

Tutorial 50: CVLof2CPLzcorrelate and SGof2CPLzcorrelate

2020-08-28: Tim C. Lueth, Professor at Technische Universität München, Germany (URL: <http://www.SG-Lib.org>) - Last Change: 2020-08-28

Contents

- [Complete List of all Tutorials with Publishable MATLAB Files of this Solid-Geometries Toolbox](#)
- [Motivation for this tutorial: \(Originally SolidGeometry 5.0 required\)](#)
- [CPLsample creates two complex CPLs for creating a solid](#)
- [CPLcorrelates\(CPLA,CPLB\) find the best fitting pairs of contour of CPLA and CPLB](#)
- [CPLcorrelates\(CPLA,CPLB\) the best non exclusive fitting pairs of contour of CPLA and CPLB](#)
- [PLtransform creates an intermediate contour between two single contours \(2020 Version_](#)
- [There are tricky situations if the number of points and the orientation differ](#)
- [Therefore PLtransform_2020 replaces the former version PLtransform_2020](#)
- [CPLtransform shows an intermediate transformation level between the two CPLs](#)
- [CVLof2CPLzcorrelate creates correlated and fused intermediate layers between the two CPLs](#)
- [CVLof2CPLzcorrelate creates even stack in z](#)
- [CVLof2CPLzcorrelate is quite robust in both directions](#)
- [SGof2CPLzcorrelate creates CVLz stack first and then creates the surfaces](#)
- [SGof2CPLzcorrelate creates CVLz stack first and then creates the surfaces](#)

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 Coordinate 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 Titles, Endtitles 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: Create a Solid by two arbitrary CPLs and a distance
- Tutorial 50: CVLof2CPLzcorrelate and SGof2CPLzcorrelate
- Tutorial 51: Creating Parallel Tasks for batch processing
- 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

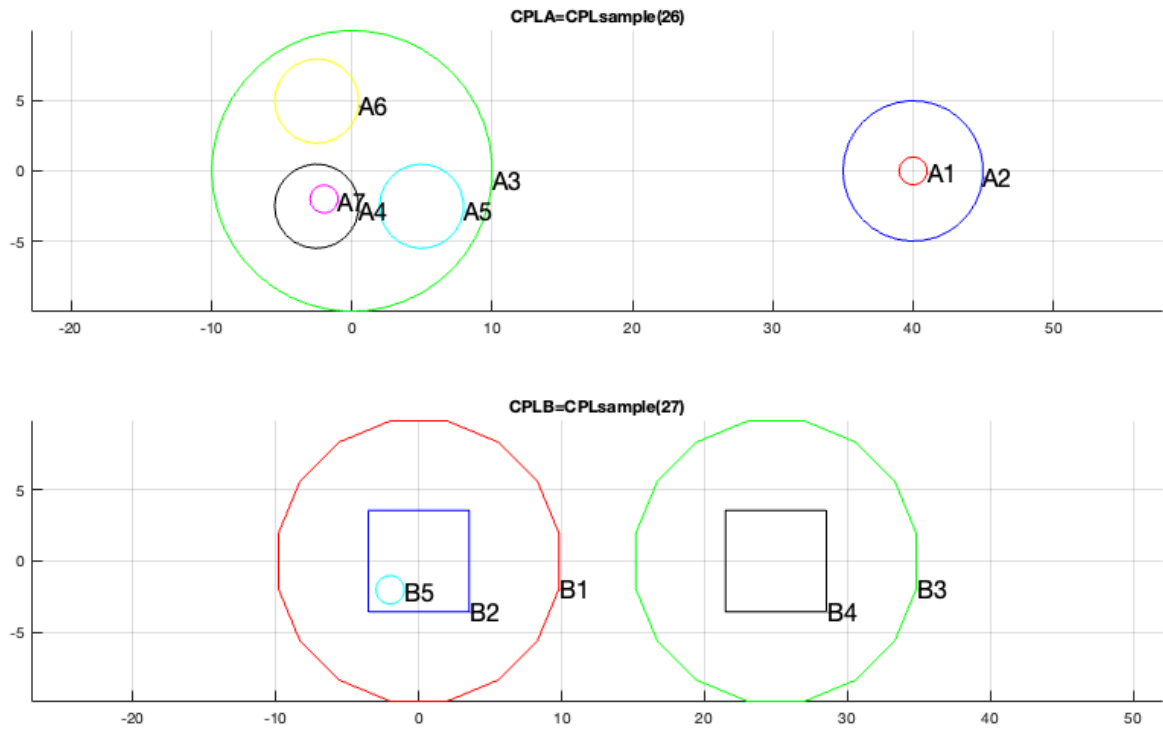
Motivation for this tutorial: (Originally SolidGeometry 5.0 required)

```
CPLplot plot all contour in one color and closed
CVLplot plot all contour in one color and open
CVLplots plot all contour in one color per segment and closed
CVLzplot plot all contour in one color and open
CPLplotcolors plots all contour in different colors and closed
```

CPLsample creates two complex CPLs for creating a solid

```
dbprintf('00 - Create the test condition');
SGfigure(0,90);
CPLA=CPLsample(26);
CPLB=CPLsample(27);
subplot(2,1,1); CPLplotcolors(CPLA,'',2); title('CPLA=CPLsample(26)'); viewCPL; textCPL(CPLA,'','','A');
subplot(2,1,2); CPLplotcolors(CPLB,'',2); title('CPLB=CPLsample(27)'); viewCPL; textCPL(CPLB,'','','B');
```

```
VLFL_EXP50: 00 - Create the test condition
```

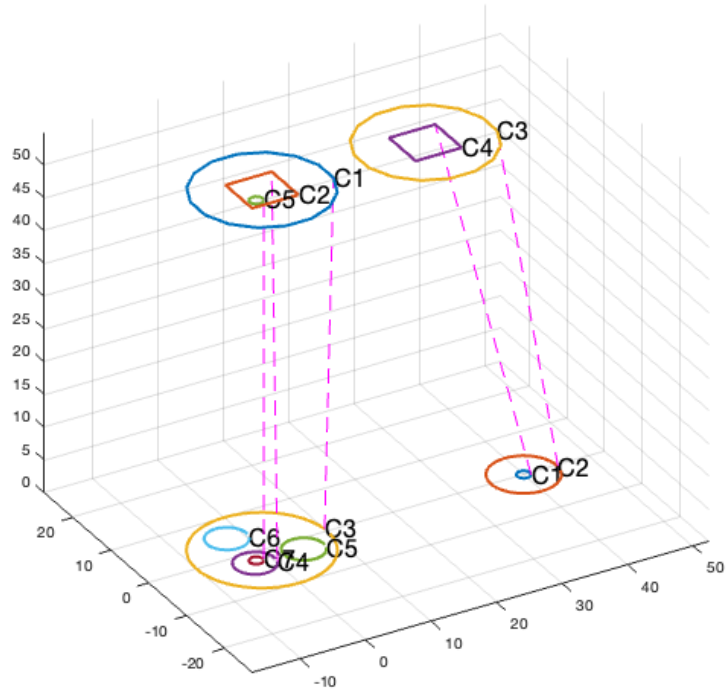


CPLcorrelates(CPLA,CPLB) find the best fitting pairs of contour of CPLA and CPLB

```
dbprintf('01 - CPLcorrelate(CPLA,CPLB,false)');
CPLcorrelate(CPLA,CPLB,false)
drawnowvid;
```

```
VLFL_EXP50: 01 - CPLcorrelate(CPLA,CPLB,false)
ans =
    3     1     0     0     0
    2     3     0     0     0
    4     2     1     3     1
    1     4     1     2     3
    7     5     2     4     2
```

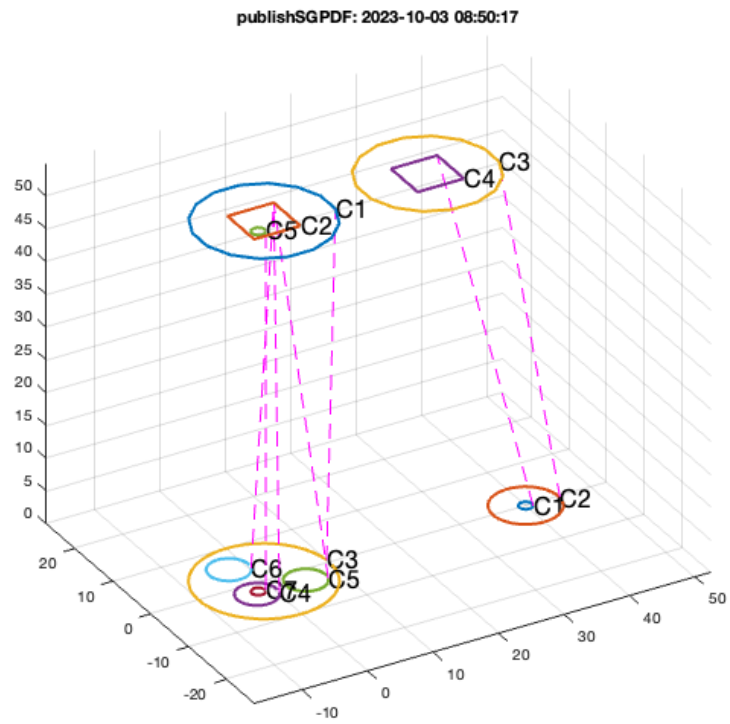
publishSGPDF: 2023-10-03 08:50:16



CPLcorrelates(CPLA,CPLB) the best non exclusive fitting pairs of contour of CPLA and CPLB

```
dbprintf('02 - CPLcorrelate(CPLA,CPLB,true)');
CPLcorrelate(CPLA,CPLB,true)
drawnowvid;
```

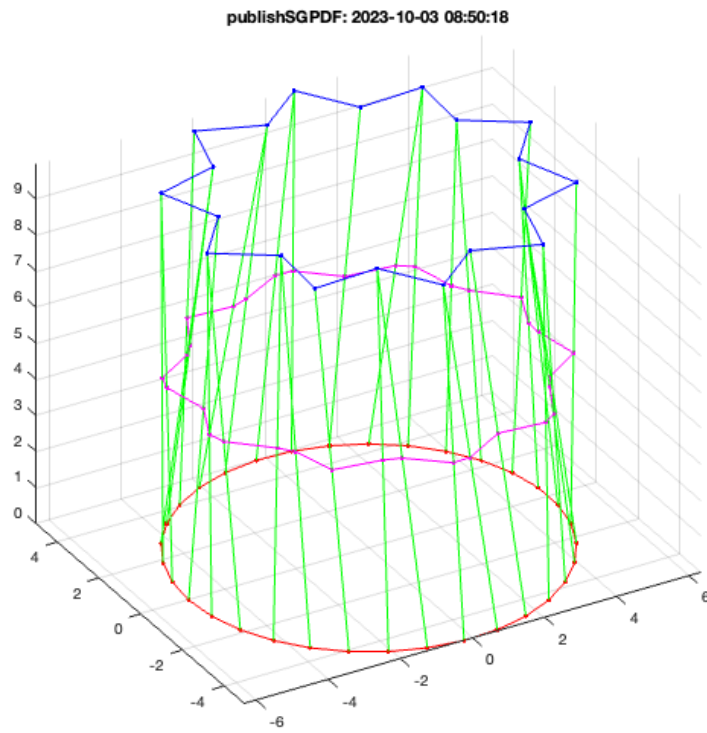
```
VLFL_EXP50: 02 - CPLcorrelate(CPLA,CPLB,true)
ans =
     3     1     0     0     0
     2     3     0     0     0
     4     2     1     3     1
     5     2     1     3     1
     6     2     1     3     1
     1     4     1     2     3
     7     5     2     4     2
```



PLtransform creates an intermediate contour between two single contours (2020 Version_

```
dbprintf('03 - PLtransform(PLcircle(5),PLstar(5,20)); % Version LIB');
PLtransform(PLcircle(5),PLstar(5,20));
drawnowvid;
```

```
VLFL_EXP50: 03 - PLtransform(PLcircle(5),PLstar(5,20));
```

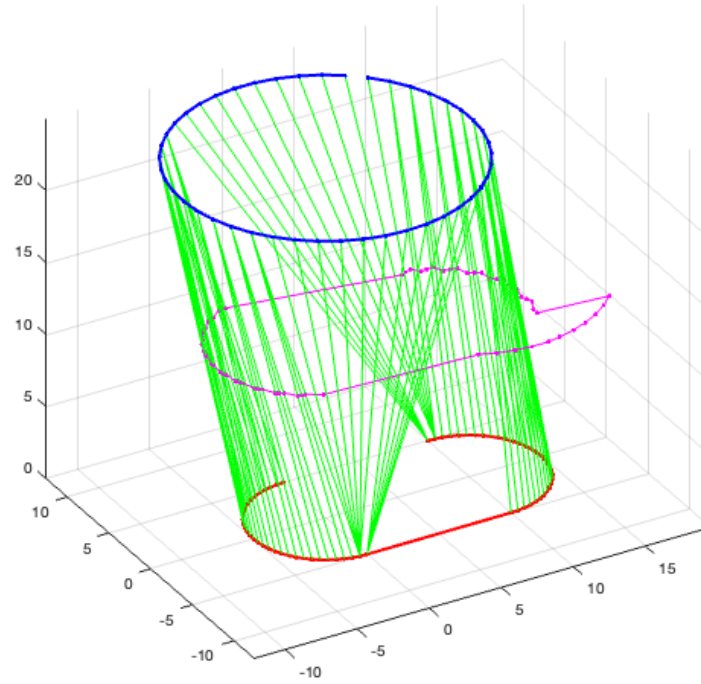


There are tricky situations if the number of points and the orientation differ

```
dbprintf('04 - PLtransform_2018(PLcircle(5),PLstar(5,20));');
PLtransform_2018(PLcircle(5),'',10),PLcircle(10,'','','',1));
drawnowvid;
```

```
VLFL_EXP50: 04 - PLtransform_2018(PLcircle(5),PLstar(5,20));
PLcorrelationcheck: [publishSGPDF] ERROR CL contains criss cross links: [32; 44; 76]
```

publishSGPDF: 2023-10-03 08:50:20

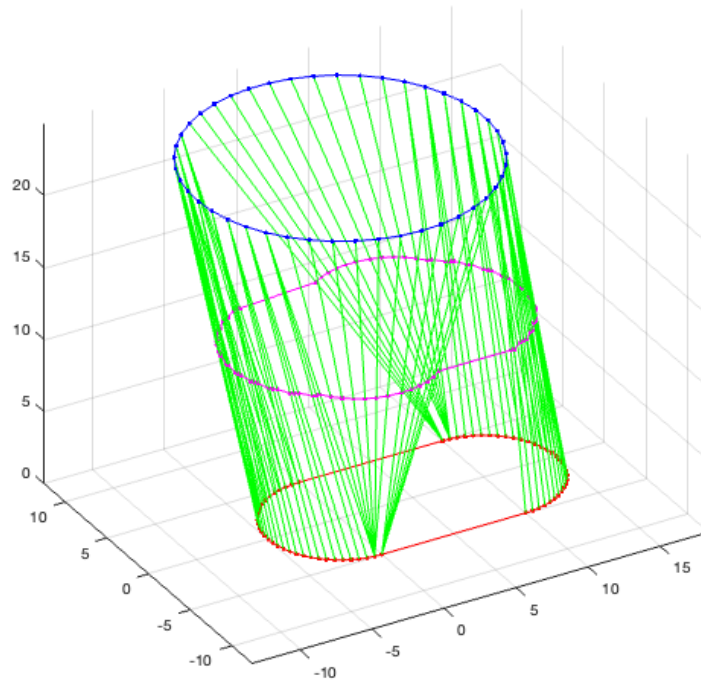


Therefor PLtransform_2020 replaces the former version PLtransform_2020

```
dbprintf('04 - PLtransform_2020(PLcircle(5),PLstar(5,20));');
PLtransform_2020(PLcircleoval(5,'',10),PLcircle(10,'',' ',1));
drawnowvid;
```

```
VLFL_EXP50: 04 - PLtransform_2020(PLcircle(5),PLstar(5,20));
PLcorrelationcheck: [publishSGPDF] ERROR CL contains criss cross links: [32; 44; 76]
s =
    24.9974
ans =
     1     11
     2     11
     3     11
     4     12
     5     12
     6     12
     7     13
     8     13
     9     13
    10     14
```

publishSGPDF: 2023-10-03 08:50:21



CPLtransform shows an intermediate transformation level between the two CPLs

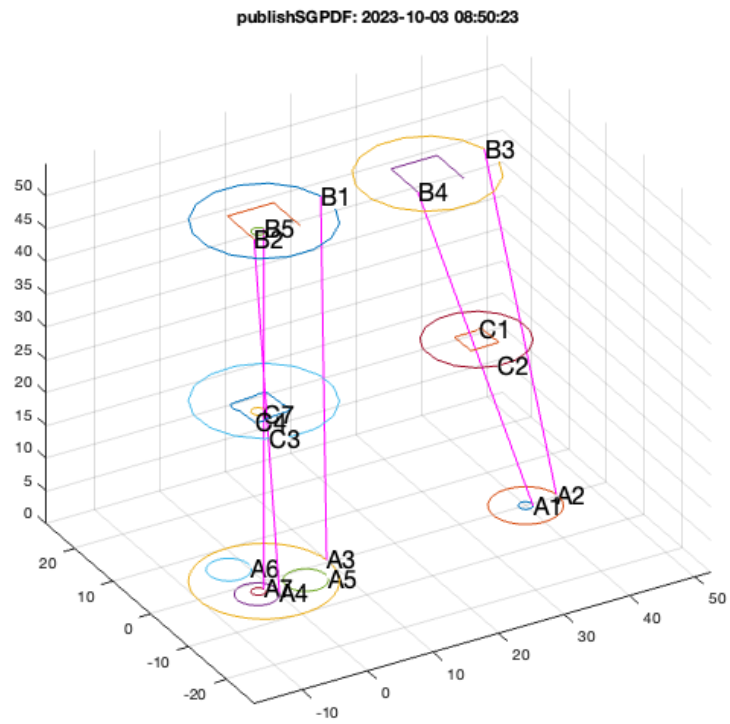
Each separated contour is transformed to another using CPLcorrelate and PLtransform Nevertheless the contours are not merged if they overlapp

```
dbprintf('05 - CPLtransform(CPLA,CPLB);');
CPLtransform(CPLA,CPLB);
drawnowvid;
```

```
VLFL_EXP50: 05 - CPLtransform(CPLA,CPLB);
PLcorrelatingsegments: CPLA has 45 points, CPLB has 16 points.
PLcorrelatingsegments: CPLA has 12 segments, CPLB has 12 segments by using angle 30.0.
PLcorrelatingsegments: CPLA has 33 points, CPLB has 16 points.
PLcorrelatingsegments: CPLA has 12 segments, CPLB has 12 segments by using angle 30.0.
PLcorrelatingsegments: CPLA has 25 points, CPLB has 4 points.
PLcorrelatingsegments: CPLA has 12 segments, CPLB has 3 segments by using angle 30.0.
PLcorrelatingsegments: CPLA has 15 points, CPLB has 4 points.
PLcorrelatingsegments: CPLA has 12 segments, CPLB has 3 segments by using angle 30.0.
PLcorrelatingsegments: CPLA has 15 points, CPLB has 15 points.
PLcorrelatingsegments: CPLA has 12 segments, CPLB has 12 segments by using angle 30.0.
```

CLL_tab =
5x5 table

Index_A	Index_B	Enclosure_Index	Parent_A	Parent_B
3	1	0	0	0
2	3	0	0	0
4	2	1	3	1
1	4	1	2	3
7	5	2	4	2

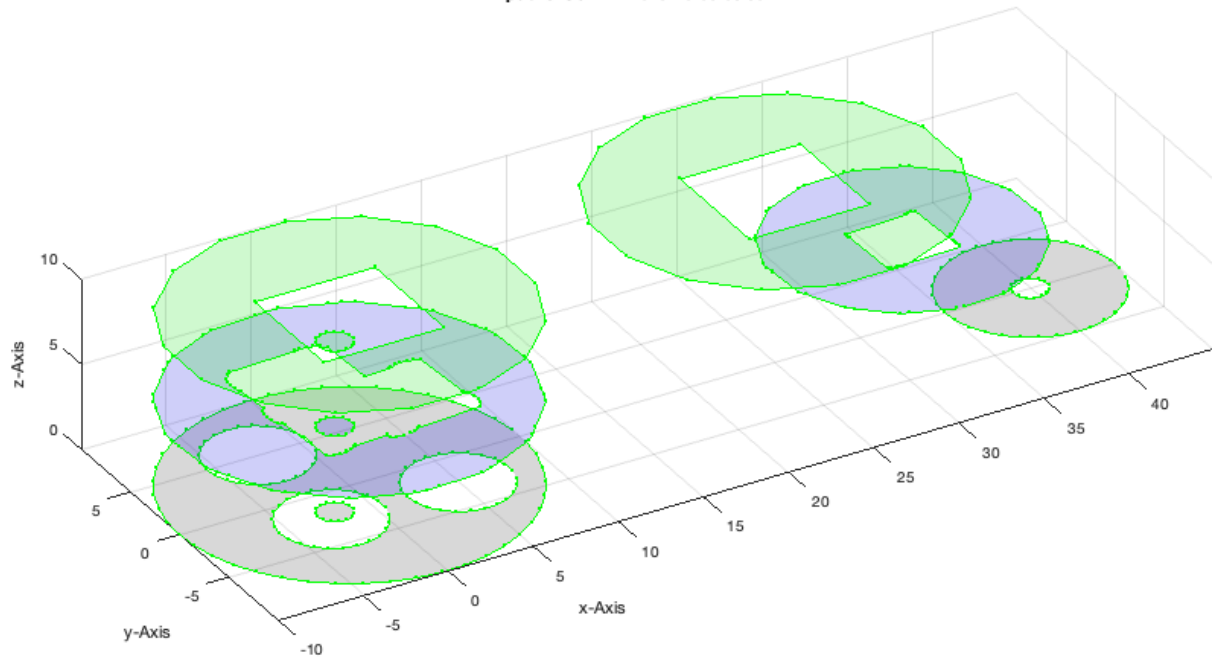


CVLoF2CPLzcorrelate creates correlated and fused intermediate layers between the two CPLs

```
dbprintf('06 - CVLoF2CPLzcorrelate(CPLA,CPLB);');
CVLoF2CPLzcorrelate(CPLA,CPLB);
drawnowvid;
```

```
VLFL_EXP50: 06 - CVLoF2CPLzcorrelate(CPLA,CPLB);
```

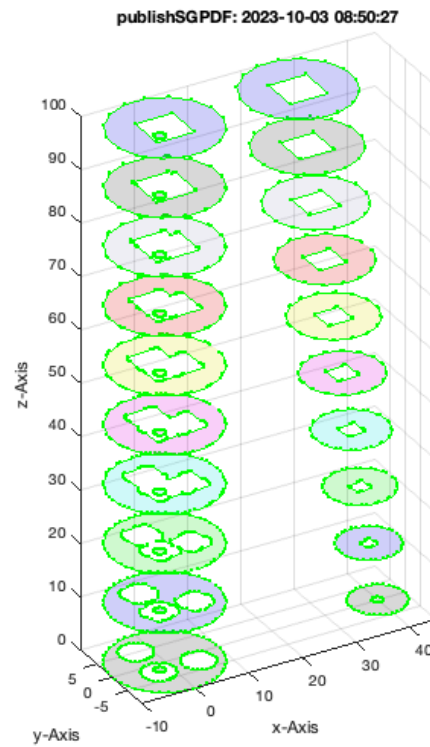
publishSGPDF: 2023-10-03 08:50:24



CVLoF2CPLzcorrelate creates even stack in z

```
dbprintf('07 - CVLoF2CPLzcorrelate(CPLA,CPLB,[100 8]); ');
CVLoF2CPLzcorrelate(CPLA,CPLB,[100 8]);    % z= 100  n=1 + 8 + 1
drawnowvid;
```

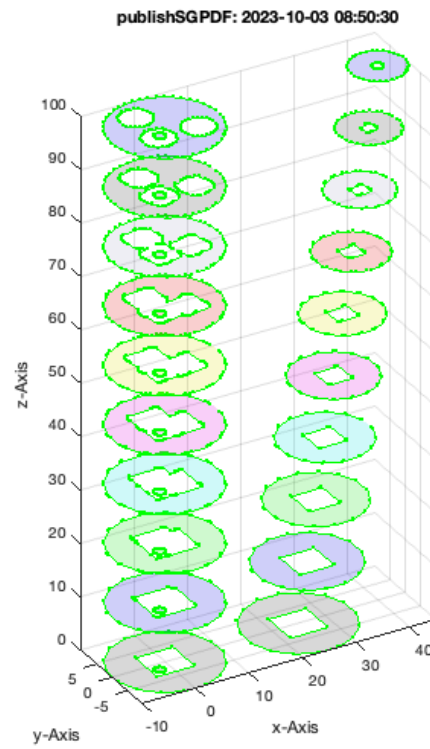
```
VLFL_EXP50: 07 - CVLoF2CPLzcorrelate(CPLA,CPLB,[100 8]);
CVLoF2CPLzcorrelate: 20%Warning: Intersecting edge constraints have been split, this may have added new
points into the triangulation.
Warning: Intersecting edge constraints have been split, this may have added new
points into the triangulation.
Warning: Intersecting edge constraints have been split, this may have added new
points into the triangulation.
```



CVLoF2CPLzcorrelate is quite robust in both directions

```
dbprintf('08 - CVLoF2CPLzcorrelate(CPLB,CPLA,[100 8]); ');
CVLoF2CPLzcorrelate(CPLB,CPLA,[100 8]);    % z= 100  n=1 + 8 + 1
drawnowvid;
```

```
VLFL_EXP50: 08 - CVLoF2CPLzcorrelate(CPLB,CPLA,[100 8]);
CVLoF2CPLzcorrelate: 20%Warning: Intersecting edge constraints have been split, this may have added new
points into the triangulation.
Warning: Intersecting edge constraints have been split, this may have added new
points into the triangulation.
Warning: Intersecting edge constraints have been split, this may have added new
points into the triangulation.
Warning: Intersecting edge constraints have been split, this may have added new
points into the triangulation.
```

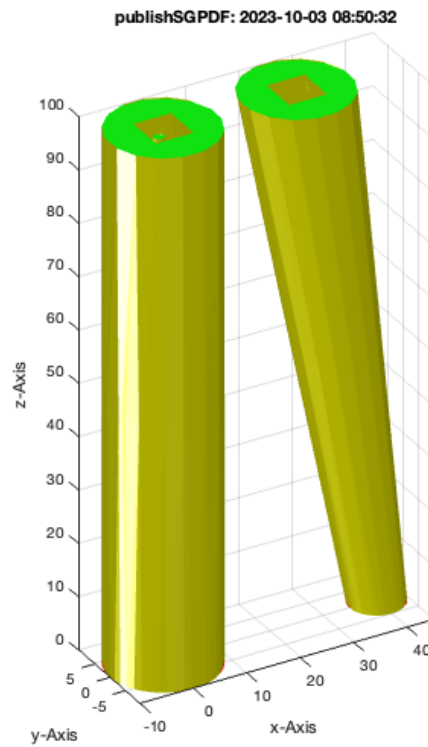


SGof2CPLzcorrelate creates CVLz stack first and then creates the surfaces

```
dbprintf('09 - SGof2CPLzcorrelate(CPLA,CPLB,100);');
SGof2CPLzcorrelate(CPLA,CPLB,100)
drawnowvid;
```

```
VLFL_EXP50: 09 - SGof2CPLzcorrelate(CPLA,CPLB,100);
ans =
  struct with fields:

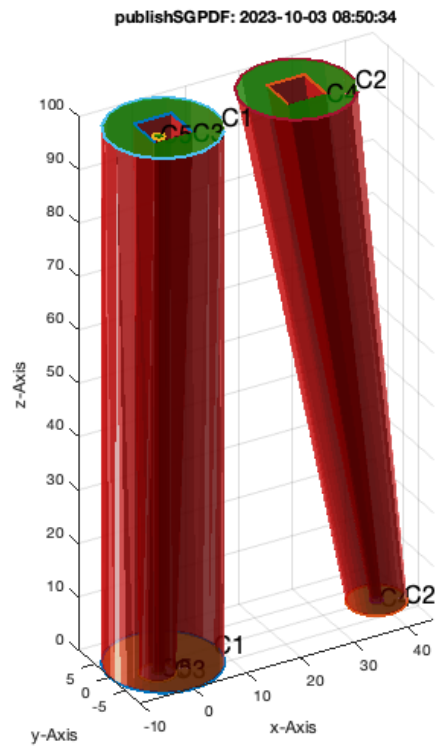
    VL: [593x3 double]
    FL: [1190x3 double]
    FC: [1190x3 double]
```



SGof2CPLzcorrelate creates CVLz stack first and then creates the surfaces

```
dbprintf('10 - SGof2CPLzheurist(CPLA,CPLB)');
SGof2CPLzheurist(CPLA,CPLB,100)
drawnowvid;
```

```
VLFL_EXP50: 10 - SGof2CPLzheurist(CPLA,CPLB)
ans =
  struct with fields:
    VL: [188x3 double]
    FL: [372x3 double]
    col: 'w'
    alpha: 0.9000
```



Published with MATLAB® R2023a