Tutorial 17: Filling and Bending of Polygons and Solids

2017-03-29: Tim C. Lueth, Professor at Technische Universität München, Germany (URL: http://www.SG-Lib.org) - Last Change: 2020-08-25

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

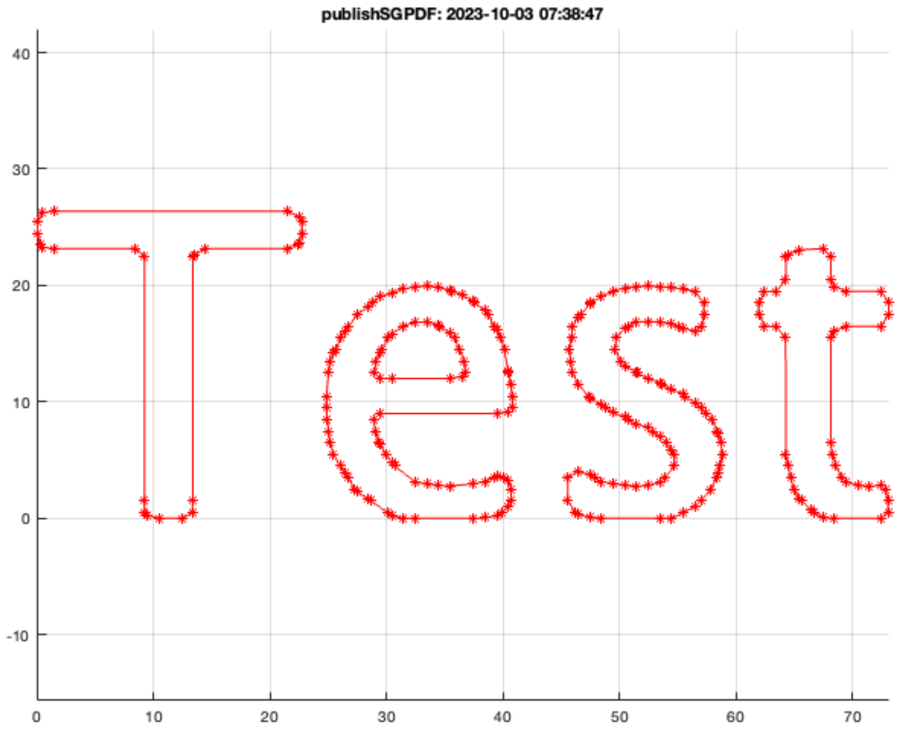
- Tutorial 36: Creating a Patient-Individual Arm-Skin Protector-Shell
- Tutorial 37: Dimensioning of STL Files and Surface Data
- Tutorial 38: Some more solid geometry modelling function
- Tutorial 39: HEBO Modules robot design
- Tutorial 40: JACO Robot Simulation and Control
- Tutorial 41: Inserting 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 54: Processing Stacks of Slices = CVLz
- 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 3.7 required)

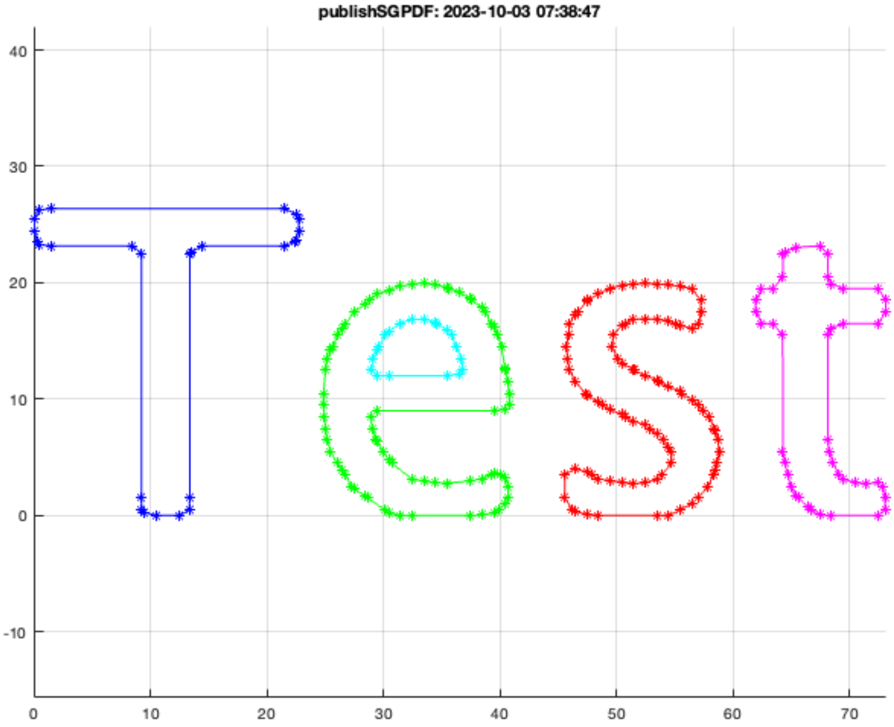
```
% For robotics design,very often we have a wish to extrude a CPL not only
% in an orthognal z direction to the xy-plane, but in an any desired direction
% even along a path in 3D space. Intutively we expect a result, but this is
% not easy to achieve automatically. Anyway, for those tasks we have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

1. Creating Closed Polygon Line

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

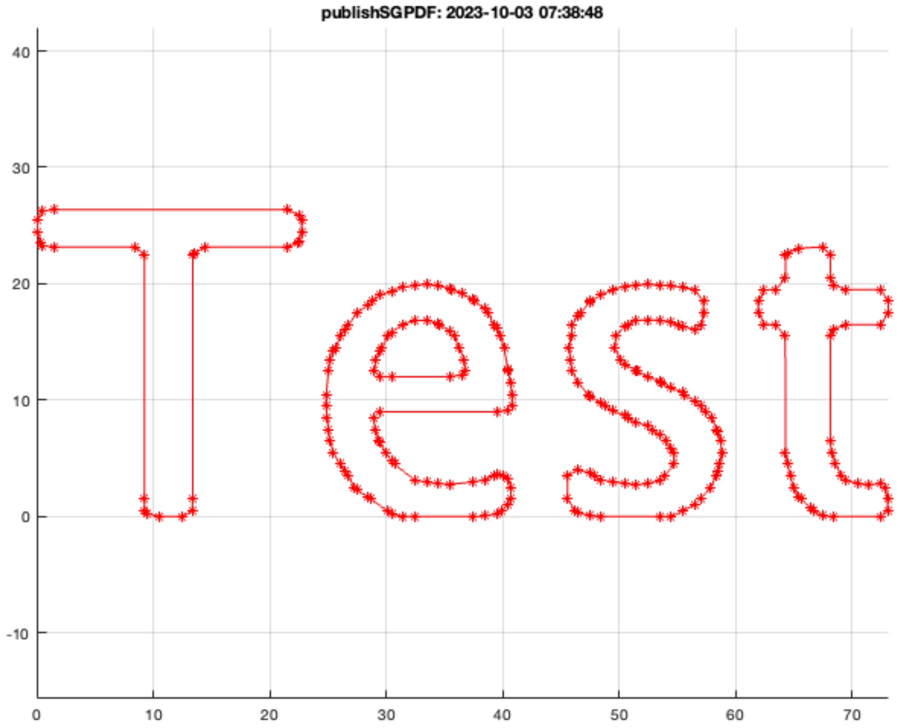


```
SGfigure; view(0,90); CVLplot(CPL);
```

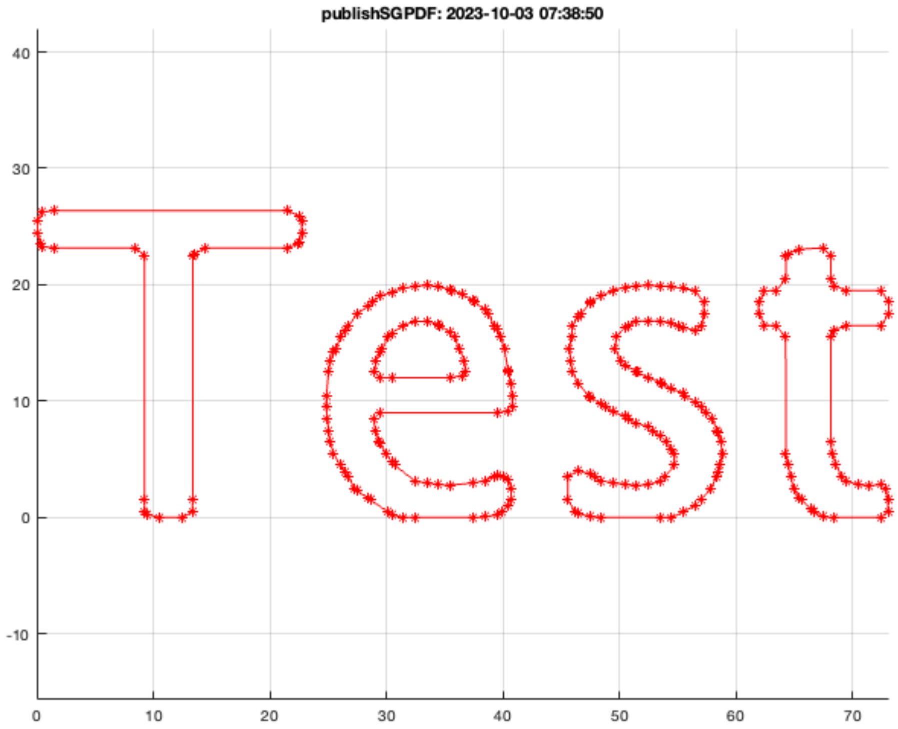


2. Converting CPL into PL EL

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

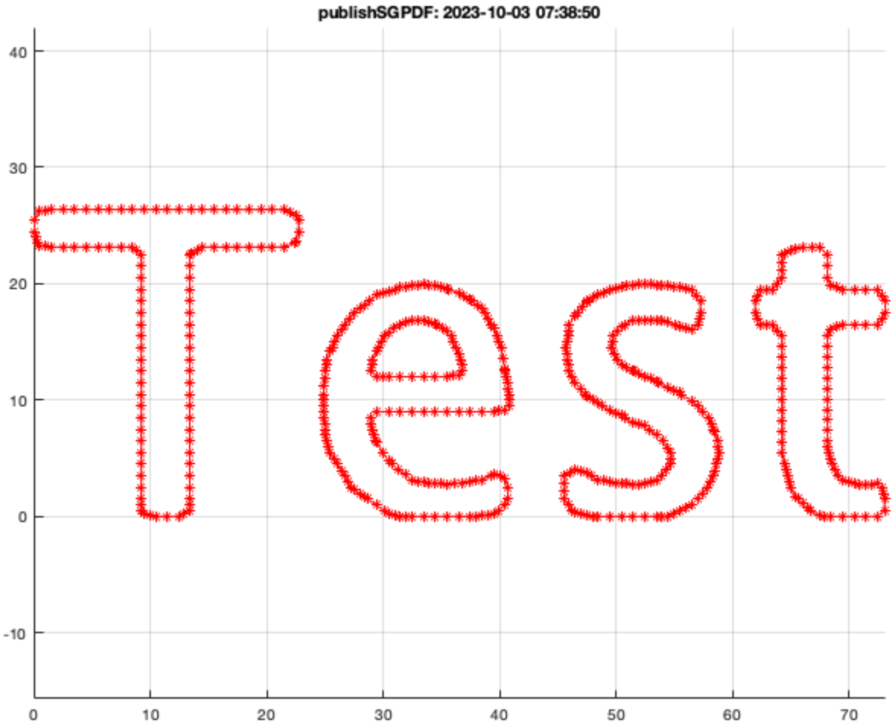


```
CPL=CPLofPLEL(PL,EL);
SGfigure; view(0,90); CPLplot(CPL);
```

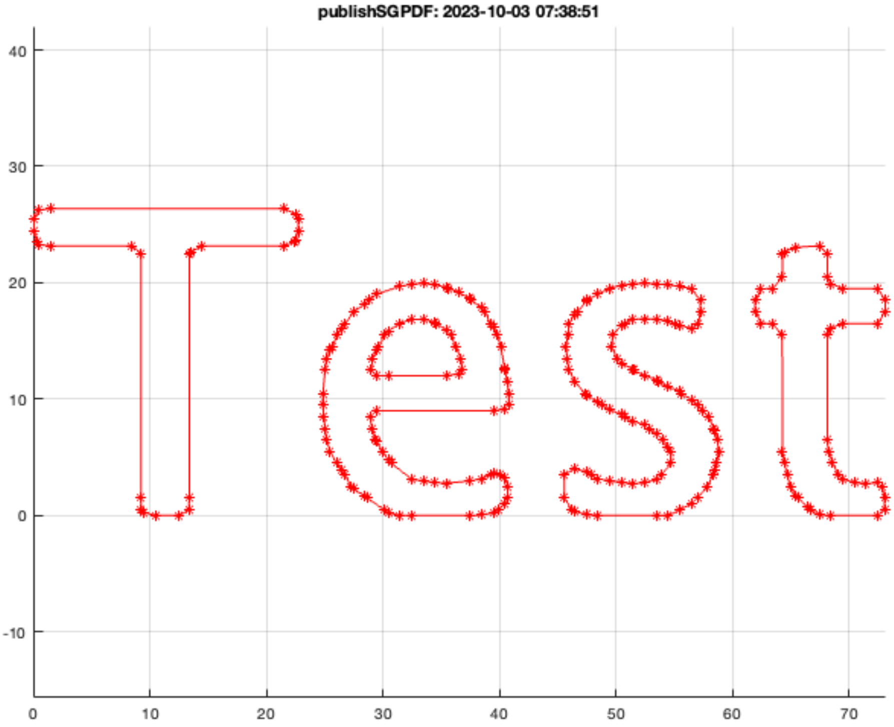


3. Adding and removing points on the contour

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

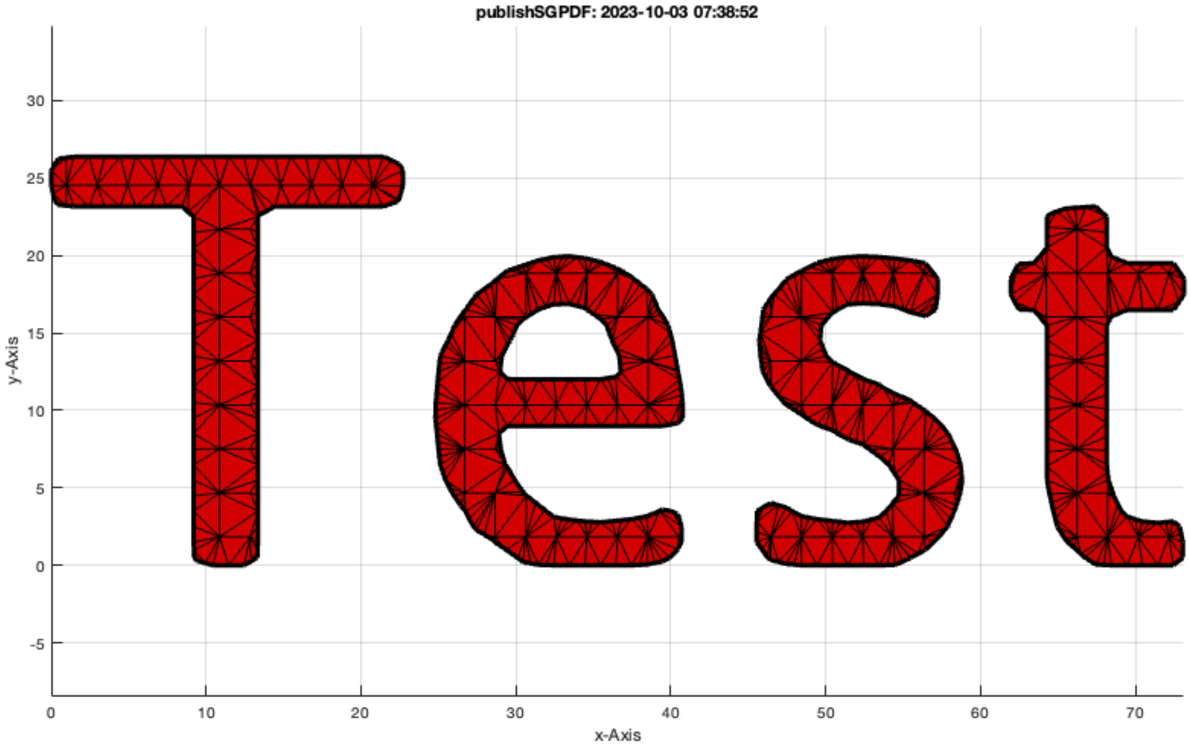


```
CPLB=CPLremstraight(CPL);
SGfigure; view(0,90); CPLplot(CPLB);
```



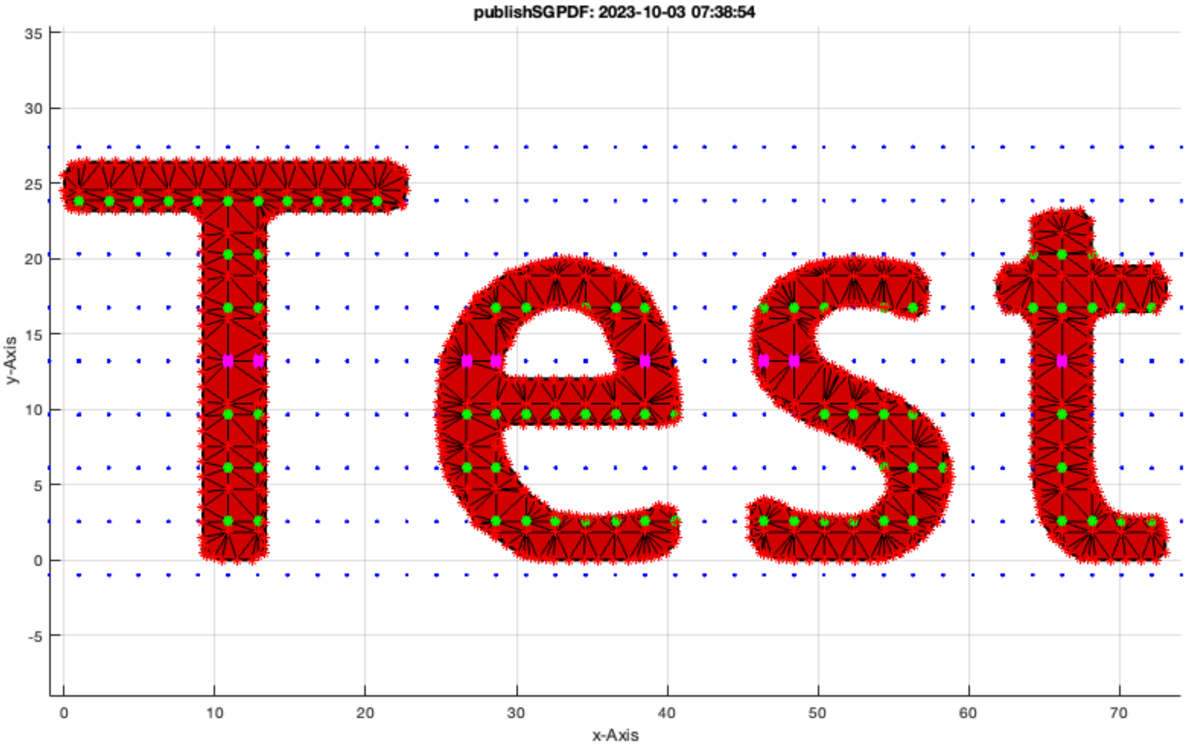
4. Adding and removing points inside of the contour

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



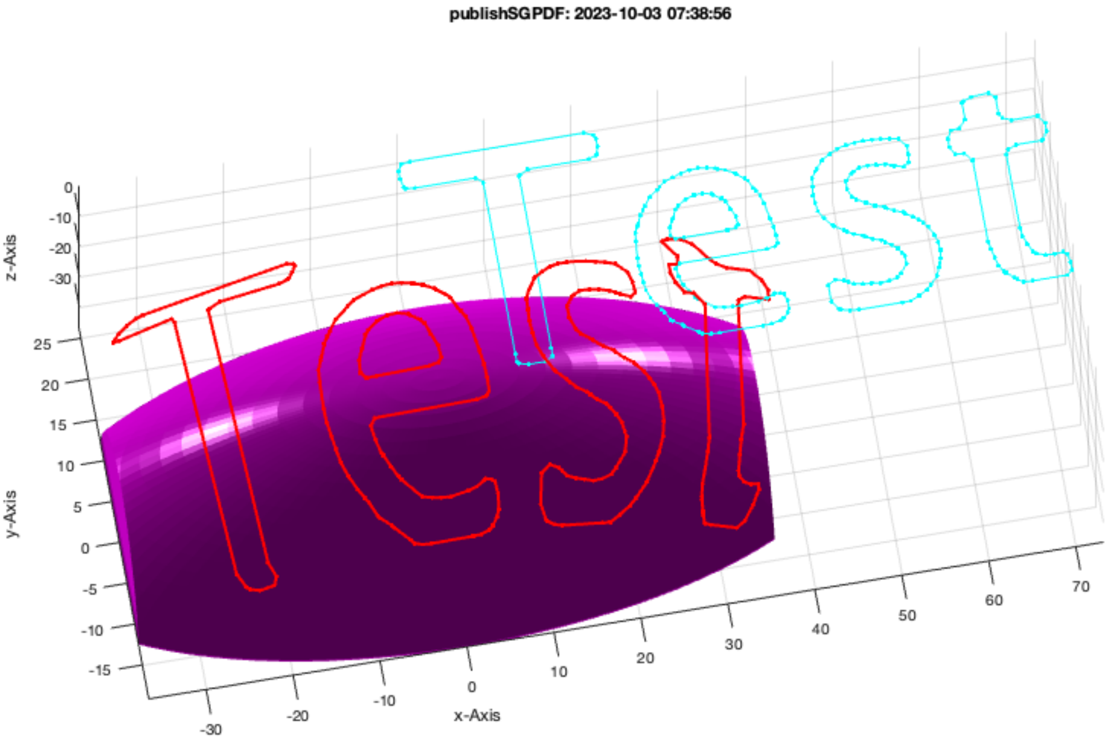
5. Calculate Grid Points

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

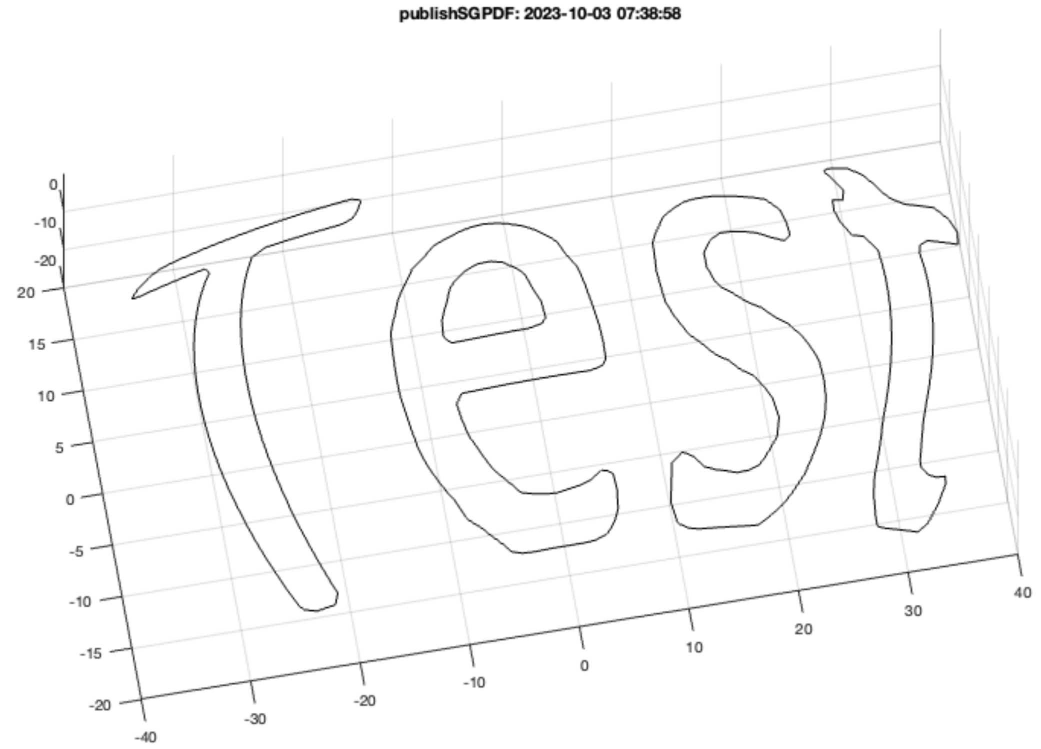


6. Bending of contours

```
% Without auxiliary points

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

% With auxiliary points
```

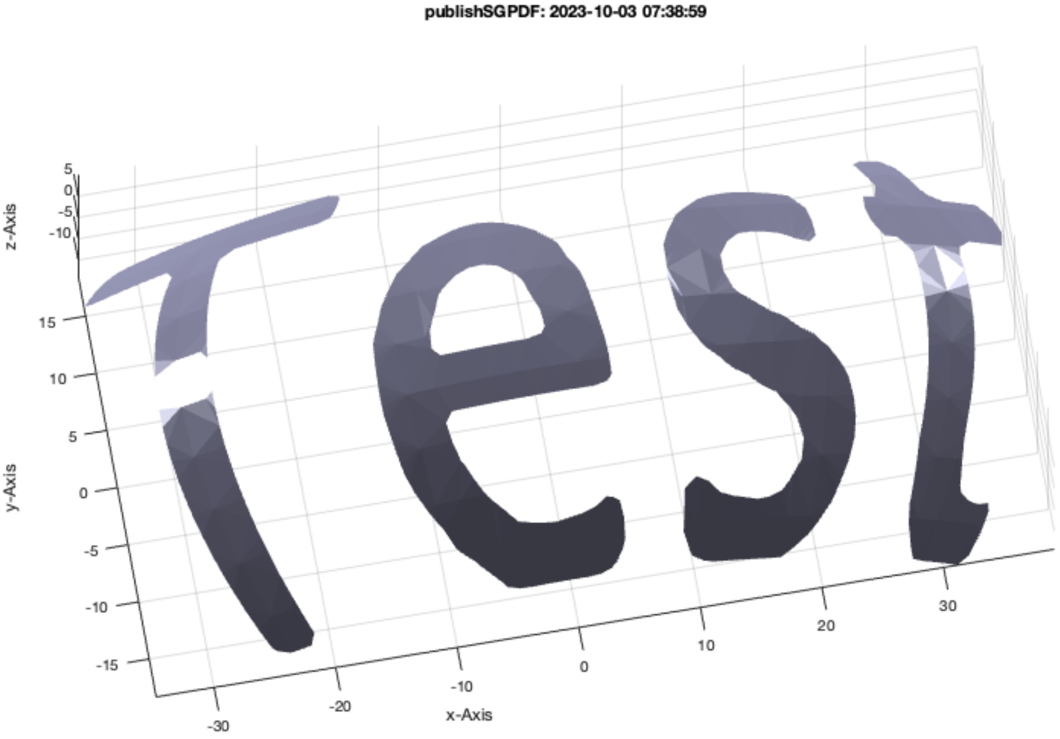


```
[PL,FL,EL]=PLFLofCPLdelaunayGrid(CPL,1,2,3);
NPL=PLbending(PL,50,10,20);
SGfigure; view(-10,70); VLELplot(NPL,EL,'k-');
```



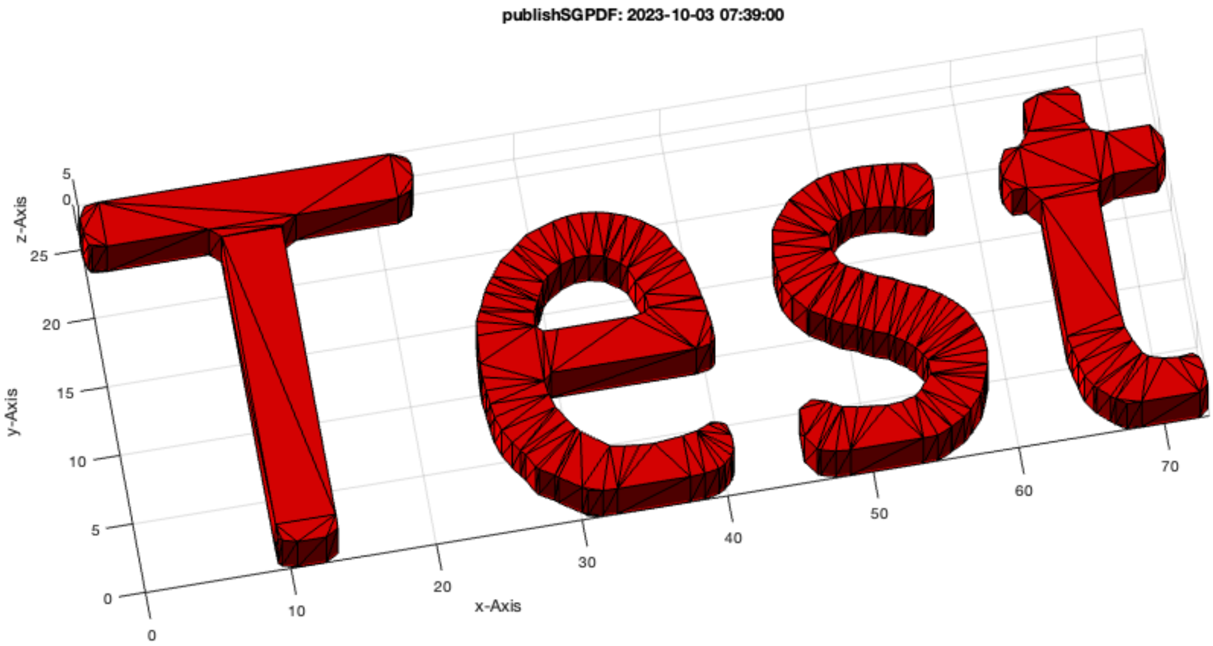
7. Bending of closed contour surfaces

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

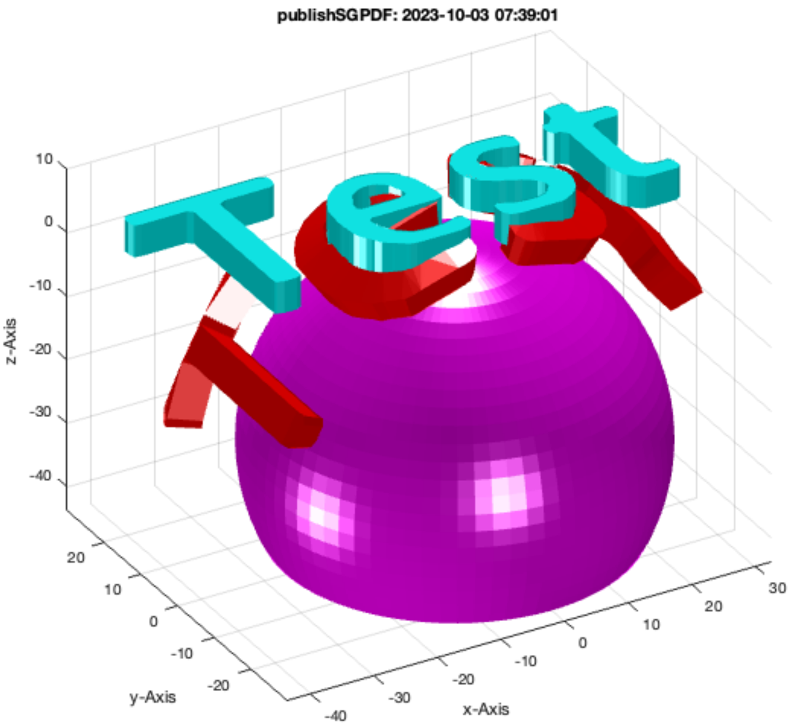


8. Bending of solid geometries

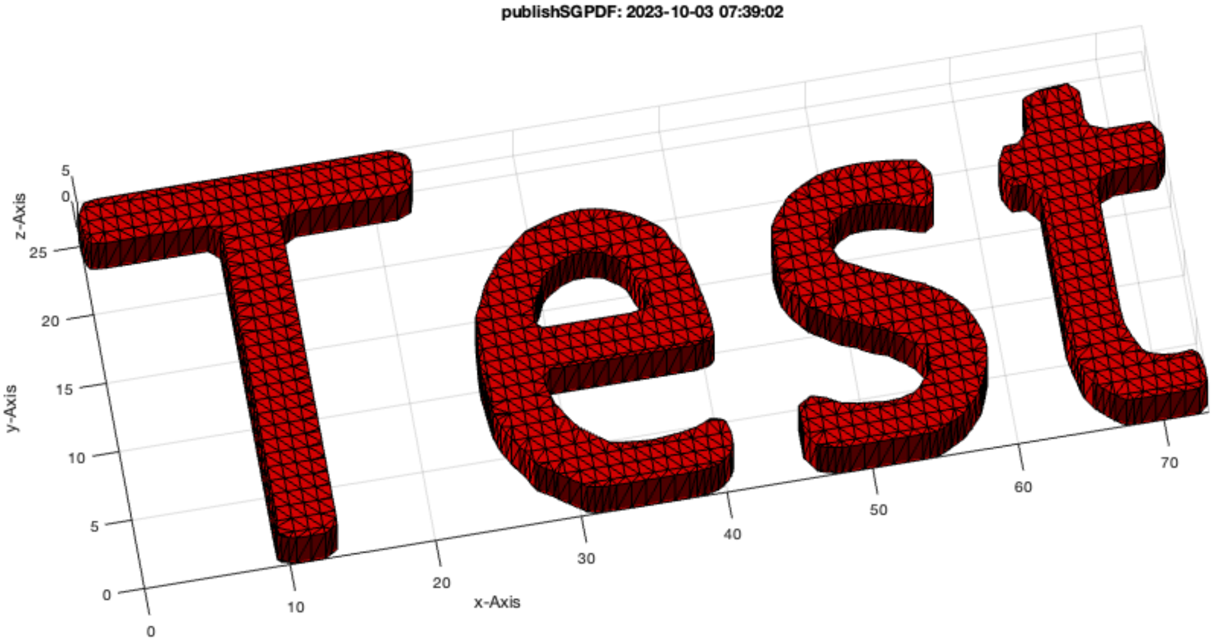
```
% Without auxiliary points

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



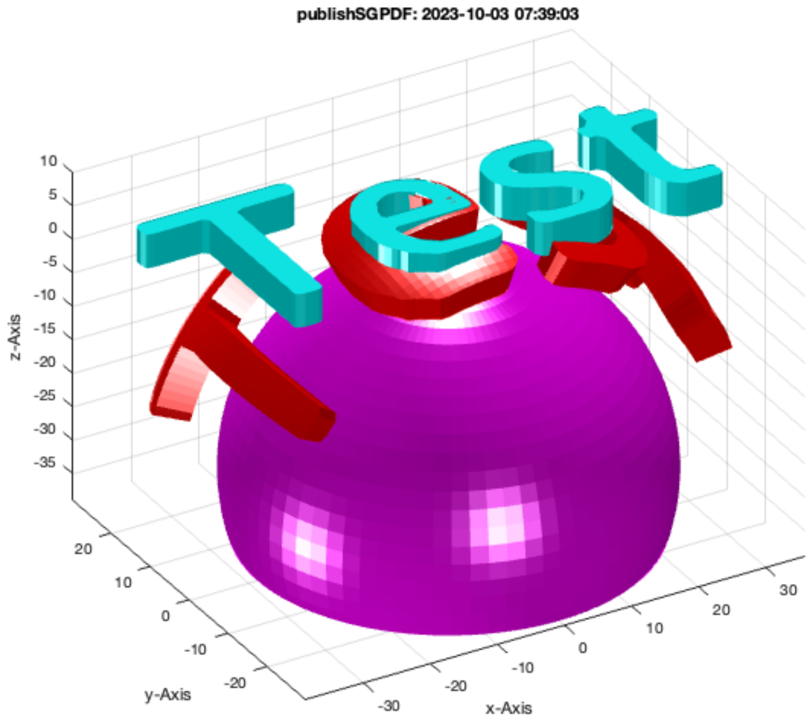
```
SGbending(SG,30,5,30); VLFLplotlight(1,1);
```



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

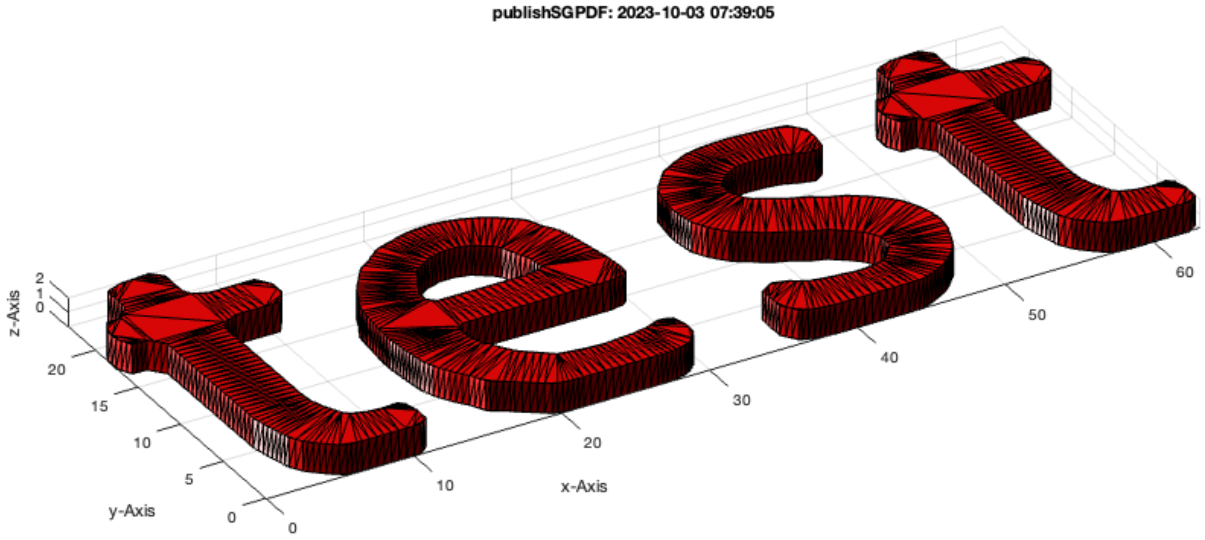


```
SGbending(SG,30,5,30); VLFLplotlight(1,1);
```

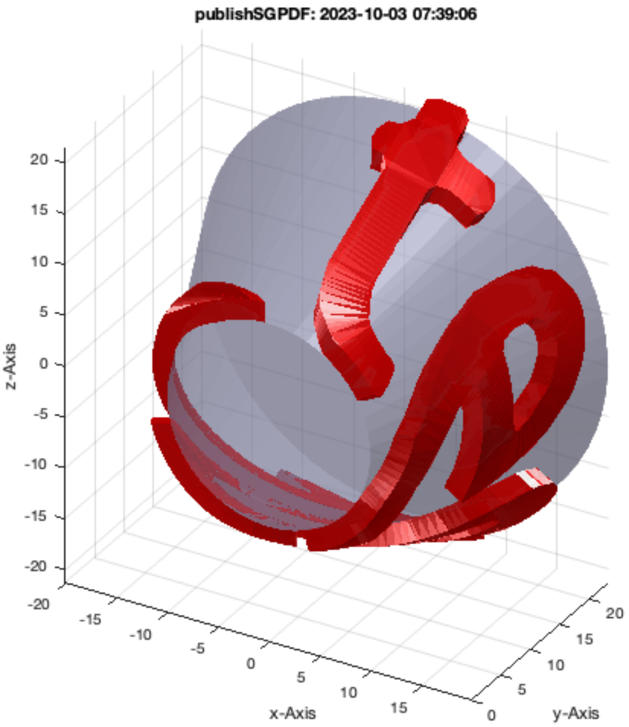


The following Example is from Nicholas Moser, Jan 2020

```
SGofCPLzdelaunayGrid(CPLoftext('test'),2,.5); SG=ans;
```



```
SGbendcone(SG,10,20); view(30,30)
```



Final remarks on toolbox version and execution date

```
VLFLlicense
close all
```

```
This VLFL-Lib, Rel.  (2023-Oct-03), is for limited non commercial educational use only!
Licensee: Tim Lueth (Development Version)!
Please contact Tim Lueth, Professor at TU Munich, Germany!
WARNING: This VLFL-Lib (Rel. ) license will exceed at 06-Jul-2078 07:39:06!
Executed 03-Oct-2023 07:39:08 by 'timlueth' on a MACI64 using Mac OSX 13.6 | R2023a Update 5 | SG-Lib 5.4
======================================== Used Matlab products: ======================================
distrib_computing_toolbox
map_toolbox
matlab
=====================================================================================================
```

- *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® R2023a