

Tutorial 57: Processing Stacks of Slices = CVLz

2020-08-23: Tim C. Lueth, Professor at Technische Universität München, Germany (URL: <http://www.SG-Lib.org>) - Last Change: 2020-08-23

Contents

- [Complete List of all Tutorials with Publishable MATLAB Files of this Solid-Geometries Toolbox](#)
- [Motivation for this tutorial: \(Originally SolidGeometry 4.9 required\)](#)
- [While Vertex Lists \(VL\) are unsorted co-ordinates lists, Contour Vertex Lists \(CVL\) describe closed contours or paths in space separated by nan nan nan. A special form are planar contours, parallel to the x/y-plane, where the coordinates of each individual contour have the same z-value and can be easily identified by the z-value. Such CVLz - lists are generated by the slicing functions.](#)
- [This tutorial describes some functions for handling this data model](#)
- [Final Remarks](#)

Complete List of all Tutorials with Publishable MATLAB Files of this Solid-Geometries Toolbox

The following topics are covered and 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 Lightweight-structures
- Tutorial 14: Manipulation Functions for Closed Polygons and Laser Cutting (SVG)
- Tutorial 15: Create a Solid by 2 Closed Polygons
- Tutorial 16: Create Tube-Style Solids by Succeeding Polygons
- Tutorial 17: Filling and Bending of Polygons and Solids
- Tutorial 18: Analyzing and modifying STL files from CSG modeler (Catia)
- Tutorial 19: Creating drawing templates and dimensioning from polygon lines
- Tutorial 20: Programmatically Interface to SimMechanics Multi-Body Toolbox
- Tutorial 21: Programmatically Convert Joints into Drives (SimMechanics)
- Tutorial 22: Adding Simulink Signals to Record Frame Movements
- Tutorial 23: Automatic Creation of a Missing Link and 3D Print of a Complete Model
- Tutorial 24: Automatic Creation of a Joint Limitations
- Tutorial 25: Automatic Creation of Video Titels, Endtitels and Textpages
- Tutorial 26: Create Mechanisms using Universal Planar Links
- Tutorial 27: Fourbar-Linkage: 2 Pose Syntheses and Linkage Export for 3D Printing
- Tutorial 28: Fourbar-Linkage: 3 Pose Syntheses and Linkage Export for 3D Printing
- Tutorial 29: Create a multi body simulation using several mass points
- Tutorial 30: Creating graphical drawings using point, lines, surfaces, frames etc.
- Tutorial 31: Importing 3D Medical DICOM Image Data and converting into 3D Solids
- Tutorial 32: Exchanging Data with a FileMaker Database
- Tutorial 33: Using a Round-Robin realtime multi-tasking system
- Tutorial 34: 2D Projection Images and Camera Coordinate System Reconstruction
- Tutorial 35: Creation of Kinematic Chains and Robot Structures
- Tutorial 36: Creating a Patient-Individual Arm-Skin Protector-Shell
- Tutorial 37: Dimensioning of STL Files and Surface Data
- Tutorial 38: Some more solid geometry modelling function
- Tutorial 39: HEBO Modules robot design
- Tutorial 40: JACO Robot Simulation and Control

- Tutorial 41: Inserting Blades, Cuts and Joints into Solid Geometries
- Tutorial 42: Performing FEM Stress and Displacement Analysis and Structural Optimization of Solids
- Tutorial 43: Performing FEM Structural Optimization (CAO) and Topological Optimization (SKO) of Solids
- Tutorial 44: Creation of solids and kinematics from 3D curves and transformation matrices
- Tutorial 45: Creation of Solids using the SG-Coder - SGofCPLcommand
- Tutorial 46: Creating Fischertechnik compatible gear boxes using SGofCPLcommand
- Tutorial 47: Creating four-joints by 3 pose synthesis
- Tutorial 52: CPL Buffers and cw/ccw Orientation
- Tutorial 53: SKOL - Soft Kill Option for Large Displacement by Yilun Sun
- Tutorial 54: Automated Design of Precision Joints by Screws or Ball Bearings
- Tutorial 55: Automated Design of Manipulators with Screws or Ball Bearing
- Tutorial 56: Checking Functions for Solids
- Tutorial 57: Processing Stacks of Slices = CVLz

Motivation for this tutorial: (Originally SolidGeometry 4.9 required)

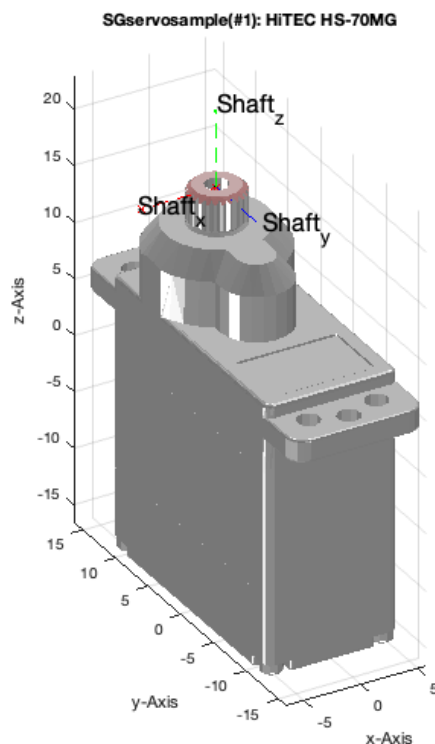
While Vertex Lists (VL) are unsorted co-ordinates lists, Contour Vertex Lists (CVL) describe closed contours or paths in space separated by nan nan nan. A special form are planar contours, parallel to the x/y-plane, where the coordinates of each individual contour have the same z-value and can be easily identified by the z-value. Such CVLz - lists are generated by the slicing functions.

This tutorial describes some functions for handling this data model

```
function VLFL_EXP57
```

```
SGservosample(1); SG=ans;
```

```
SGservosample: Non-manifold edges of this solid: 116
```



```
CVLzofSGsliceauto(SG); CVLz=ans;
```

```
z =
-16.6250
-16.6250
```

-16.6241
-16.6241
-16.6110
-16.6109
-16.6098
-16.6040
-16.6021
-16.5909
-16.5909
-16.5665
-16.5664
-16.5663
-16.5663
-16.5495
-16.5495
-16.5495
-16.5373
-16.5371
-16.5272
-16.5272
-16.5015
-16.4987
-16.4985
-16.4985
-16.4958
-16.4957
-16.4957
-16.4957
-16.4865
-16.4864
-16.4864
-16.4785
-16.4478
-16.4478
-16.4477
-16.4398
-16.4352
-16.4303
-16.4250
-16.4199
-16.4182
-16.4139
-16.4139
-16.4139
-16.4057
-16.4056
-16.3957
-16.3956
-16.3889
-16.3889
-16.3889
-16.3842
-16.3840
-16.3638
-16.3570
-16.3563
-16.3250
-16.1250
-15.7714
-15.6250
-9.6250
1.3750
4.2750
4.2887
4.2891
4.2902
4.3223
4.3253
4.3336
4.3337
4.3687
4.3985
4.4015
4.4750
6.0750
6.1485
6.2163
6.2164
6.2270
6.2277

```

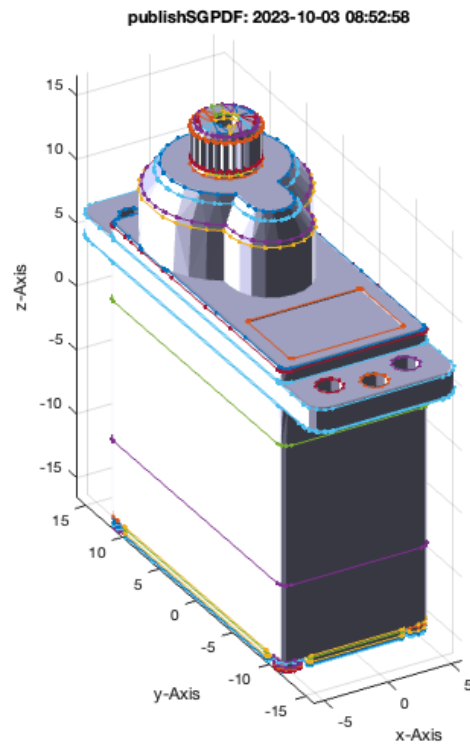
6.2609
6.2750
7.0750
7.1050
7.1545
7.1750
7.1854
7.1913
7.2501
7.2871
7.3051
7.3062
7.3364
7.3540
7.3750
11.3750
11.5636
11.5887
12.8743
13.3750
13.5883
13.7850
13.8743
16.1250
16.4114
16.6250

```

```

CPLofSGslice3: 5%Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculWarning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free

```

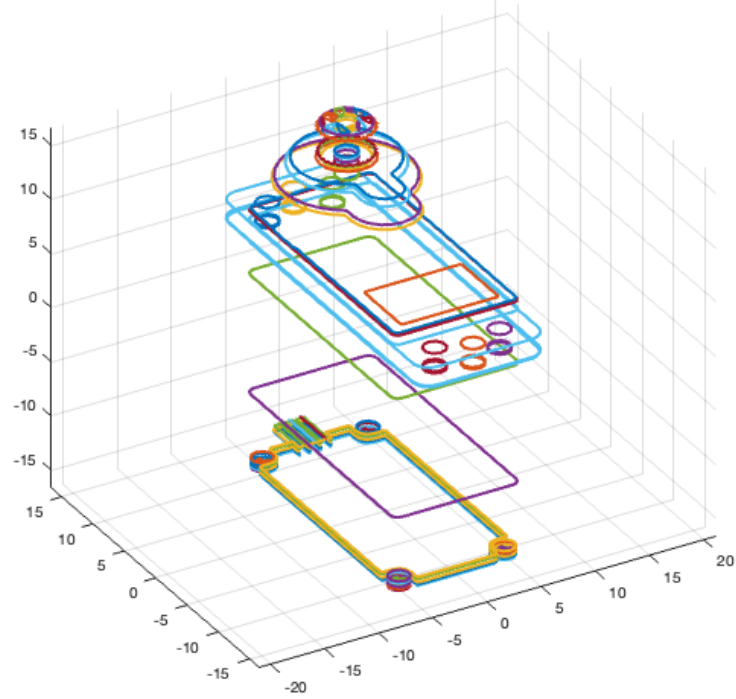


```

SGfigure(-30,30); CVLzplot(CVLz,'-',2); % plot slices as contour

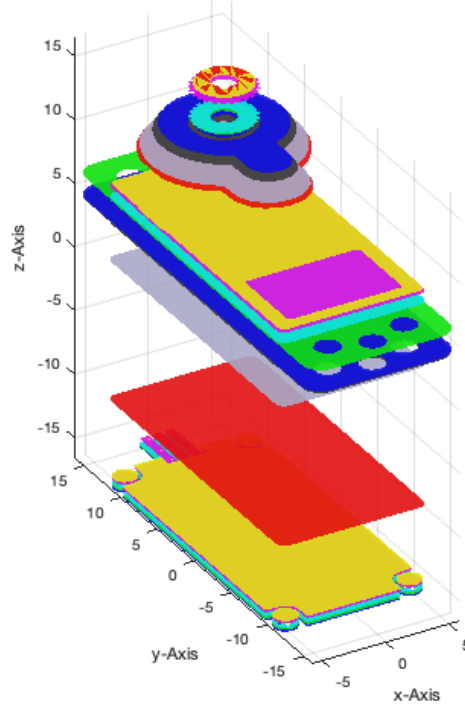
```

publishSGPDF: 2023-10-03 08:52:59



```
SGfigure(-30,30); CVLzplotasCPS(CVLz, 'r',0.9); % plot slices as polyshapes
```

publishSGPDF: 2023-10-03 08:53:01

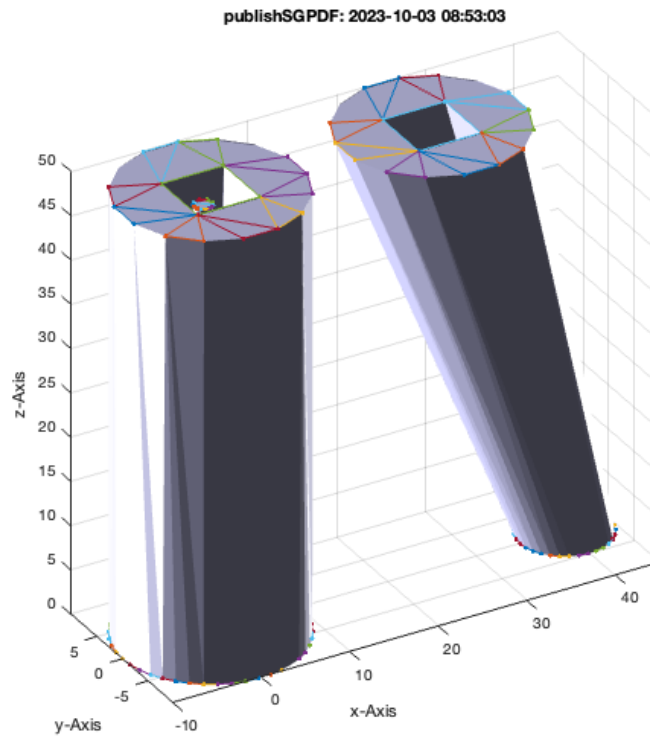


```
CVLzofSGsliceauto(SGsample(48)); CVLz=ans; % nutzt nur die exisiterenden z werte
```

z =
0

50

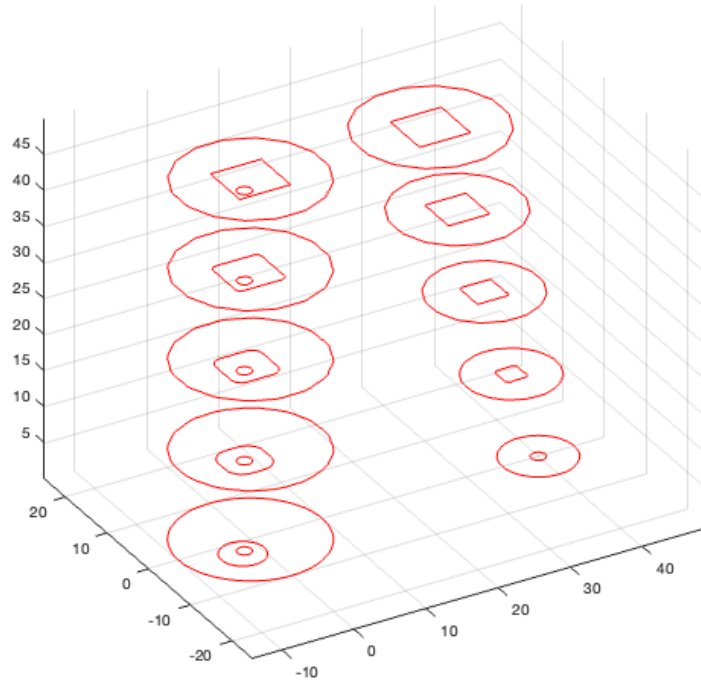
```
CPLofSGslice3: 50%Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free
Warning: Crossing plane cannot be calculated error-free
```



```
CVLzofSGslices(SGsample(48),5); CVLz=ans; % nimmt systematisch genau 5 z-Werte
```

```
CVLzofSGslices: 20% 40% 60% 80% 100%
```

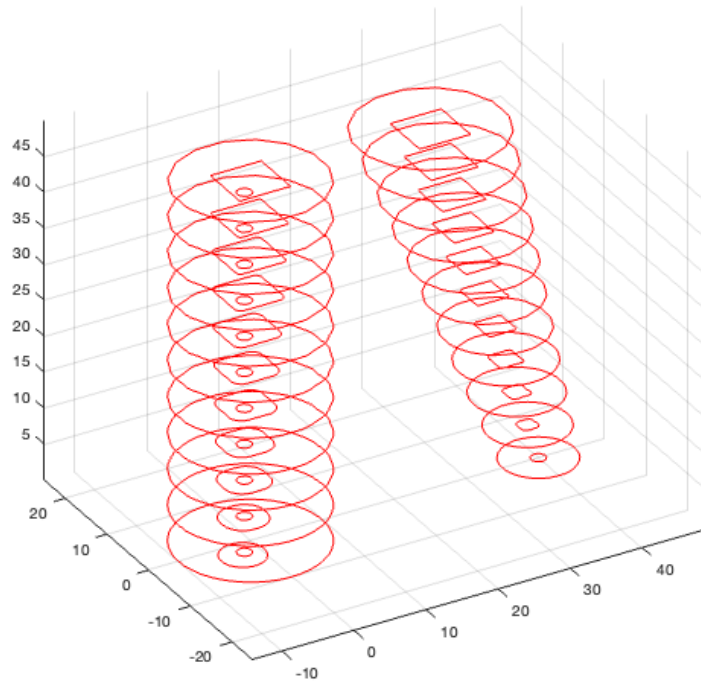
publishSGPDF: 2023-10-03 08:53:04



```
CVLzofSGslices(SGsample(48),5.01); CVLz=ans; % nimmt z-Werte im Abstand von 5.01
```

CVLzofSGslices: 10% 20% 25% 35% 45% 55% 65% 75% 80% 90% 100%

publishSGPDF: 2023-10-03 08:53:06



Final Remarks

```
close all
VLFLLicense
```

```
This VLFL-Lib, Rel. (2023-Oct-03), is for limited non commercial educational use only!
Licensee: Tim Lueth (Development Version)!
Please contact Tim Lueth, Professor at TU Munich, Germany!
WARNING: This VLFL-Lib (Rel. ) license will exceed at 06-Jul-2078 08:53:06!
Executed 03-Oct-2023 08:53:08 by 'timlueth' on a MACI64 using Mac OSX 13.6 | R2023a Update 5 | SG-Lib 5.4
===== Used Matlab products: =====
database_toolbox
distrib_computing_toolbox
fixed_point_toolbox
image_toolbox
map_toolbox
matlab
optimization_toolbox
pde_toolbox
simmechanics
simscape
simulink
=====
```

Published with MATLAB® R2023a