Exercise 02: Using the VLFL-Toolbox for STL-File Export and Import

2014-11-18: 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. Import and export of STL-files in ASCII format and binary format
- 3. Checking surface volume data and STL-files
- 4. Generation of text, numbers, characters and formulas as solid volume
- 5. Turning and mirroring of solids by manipulating the vertex lists (VL)
- 6. Spatial transformation of solids by manipulating the vertex lists (VL)
- Final remarks on toolbox version and execution date

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

In Experiment 1 we've learned about

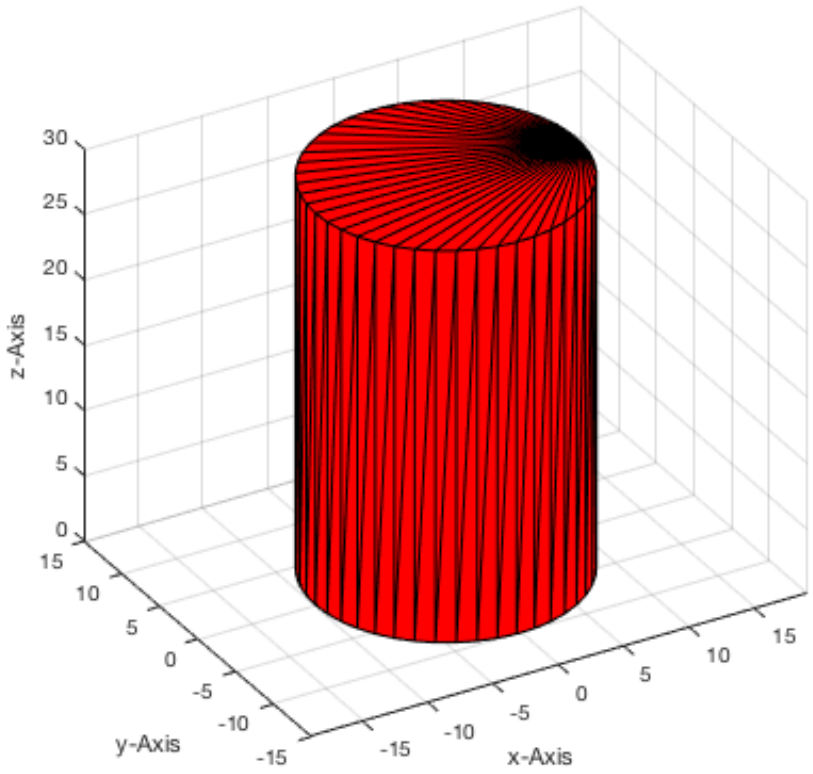
- Point lists (PL) as nx2 coordinate arrays that could be closed to contours.
- Vertex lists (VL) as nx3 coordiante arrays that could be closed to contours.
- Facet lists (FL) as nx3 index arrays that describe traingle facets of a surface.
- Functions for the interactive input of vertex lists.
- Functions for displaying PL and VL in 3D in different color and transparency.
- Functions for generating Bezier-curves or circular-curves.
- Functions for extruding solid volumes based on a closed convex contour.

2. Import and export of STL-files in ASCII format and binary format

Often it is useful to import surface data fo solid volumes from STL-Files generated by other programs such as CATIA, ProEngineer, Solidworks etc. On the other hand we want to export our data for documentation, 3D-printing or the exchange with other users. The STL-File format is the most common file format for surface models. It supports ascii-text-format and binary formatted files. For export and import we need a couple of functions:

- **VLFLwriteSTL** for writing STL-files in ascii file format.
- **VLFLwriteSTLb** for writing STL-files in binary file format.

```
close all;
PL=PLcircle(10);
[VL,FL]=VLFLofPLz (PL,30);
VLFLplot(VL,FL); view (-30,30); grid on;
```




```
VLFLwriteSTL(VL,FL,'STL-ASCII','by My Name');
VLFLwriteSTLb(VL,FL,'STL-BINAR','by My Name');
```

```
WRITING STL FILE /Users/lueth/Desktop/STL-ASCII.STL in ASCII MODE
completed.
WRITING STL (90 vertices, 176 facets) FILE /Users/lueth/Desktop/STL-BINAR.STL in BINARY MOD
E
completed.
```

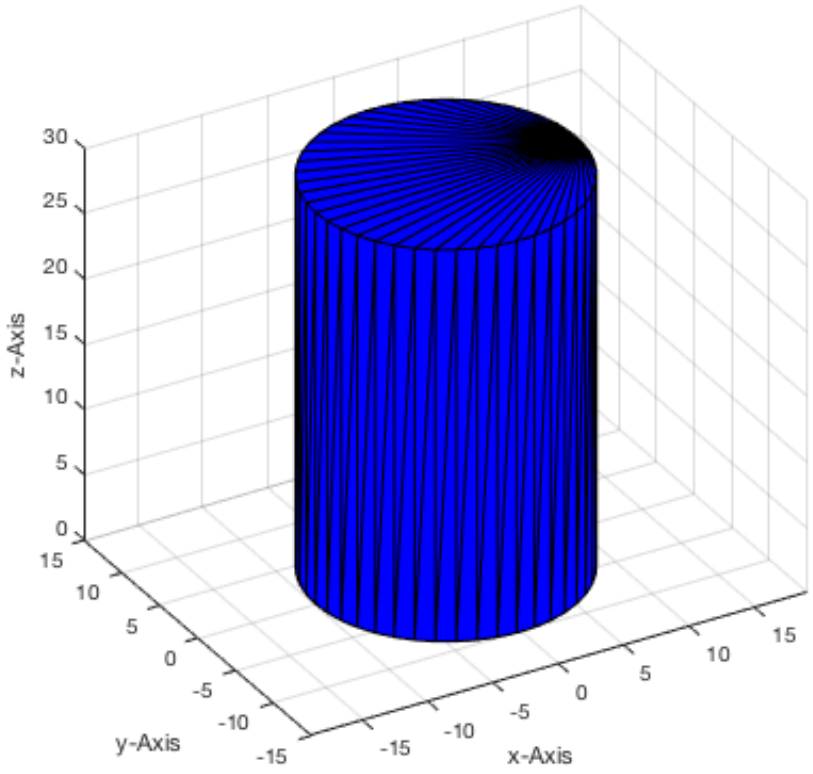
Similar it is possible to read the files in again

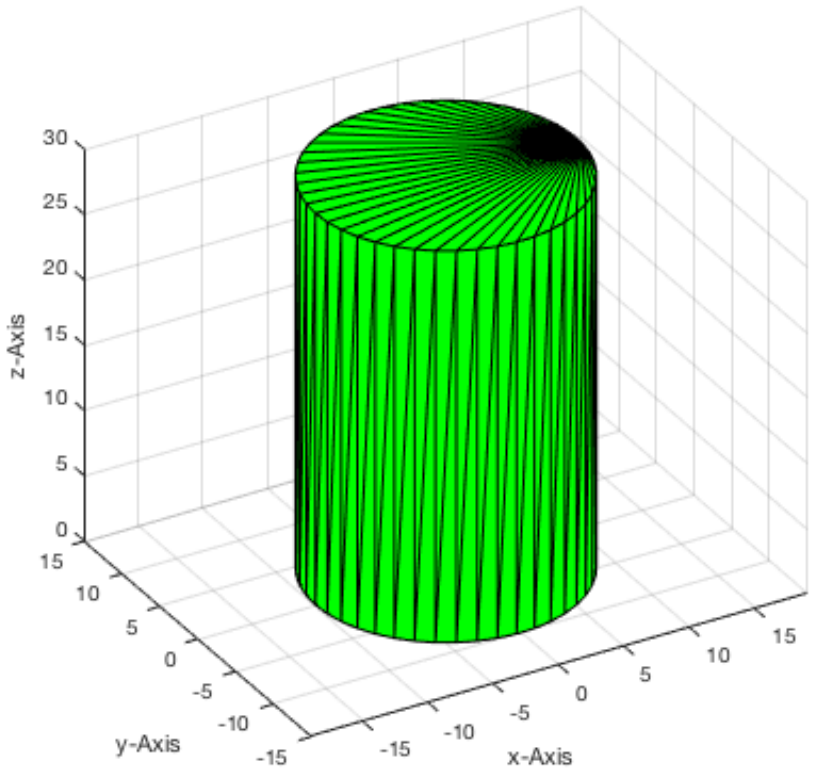
- **VLFLreadSTL** for importing STL-files in ascii file format.
- **VLFLreadSTLb** for importing STL-files in binary file format.

```
close all;
[VL,FL]=VLFLreadSTL ('STL-ASCII');
figure(1); VLFLplot(VL,FL,'b'); view (-30,30); grid on;
[VL,FL]=VLFLreadSTLb ('STL-BINAR');
figure(2); VLFLplot(VL,FL,'g'); view (-30,30); grid on;
```

```
LOADING ASCII STL-File: /Users/lueth/Desktop/STL-ASCII.STL
Processing 1234 lines:
Finishing solid AOI-LIB:"STL-ASCII by My Name" 22-Mar-2017 22:57:46 22-Mar-2017 22:57:46
```

```
VLFLchecker: 90 vertices and 176 facets.
    0 FACET PROBLEMS DETECTED (ERRORS)
    0 VERTEX PROBLEMS DETECTED (OBSOLETE WARNING)
    0 EDGE PROBLEMS DETECTED (NON MANIFOLD WARNING)
    0 SOLID/EDGE PROBLEMS DETECTED (OPEN SOLID WARNING)
LOADING BINARY STL-File: /Users/lueth/Desktop/STL-BINAR.STL
Header: AOI-LIB:"STL-BINAR by My Name" 22-Mar-2017 22:57:46
Number of facets: 176
0..
Finishing and Checking
VLFLchecker: 90 vertices and 176 facets.
    0 FACET PROBLEMS DETECTED (ERRORS)
    0 VERTEX PROBLEMS DETECTED (OBSOLETE WARNING)
    0 EDGE PROBLEMS DETECTED (NON MANIFOLD WARNING)
    0 SOLID/EDGE PROBLEMS DETECTED (OPEN SOLID WARNING)
```





3. Checking surface volume data and STL-files

Especially, when reading in STL-Files that are generated by other programs and libraries it makes sense to check the data quality. For that purpose there is a function that will be explained later in more detail. This function is called at the end of each STL import.

- **VLFLchecker** is used to analyze vertex list (VL) and facet list (FL)

```
VLFLchecker(VL,FL);     % Check the data structure

% There are some more procedures to view and analyze solid volumee data
```

```
VLFLchecker: 90 vertices and 176 facets.
    0 FACET PROBLEMS DETECTED (ERRORS)
    0 VERTEX PROBLEMS DETECTED (OBSOLETE WARNING)
    0 EDGE PROBLEMS DETECTED (NON MANIFOLD WARNING)
    0 SOLID/EDGE PROBLEMS DETECTED (OPEN SOLID WARNING)
```

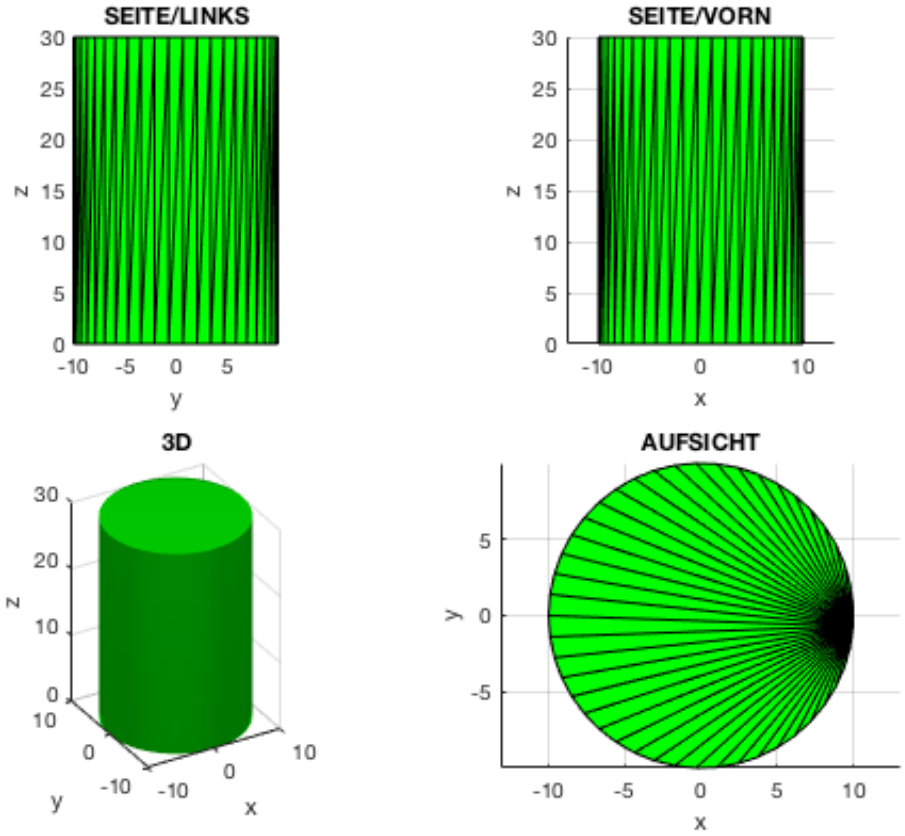
- **BBofVL** generates the bounding box dimensions of the solid
- **VLFLui** is simple user interface to open an STL file
- **VLFLminimize** eliminates doubles in VL and FL
- **VLFLnormf** calucates norm vector direction and size
- **VLFLplot4** figure with 4 subplots
- **VLFLselect** selected vertex list for a given facet list

- **VLFLseparate** find different independen objects in VL and FL
- **VLFLshort** remove unused vertices from VL
- **VLFLsurface** returns only vertex list and facet list for one surface
- **VLFLvertexfusion** shrinks vertex list by merging extremy near vertices

```
BBofVL(VL)
close all; VLFLplots4 (VL,FL,'g');
```

```
ans =

  -10.0000    9.9756   -9.9939    9.9939         0   30.0000
```

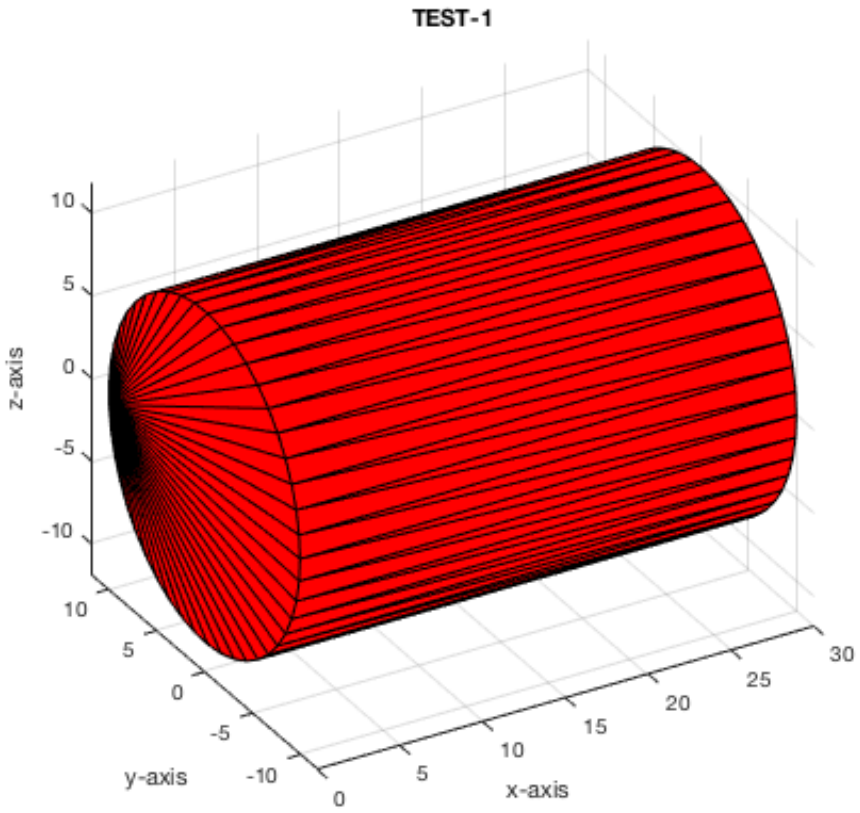



```
close all; VLFLseparate(VL,FL);
```

```
Analyzing 90 facets for separation z=[0.0mm|30.0mm]
Object TEST-1 with 176 facets

MVL =

     0   -10     0
```



4. Generation of text, numbers, characters and formulas as solid volume

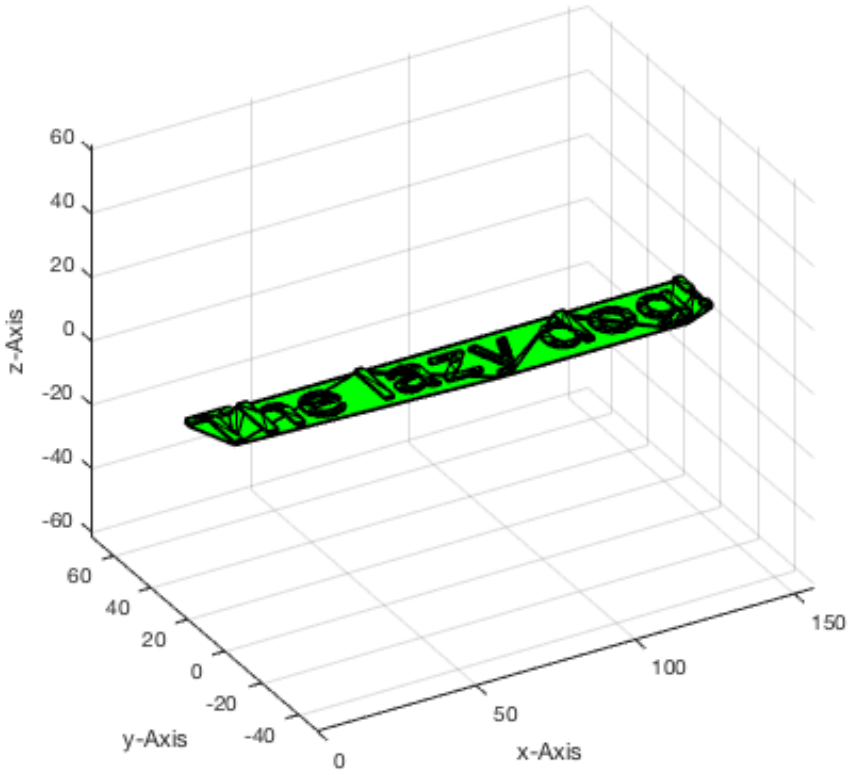
Often you want to write some numbers or code on top of a solid object. For that purpose there is a currently slow function that is able to convert a Matlab-string (even with LaTex-code) into a solid object.

- **VLFLtextimage** writes a line using the text command and converts it into a solid volume
- **VLFLtext** does the same for a very limited number of characters

```
close all;
[VL,FL]=VLFLtextimage('The lazy dog!');
VLFLplot (VL,FL,'g'); view (-30,30);

[VL,FL,d]=VLFLtext('TL-MMXI-XII-XVII');
VLFLwriteSTL (VL,FL,'exp_2011_12_17', 'by Tim C Lueth');
```

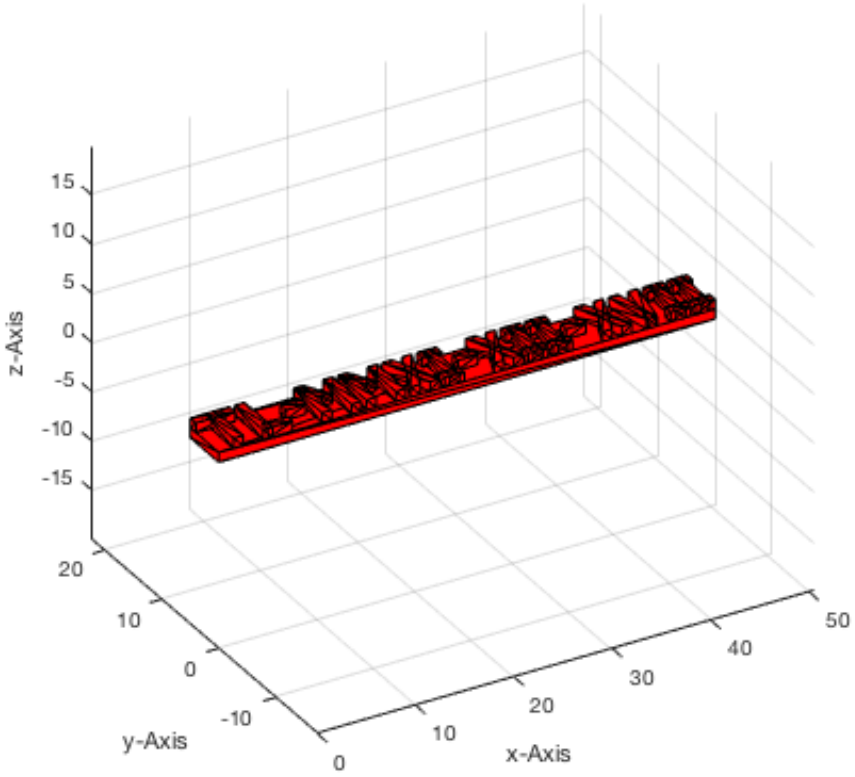
```
WRITING STL FILE /Users/lueth/Desktop/exp_2011_12_17.STL in ASCII MODE
completed.
```



5. Turning and mirroring of solids by manipulating the vertex lists (VL)

Turning an object and mirroring is quite simple by exchanging a column of the vertex list to change the sign of a column. To show the use of the functions we generate first a simple roman date string as solid volume.

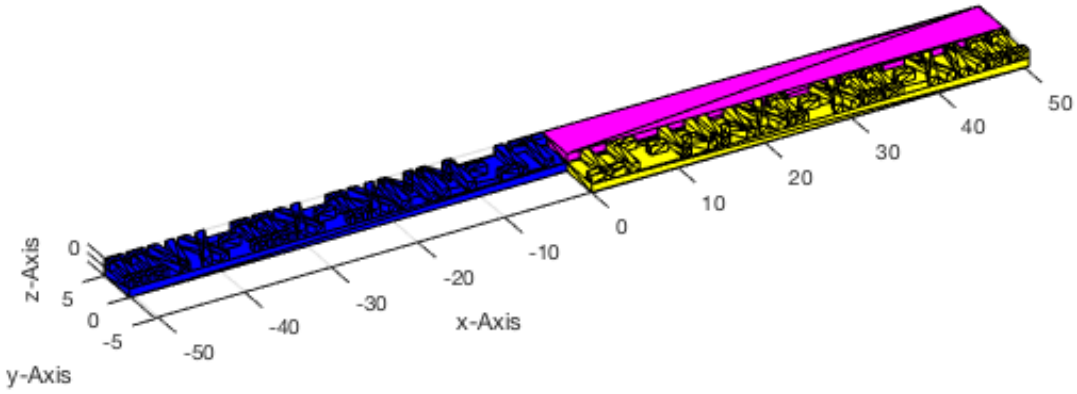
```
close all;
[VL,FL,d]=VLFLtext('TL-MMXI-XII-XVII'); VLFLplot(VL,FL,'r'); view(-30,30);
```



The functions for mirroring solid objects by manipulating the vertex list are the following:

- **VLswapX** mirrors the solid at the x-axis (y/z-plane).
- **VLswapY** mirrors the solid at the y-axis (x/z-plane).
- **VLswapZ** mirrors the solid at the z-axis (x/y-plane).

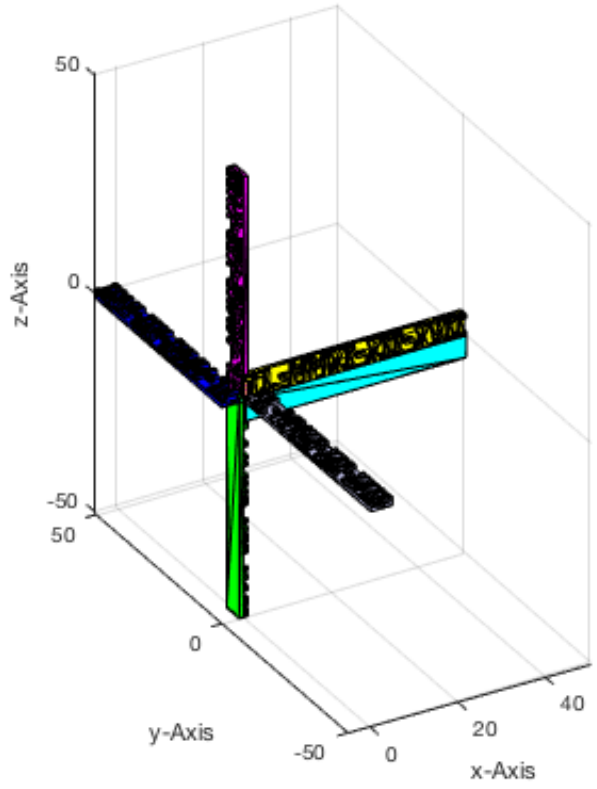
```
close all, view (-30,30); grid on;
VLFLplot(VLswapX(VL),FL,'b');       % mirror at x-axis
VLFLplot(VLswapY(VL),FL,'y');       % mirror at y-axis
VLFLplot(VLswapZ(VL),FL,'m');       % mirror at z-axis
```



The functions for turning solid objects by manipulating the vertex list are the following:

- **VLswapXY** turn the x-axis to the y-axis.
- **VLswapXZ** turn the x-axis to the z-axis.
- **VLswapYX** turn the y-axis to the x-axis.
- **VLswapYZ** turn the y-axis to the z-axis.
- **VLswapZX** turn the z-axis to the x-axis.
- **VLswapZY** turn the z-axis to the y-axis.

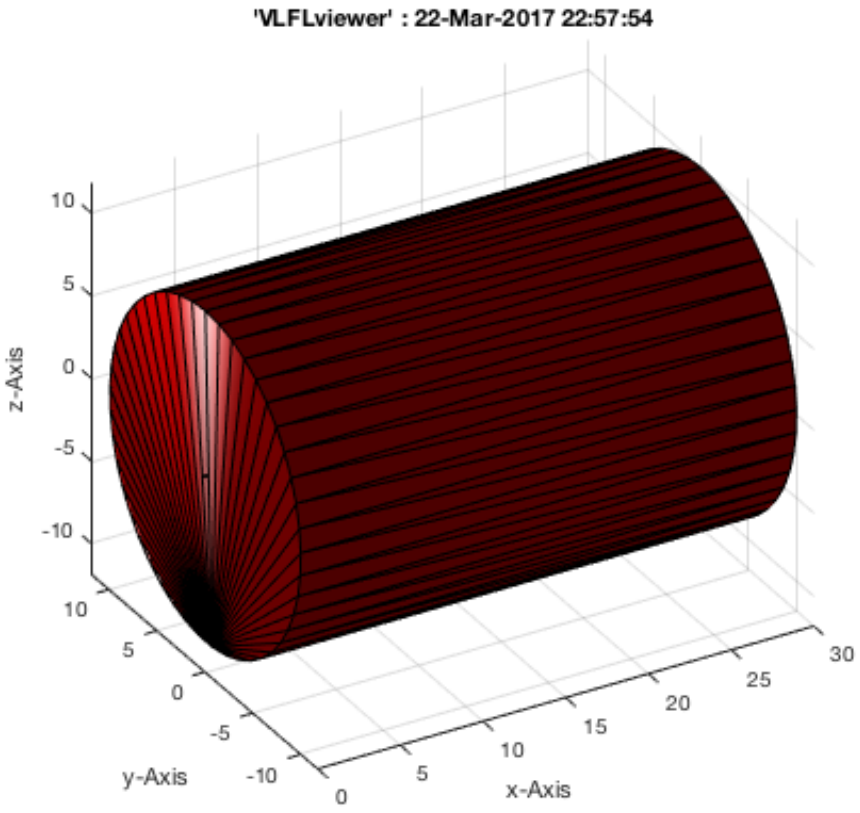
```
close all, view (-30,30); grid on
VLFLplot(VL,FL,'r');                % original solid
VLFLplot(VLswapXY(VL),FL,'b');      % turn the x-axis to the y-axis
VLFLplot(VLswapXZ(VL),FL,'m');      % turn the x-axis to the z-axis
VLFLplot(VLswapYZ(VL),FL,'y');      % turn the y-axis to the z-axis
VLFLplot(VLswapZY(VL),FL,'c');      % turn the z-axis to the y-axis
VLFLplot(VLswapZX(VL),FL,'g');      % turn the z-axis to the x-axis
VLFLplot(VLswapYX(VL),FL,'w');      % turn the y-axis to the x-axis
```



6. Spatial transformation of solids by manipulating the vertex lists (VL)

All solid objects consisting of vertices and facets can be moved and rotated by only manipulating the vertex list (VL). Since the facet list is an index list, the facet list (FL) is not affected by a transformation of the vertex list. The following example generates a cylinder and perform different postion and orientation transformations.

```
closeall;
VLFLviewer([]);
PL=PLcircle(10);                              % define a base-contour
[VL,FL]=VLFLofPLz (PL,30);                    % extrude to a solid volume
VL=VLswapZX (VL);                             % swap X and X axis
VLFLplot(VL,FL); view (-30,30); grid on;      % plot as red cylinder
```



In detail, there are five basic transformation functions for manipulation a vertex list (VL)

- **VLtrans0** for translating the solid in the coordiante system origin.
- **VLtrans1** for translating the solid into quadrant 1.
- **VLtransP** for translating the solid using a translation vector.
- **VLtransR** for rotation the solid using a rotation matrix.
- **VLtransT** for transformating using an homogenous transformation matrix.

In addition to the already existing matlab functions rotx, roty, and rotz, two new functions are useful.

- **rot** for generating a 3x3 rotation matrix for x y z given in rad.
- **rotdeg** for generating a 3x3 rotation matrix for x y z given in degree.

```
VL=VLtrans0 (VL);                   % Transformation into the origin (blue)
VLFLplot(VL,FL,'b'); view (-30,30);

VL=VLtrans1 (VL);                    % Transformation into quadrant 1 (black)
VLFLplot(VL,FL,'k'); view (-30,30);

VL=VLtransP (VL,[0 ;0; 30]);         % Transformation upwards 30 mm (yellow)
VLFLplot(VL,FL,'y'); view (-30,30);

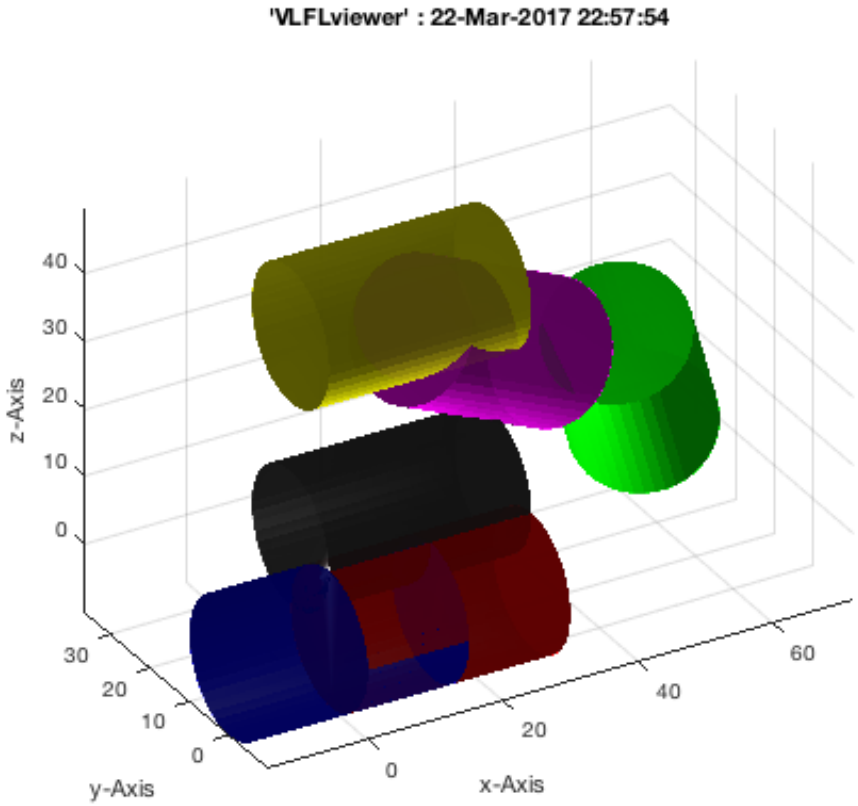
VL=VLtransR (VL,rotdeg(0,30,15));    % Rotate 30 degree around y and 15 around z (magenta)
VLFLplot(VL,FL,'m'); view (-30,30);

T=[rotdeg(0,30,15), [20;0;0];[0 0 0 1]] % define a homogenous transformation matrix (green)
```

```
VL=VLtransT (VL,T);                % Transformation using an HT matrix
VLFLplot(VL,FL,'g'); view (-30,30); grid on;
VLFLplotlight (1,0.9); grid on;
```

```
T =

    0.8365   -0.2241    0.5000   20.0000
    0.2588    0.9659         0         0
   -0.4830    0.1294    0.8660         0
         0         0         0    1.0000
```



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 22:57:57 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 2014-11-19
- Tim Lueth, executed and published on 64 Bit PC using Windows with Matlab 2014b on 2014-11-19

Published with MATLAB® R2016b