Exercise 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)

Contents

- 1. Summary of the already explained aspects of the VLFL-Toolbox
- 2. Analyzing mouting faces of flat surfaces
- 3. Analyzing mouting faces of spherical/freeform surfaces
- 4. Create corresponding surfaces parallel to mounting faces
- 5. Create solids using the parallel surfaces and a plate thickness
- 6. Finding the 2D CPL of a planar 3D Surface
- 7. Replace a solid block by covering plates
- 8. Replace a solid block by covering plates with punched contours
- Final remarks on toolbox version and execution date

1. Summary of the already explained aspects of the VLFL-Toolbox

In examples 1-12 we've learned about:

- *Point lists* (PL), *Vertex lists* (VL), *Facet lists* (VL), *Edge lists* (EL)
- Functions to import, check, and export of *STL-File* data
- Functions for generating text strings as solid volumes
- *Closed polygon lists* (CPL) for boolesche operations of contours
- 2.5D extrusion of closed polyogon list (CPL) to *Solid Geometries* (SG)
- Spatial transformation and merging of solid geometry sets
- Relative positioning of solids geometries and boolean operations on solids
- Packaging and boxing of several parts into a complete assemble kit
- Attaching coordinates frames and the creation of kinematik models

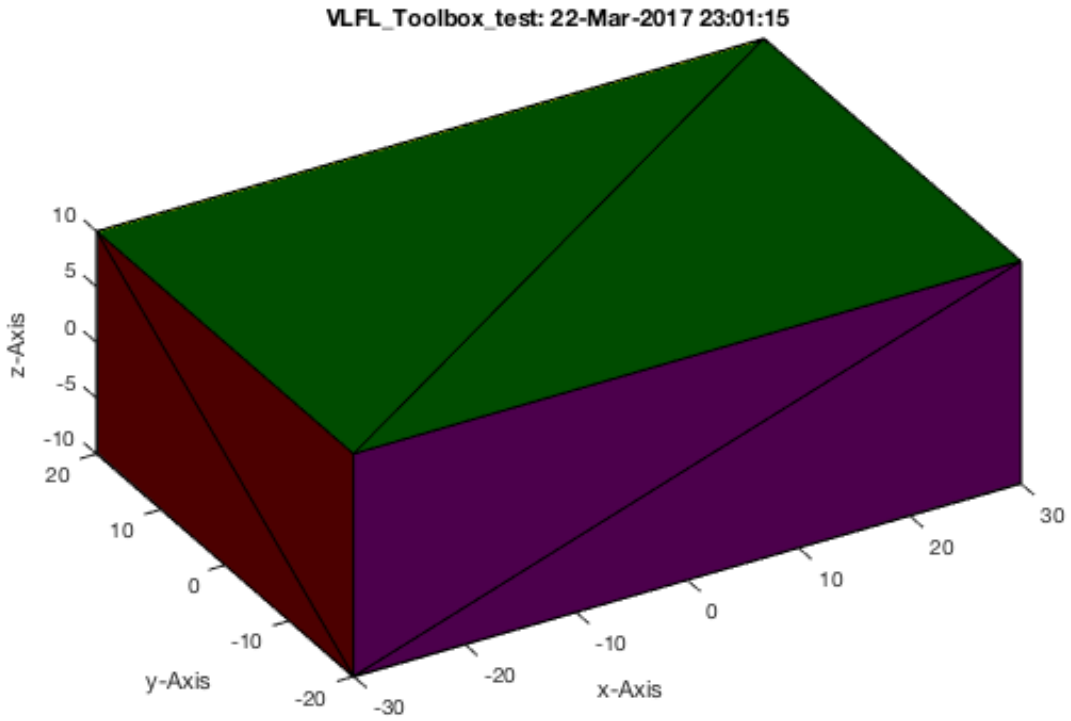
2. Analyzing mouting faces of flat surfaces

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 =

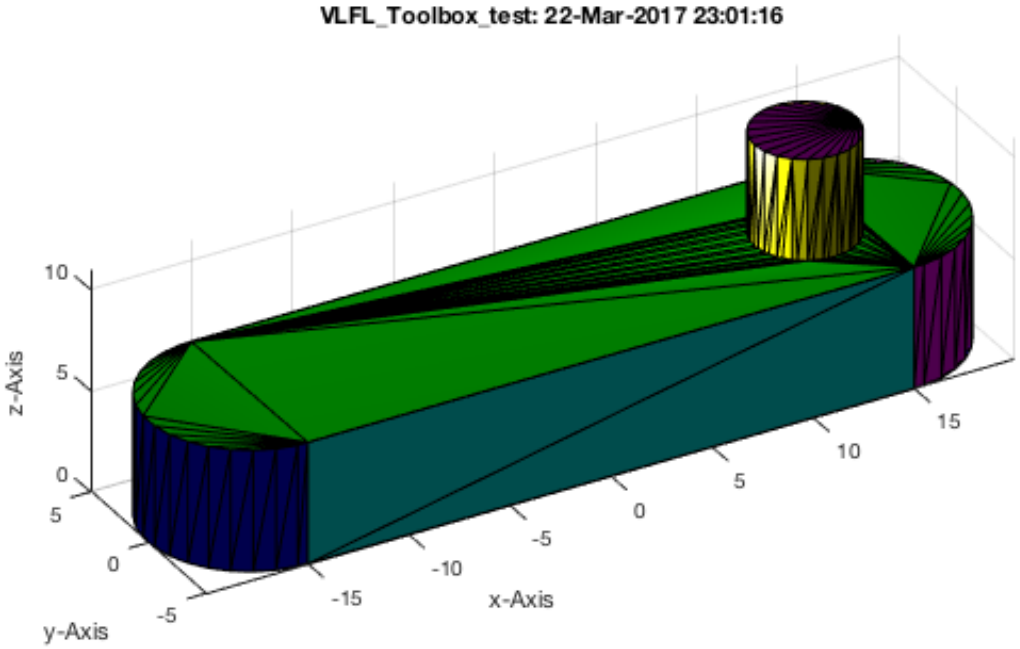
  struct with fields:

    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   -0.0000         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    0.0000         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 =

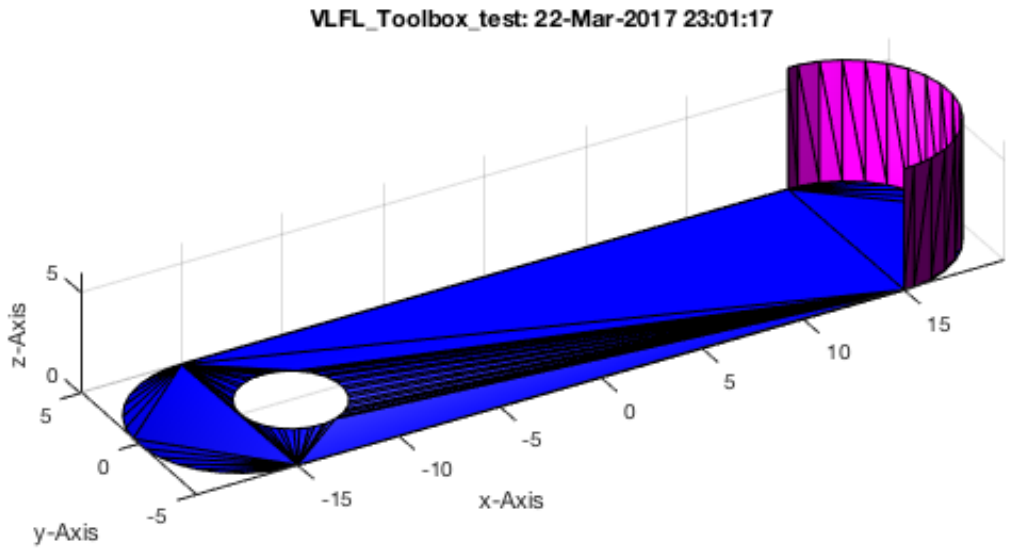
   1.0e-04 *

   -0.6543    0.2818    0.1521


z2 =

   1.0e-04 *

    0.1239    0.3370    0.0978
```



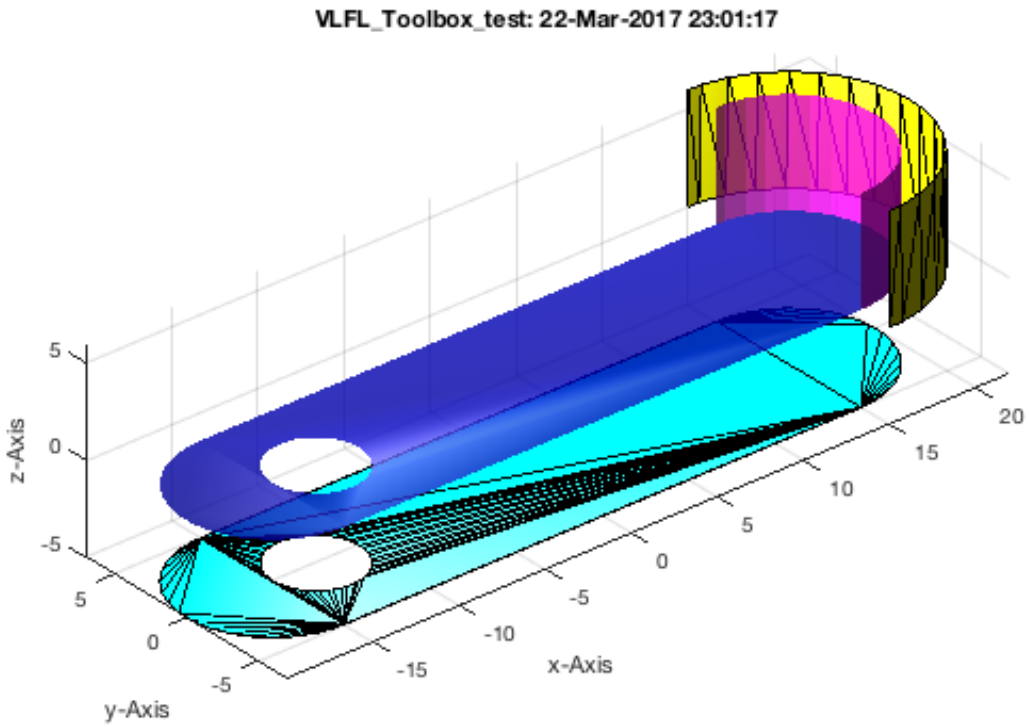
4. Create corresponding surfaces parallel to mounting faces

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');
```

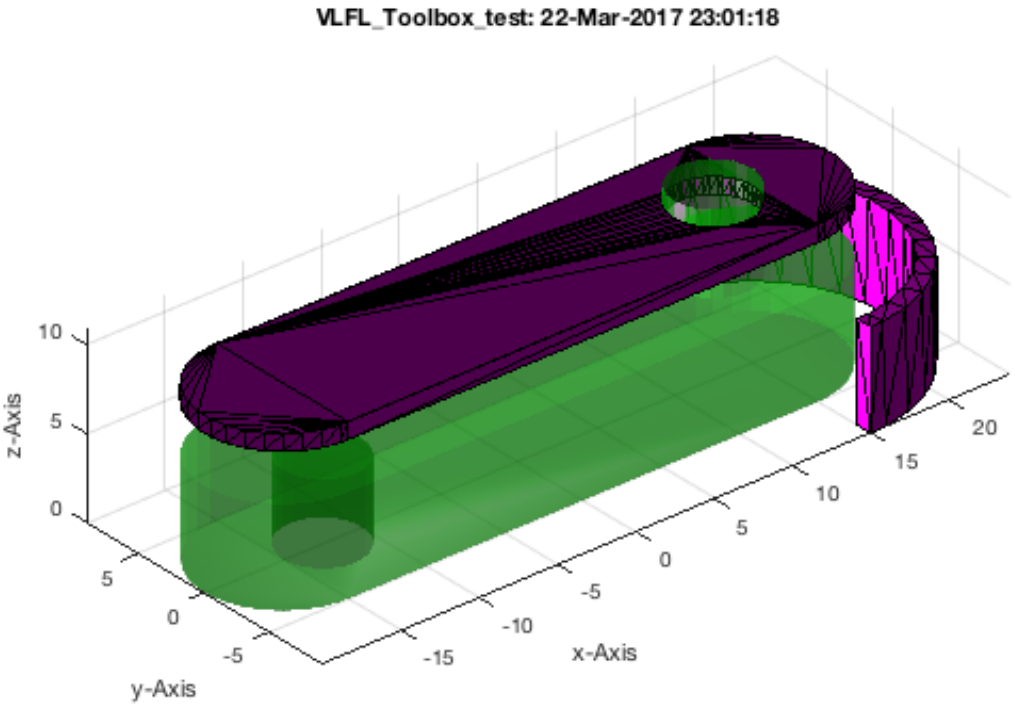


5. Create solids using the parallel surfaces and a plate thickness

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');
```

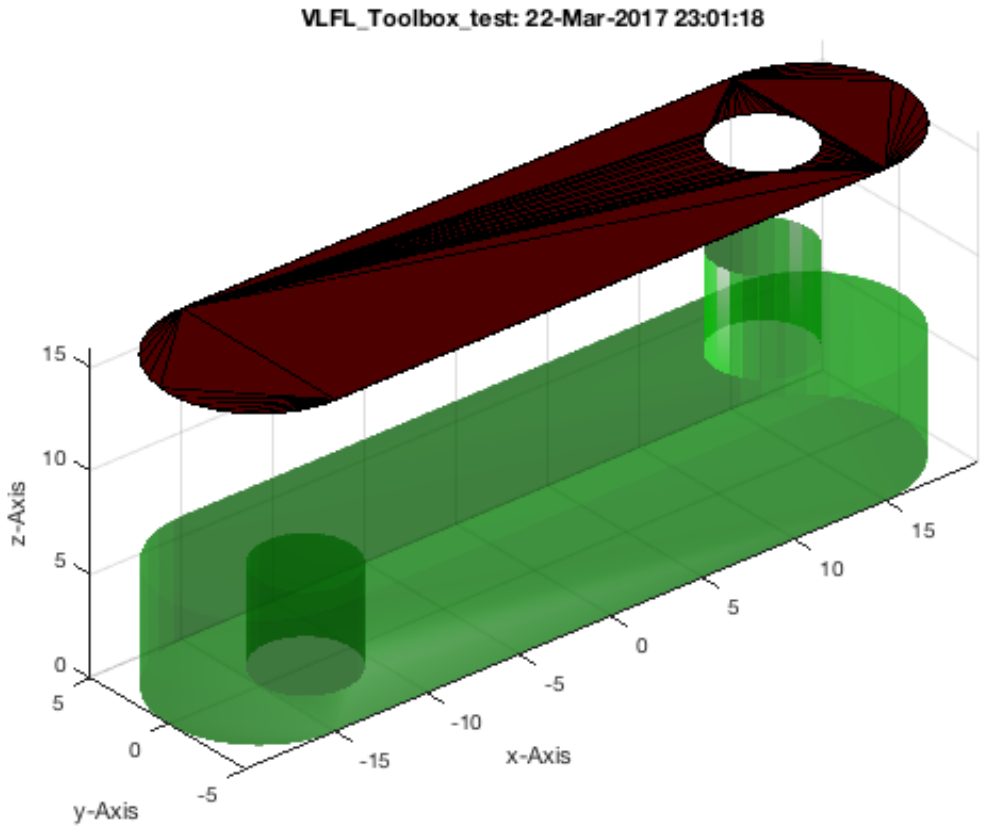


6. Finding the 2D CPL of a planar 3D Surface

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.

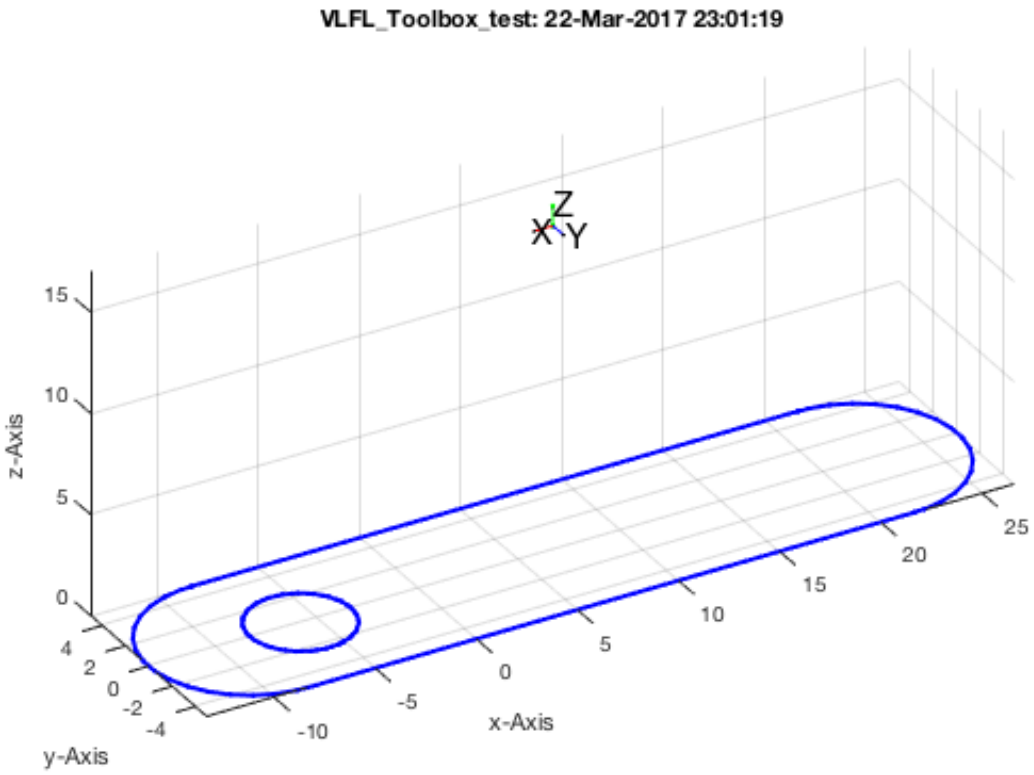
- [PL,T]=PLofVLFL(VL,FL) - returns a PL and a transformation matrix
- [CPL,T]=CPLofVLFL(VL,FL) - returns a CPL and a transformation matrix

```
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);
```



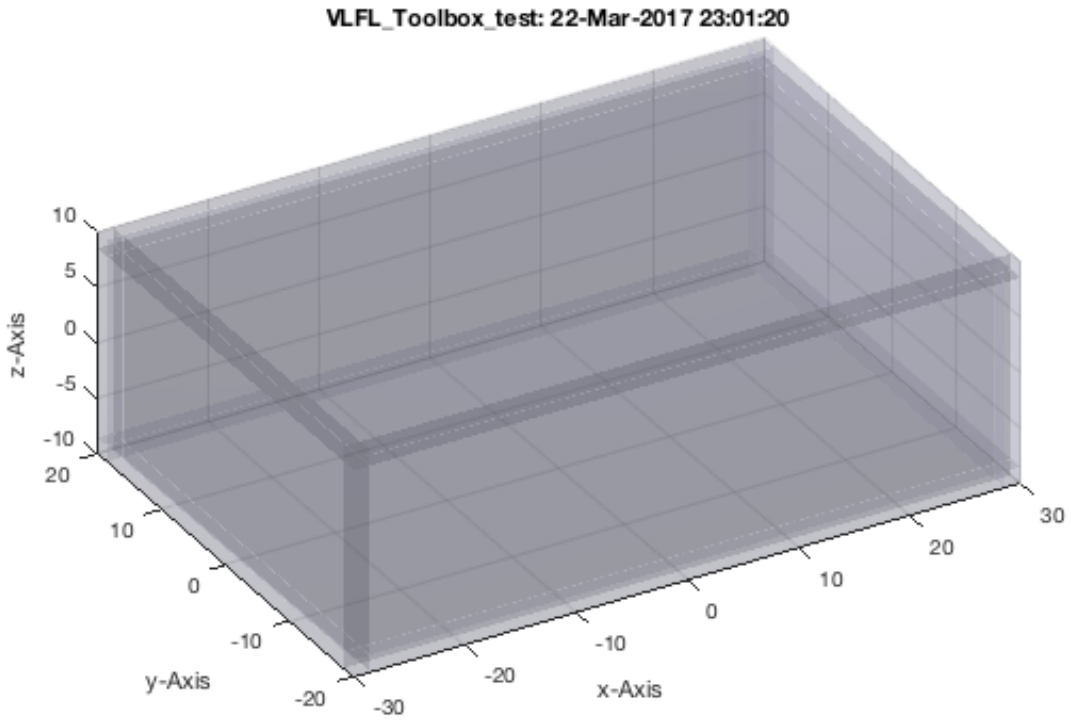
7. Replace a solid block by covering plates

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

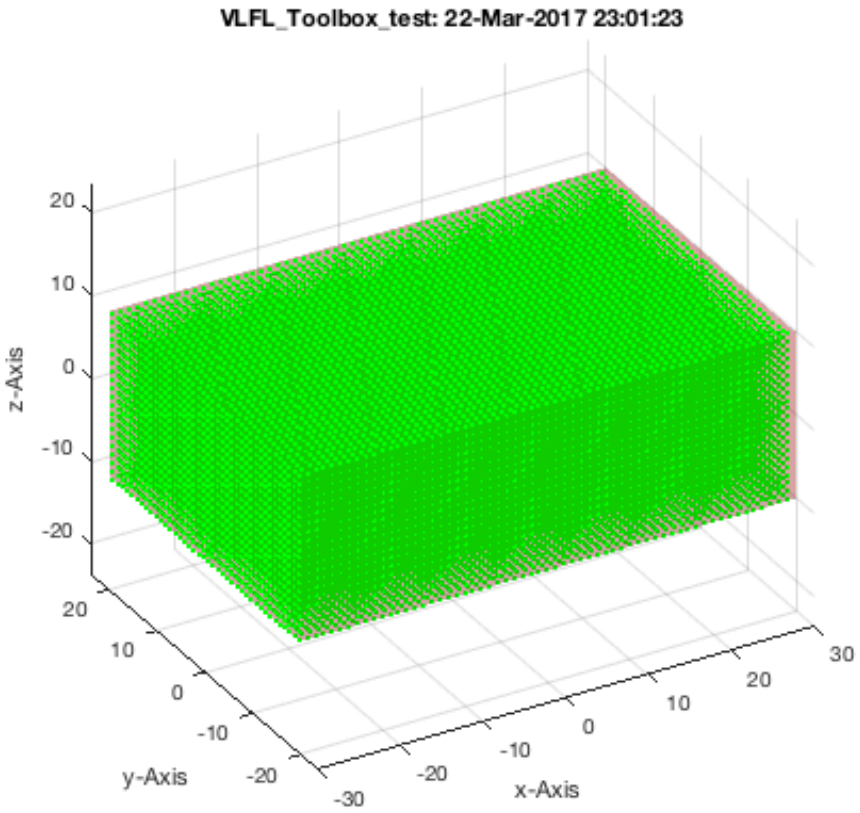
```
close all; SGfigure; view (-30,30);

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



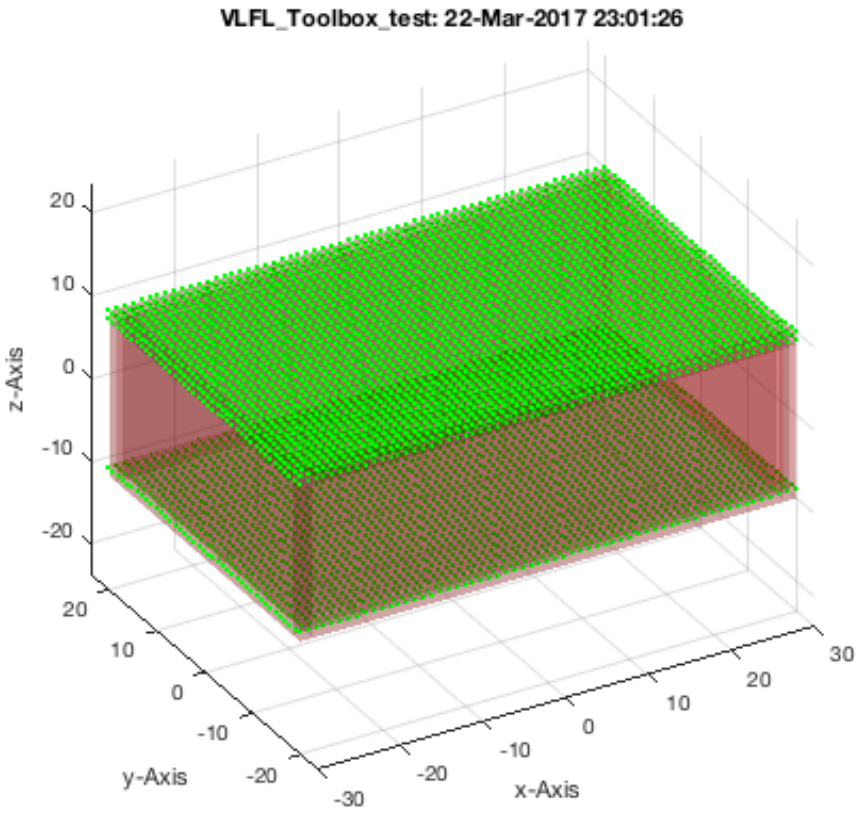

```
SGweight(SGbox([60,40,20]),[],1);
```

```
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 2.762754 seconds.
```




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

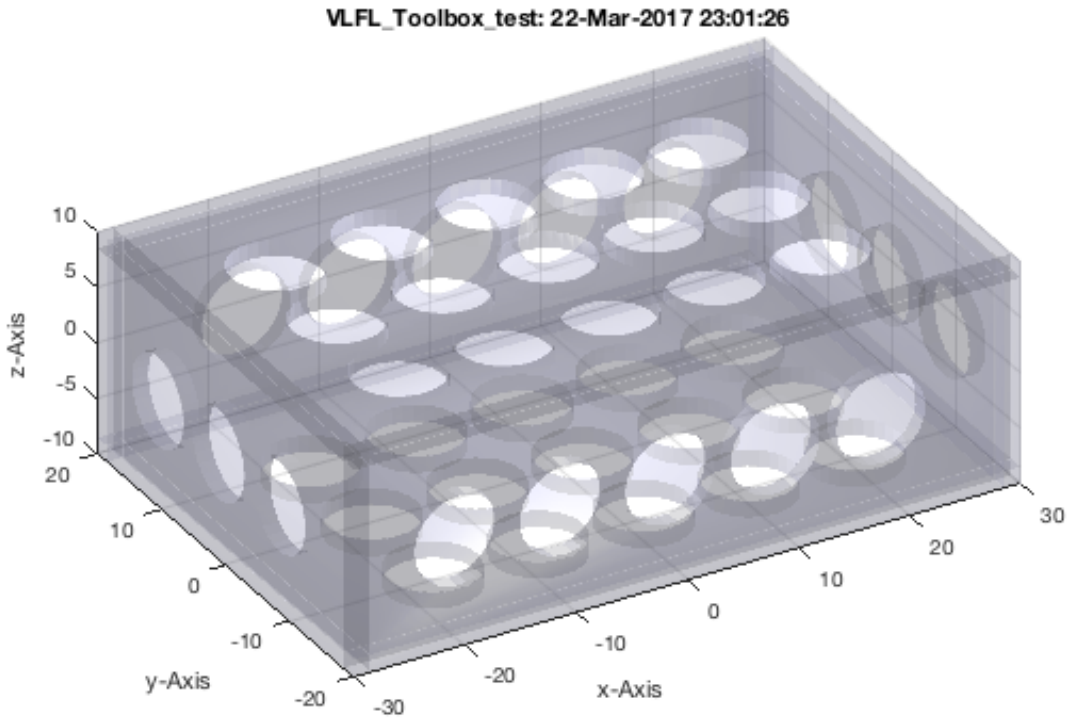
```
Using a resolution of 1.0 mm^3 (n=62307) and a specific weight of 1.15 milligramm per mm^3,
 the overall weight is ca. 9 gramm.
Elapsed time is 2.865529 seconds.
```



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

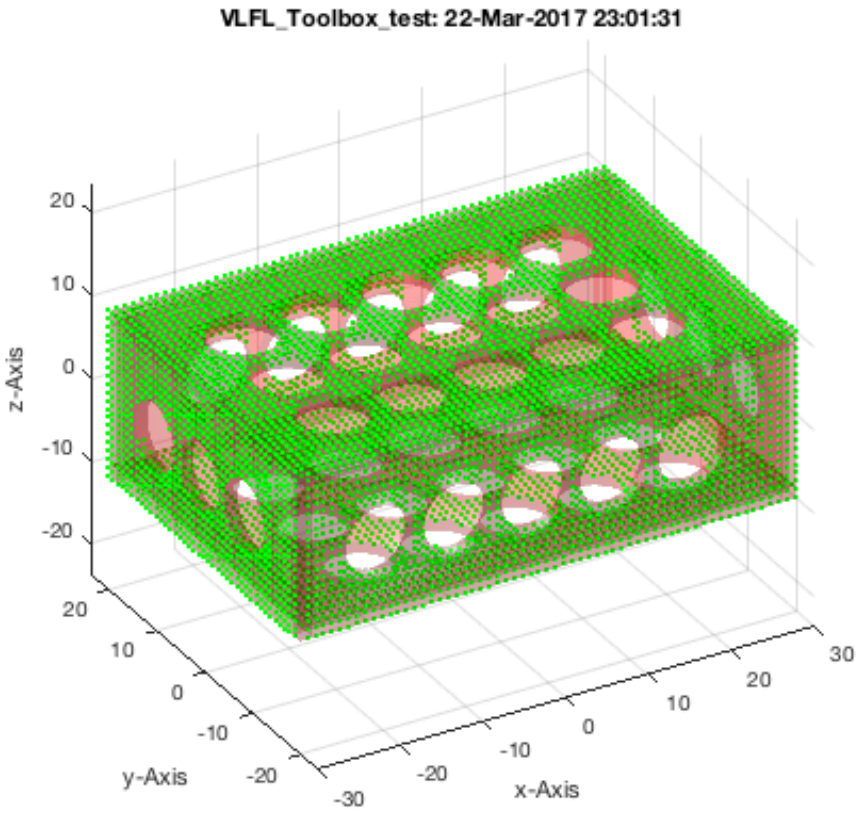
- SGplatesofSGML(SG,thickness,CPL) - convert a solid into a punched plate structure

```
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. 9 gramm.
Elapsed time is 3.097710 seconds.
```



Final remarks on toolbox version and execution date

```
VLFLlicense
```

```
This VLFL-Lib, Rel. 3.6 (2017-Mar-23), is for limited non commercial educational use only!
Licensee: 3.6 Tim Lueth for academic and education 05-Mar-2017 !
Please contact Tim Lueth, Professor at TU Munich, Germany!
WARNING: This VLFL-Lib (Rel. 3.6) license will exceed at 18-Mar-2018!
Executed 22-Mar-2017 23:01:33 by 'lueth' using Matlab 9.1.0.441655 (R2016b) on a MACI64
======================================== Used Matlab products: ============================
==========
map_toolbox
matlab
simmechanics
simscape
simulink
================================================================================================
==========
```

- *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® R2016b