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
- 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 2.5.1 required)

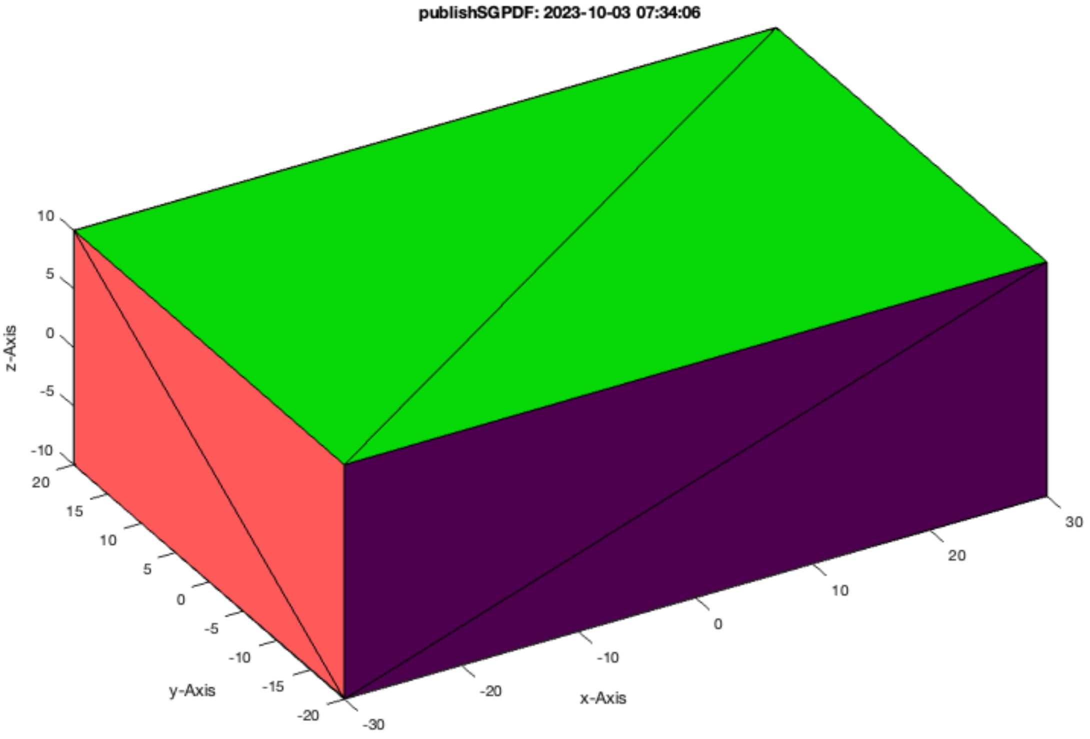
2. Analyzing mouting faces of flat surfaces

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);
```


publishSGPDF: 2023-10-03 07:34:06

- 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
```

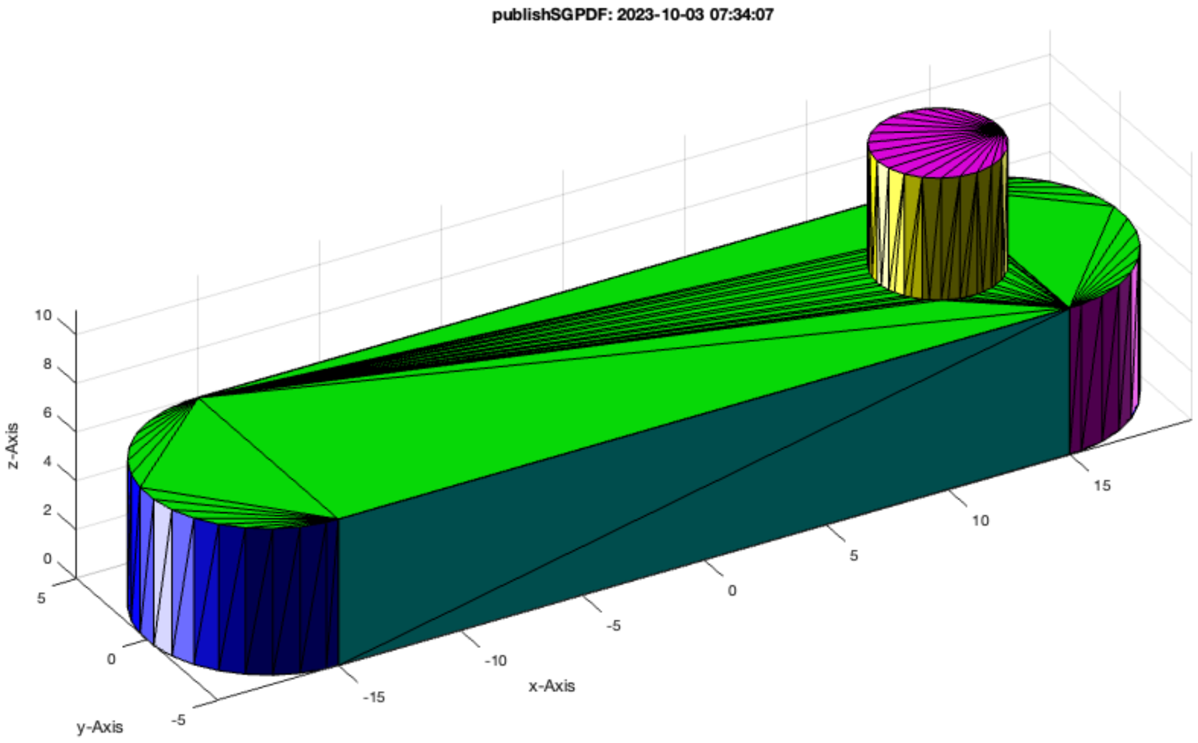
```
SG = 
  struct with fields:

       VL: [8×3 double]
       FL: [12×3 double]
    Tname: {'B'  'F'  'X+'  'X-'  'Y+'  'Y-'}
        T: {1×6 cell}
     TFiL: {[]  []  []  []  []  []}
       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

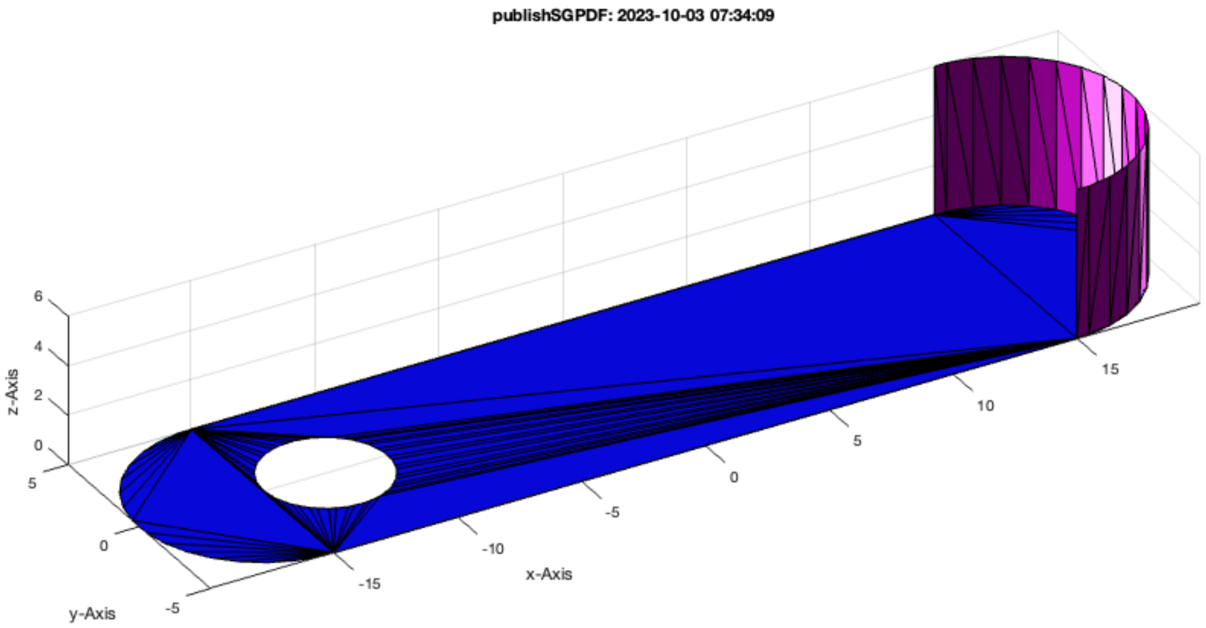
```
close all; SGfigure; view (-30,30);
load AIM_SGrobot
[ML,MA,SG]=MLofSG(SG2);
MLplot(SG);
```



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
```



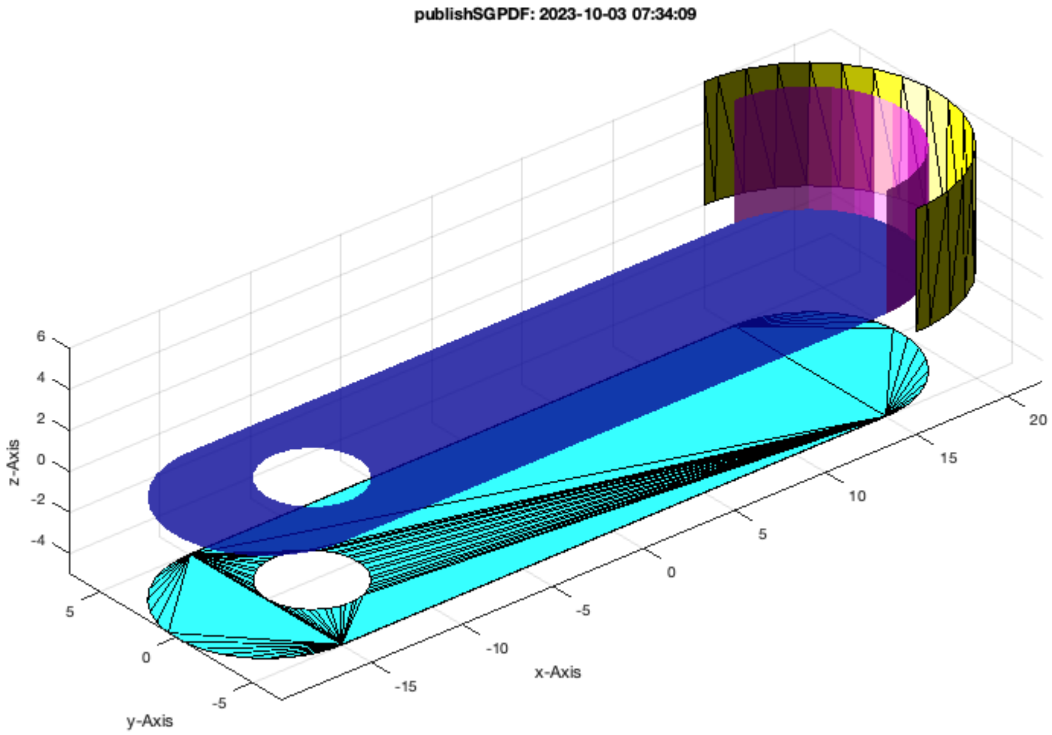
4. Create corresponding surfaces parallel to mounting faces

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');
```

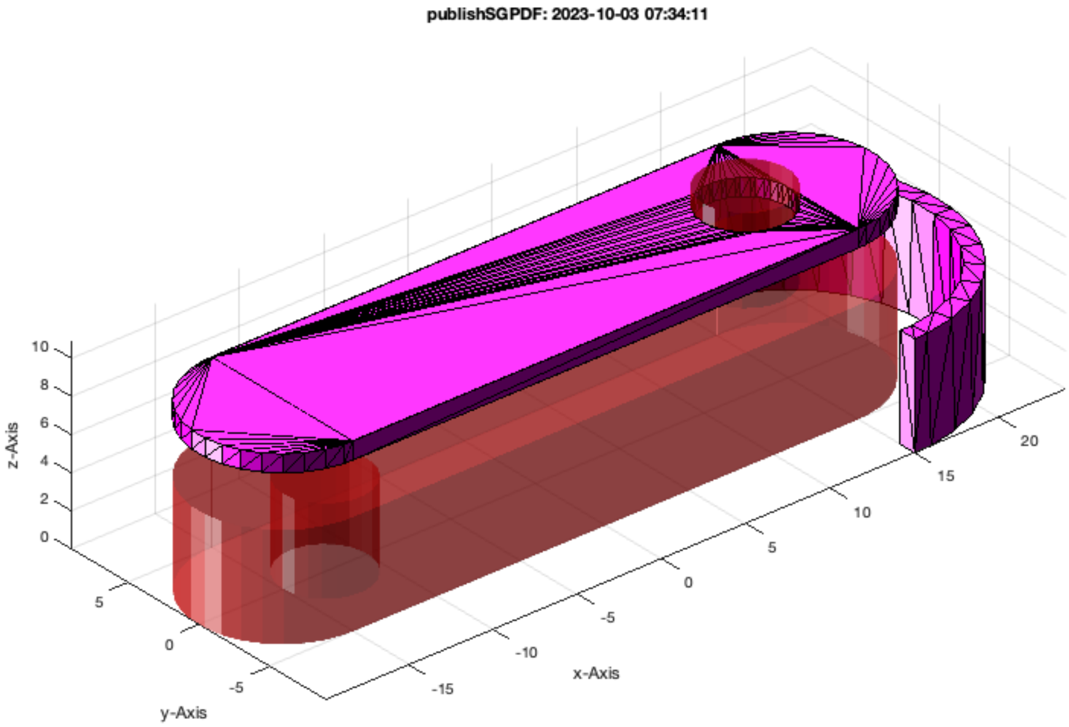


5. Create solids using the parallel surfaces and a plate thickness

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');
```

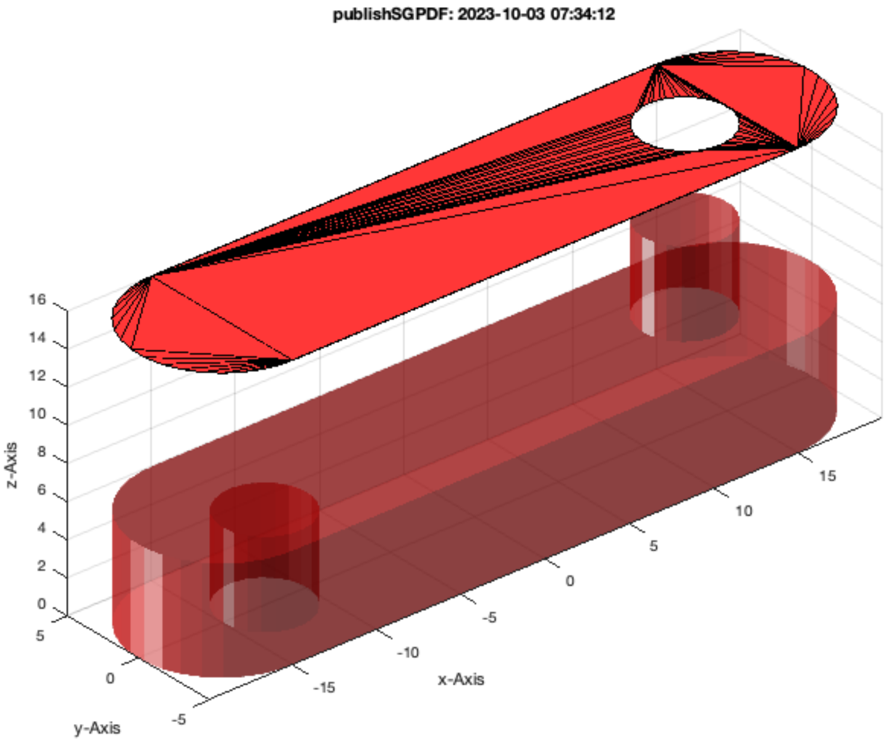


6. Finding the 2D CPL of a planar 3D Surface

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.

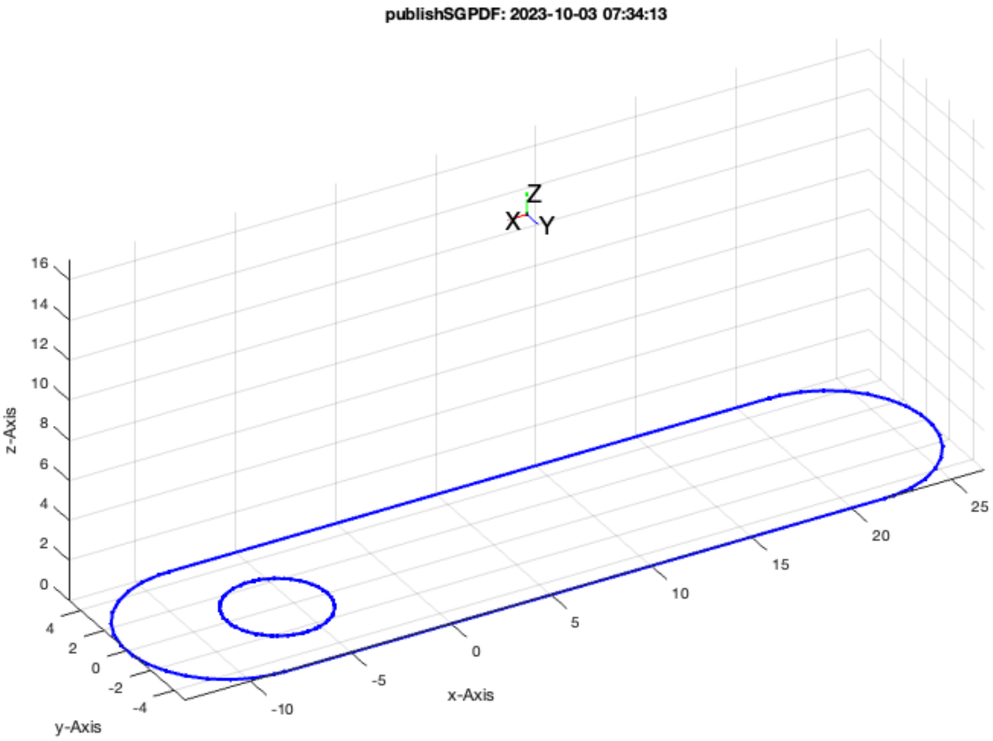
- [PL,T]=PLofVLFL(VL,FL) - returns a PL and a transformation matrix
- [CPL,T]=CPLofVLFL(VL,FL) - returns a CPL and a transformation matrix

```
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);
```



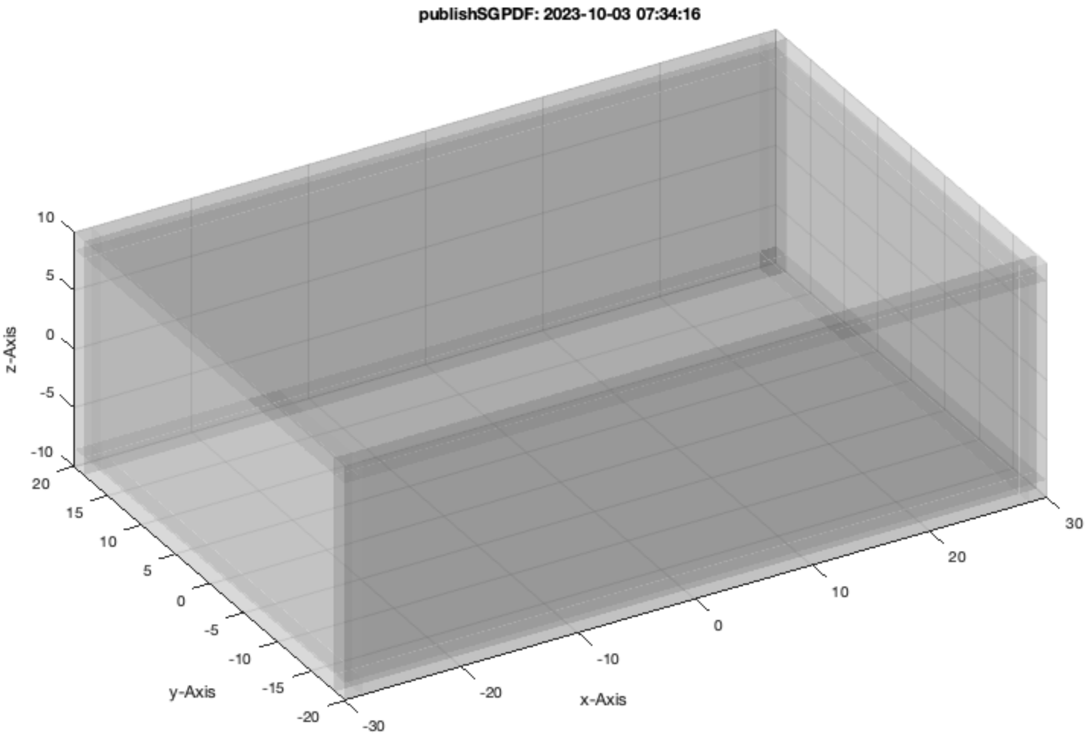
7. Replace a solid block by covering plates

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

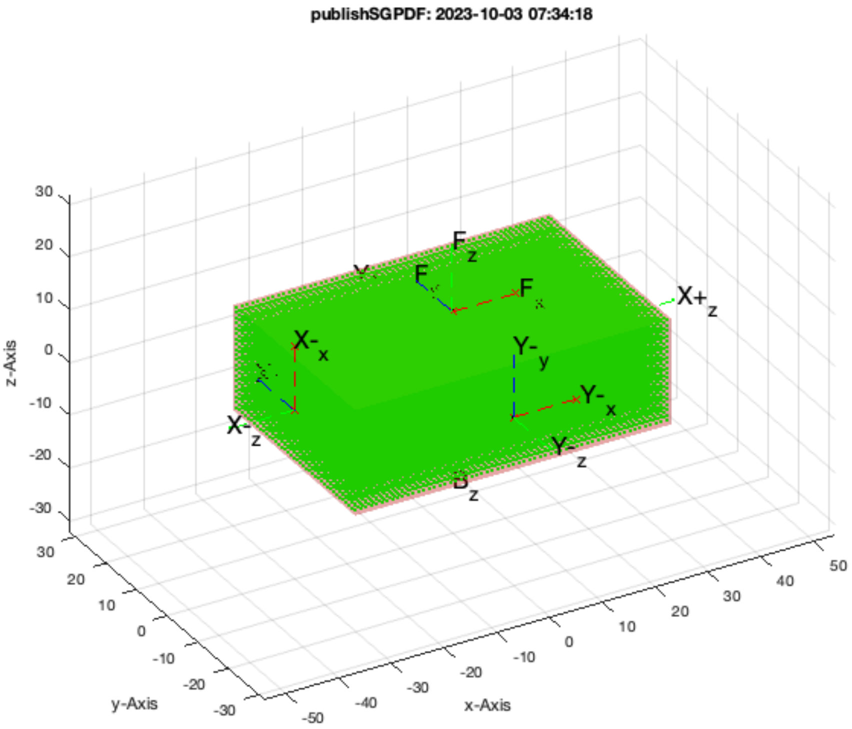
```
close all; SGfigure; view (-30,30);

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



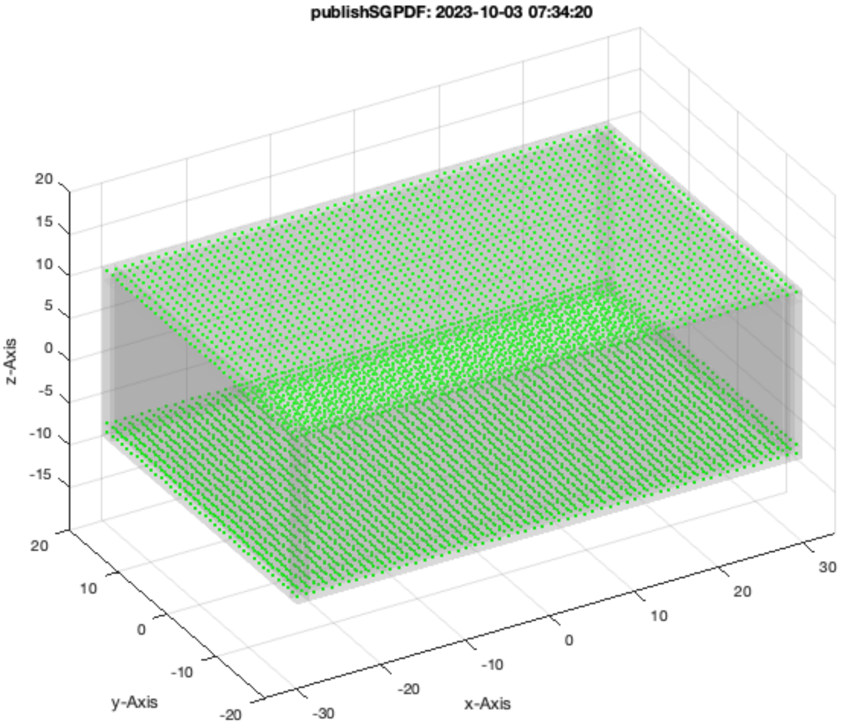
```
SGweight(SGbox([60,40,20]),[],1);
```

```
Using a resolution of 1.0 mm^3 (n=57288) and a specific weight of 1.15 milligramm per mm^3, the overall weight is ca. 55 gramm.
Elapsed time is 1.490539 seconds.
```



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

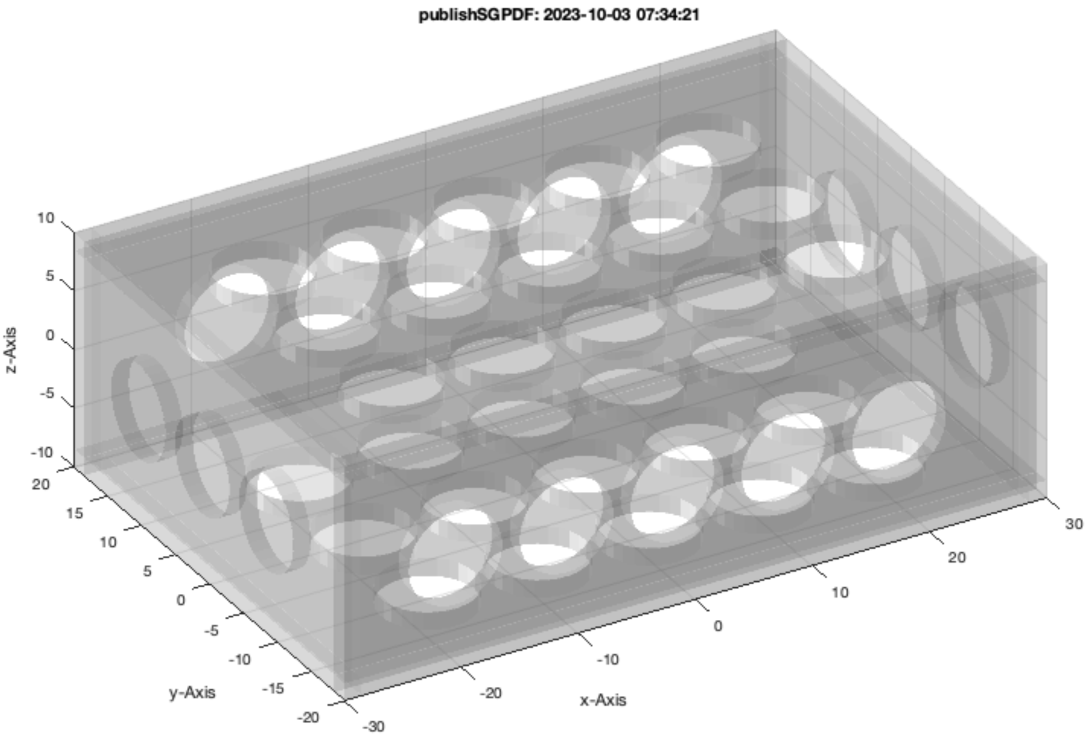
```
Using a resolution of 1.0 mm^3 (n=57288) and a specific weight of 1.15 milligramm per mm^3, the overall weight is ca. 8 gramm.
Elapsed time is 1.464263 seconds.
```



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

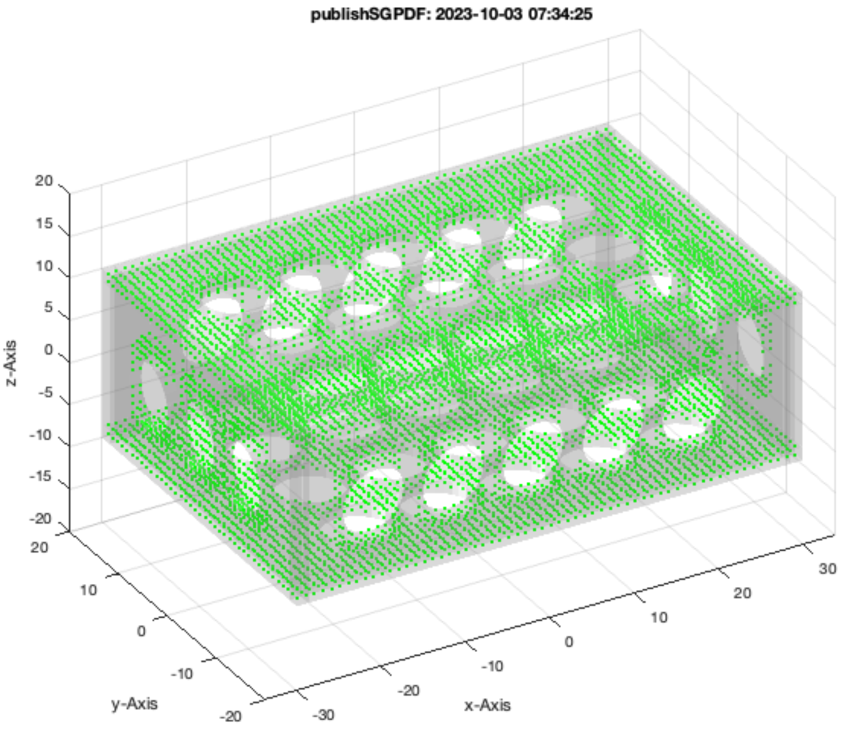
- SGplatesofSGML(SG,thickness,CPL) - convert a solid into a punched plate structure

```
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=57288) and a specific weight of 1.15 milligramm per mm^3, the overall weight is ca. 9 gramm.
Elapsed time is 3.323085 seconds.
```



Final remarks on toolbox version and execution date

```
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 07:34:25!
Executed 03-Oct-2023 07:34:27 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
phased_array_system_toolbox
signal_blocks
signal_toolbox
simmechanics
simscape
simulink
statistics_toolbox
=====================================================================================================
```

- *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® R2023a