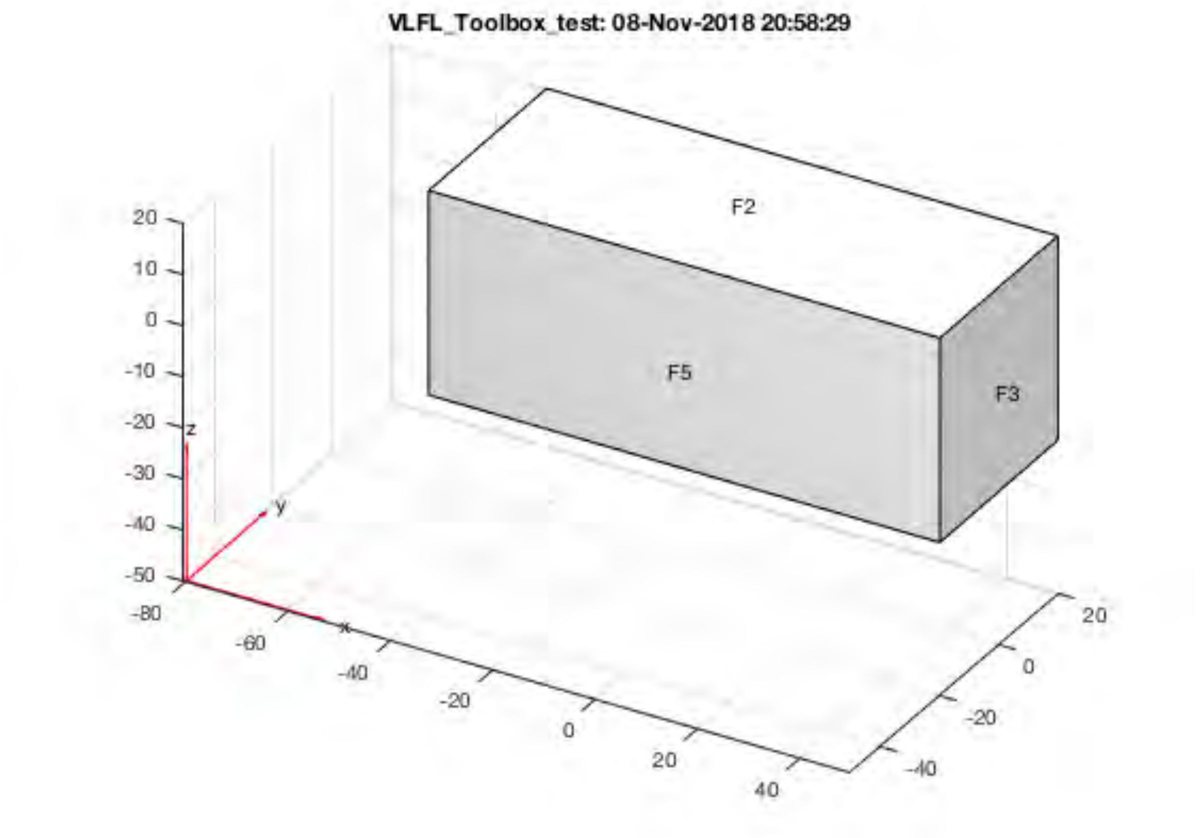
```
SGfigure; view(30,30);
FSplot(A);
```

```
6 Feature Surfaces found! Only the largest 99.90% (4.000 .. 4000.0mm^2), i.e. 6 of 6 are sh
own.
```



2.2 Feature surface plot on pde model lebel

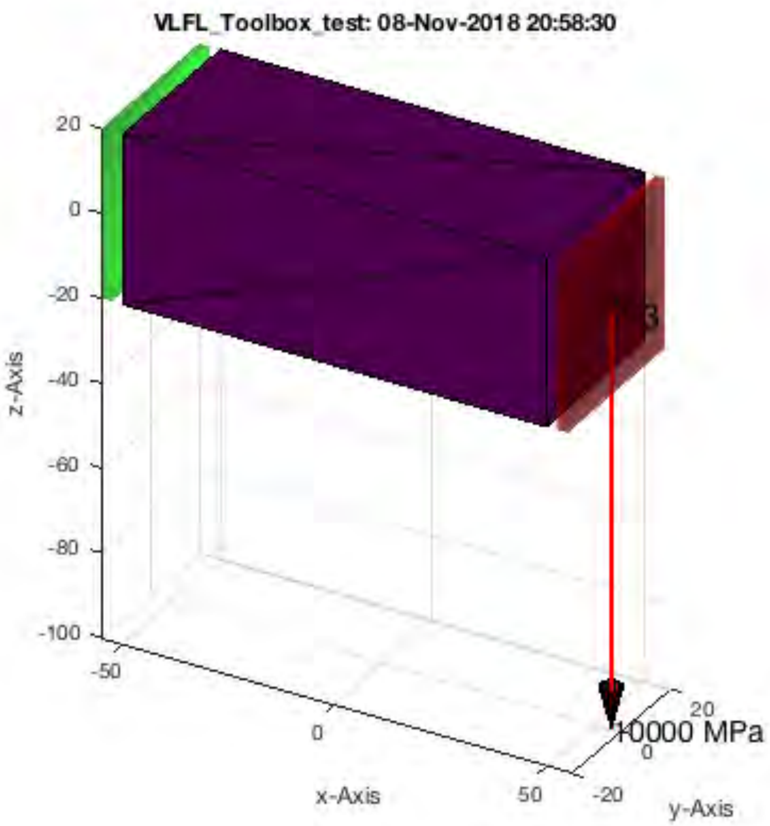
```
SGfigure; view(30,30)
pdeplotfaces(model);
```



3. Calculating surface load dependend displacement and von-Miss stress situation

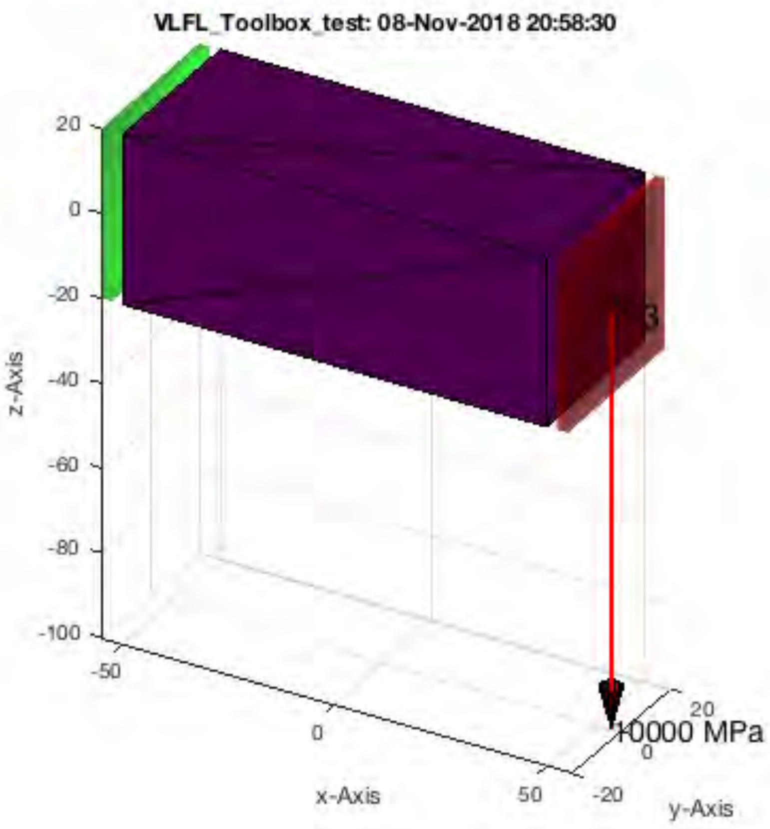
3.1 Display a loading condition Fixed facet is 4, loaded surface is 3, load vector in z using Propertynames

```
SGfigure; SGplot(A,'m'); view(30,30);
SGplotsurfaceload (A,'FixedFaceIndices',4,'LoadFaceIndices',3,'Load',[0 0 -1e4]);
```



3.2 Display a loading condition Fixed facet is 4, loaded surface is 3, load vector in z using varargin

```
SGfigure; SGplot(A,'m'); view(30,30);
SGplotsurfaceload (A,4,3,[0 0 -1e4]);
```

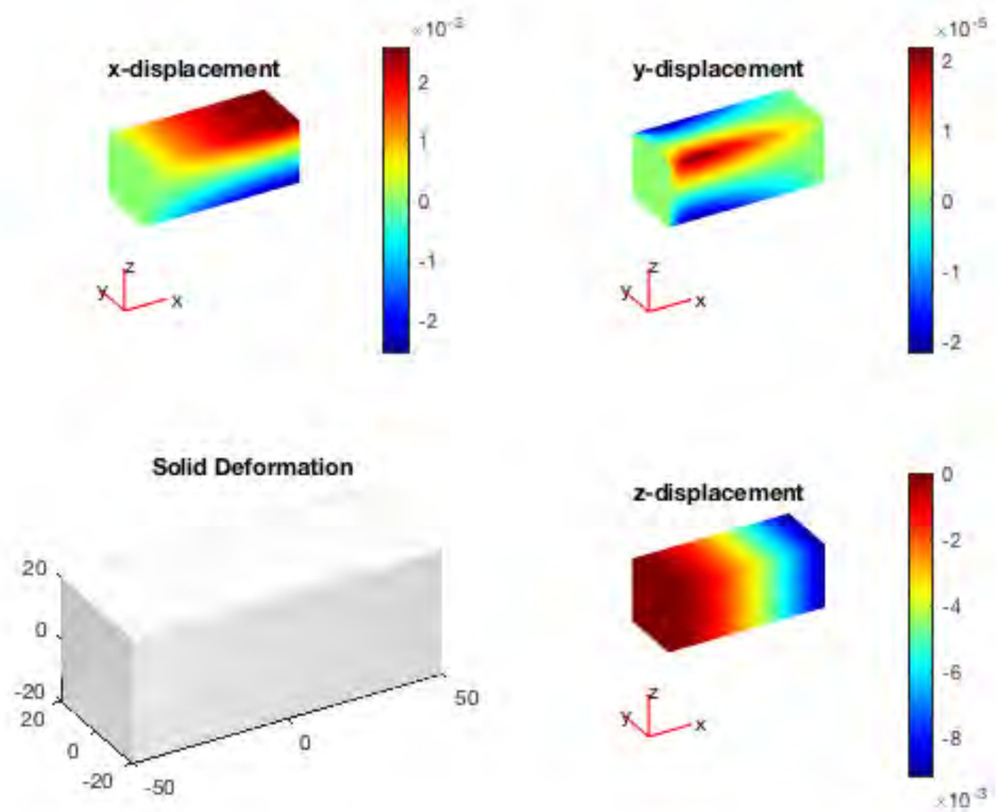


3.3 Fixed facet is 4, loaded surface is 3, load vector in z using varargin

```
pdesolvesurfaceload(model,4,3,[0 0 -1e4]);
```

```
ATTENTION: The already existing pde BoundaryConditions are deleted first
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
```

```
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```

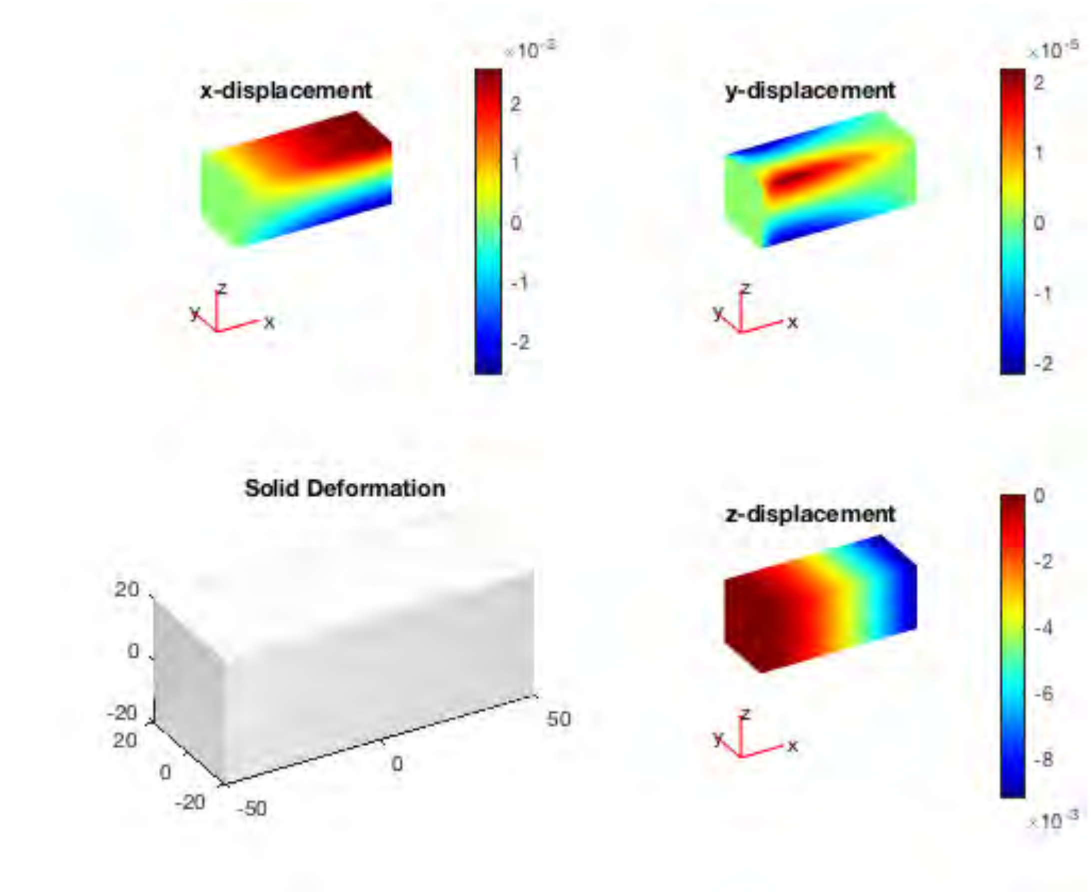


3.4 Fixed facet is 4, loaded surface is 3, load vector in z using Propertynames

```
pdesolvesurfaceload(model,'FixedFaceIndices',4,'LoadFaceIndices',3,'Load',[0 0 -1e4]);
```

```
ATTENTION: The already existing pde BoundaryConditions are deleted first
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
```

```
future release, it will be an error.
```

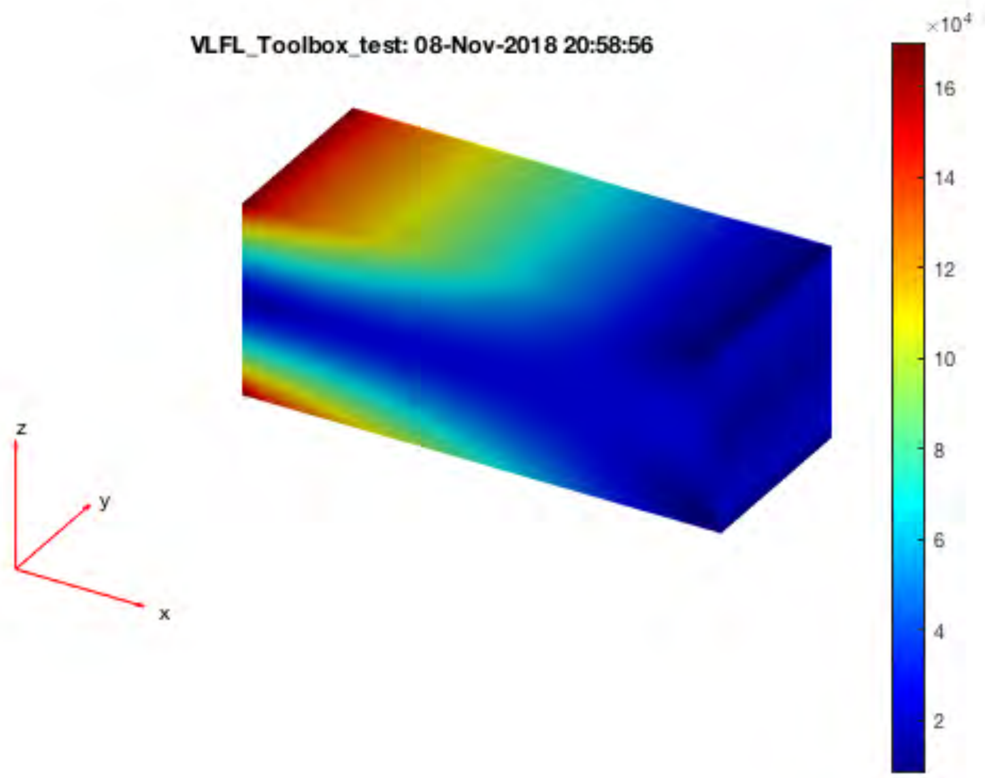


3.5 Show von-mises-Stress for load condition

```
[result,model]=pdesolvesurfaceload(model,'FixedFaceIndices',4,'LoadFaceIndices',3,'Load',[0
 0 -1e4]);
pdestressstatic(model,result);
```

```
ATTENTION: The already existing pde BoundaryConditions are deleted first
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```

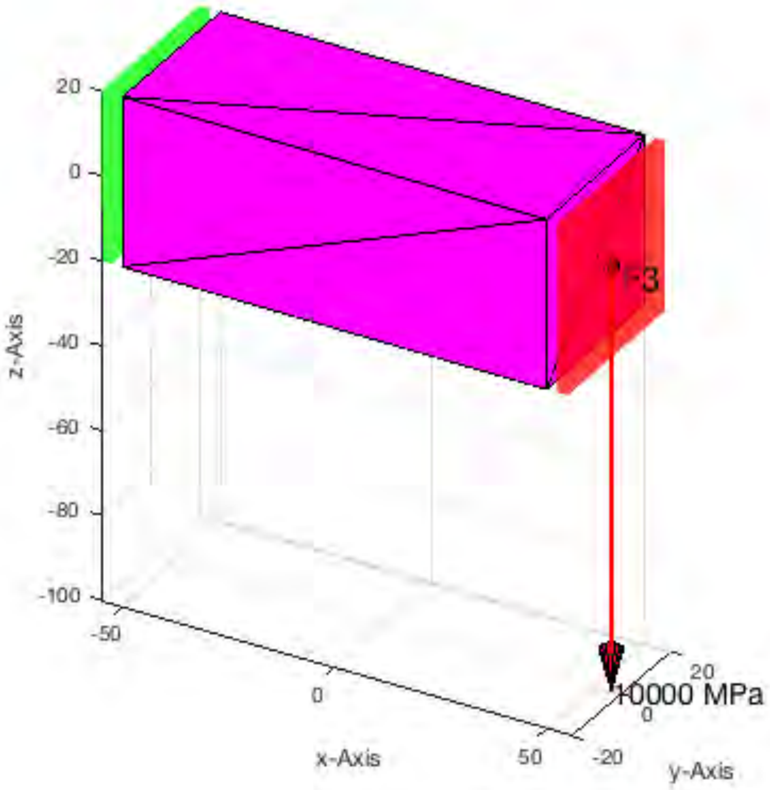
```
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```

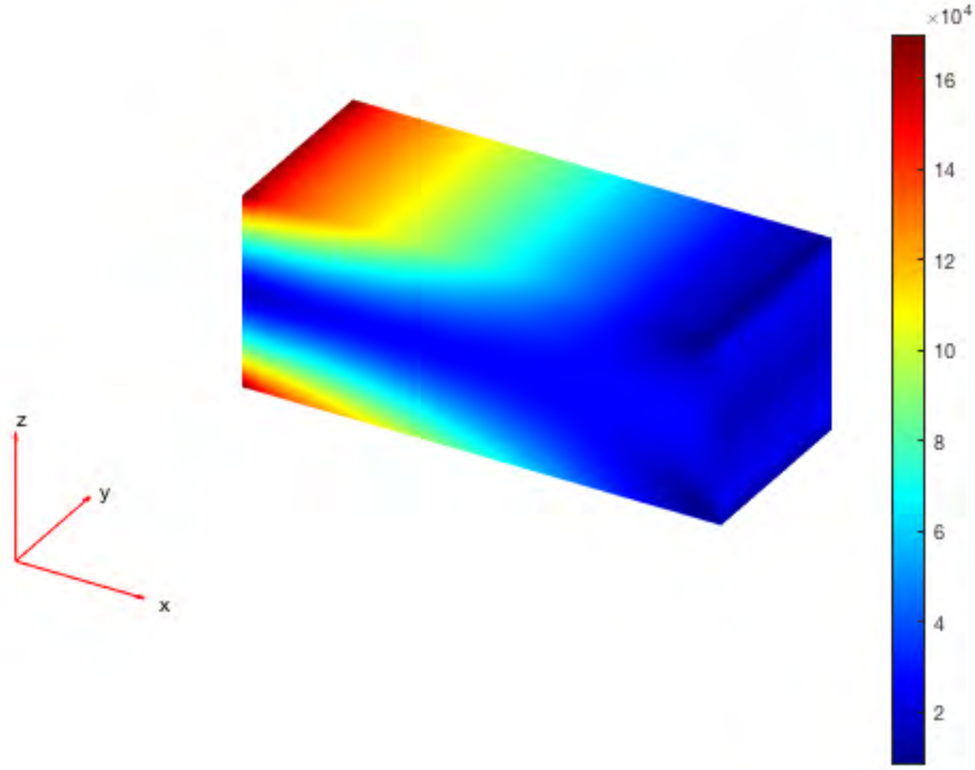


3.6 Show von-mises-Stress and load condition

```
close all; figure(1);  view(30,30); SGplot(A,'m');
SGplotsurfaceload (A,'FixedFaceIndices',4,'LoadFaceIndices',3,'Load',[0 0 -1e4]);
figure(2); view(30,30);
[~,stress]=pdestressstatic(model,result);
pdeplot3D(model,'colormapdata',stress);
```

```
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```



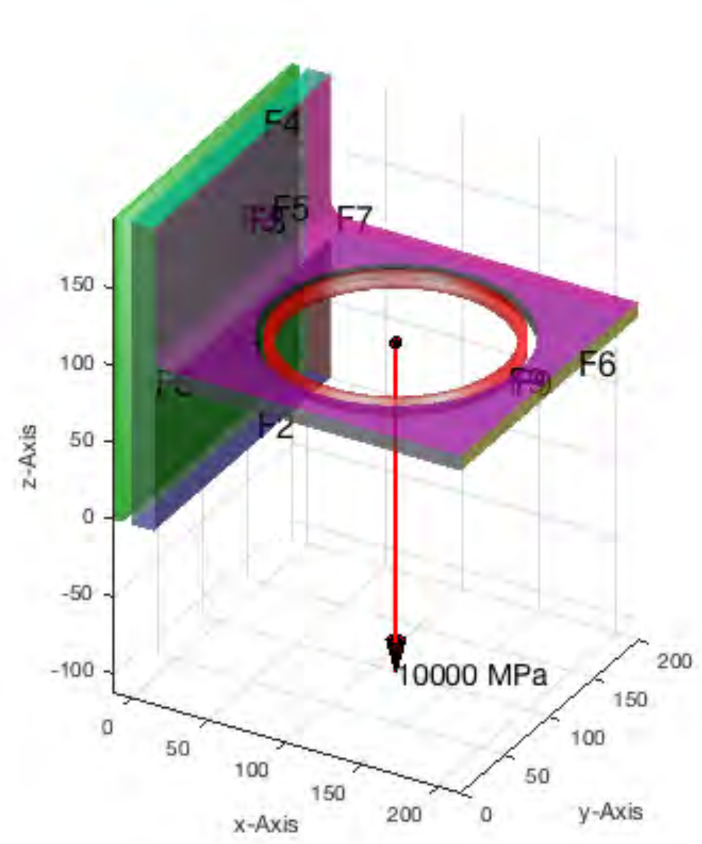


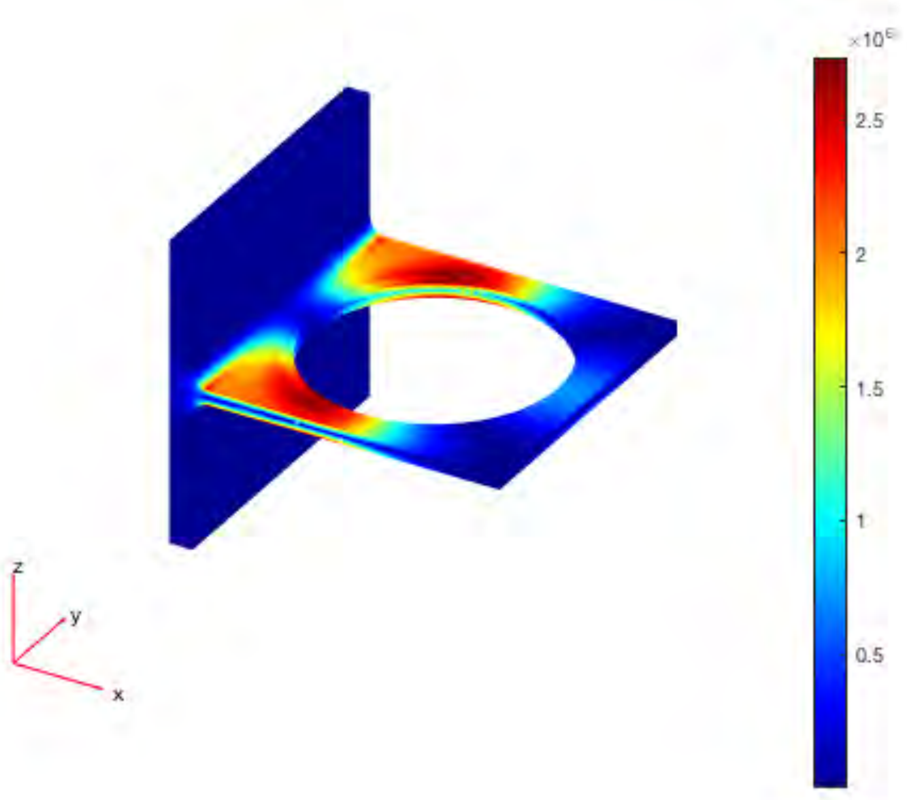
3.7 Do the same for the matlab standard fem solid: BracketWithHole

```
A=SGreadSTL(which('BracketWithHole.stl'),1000);
model=pdemodelofSG(A);
[result,model]=pdesolvesurfaceload(model,'FixedFaceIndices',3,'LoadFaceIndices',9,'Load',[0
 0 -1e4]);
close all; figure(1);  view(30,30); FSplot(A);
SGplotsurfaceload (A,'FixedFaceIndices',3,'LoadFaceIndices',9,'Load',[0 0 -1e4]);
figure(2); view(30,30);
[~,stress]=pdestressstatic(model,result);
pdeplot3D(model,'colormapdata',stress);
```

```
LOADING ASCII STL-File: /Applications/MATLAB_R2018a.app/toolbox/pde/pdedata/BracketWithHole
.stl scaling factor: 1000
Processing 2102 lines:
Finishing solid bracket_with_hole_meters
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
ATTENTION: The already existing pde BoundaryConditions are deleted first
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
```

```
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
9 Feature Surfaces found! Only the largest 99.90% (39.433 .. 39433.2mm^2), i.e. 9 of 9 are
shown.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```





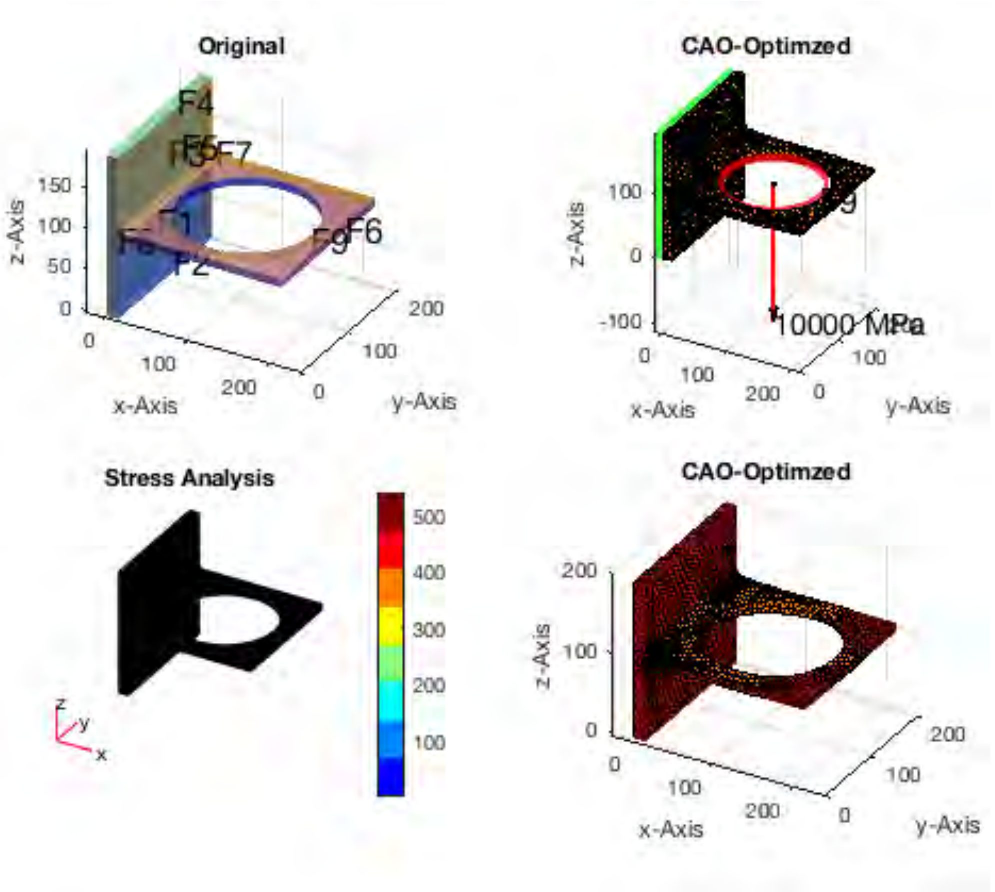
4 Structural Optimization

4.1 CAO Optimization using load face 9

```
SGshapeOptiCAO(A,'FixedFaceIndices',3,'LoadFaceIndices',9,'Load',[0 0 -1e4]);
```

```
Iteration   0: Warning: A value of class "logical" was indexed with no subscripts specified
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Volume of SG is 801143.4667
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
```

```
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration   1: Warning: A value of class "logical" was indexed with no subscripts specified
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
CAO end: CAO process stops because meshing size does not fit.

******************** CAO result *********************
Original volume: 801143.4667 mm^3
Optimized volume: 801143.4667 mm^3
Original maximal von Mises stress: 541.644 N/mm^2
Optimized maximal von Mises stress: 541.644 N/mm^2
9 Feature Surfaces found! Only the largest 99.90% (39.433 .. 39433.2mm^2), i.e. 9 of 9 are
shown.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```



4.2 CAO Optimization using load face 6

```
SGshapeOptiCAO(A,'FixedFaceIndices',3,'LoadFaceIndices',6,'Load',[0 0 -1e4]);
```

```
Iteration   0: Warning: A value of class "logical" was indexed with no subscripts specified
```

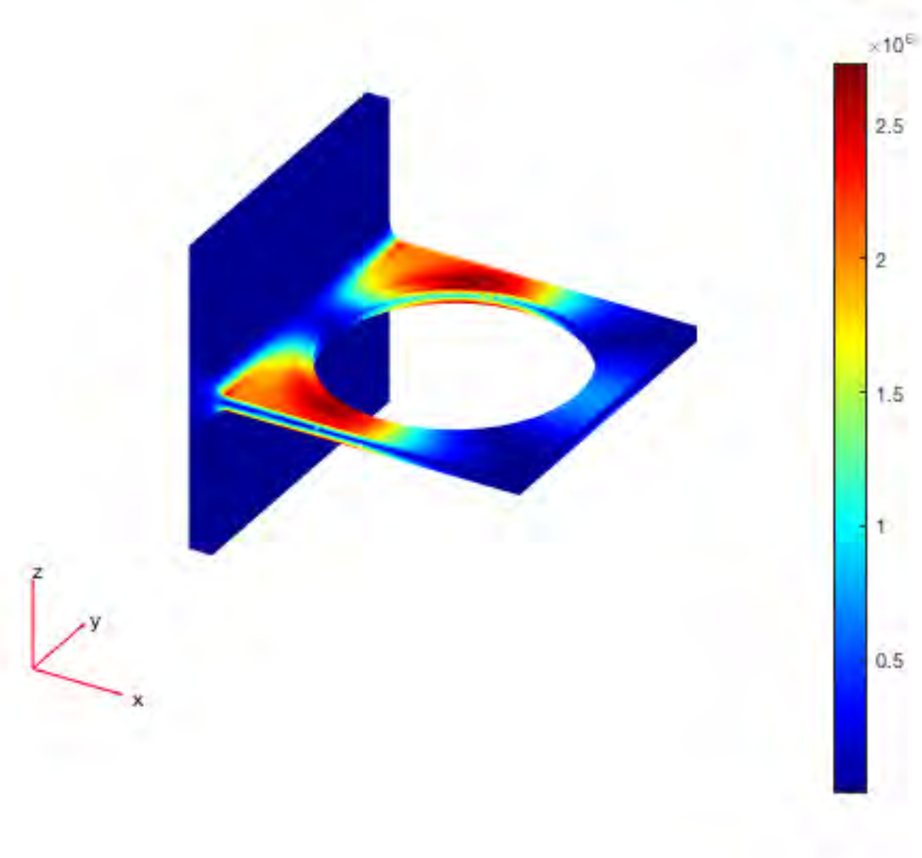
```
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Volume of SG is 801084.6513
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration   1: Warning: A value of class "logical" was indexed with no subscripts specified
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
```

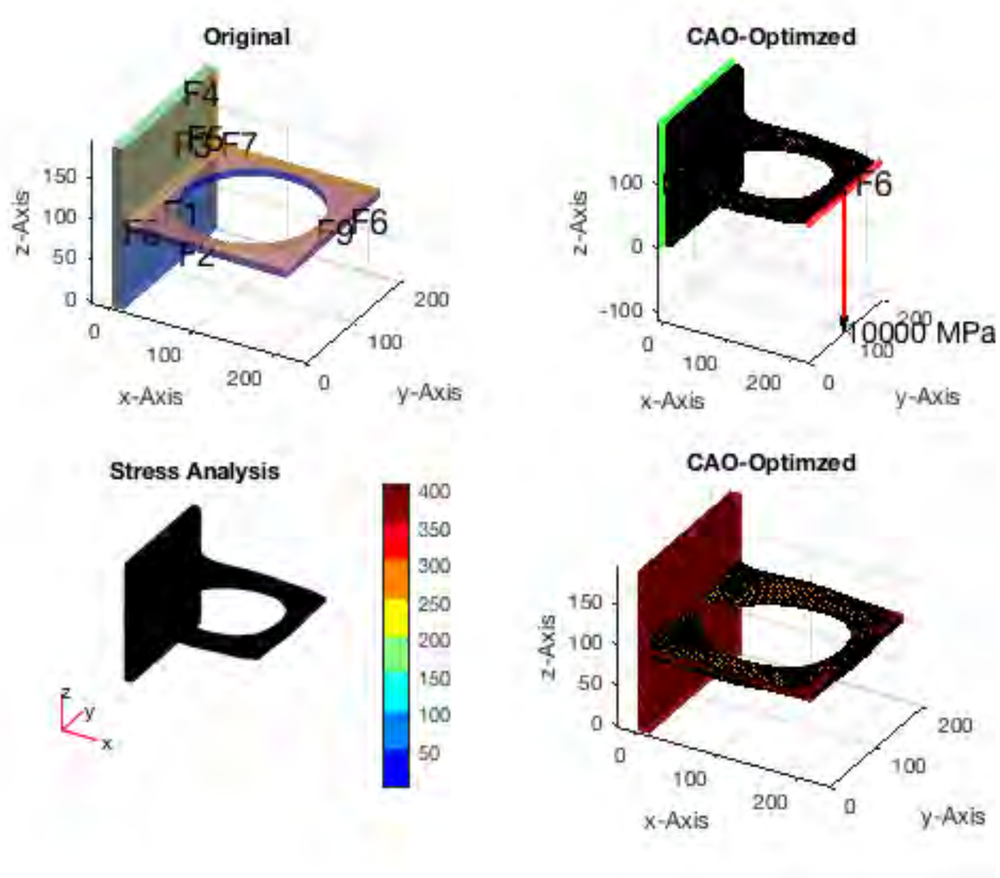
```
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Volume of SG is 789944.378
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration   2: Warning: A value of class "logical" was indexed with no subscripts specified
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
```

```
future release, it will be an error.
Volume of SG is 779041.5887
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration   3: Warning: A value of class "logical" was indexed with no subscripts specified
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Volume of SG is 770902.8223
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
```

```
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration   4: Warning: A value of class "logical" was indexed with no subscripts specified
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Volume of SG is 764095.7351
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```

```
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration   5: Warning: A value of class "logical" was indexed with no subscripts specified
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
CAO end: CAO process stops because meshing size does not fit.

******************** CAO result *********************
Original volume: 801084.6513 mm^3
Optimized volume: 764095.7351 mm^3
Original maximal von Mises stress: 1679.0079 N/mm^2
Optimized maximal von Mises stress: 409.3249 N/mm^2
9 Feature Surfaces found! Only the largest 99.90% (39.433 .. 39433.2mm^2), i.e. 9 of 9 are
shown.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```





4.3 CAO Optimization using load face 5

SGshapeOptiCAO(A,'FixedFaceIndices',3,'LoadFaceIndices',5,'Load',[0 0 -1e4]);

4.4 CAO Optimization using load face 1

SGshapeOptiCAO(A,'FixedFaceIndices',3,'LoadFaceIndices',1,'Load',[0 0 -1e4]);

4.5 CAO Optimization of a simple bar

```
A=SGbox([100,40,40])
[B,result,model]=SGshapeOptiCAO(A,'FixedFaceIndices',4,'LoadFaceIndices',3,'Load',[0 0 -1e4
]);
SGplot4(B,'m');
subplot(2,2,3); SGplotsurfaceload(A,'FixedFaceIndices',4,'LoadFaceIndices',3,'Load',[0 0 -1
e4]);
```

```
A =

  struct with fields:

    VL: [8×3 double]
    FL: [12×3 double]

Iteration   0: Warning: A value of class "logical" was indexed with no subscripts specified
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Volume of SG is 160000
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
```

```
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration   1: Warning: A value of class "logical" was indexed with no subscripts specified
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Volume of SG is 159744.8305
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
```

```
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration   2: Warning: A value of class "logical" was indexed with no subscripts specified
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Volume of SG is 159467.0401
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```

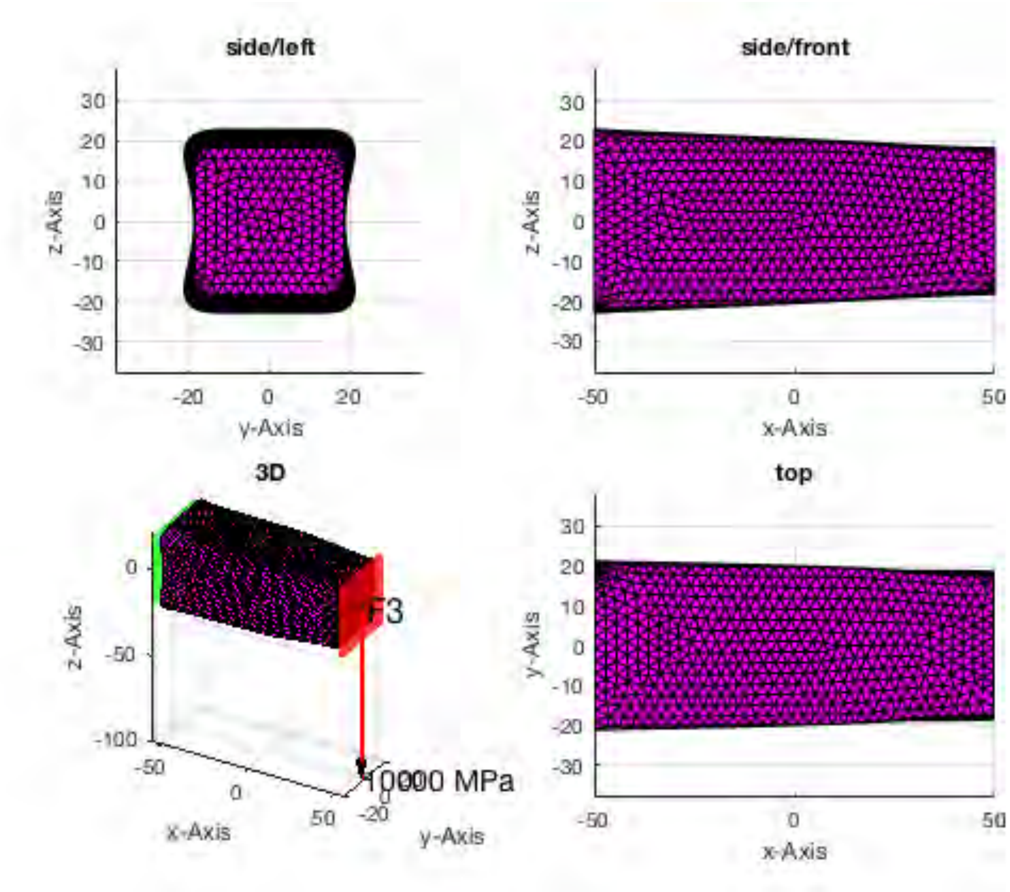
```
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration   3: Warning: A value of class "logical" was indexed with no subscripts specified
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Volume of SG is 159107.6043
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
```

```
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration   4: Warning: A value of class "logical" was indexed with no subscripts specified
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Volume of SG is 158752.6601
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
```

```
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration   5: Warning: A value of class "logical" was indexed with no subscripts specified
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Volume of SG is 158360.4781
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```

```
Iteration   6: Warning: A value of class "logical" was indexed with no subscripts specified
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Volume of SG is 158060.6989
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration   7: Warning: A value of class "logical" was indexed with no subscripts specified
.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```

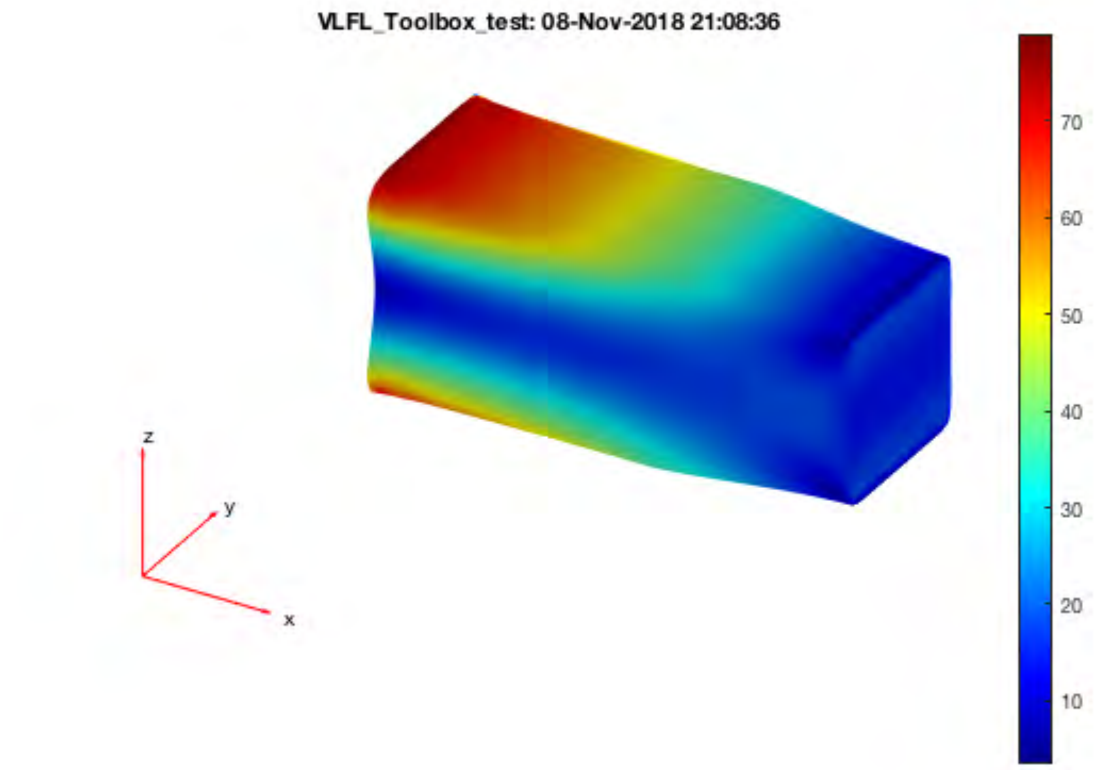
```
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
CAO end: CAO process stops because meshing size does not fit.

******************** CAO result *********************
Original volume: 160000 mm^3
Optimized volume: 158060.6989 mm^3
Original maximal von Mises stress: 110.75 N/mm^2
Optimized maximal von Mises stress: 76.6522 N/mm^2
```



4.6 Show the stress distribution in the CAO optimized shape

```
SGfigure; pdestressstatic(model,result);
view(30,30);
```

```
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```



Final Remarks

```
close all
VLFLlicense
```

```
Warning: Error creating or updating Patch
 Error in value of property  <a
 href="matlab:helpUtils.reference.showPropertyHelp('matlab.graphics.primitive.Patch','FaceV
ertexCData');")">FaceVertexCData</a>
  Number of colors must equal number of vertices



This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 21:08:38!
Executed 08-Nov-2018 21:08:40 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
database_toolbox
image_toolbox
map_toolbox
matlab
pde_toolbox
robotics_system_toolbox
```

```
simmechanics
simscape
simulink
video_and_image_blockset
================================================================================
==========
```

Published with MATLAB® R2018a

Tutorial 43: Performing FEM Structural Optimization (CAO) and Topological Optimization (SKO) of Solids

2018-03-08: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de) - Last Change: 2018-03-08

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

Motivation for this tutorial: (Originally SolidGeometry 4.2 required)

Yinlun Sun of TU Munich has supplemented the SG-Library with functions that allow a structural and topological optimization of geometric bodies with surface representation.

List of function introduced in this tutorial

* * * * * * * * * * * * * *

```
% function VLFL_EXP43
% clear all; close all;
```

1. Conversion between triangle surface model and tetrahedon volumen model

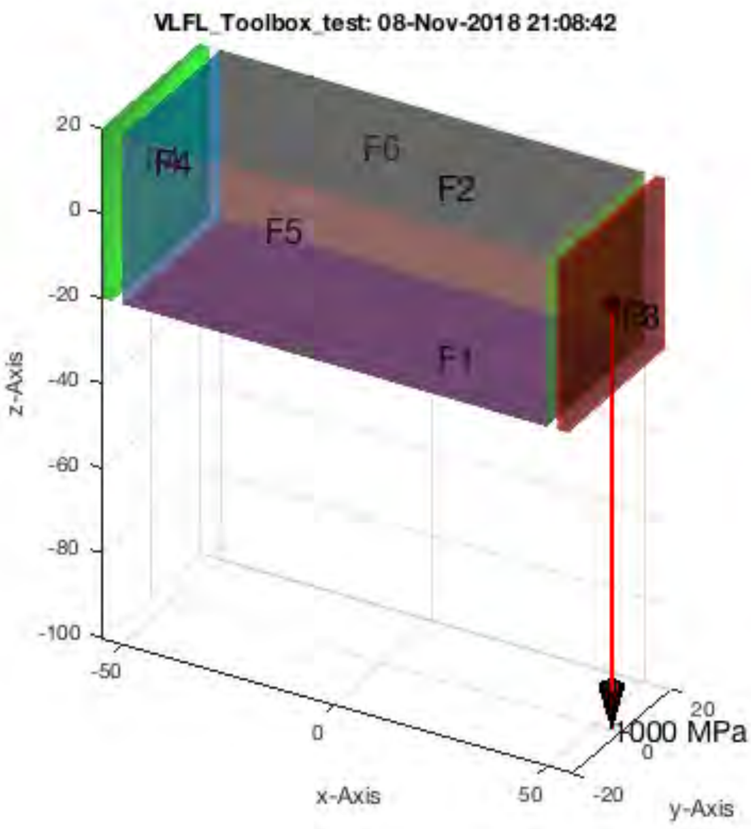
1.1 Create a simple bar type link

```
A=SGbox([100,40,40])
SGfigure; FSplot(A); view(30,30);
SGplotsurfaceload (A,'FixedFaceIndices',4,'LoadFaceIndices',3,'Load',[0 0 -1e3]);
```

```
A =

  struct with fields:
```

```
    VL: [8×3 double]
    FL: [12×3 double]
```

```
6 Feature Surfaces found! Only the largest 99.90% (4.000 .. 4000.0mm^2), i.e. 6 of 6 are sh
own.
```




```
SGshapeOptiSKO(A,'FixedFaceIndices',4,'LoadFaceIndices',3,'Load',[0 0 -1e3]); B=ans
```

```
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration 0: Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```

```
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Average stress: 2.95, Maximum stress: 10.61
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration 1: Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
```

```
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Average stress: 2.95, Maximum stress: 10.73
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration 2: Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
```

```
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Average stress: 2.95, Maximum stress: 10.85
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration 3: Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```

```
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Average stress: 2.94, Maximum stress: 10.96
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration 4: Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
```

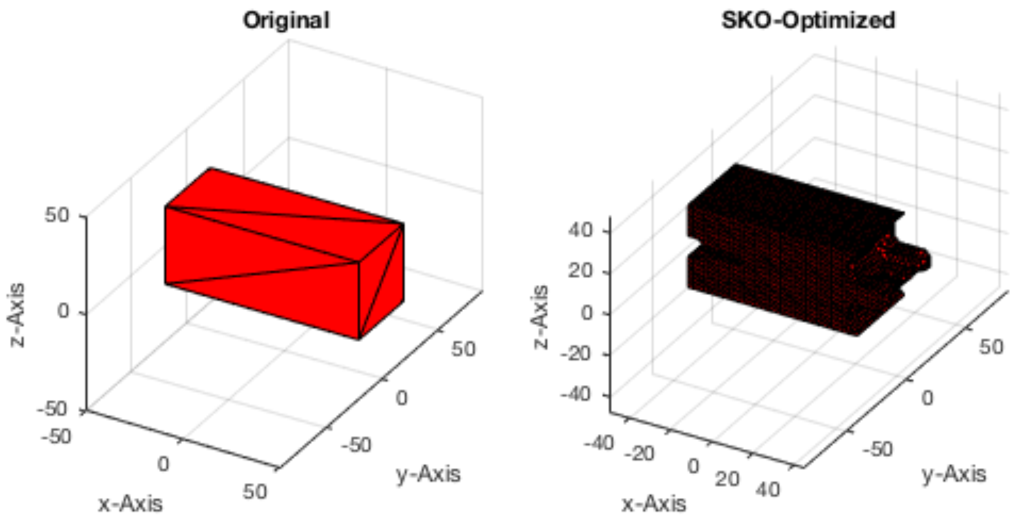
```
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Average stress: 2.94, Maximum stress: 11.08
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration 5: Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
```

```
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Average stress: 2.94, Maximum stress: 11.19
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration 6: Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
```

```
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Average stress: 2.94, Maximum stress: 11.30
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration 7: Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
```

```
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Average stress: 2.94, Maximum stress: 11.41
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Iteration 8: Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
```

```
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Warning: A value of class "logical" was indexed with no subscripts specified.
Currently the result of this operation is the indexed value itself, but in a
future release, it will be an error.
Average stress: 2.94, Maximum stress: 11.52

B =

  struct with fields:

      VL: [3034×3 double]
      FL: [6054×3 double]
    pcon: 0.8000
```



Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 21:10:01!
Executed 08-Nov-2018 21:10:03 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
database_toolbox
image_toolbox
map_toolbox
matlab
pde_toolbox
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
===========================================================================================
==========
```

Published with MATLAB® R2018a

Tutorial 44: Creation of solids and kinematics from 3D curves and transformation matrices

2018-07-24: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de) - Last Change: 2018-07-24

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

Motivation for this tutorial: (Originally SolidGeometry 4.2 required)

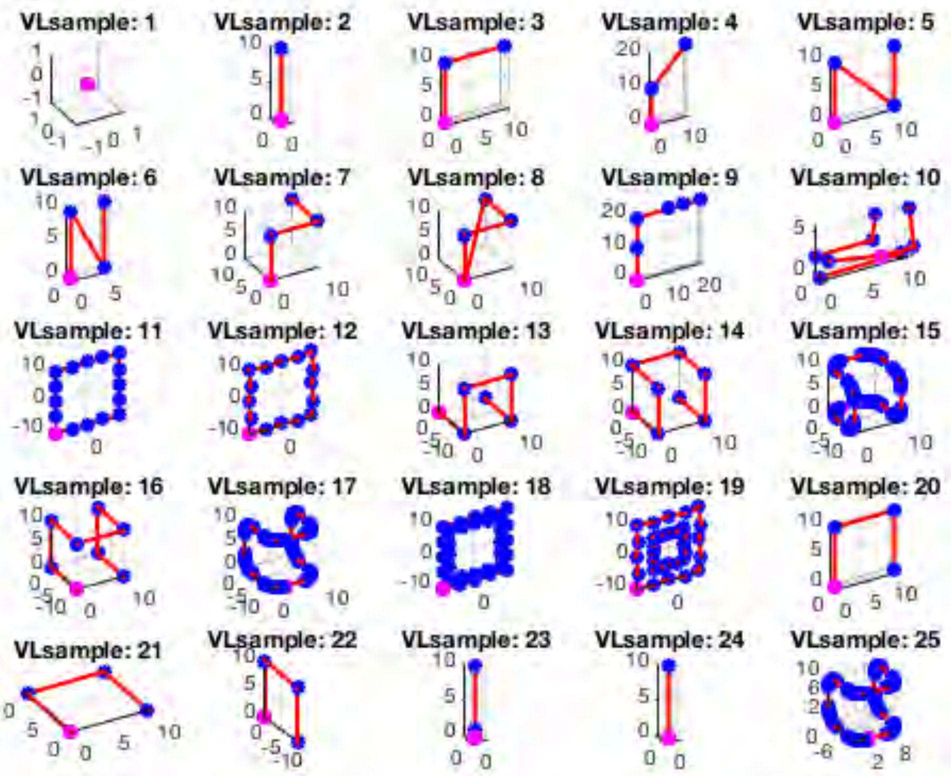
The creation of solids from space curves and a cross-section polygon is not trivial. The affected persons in the SGLib are VLsample - for generating example curves VLedgeNormal - Non trivial function for creating normal orthogonal vectors The creation of solids from space curves and a cross-section polygon is not trivial. The affected functions in the SGLib are VLsample - for generating example curves VLedgeNormal - Non-trivial function for generating normal orthogonal vectors SGcontourtube - The first function with rotating matrices (error in special cases) SGcontourtube2 - The new function with VLedgeNormal (previously error-free) SGofCPLCVVLR - Now based on SGcontourtube2 SGof2T - Now based on SGcontourtube2 SGTofDenavitHartenberg - Based on SGof2T SGTofDHset - Based on SGTofDenavitHartenberg

List of function introduced in this tutorial

* * * * * * * * * * * * * * *

Using VLsample to create example funktions

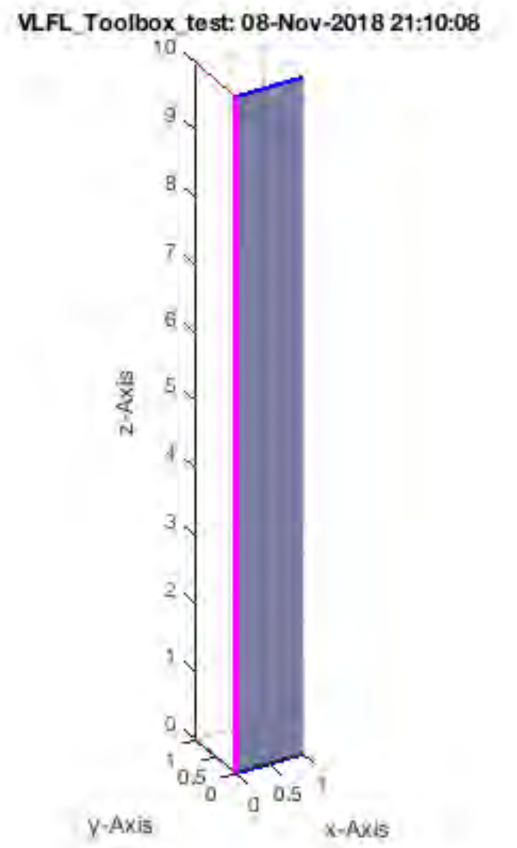
```
VLsample;
```



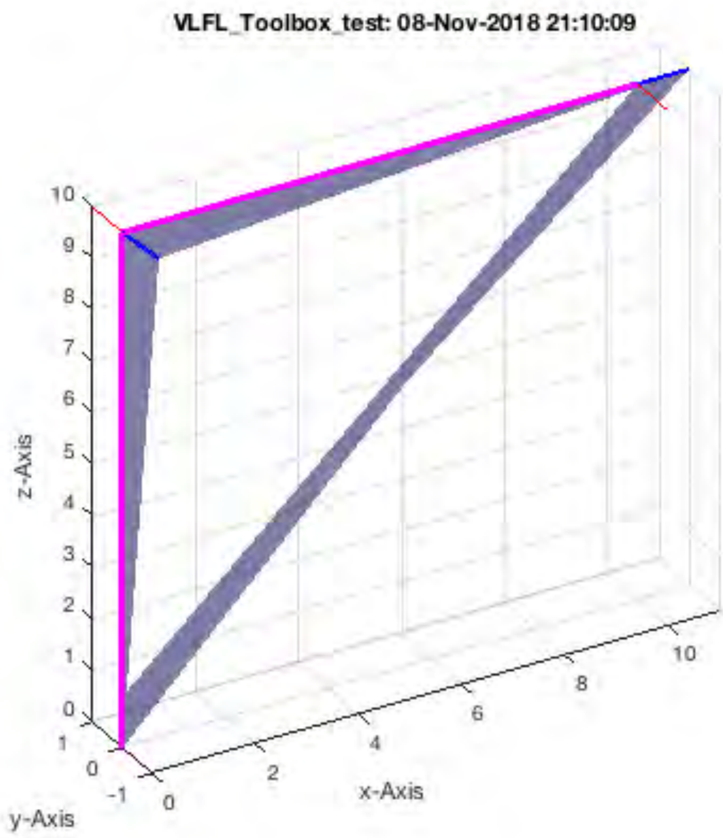
Creating edge normal function for an open spatial curve

If angles are larger than 90 degree (pi/2)

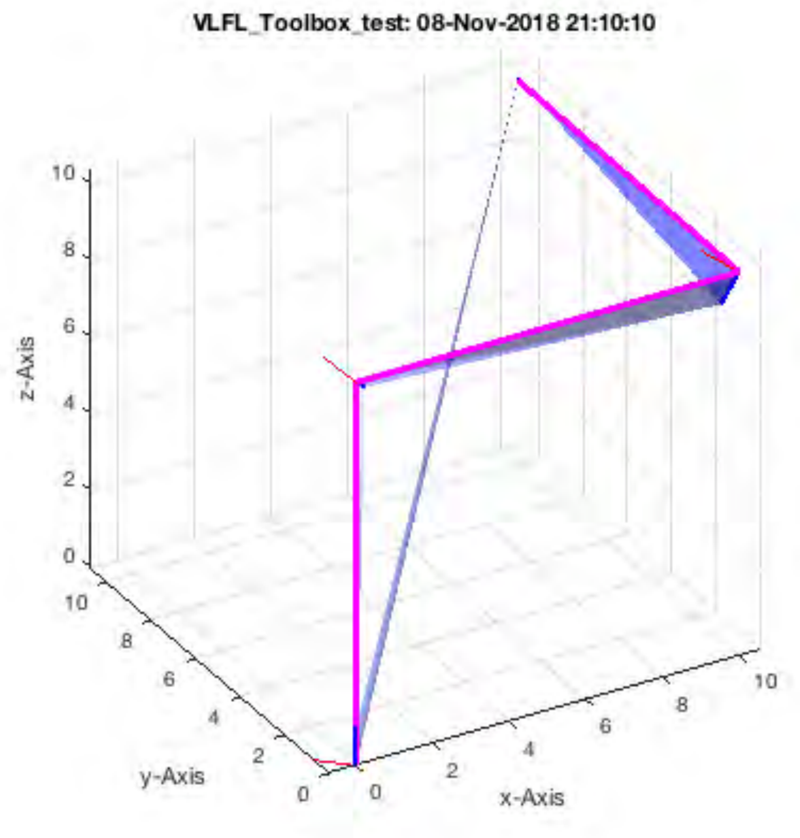
```
VLedgeNormal(VLsample(2));
```



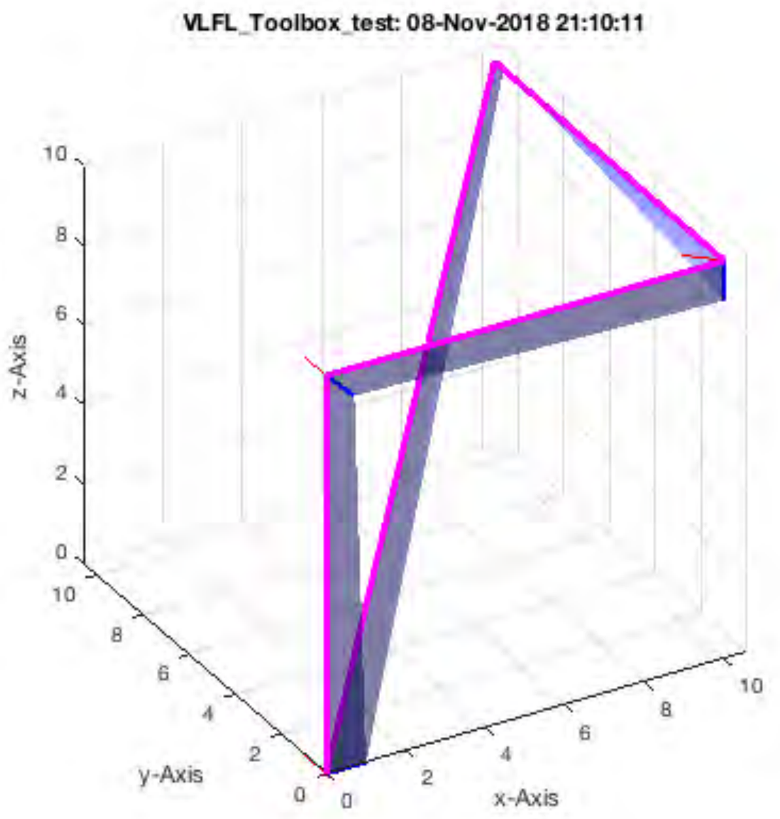

```
VLedgeNormal(VLsample(3));
```



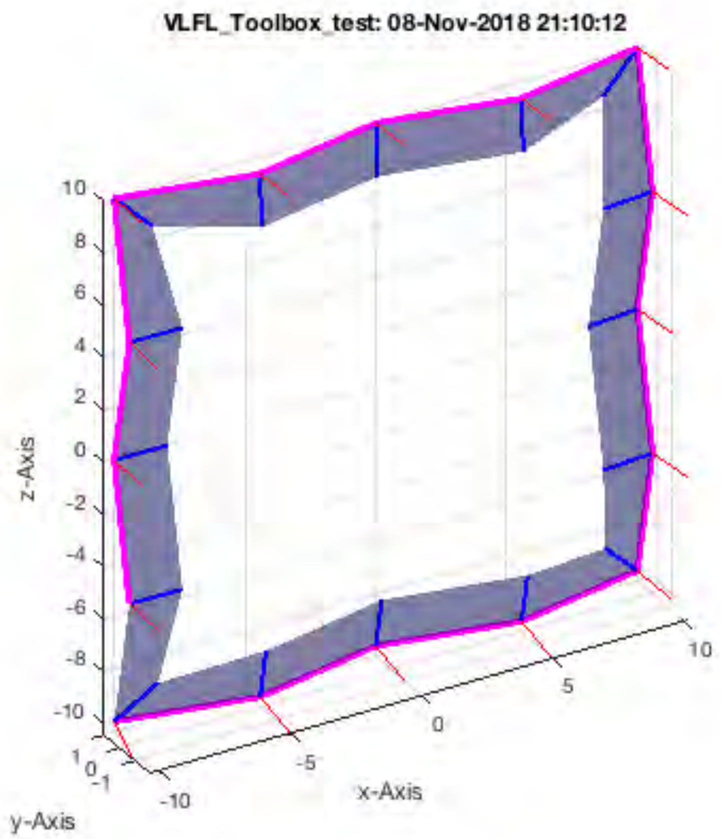

```
VLedgeNormal(VLsample(7));
```



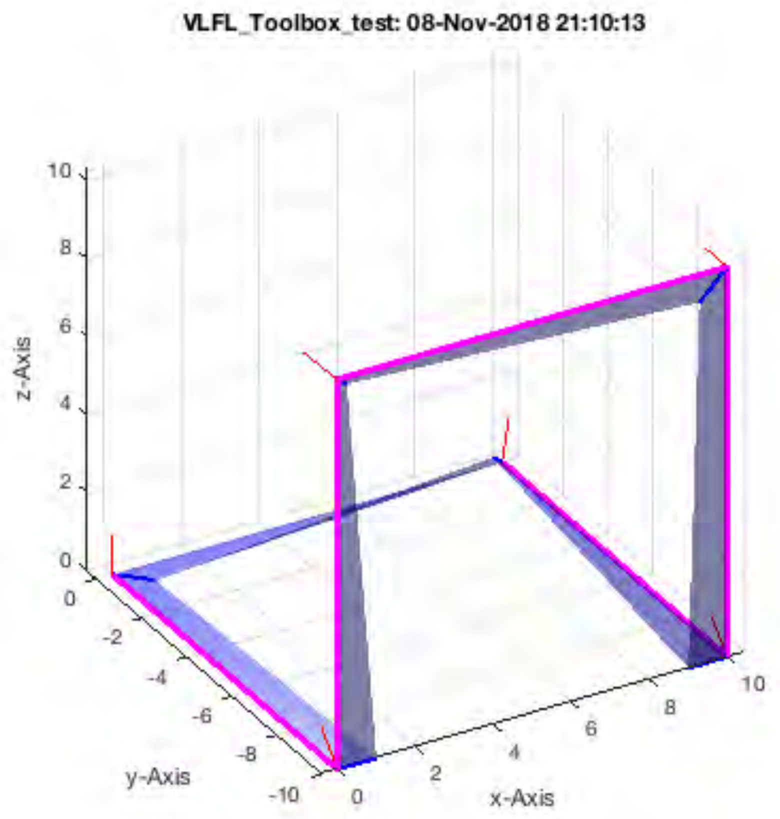

```
VLedgeNormal(VLsample(8));
```



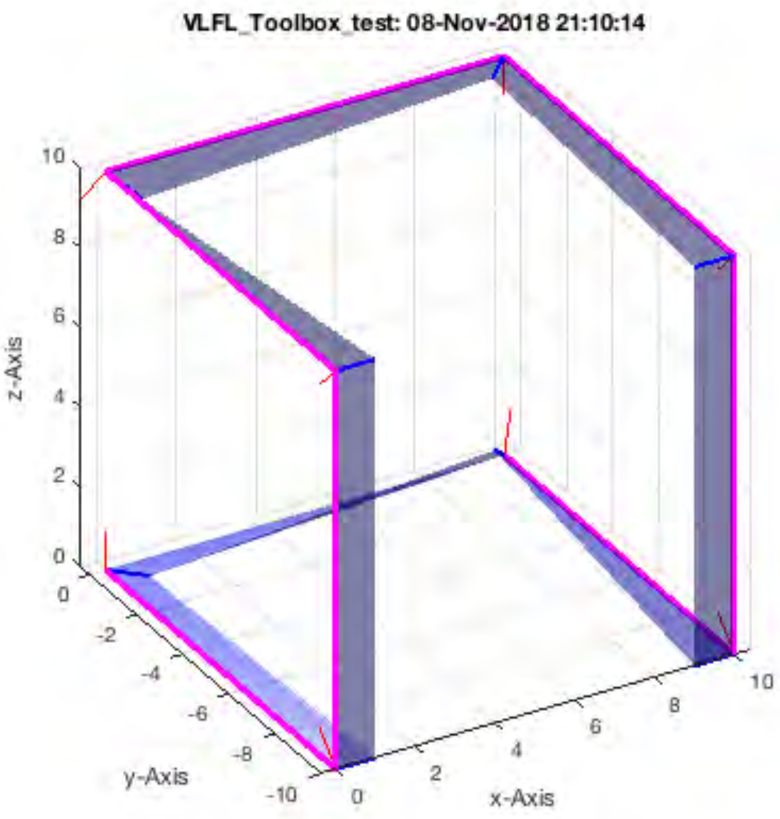

```
VLedgeNormal(VLsample(12));
```



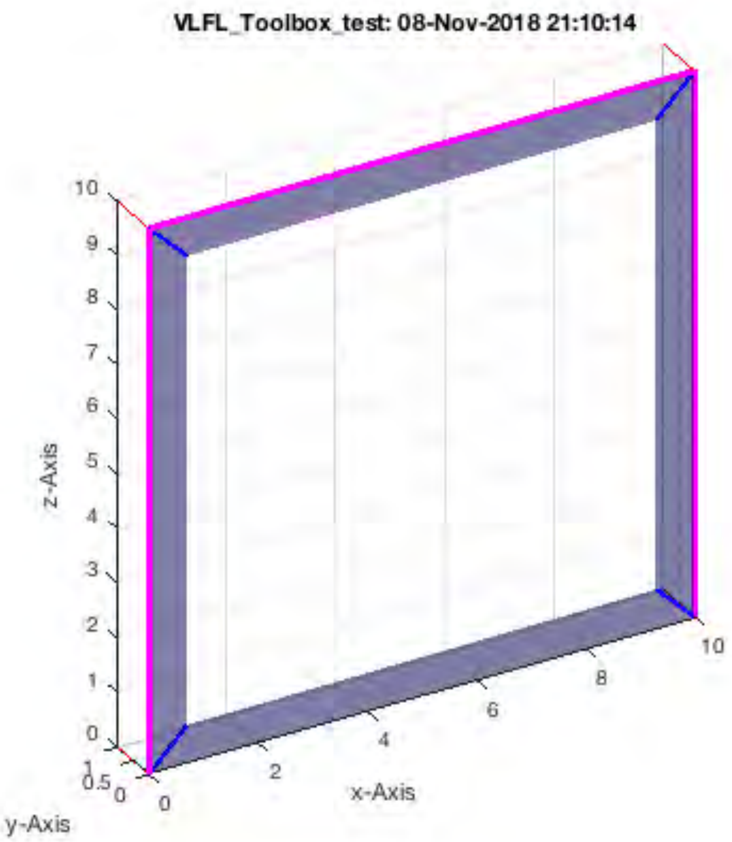

```
VLedgeNormal(VLsample(13));
```



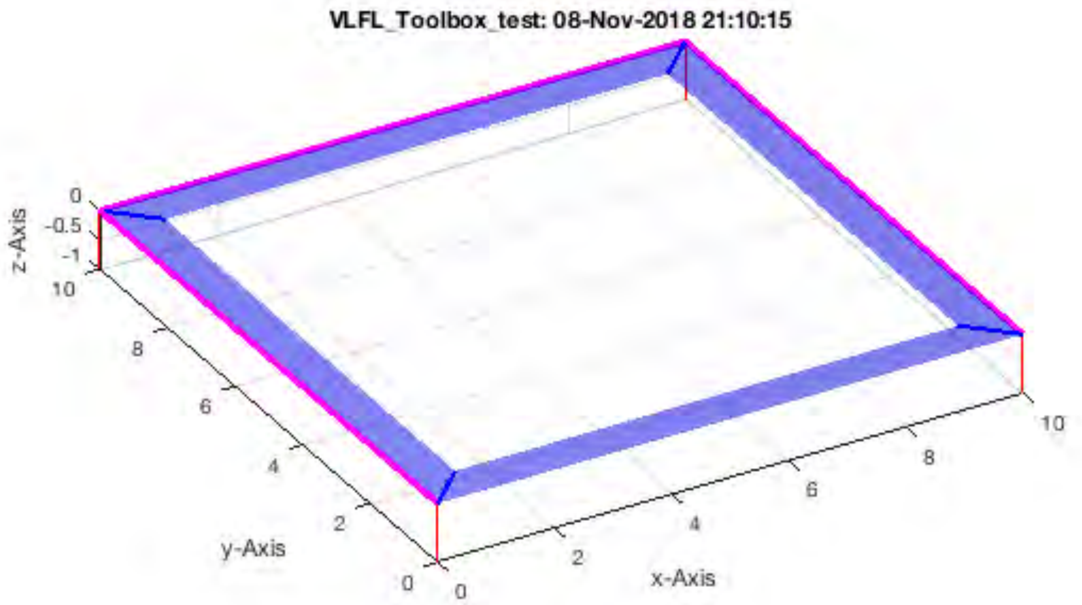

```
VLedgeNormal(VLsample(14));
```



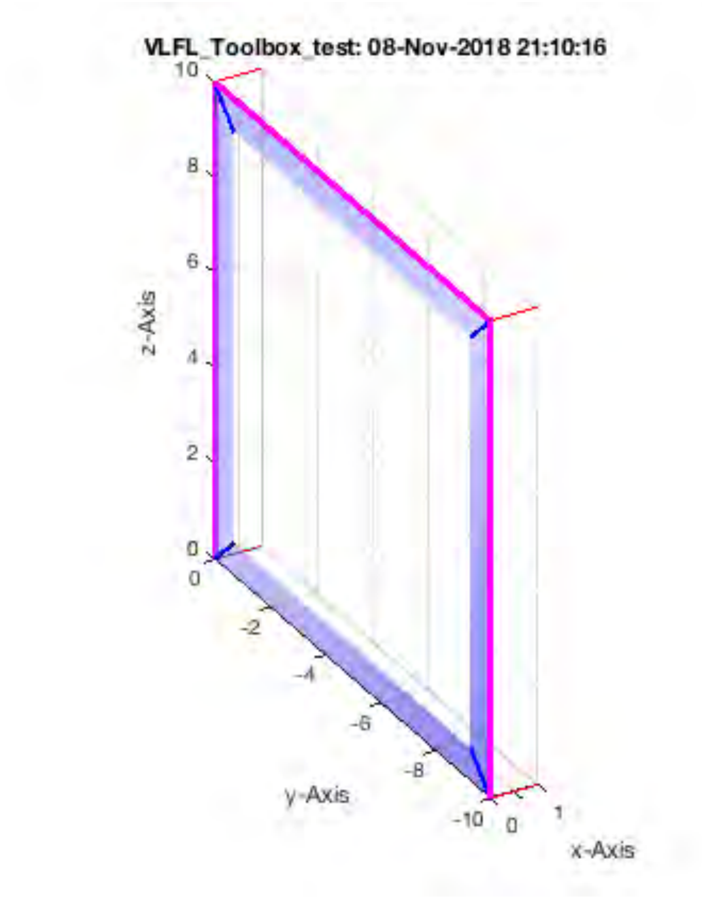

```
VLedgeNormal(VLsample(20));
```




```
VLedgeNormal(VLsample(21));
```



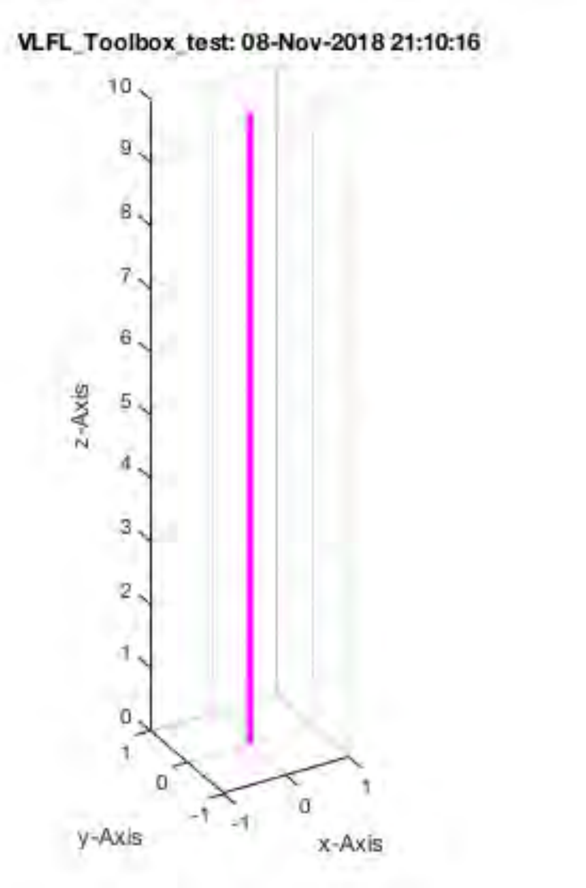

```
VLedgeNormal(VLsample(22));
```



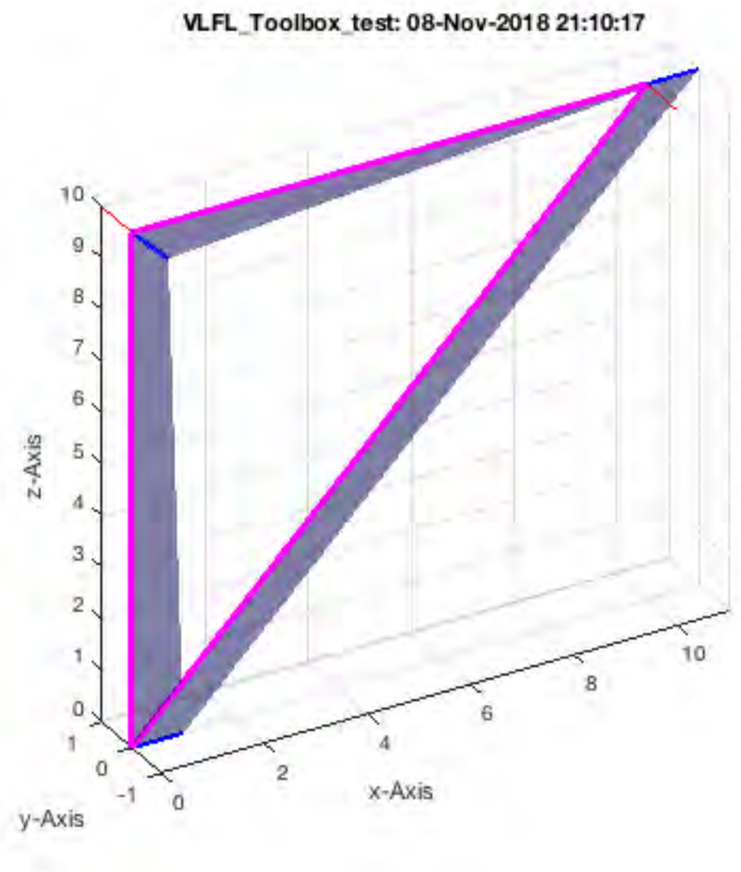
Creating normal function for a closed spatial curve

If angles are larger than 90 degree (pi/2)

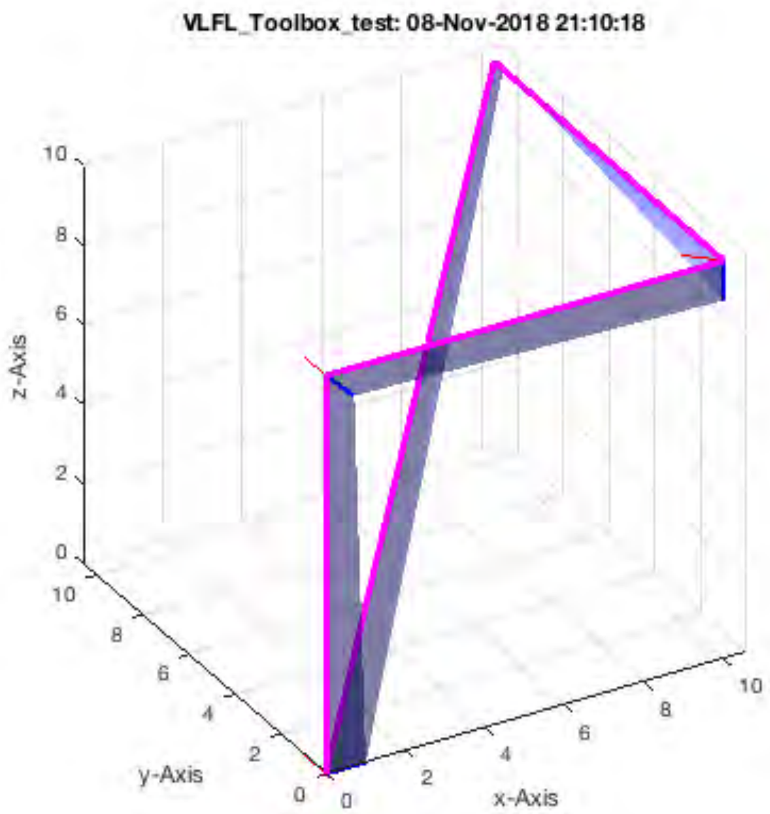
```
VLedgeNormal(CVLofVL(VLsample(2)));
```



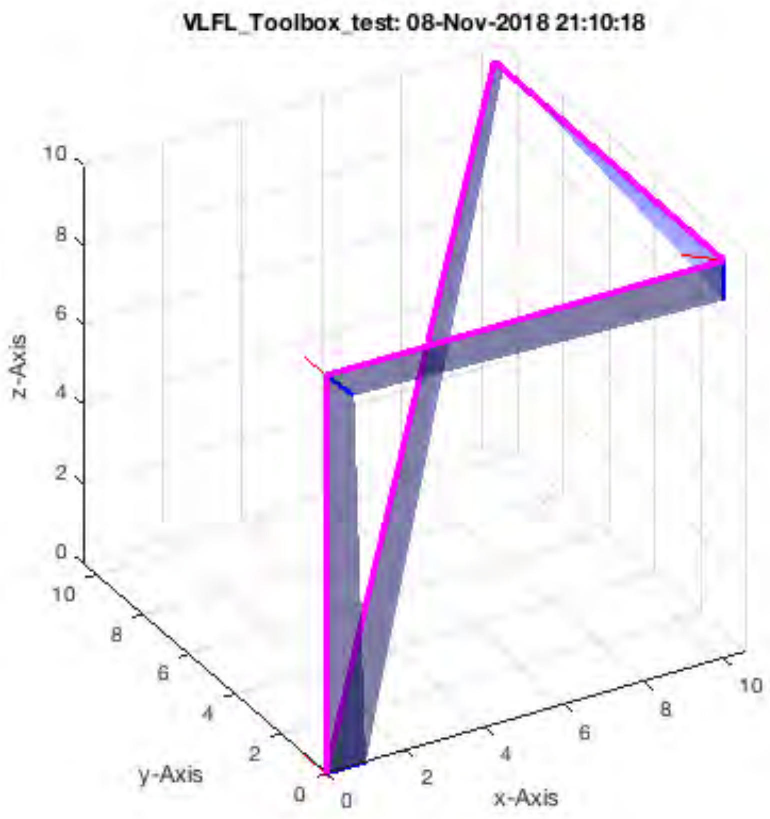

```
VLedgeNormal(CVLofVL(VLsample(3)));
```



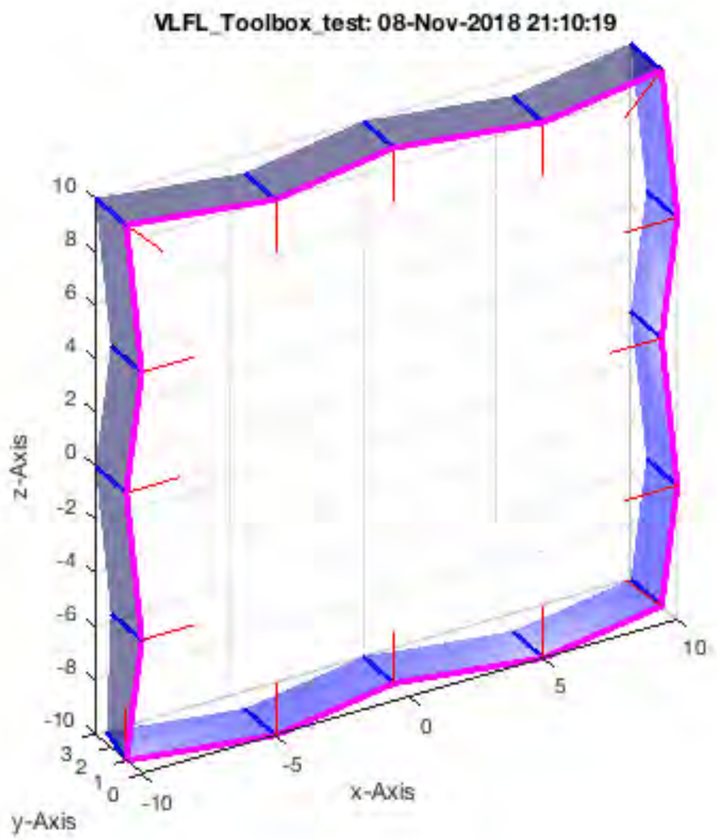

```
VLedgeNormal(CVLofVL(VLsample(7)));
```



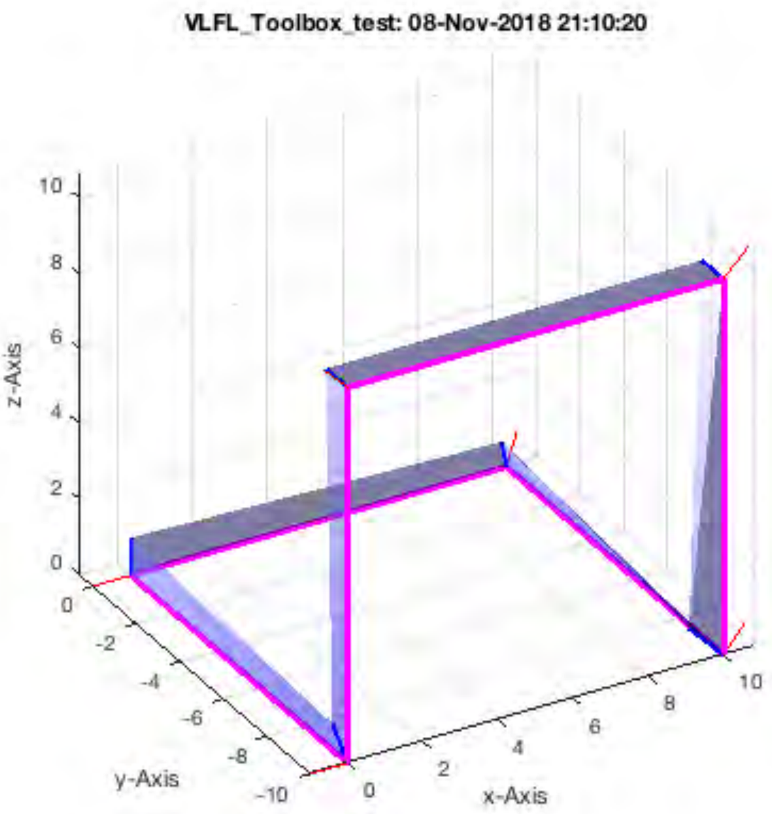

```
VLedgeNormal(CVLofVL(VLsample(8)));
```



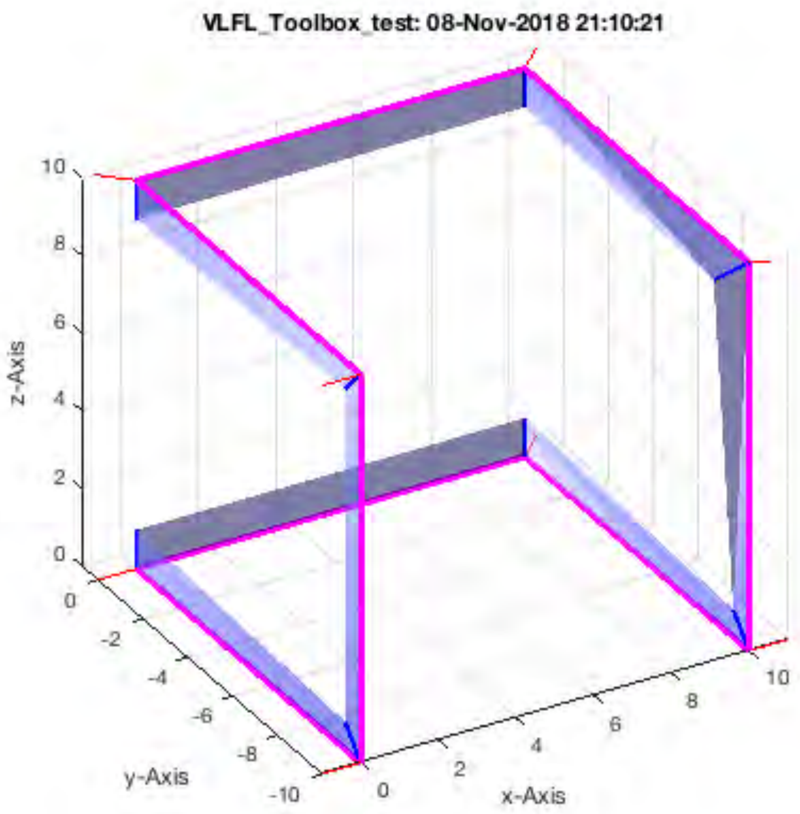

```
VLedgeNormal(CVLofVL(VLsample(12)));
```



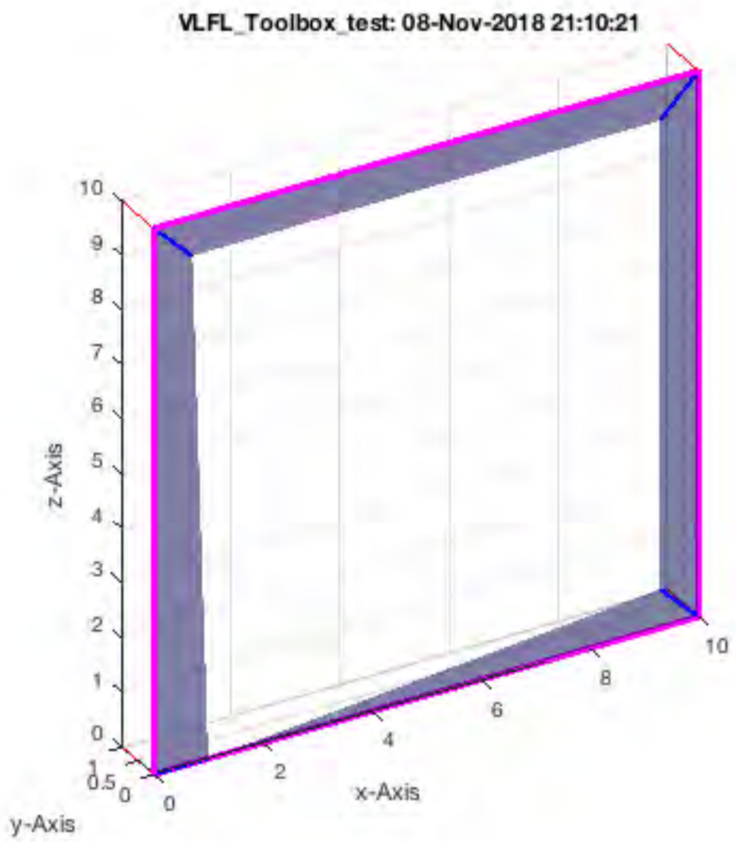

```
VLedgeNormal(CVLofVL(VLsample(13)));
```



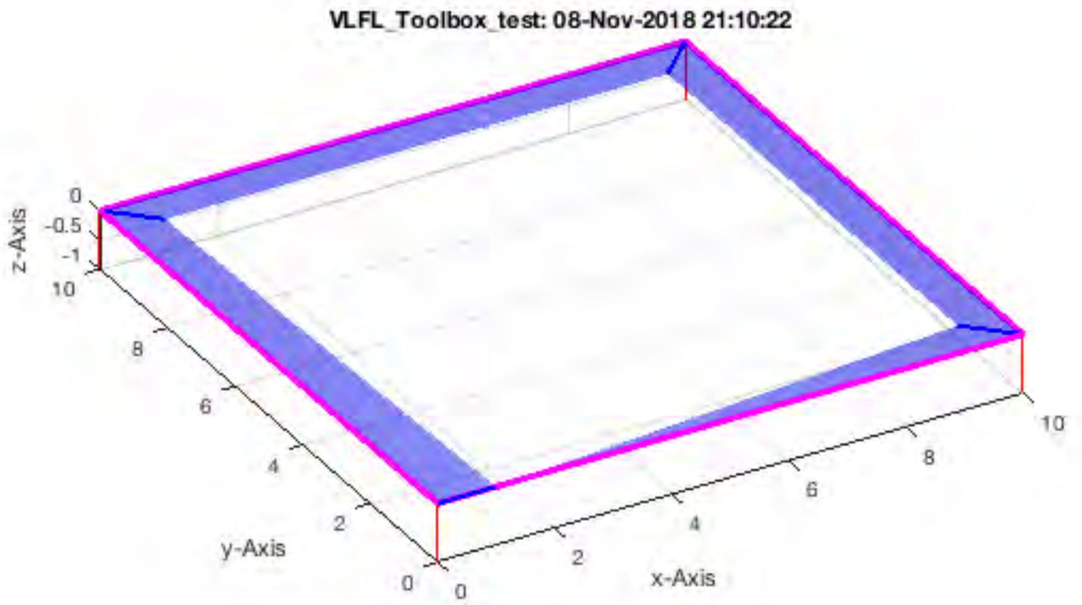

```
VLedgeNormal(CVLofVL(VLsample(14)));
```



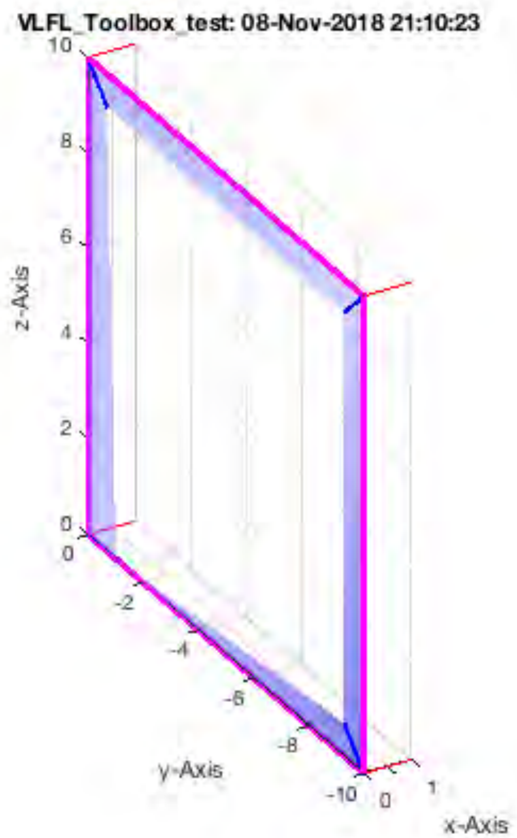

```
VLedgeNormal(CVLofVL(VLsample(20)));
```




```
VLedgeNormal(CVLofVL(VLsample(21)));
```



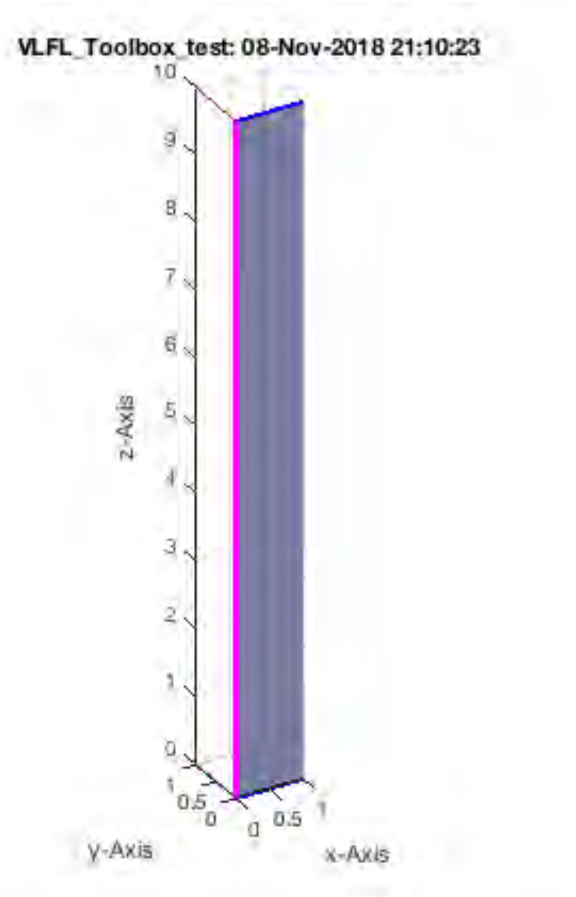

```
VLedgeNormal(CVLofVL(VLsample(22)));
```



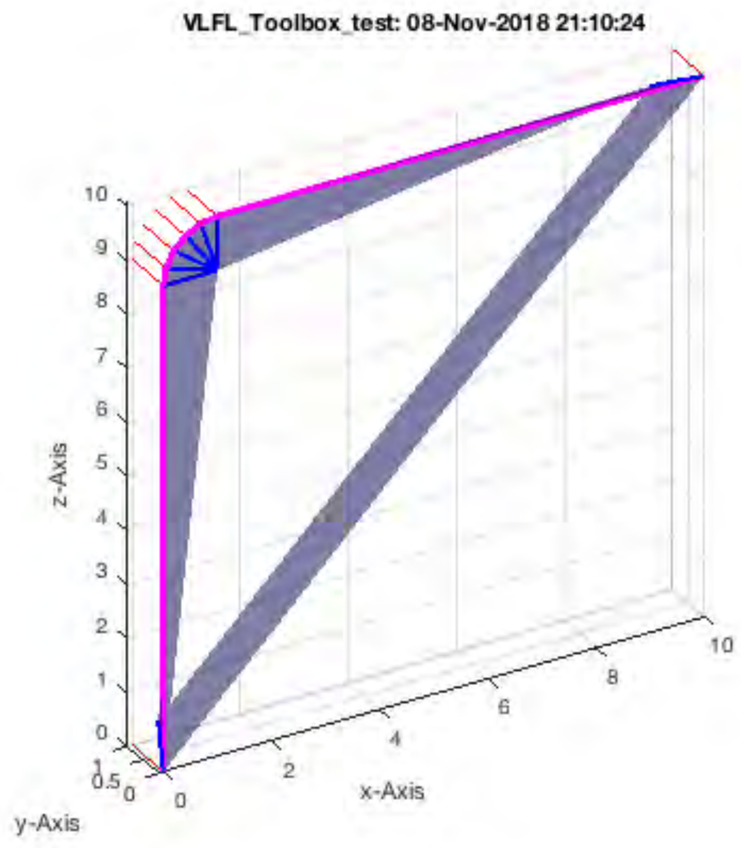
Creating normal function for open spatial radial curve

If angles are larger than 90 degree (pi/2)

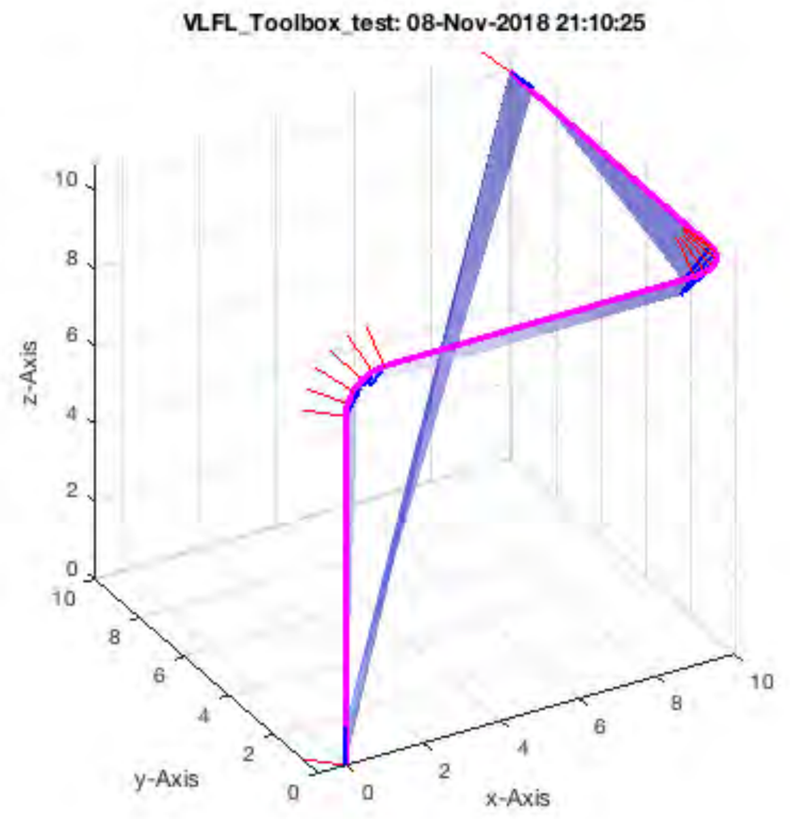
```
VLedgeNormal(VLradialEdges(VLsample(2)));
```



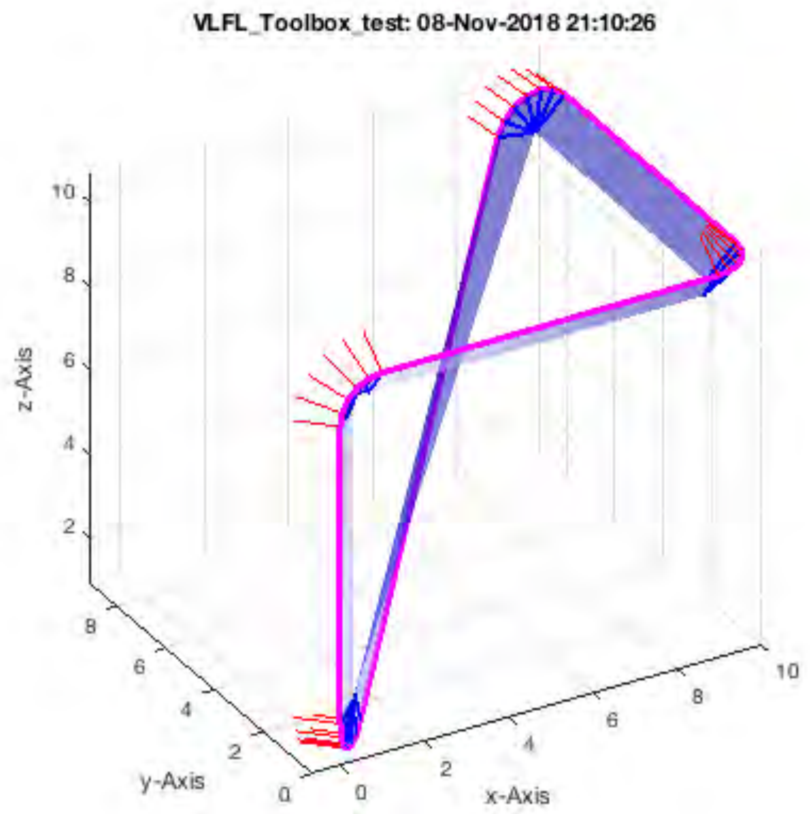

```
VLedgeNormal(VLradialEdges(VLsample(3)));
```



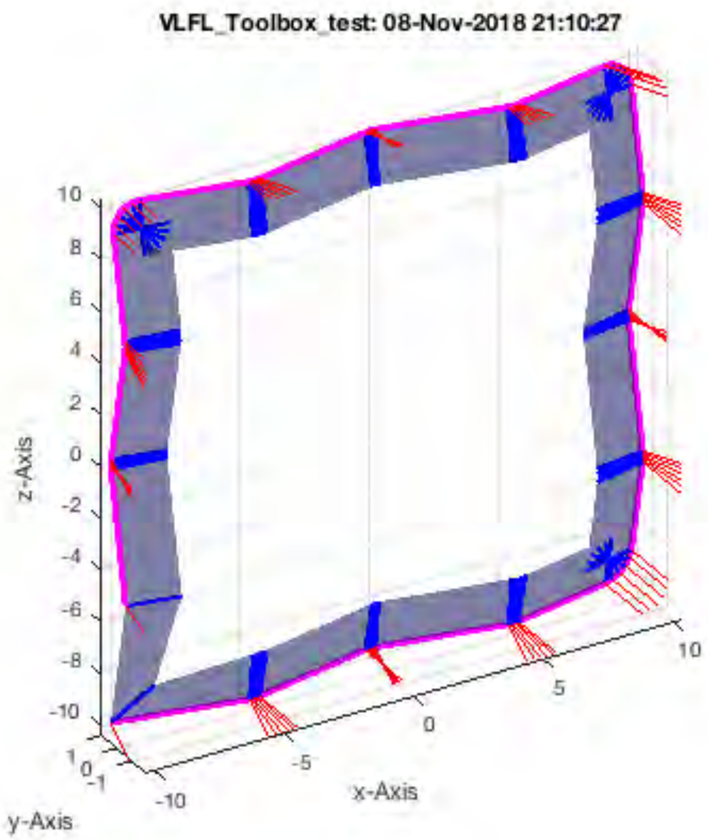

```
VLedgeNormal(VLradialEdges(VLsample(7)));
```



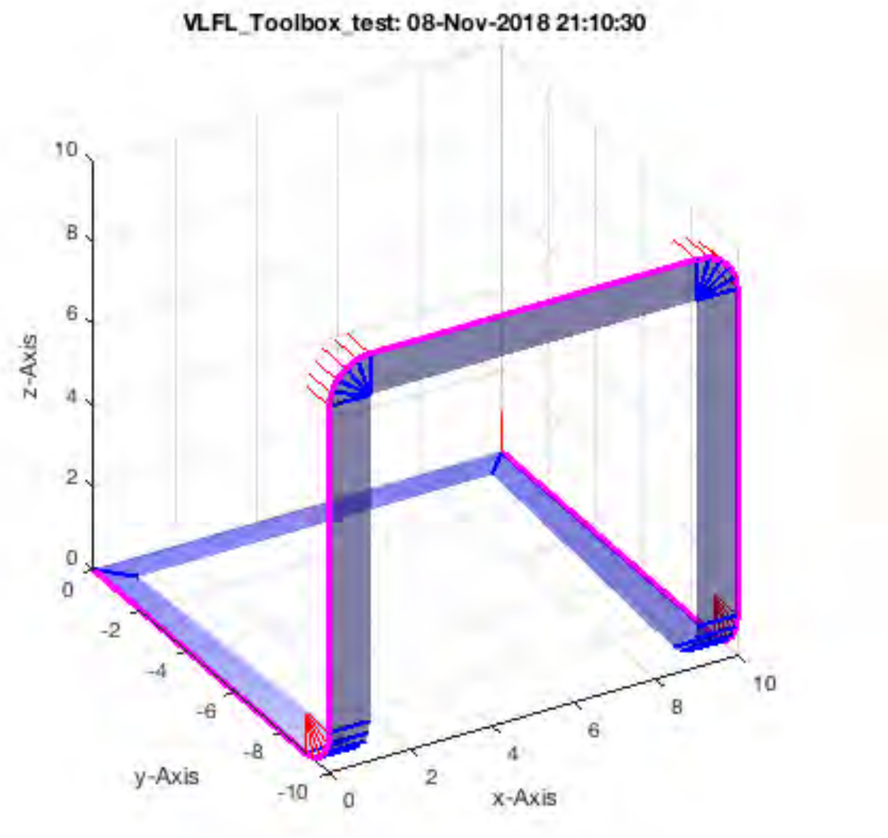

```
VLedgeNormal(VLradialEdges(VLsample(8)));
```



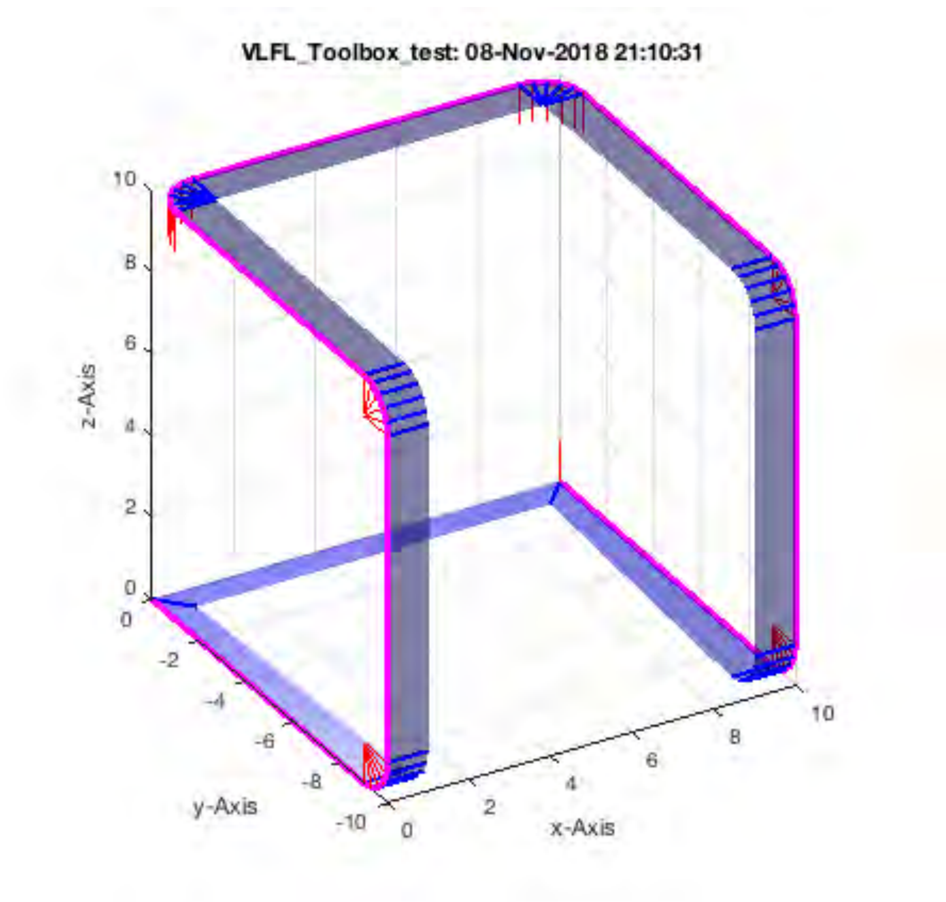

```
VLedgeNormal(VLradialEdges(VLsample(12)));
```



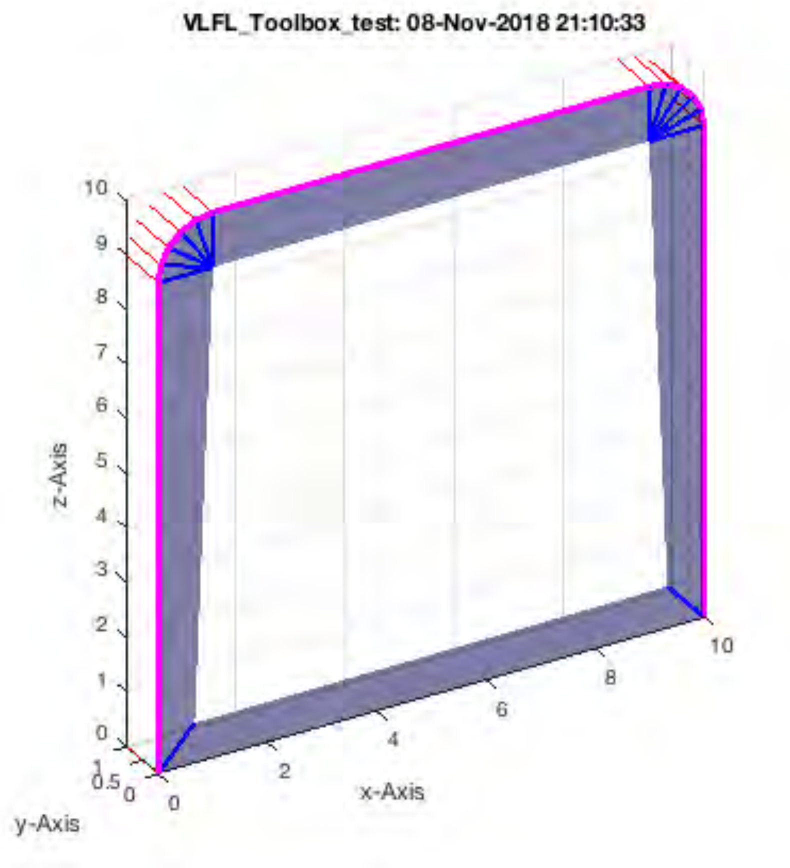

```
VLedgeNormal(VLradialEdges(VLsample(13)));
```



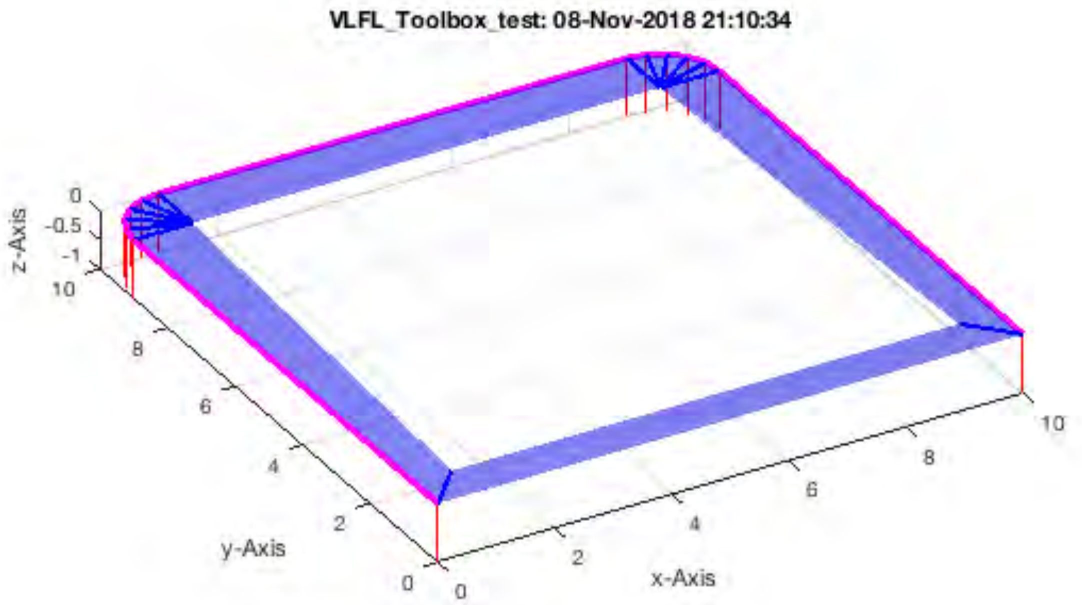

```
VLedgeNormal(VLradialEdges(VLsample(14)));
```



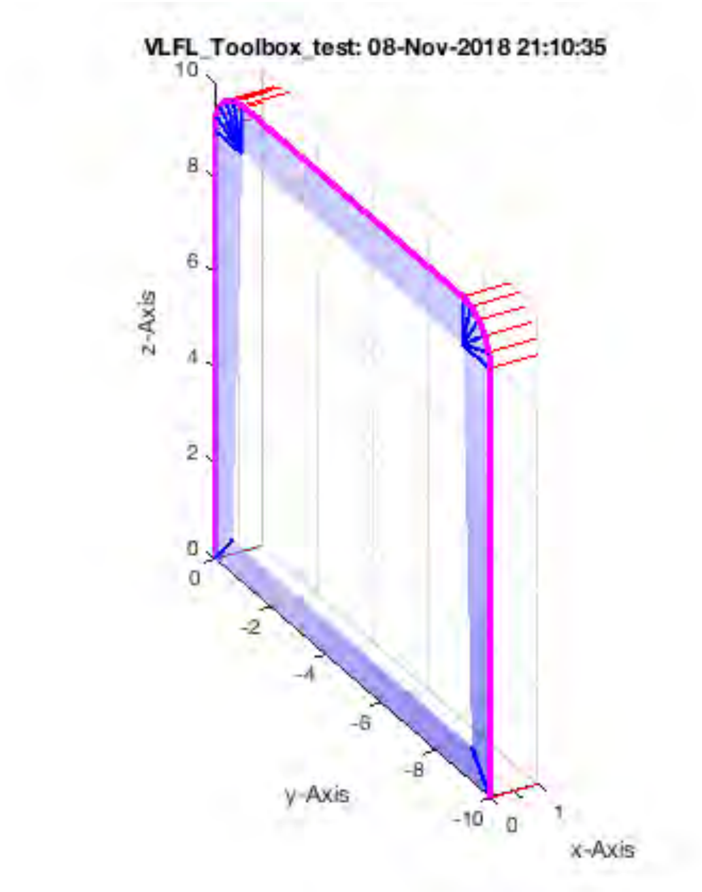

```
VLedgeNormal(VLradialEdges(VLsample(20)));
```




```
VLedgeNormal(VLradialEdges(VLsample(21)));
```



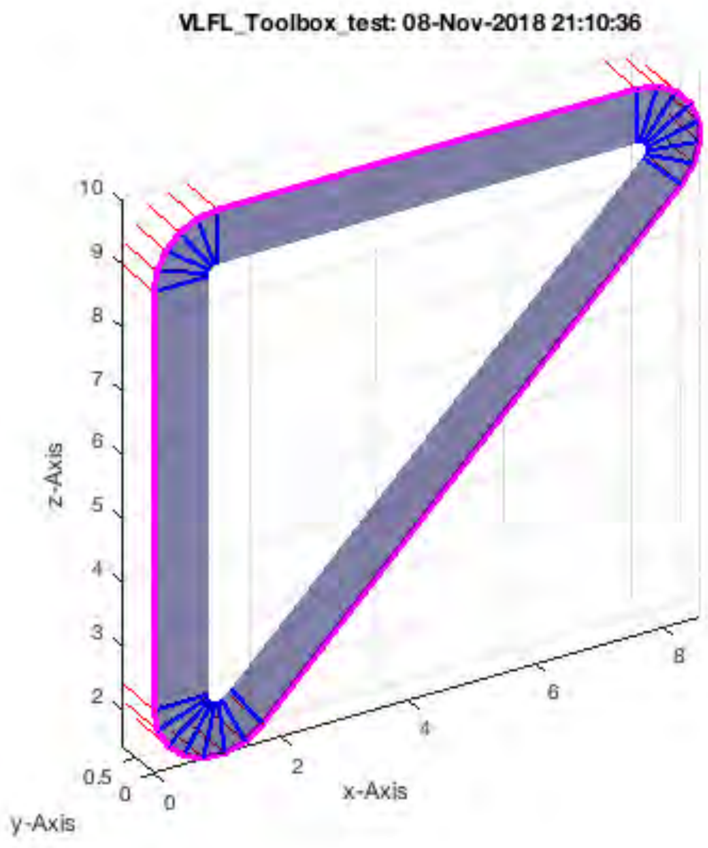

```
VLedgeNormal(VLradialEdges(VLsample(22)));
```



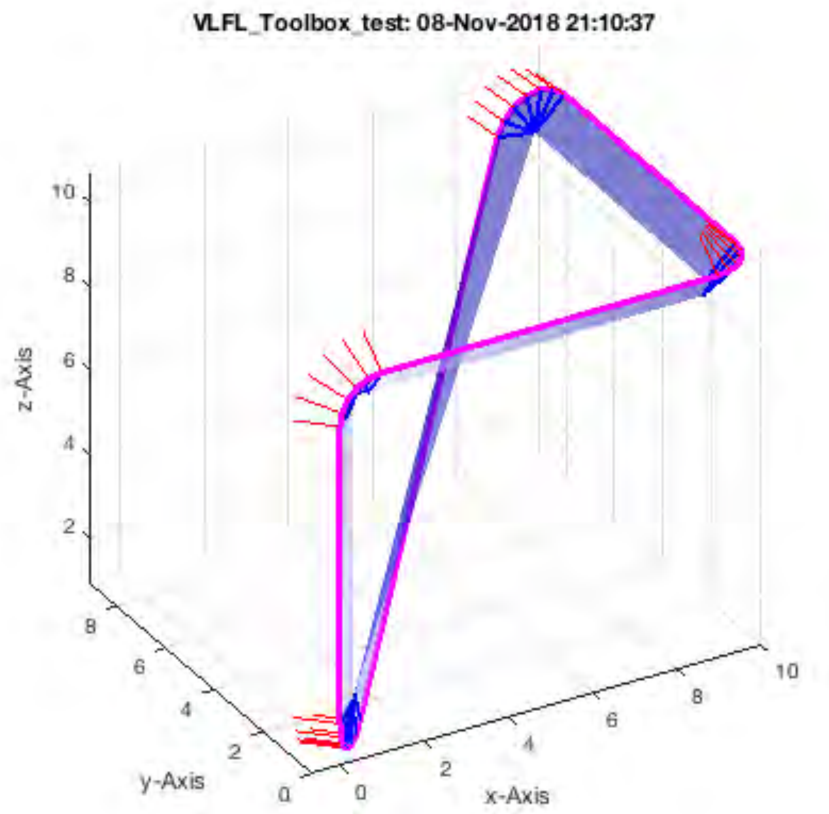
Creating normal function for closed spatial radial curve

If angles are larger than 90 degree (pi/2)

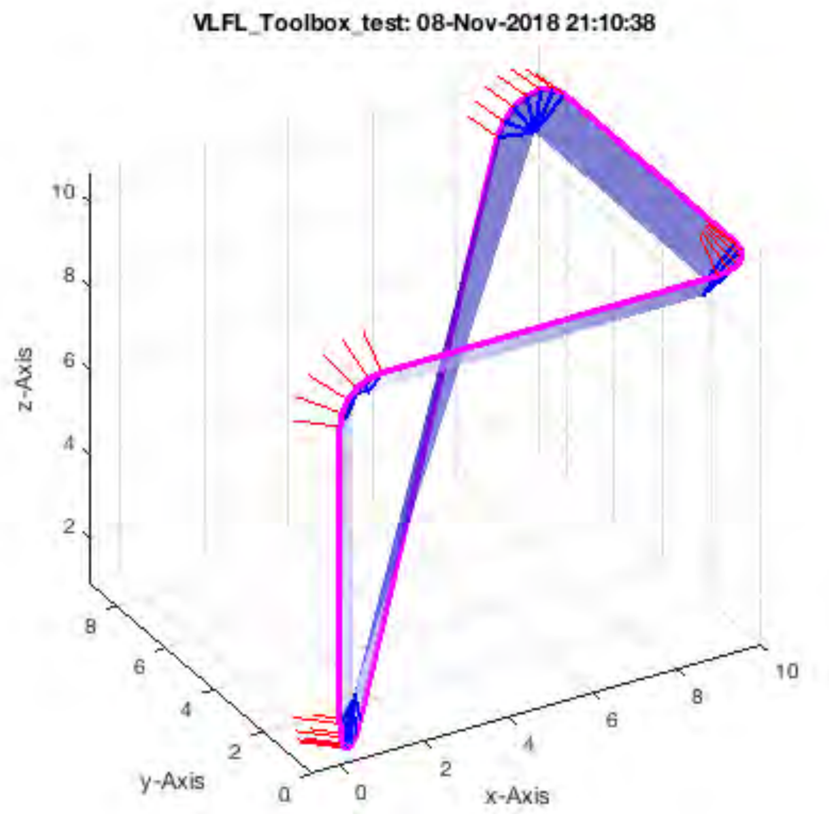
```
VLedgeNormal(VLradialEdges(CVLofVL(VLsample(3))));
```



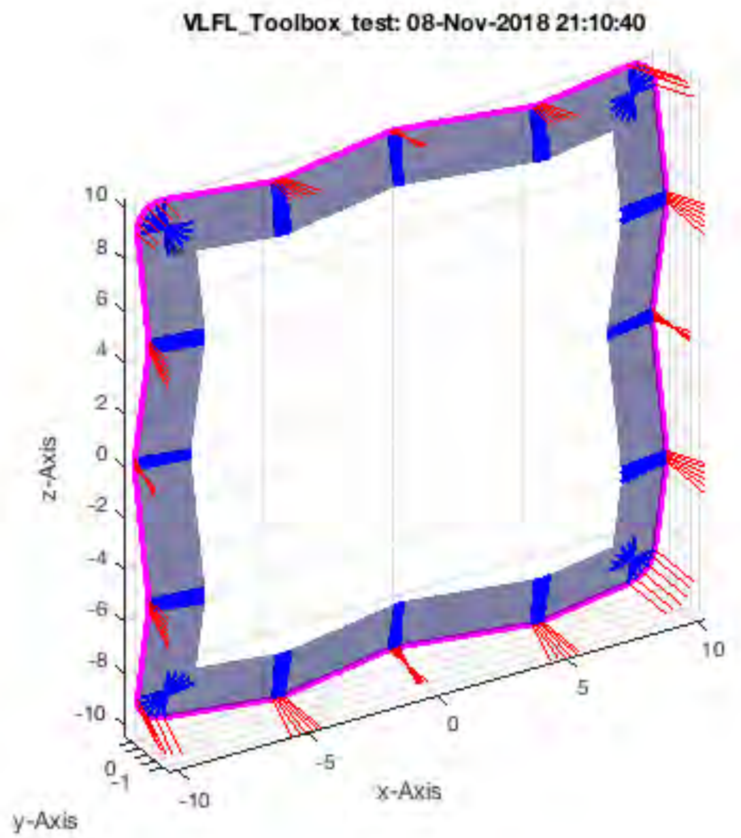

```
VLedgeNormal(VLradialEdges(CVLofVL(VLsample(7))));
```



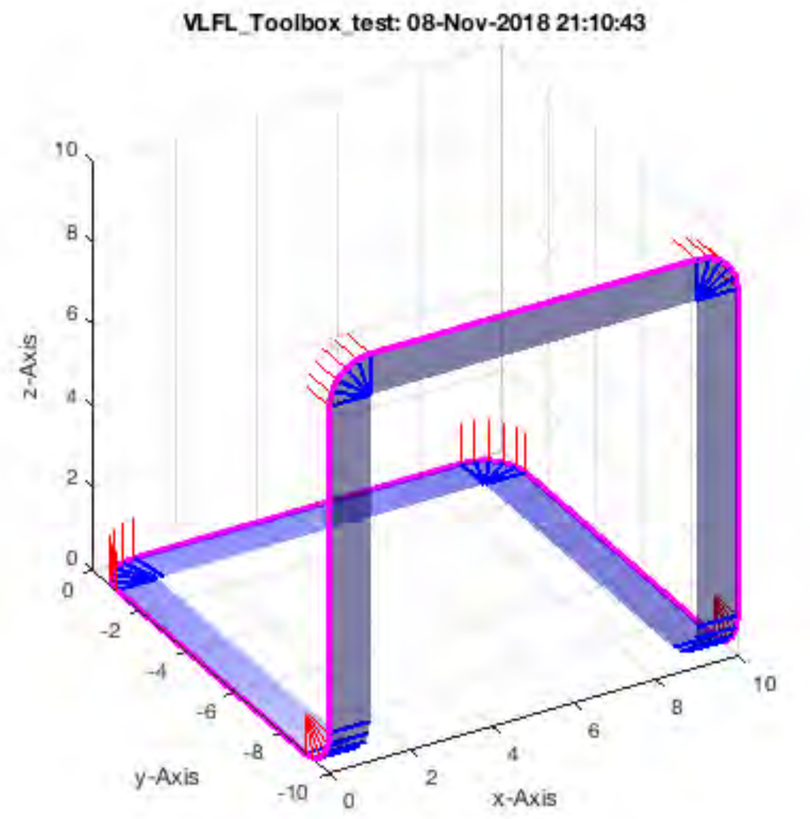

```
VLedgeNormal(VLradialEdges(CVLofVL(VLsample(8))));
```



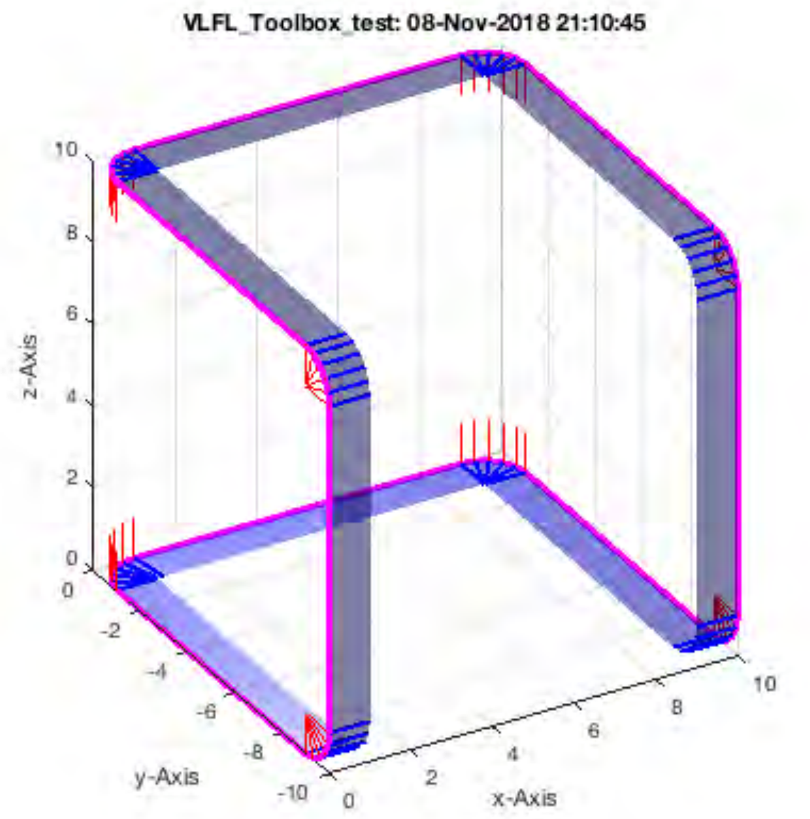

```
VLedgeNormal(VLradialEdges(CVLofVL(VLsample(12))));
```



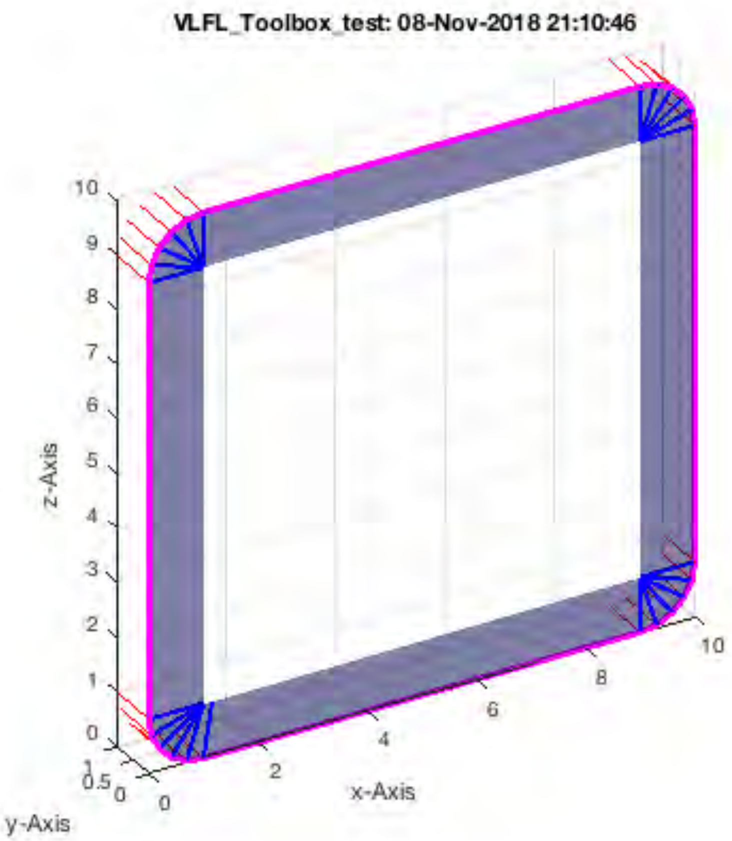

```
VLedgeNormal(VLradialEdges(CVLofVL(VLsample(13))));
```



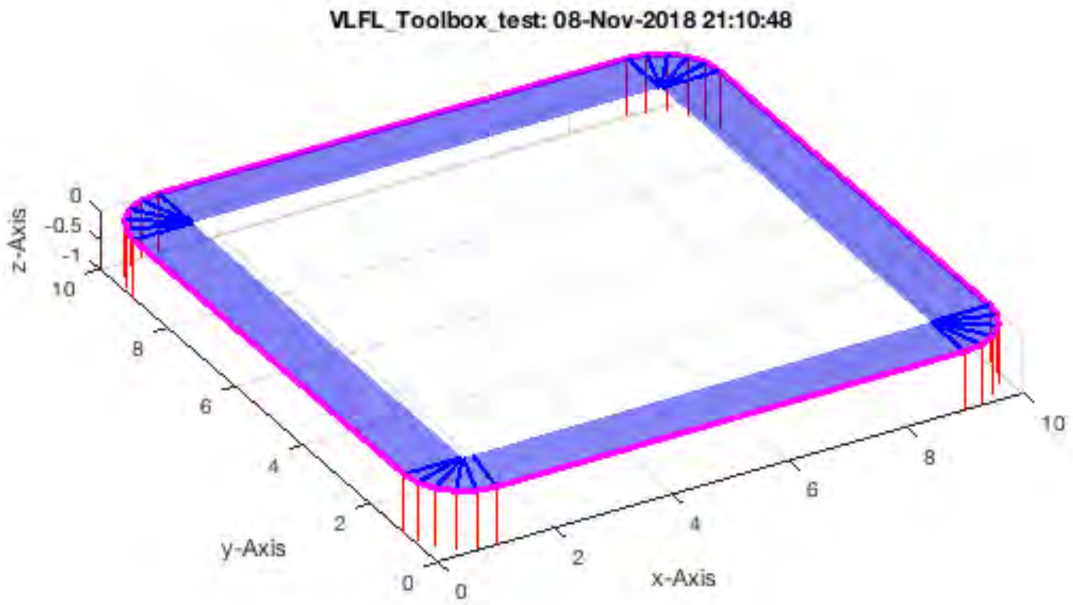

```
VLedgeNormal(VLradialEdges(CVLofVL(VLsample(14))));
```



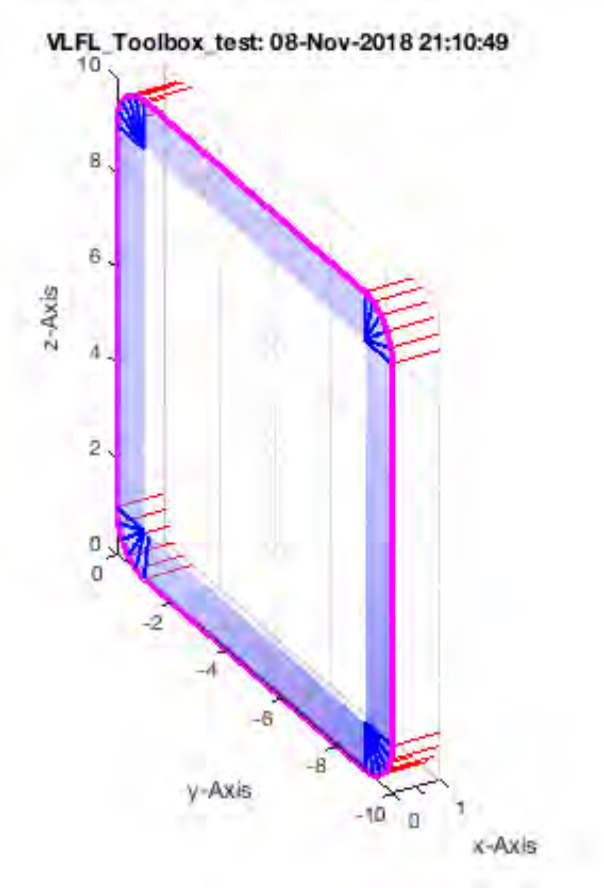

```
VLedgeNormal(VLradialEdges(CVLofVL(VLsample(20))));
```




```
VLedgeNormal(VLradialEdges(CVLofVL(VLsample(21))));
```



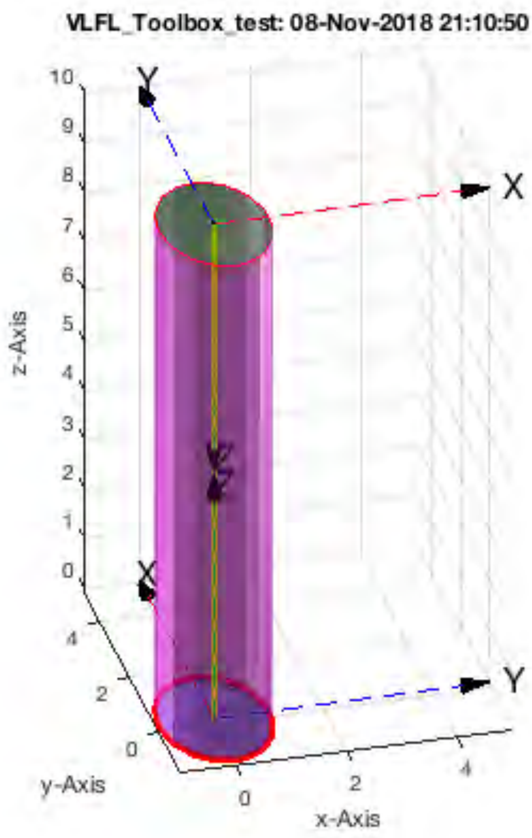

```
VLedgeNormal(VLradialEdges(CVLofVL(VLsample(22))));
```



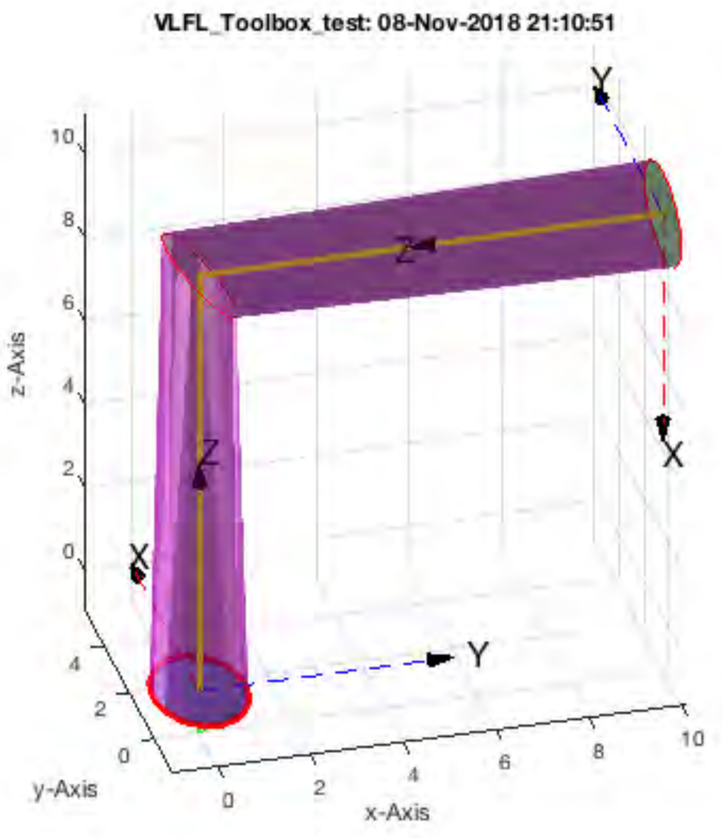
Creating Solid Geometries open

If angles are larger than 90 degree (pi/2)

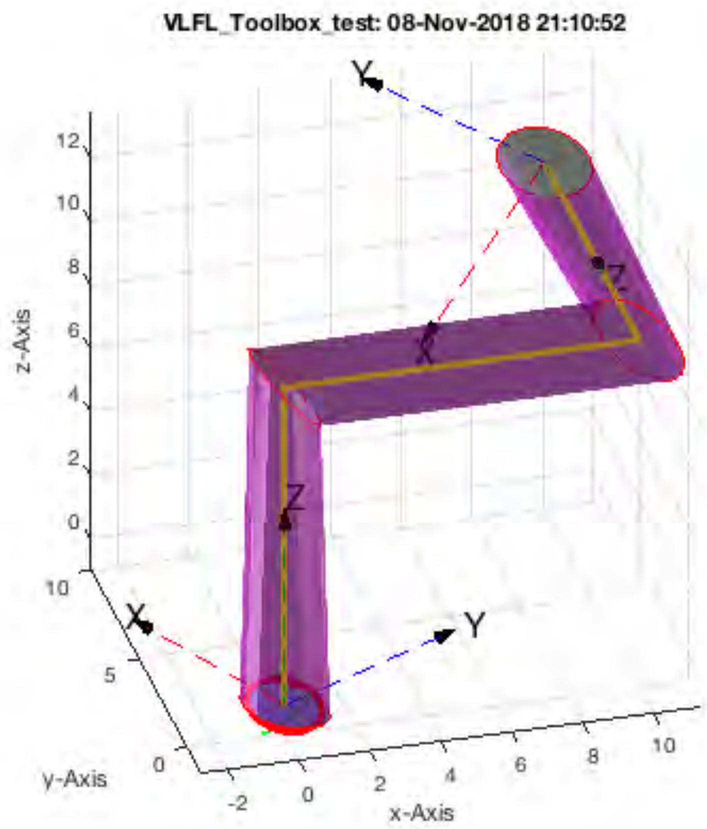
```
SGcontourtube2(PLcircle(1,'','',1.5),VLsample(2));
```



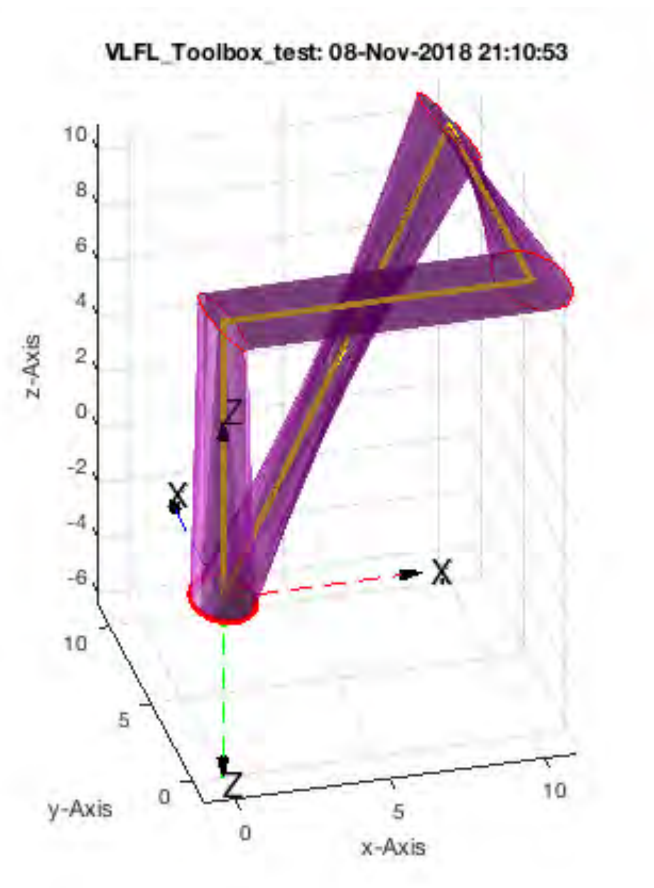

```
SGcontourtube2(PLcircle(1,'','',1.5),VLsample(3));
```



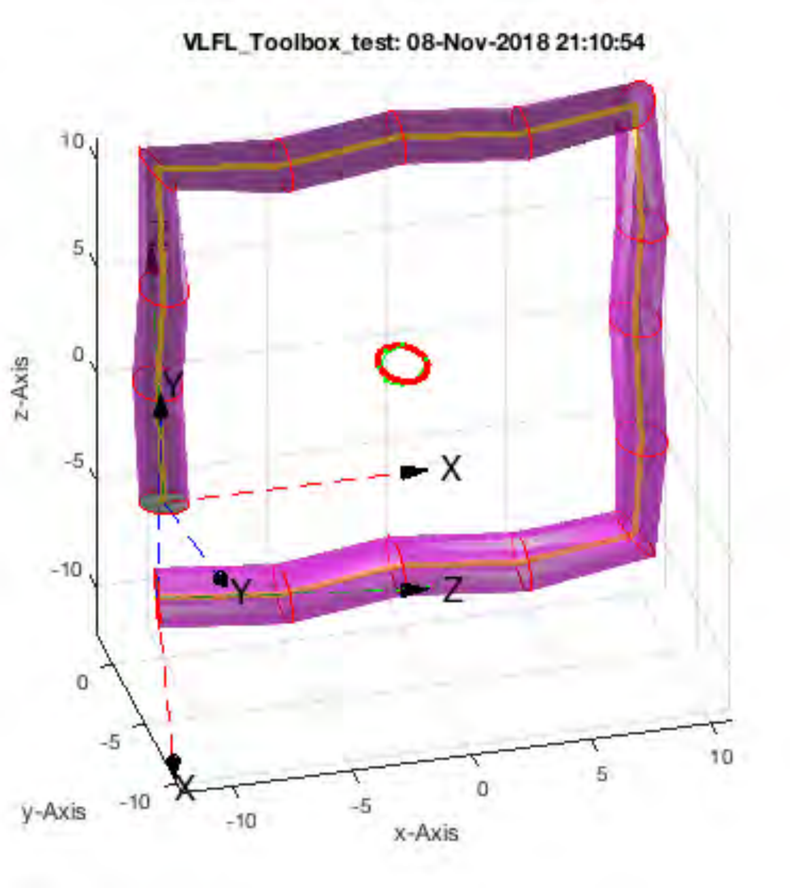

```
SGcontourtube2(PLcircle(1,'','',1.5),VLsample(7));
```



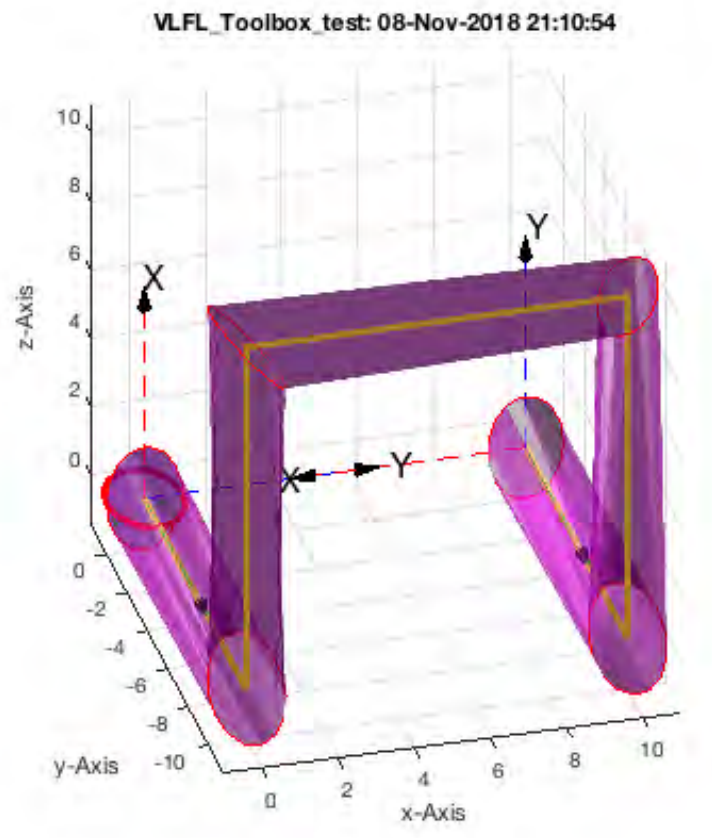

```
SGcontourtube2(PLcircle(1,'','',1.5),VLsample(8));
```



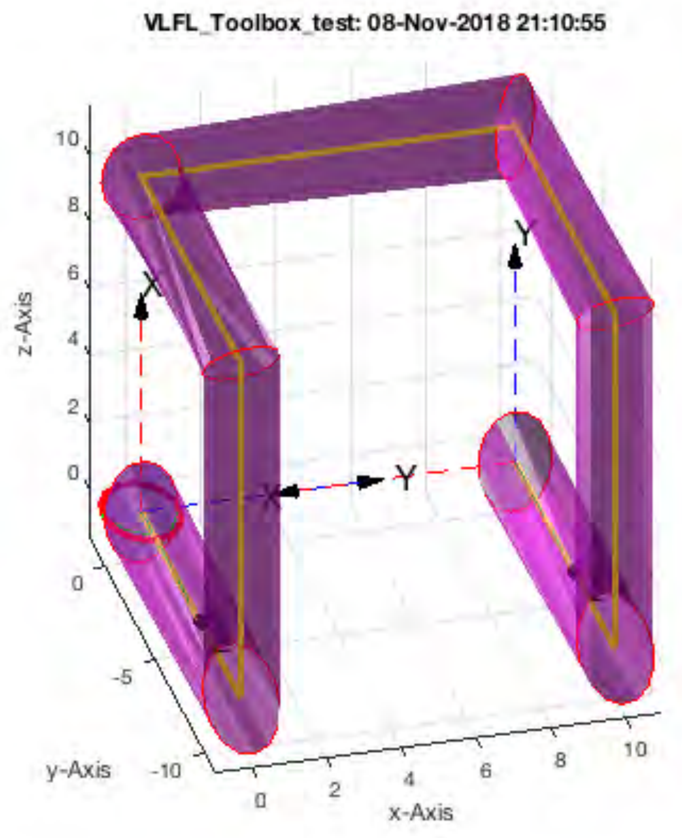

```
SGcontourtube2(PLcircle(1,'','',1.5),VLsample(12));
```



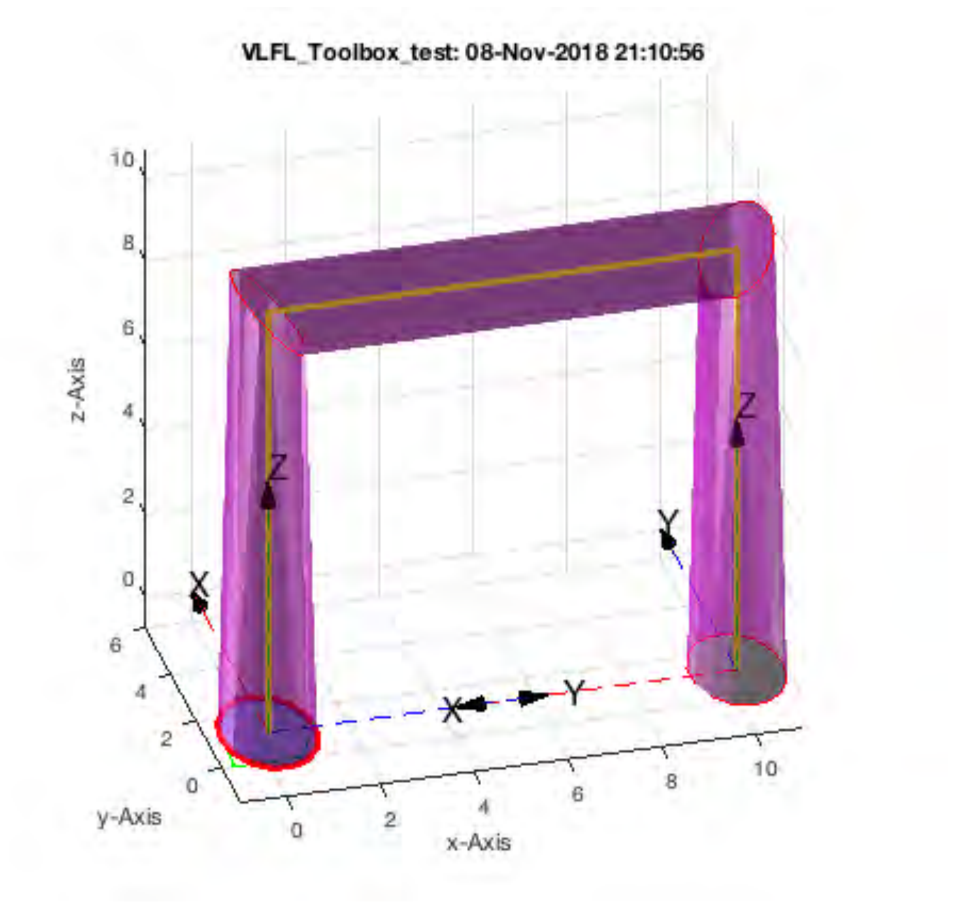

```
SGcontourtube2(PLcircle(1,'','',1.5),VLsample(13));
```



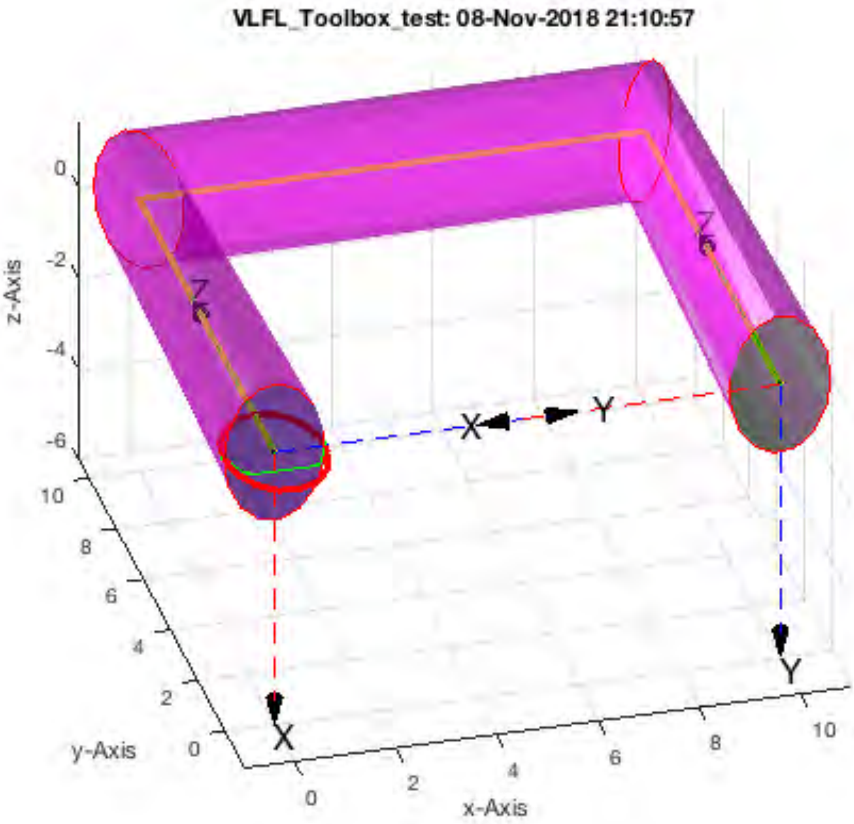

```
SGcontourtube2(PLcircle(1,'','',1.5),VLsample(14));
```



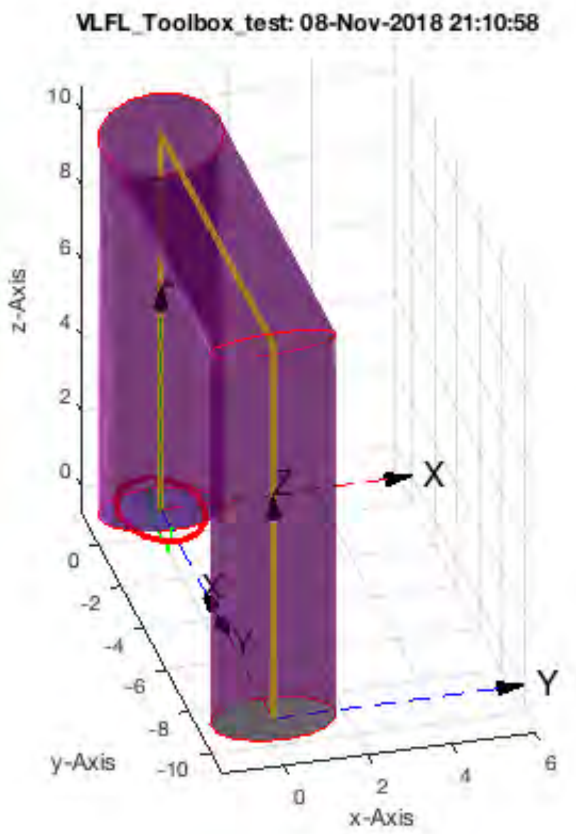

```
SGcontourtube2(PLcircle(1,'','',1.5),VLsample(20));
```




```
SGcontourtube2(PLcircle(1,'','',1.5),VLsample(21));
```



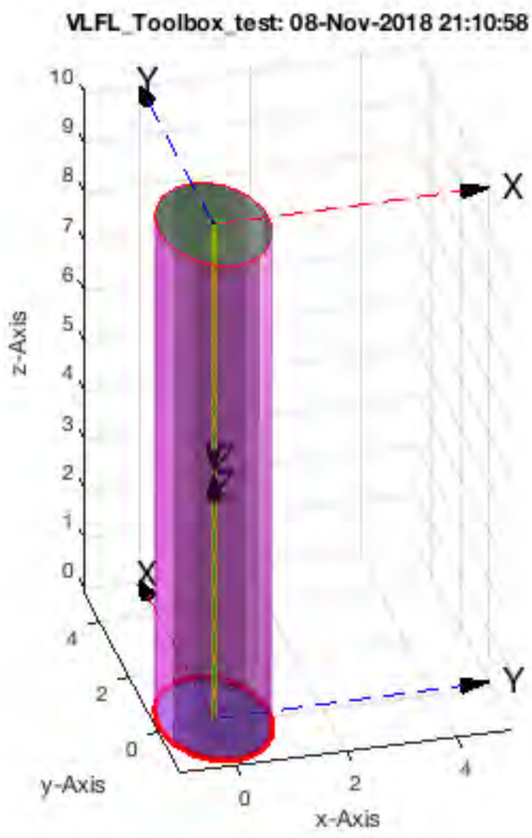

```
SGcontourtube2(PLcircle(1,'','',1.5),VLsample(22));
```



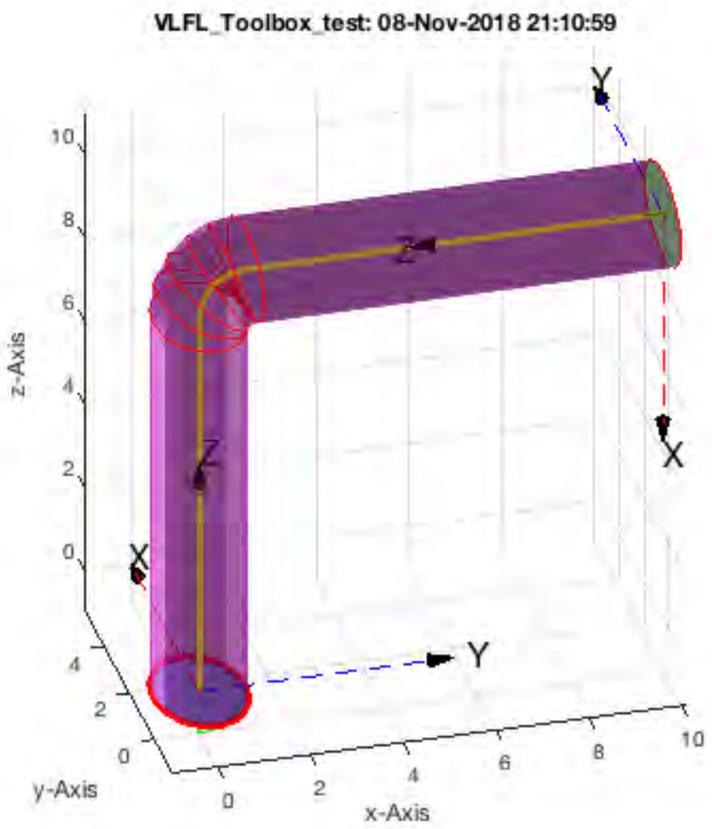
Creating Solid Geometries open

If angles are larger than 90 degree (pi/2)

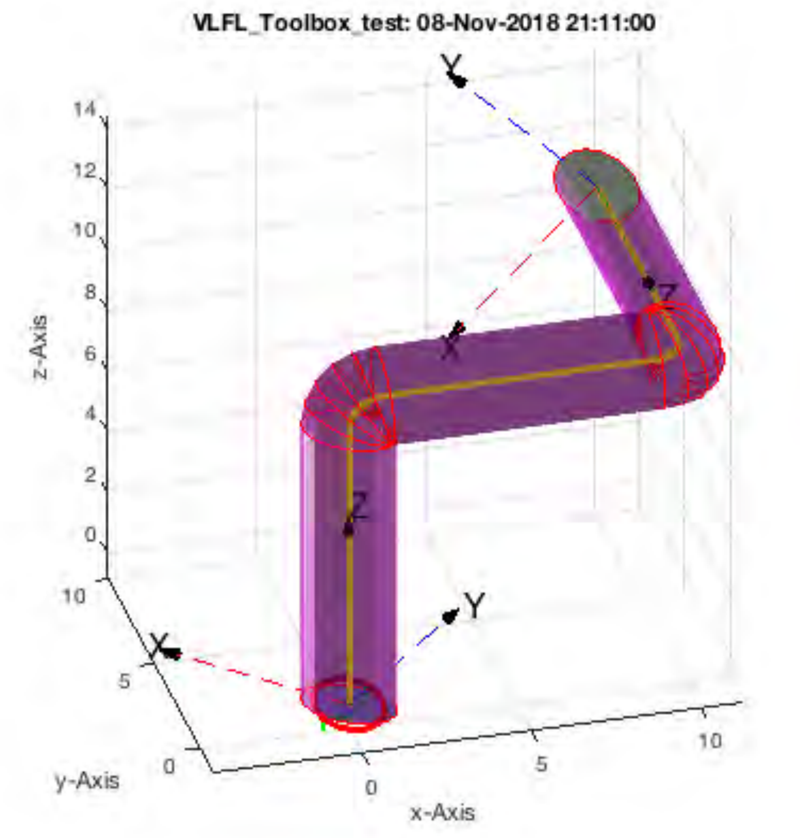
```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(VLsample(2)));
```



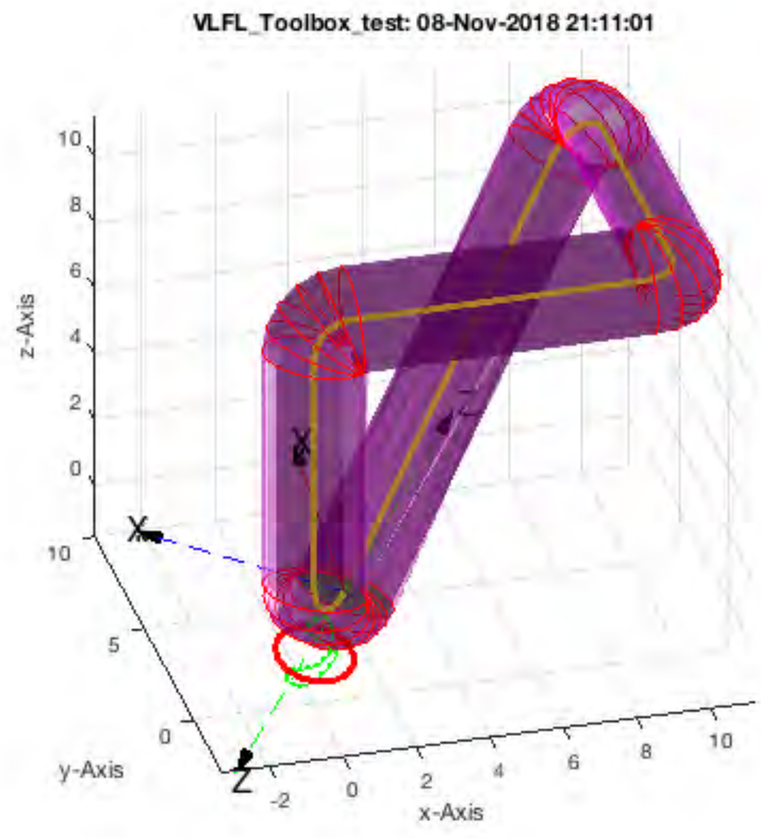

```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(VLsample(3)));
```



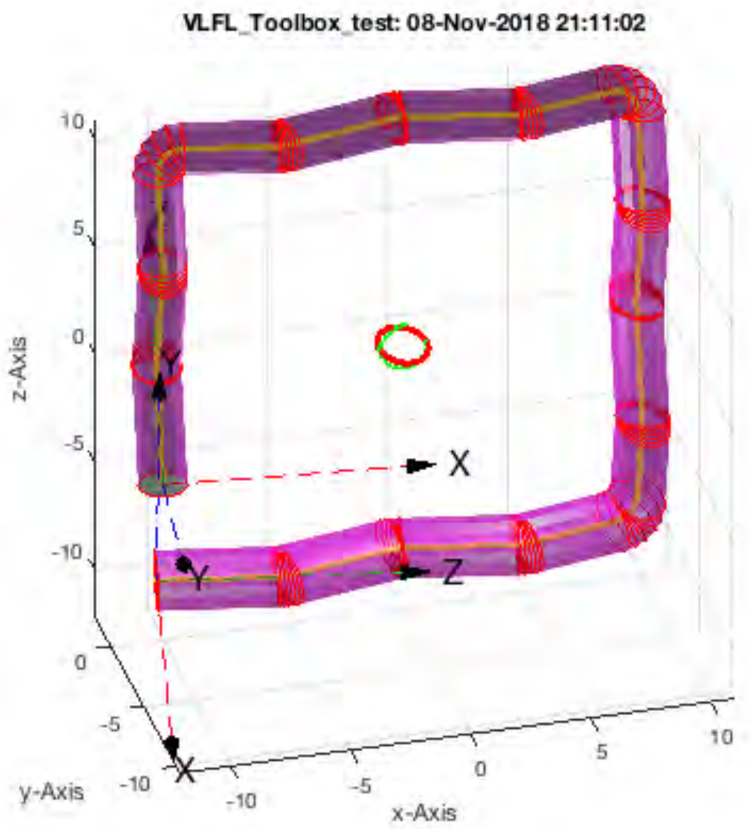

```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(VLsample(7)));
```



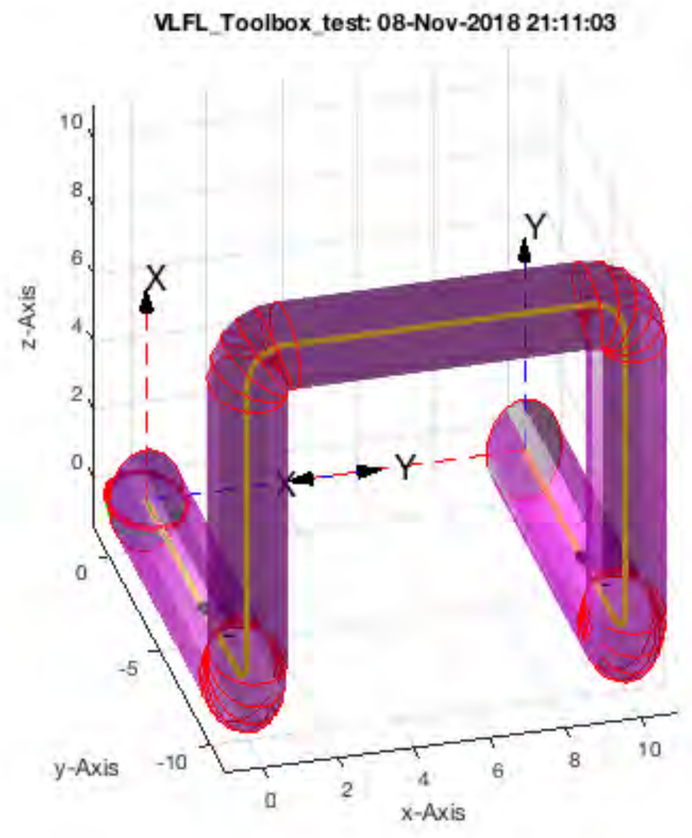

```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(VLsample(8)));
```



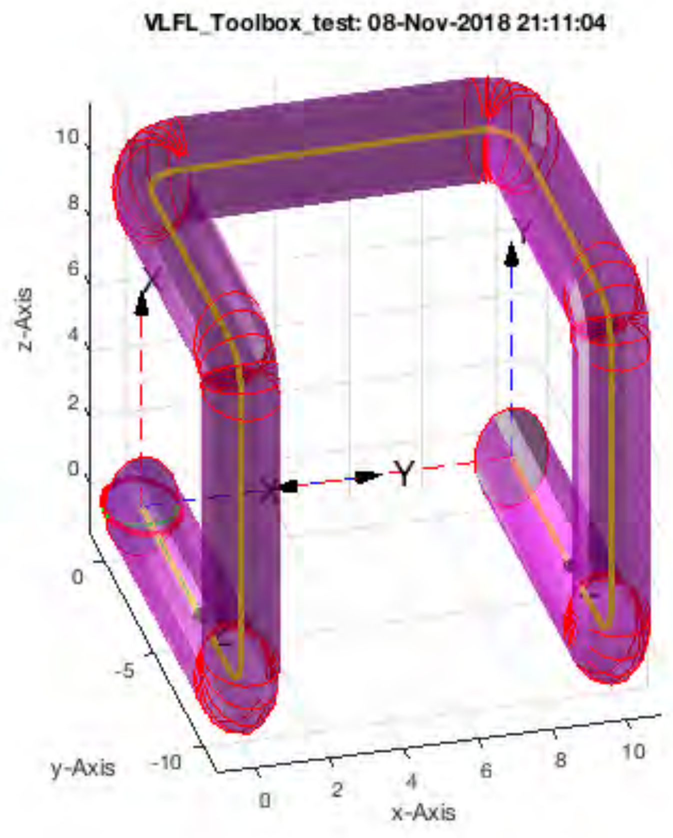

```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(VLsample(12)));
```



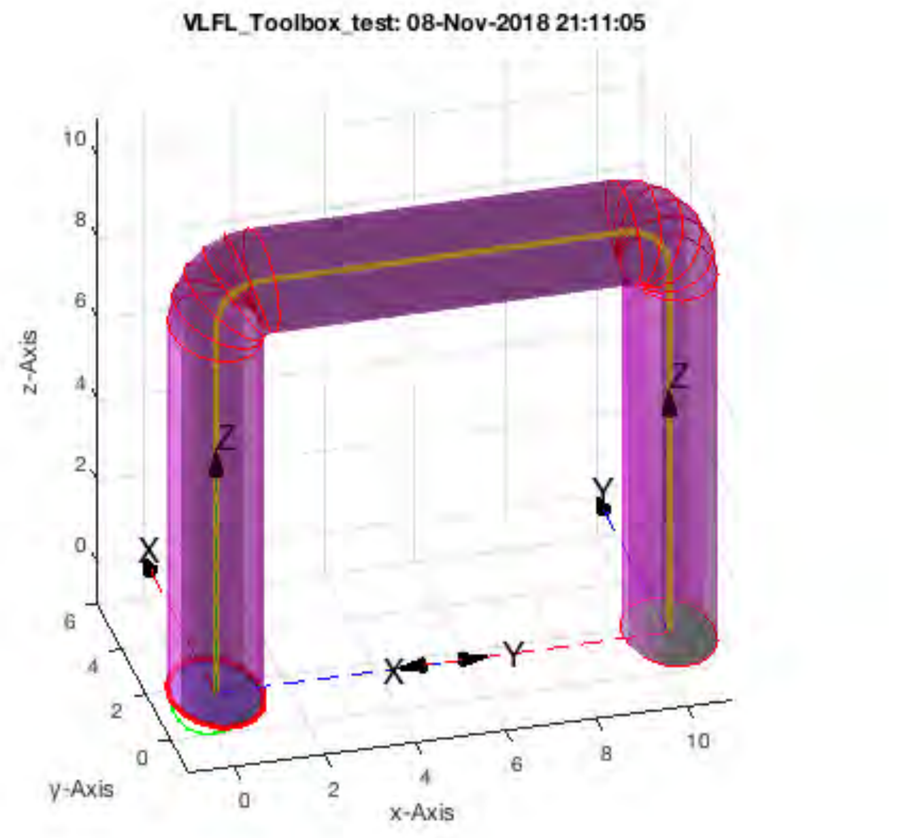

```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(VLsample(13)));
```



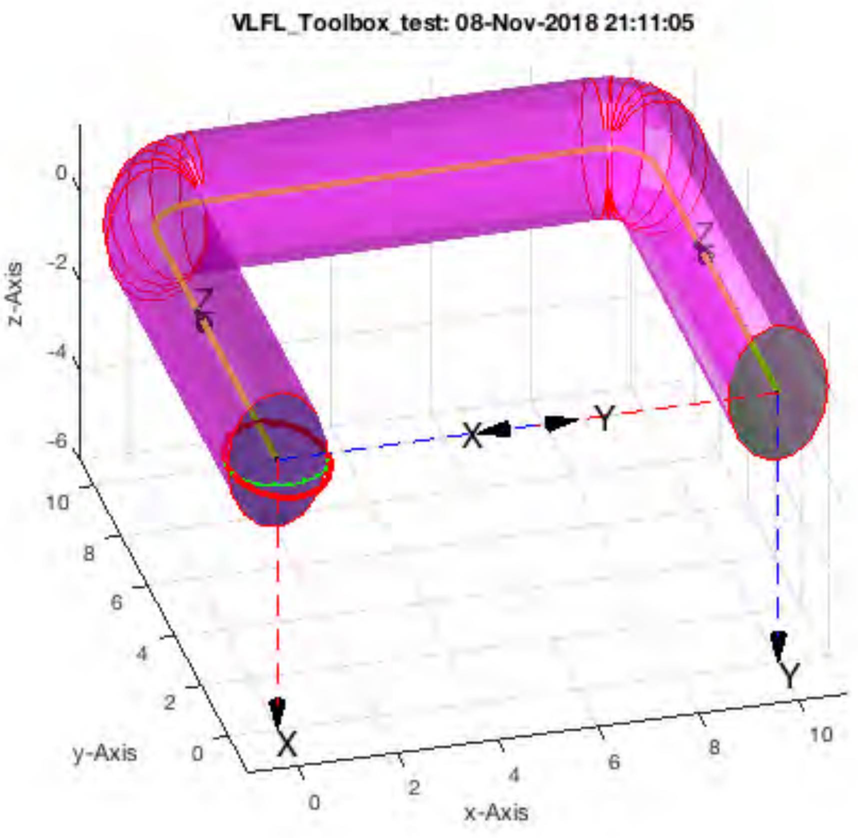

```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(VLsample(14)));
```



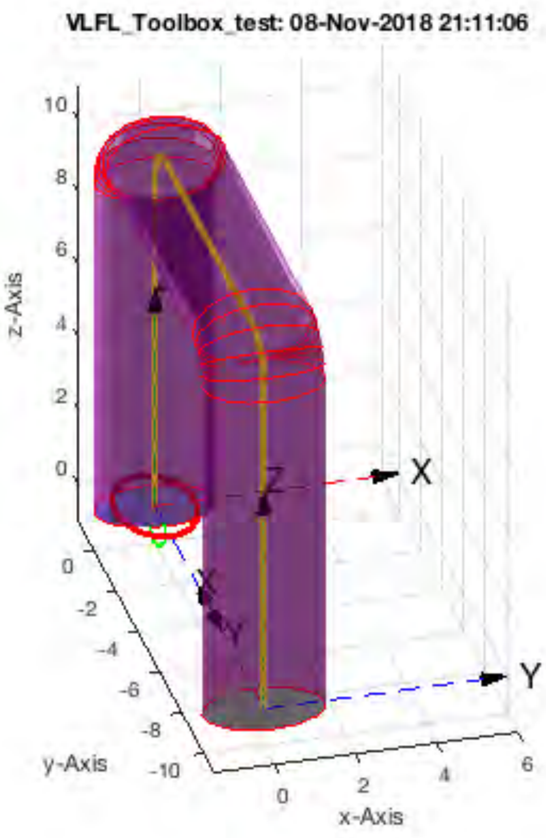

```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(VLsample(20)));
```




```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(VLsample(21)));
```



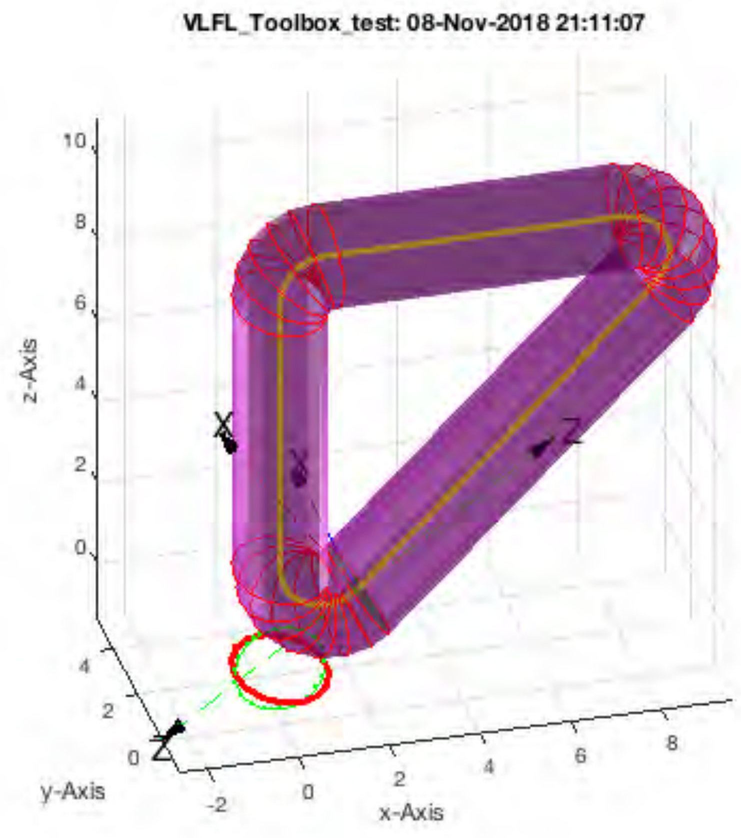

```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(VLsample(22)));
```



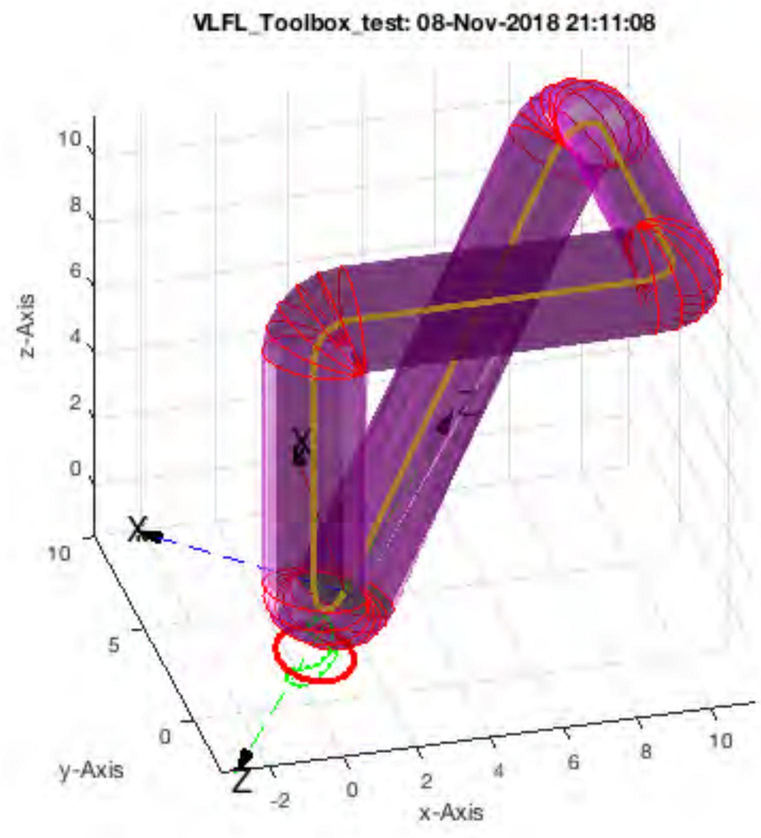
Creating Solid Geometries open

If angles are larger than 90 degree (pi/2)

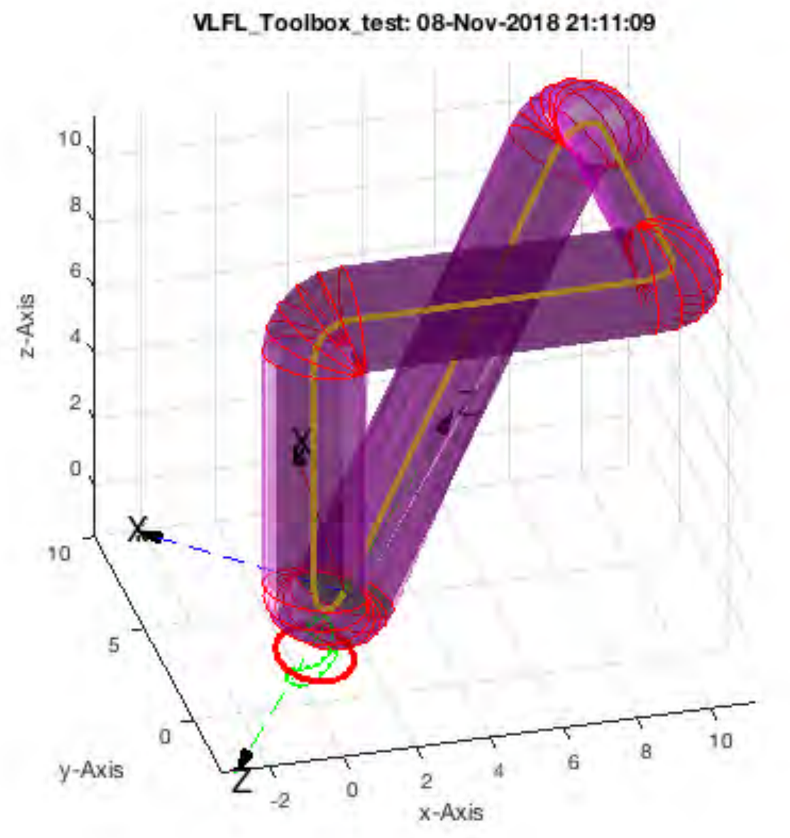
```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(CVLofVL(VLsample(3))));
```



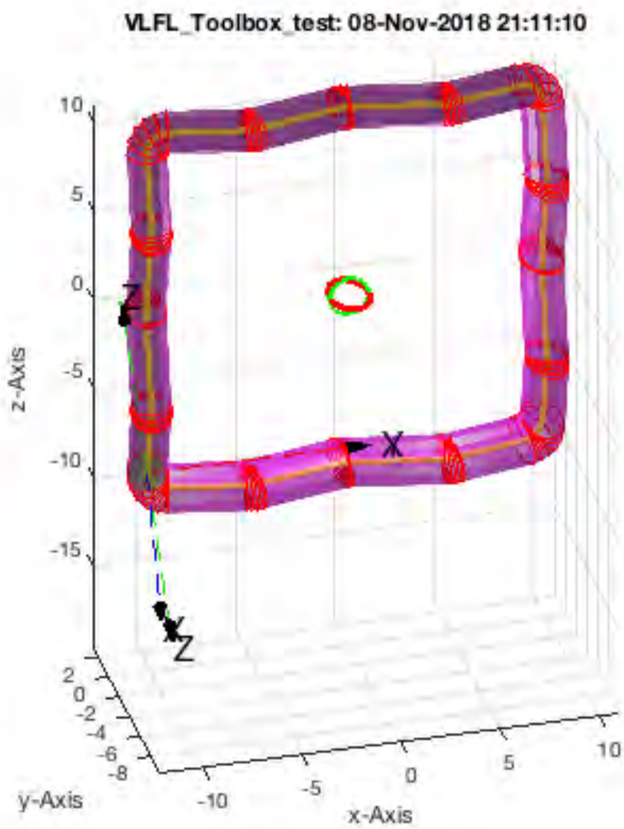

```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(CVLofVL(VLsample(7))));
```



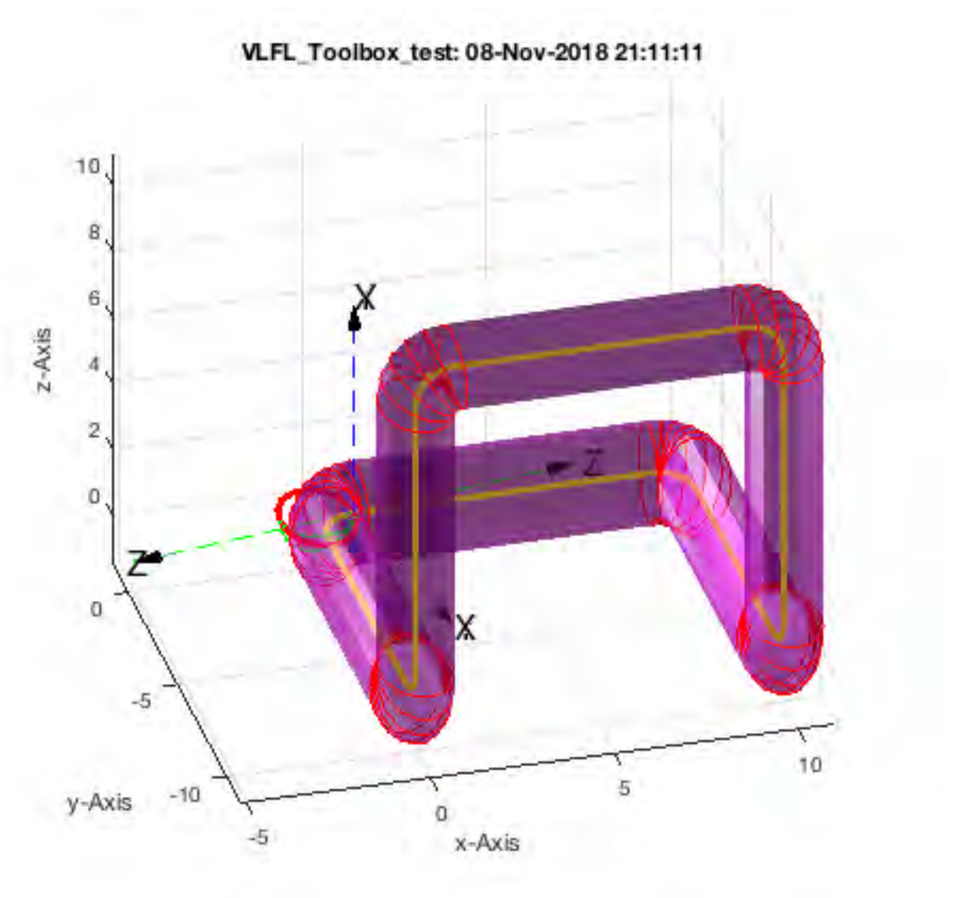

```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(CVLofVL(VLsample(8))));
```



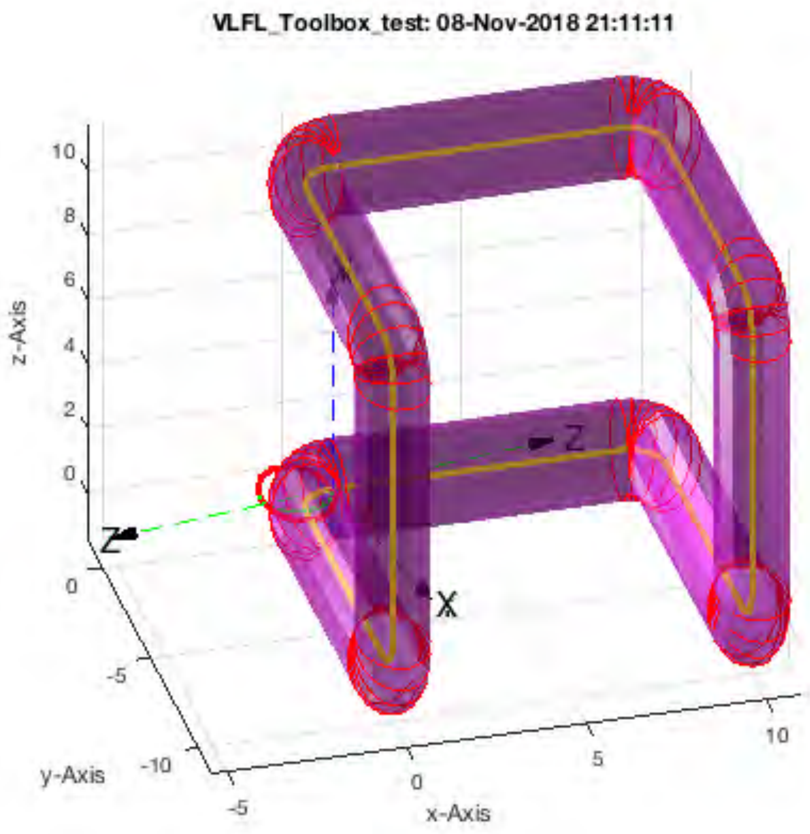

```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(CVLofVL(VLsample(12))));
```



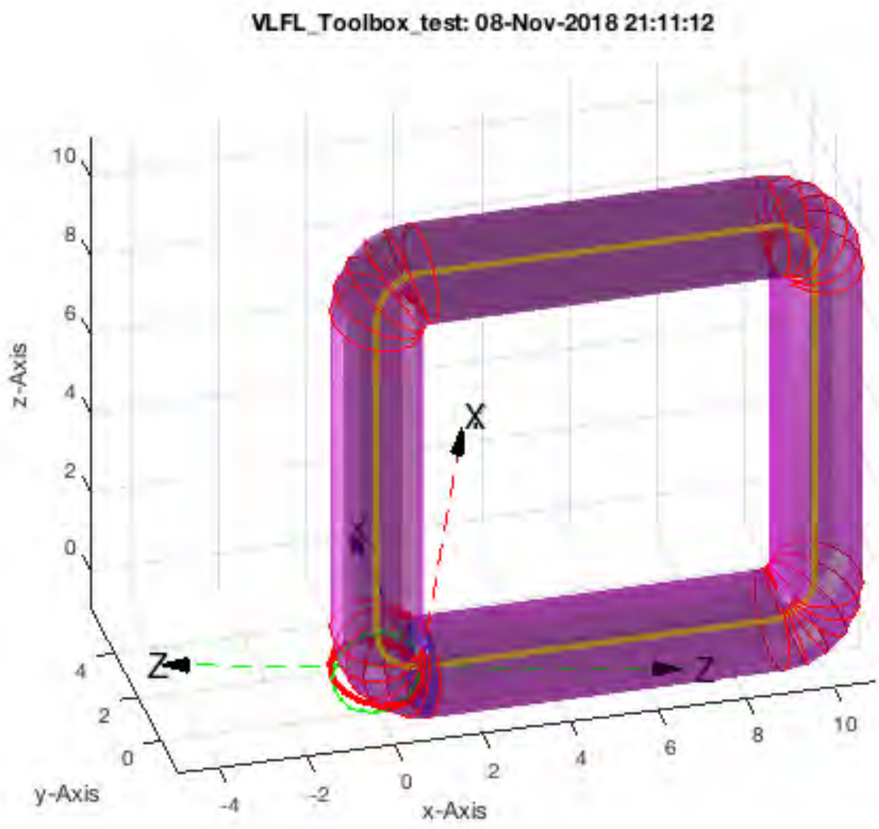

```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(CVLofVL(VLsample(13))));
```



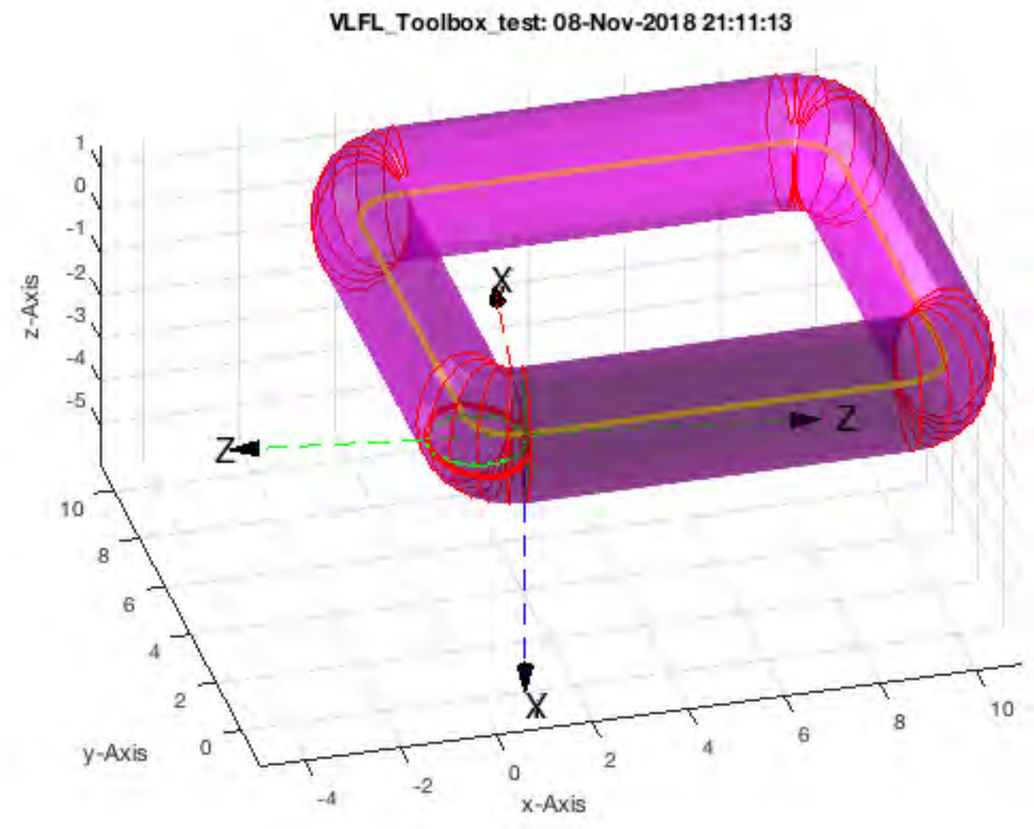

```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(CVLofVL(VLsample(14))));
```



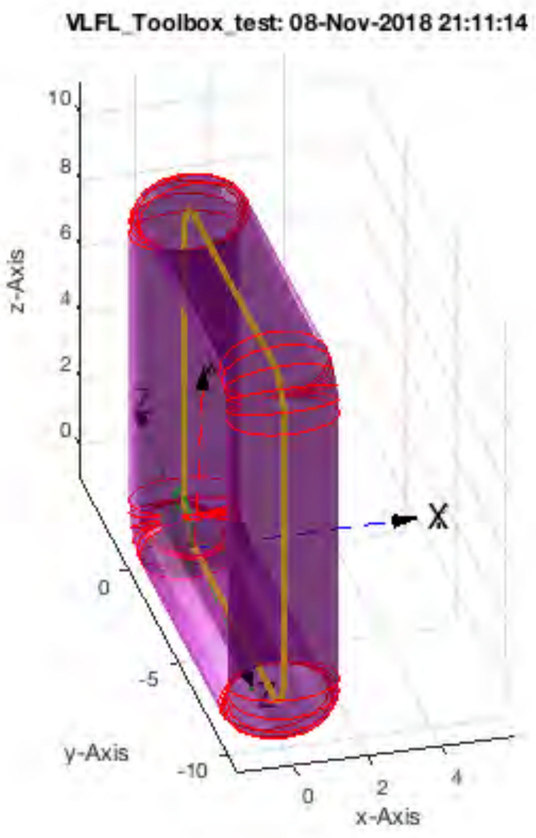

```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(CVLofVL(VLsample(20))));
```



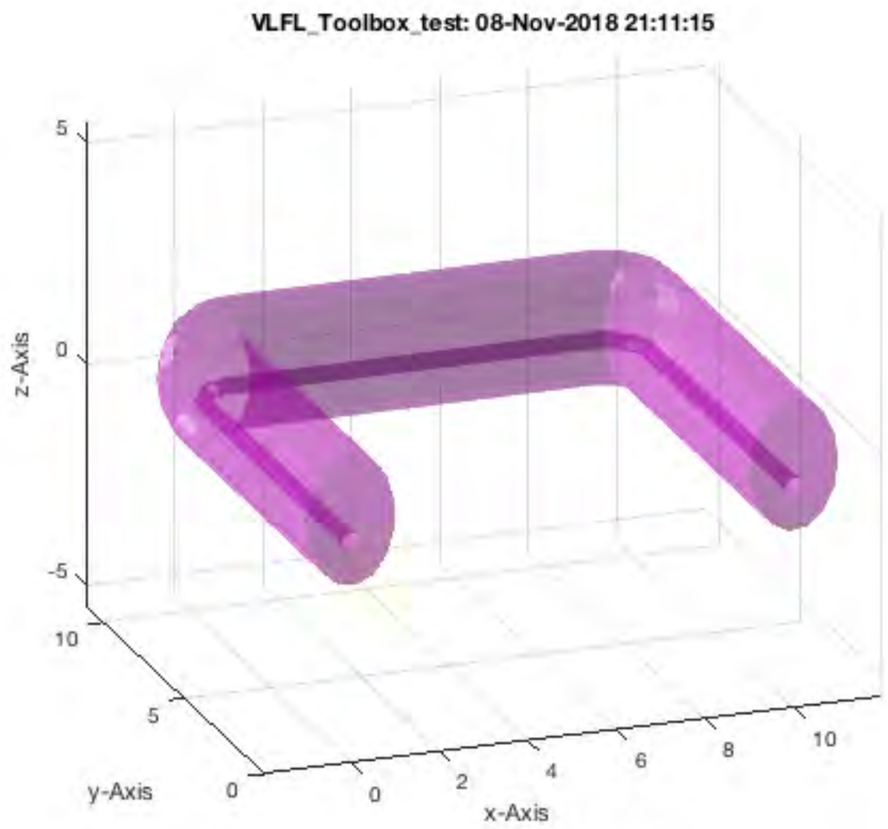

```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(CVLofVL(VLsample(21))));
```




```
SGcontourtube2(PLcircle(1,'','',1.5),VLradialEdges(CVLofVL(VLsample(22))));
```




```
SGcontourtube2([PLcircle(1,'','',1.5);NaN NaN;PLcircle(0.2)+[0 0.5]],VLradialEdges(VLsample
(21))); SG=ans;
SGfigure(SG);VLFLplotlight(1,0.3); view(-20,20);
```



1. Conversion between triangle surface model and tetrahedon volumen model

Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2018-Nov-08), 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 11-Aug-2073 21:11:16!
Executed 08-Nov-2018 21:11:18 by 'lueth' using Matlab 9.4.0.949201 (R2018a) Update 6 on a M
ACI64
======================================== Used Matlab products: ============================
==========
antenna_toolbox
database_toolbox
image_toolbox
map_toolbox
matlab
pde_toolbox
robotics_system_toolbox
simmechanics
simscape
simulink
video_and_image_blockset
==========================================================================================
==========
```

Published with MATLAB® R2018a