Exercise 06: Relative Positioning and Alignment of Solid Geometries (SG)

2014-11-24: 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. Relative spatial positioning of solid geometries (SG) using bounding boxes
- 3. Relative spatial alignment of solid geometries (SG) using bounding boxes
- Final remarks on toolbox version and execution date

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

In examples 1-5 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

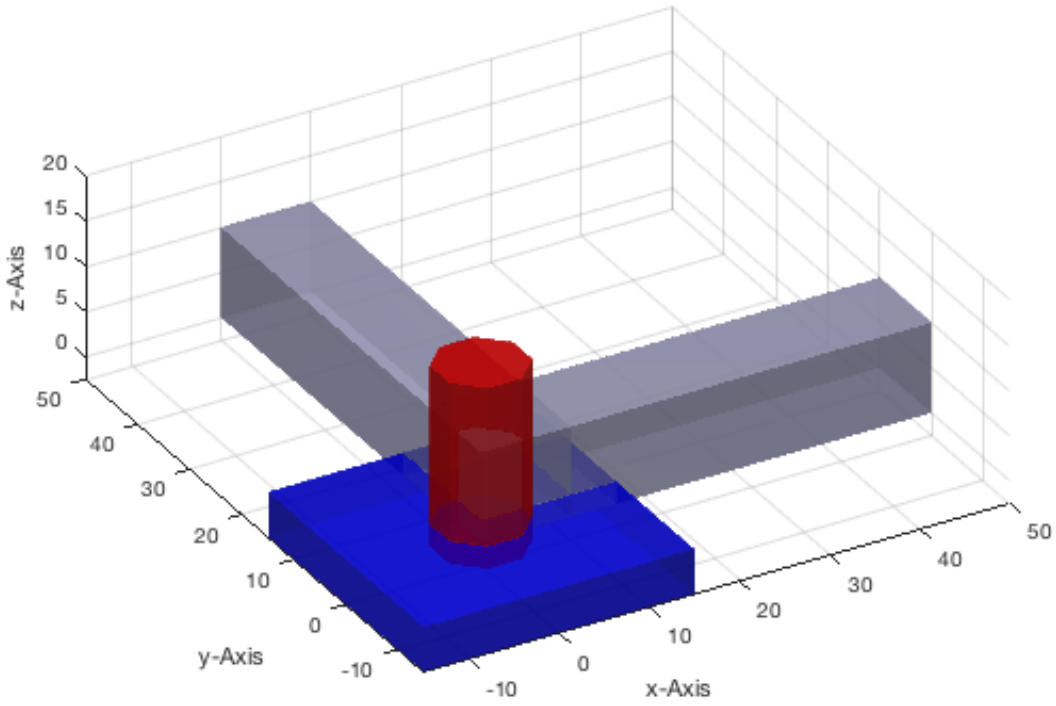
2. Relative spatial positioning of solid geometries (SG) using bounding boxes

Since it is quite convenient to use solid geometries (SG), there is a need for relative spatial positioning of these objects. For 'on top', 'under' (modifies the z-coordinates), 'in front', 'behind' (modifies the y-coordinates), 'left' and 'right' (modifies the x-coordinates), we have six different positioning functions that generate copies of the SG with just a changed vertex list. A third parameter of those functions is a gap, that can be defined. A positive gap value means a separation of thoses solids, a negative gap value means a penetration of those solids.

- **SGontop** positions a solid geometry 'A' on top of solid geometry 'B'
- **SGunder** positions a solid geometry 'A' under of solid geometry 'B'
- **SGinfront** positions a solid geometry 'A' in front of solid geometry 'B'
- **SGbehind** positions a solid geometry 'A' behind of solid geometry 'B'
- **SGleft** positions a solid geometry 'A' left of solid geometry 'B'
- **SGright** positions a solid geometry 'A' right of solid geometry 'B'

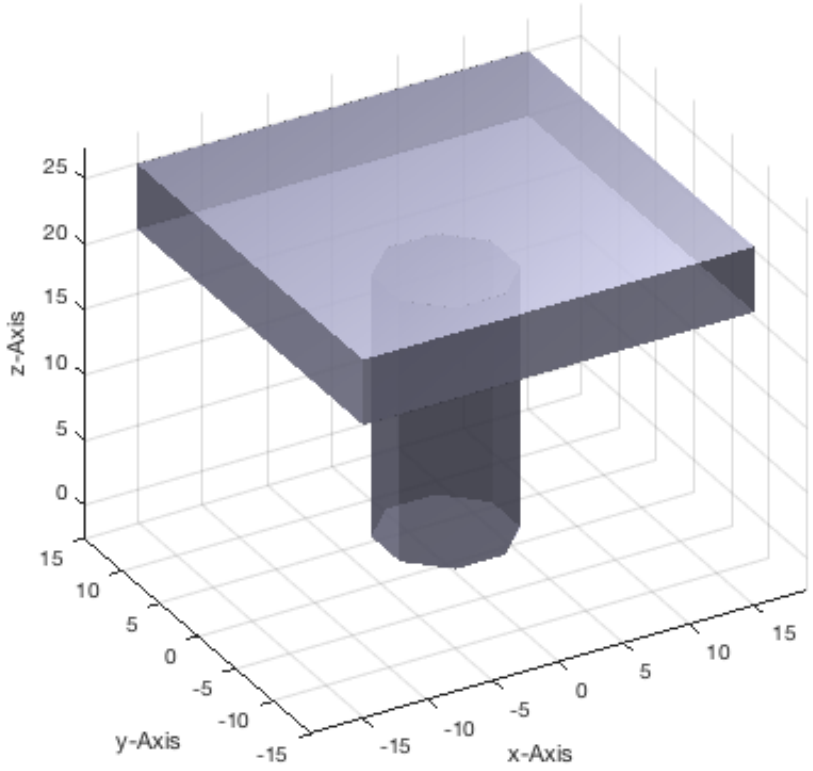
Define two solid geometrys 'A' and 'B'

```
close;
A=SGbox([30,30,5]); B=SGofCPLz(PLcircle(5,8),20);
SGplot (A,'b'); SGplot (B,'r'); VLFLplotlight(1,0.7); view (-30,30);
```



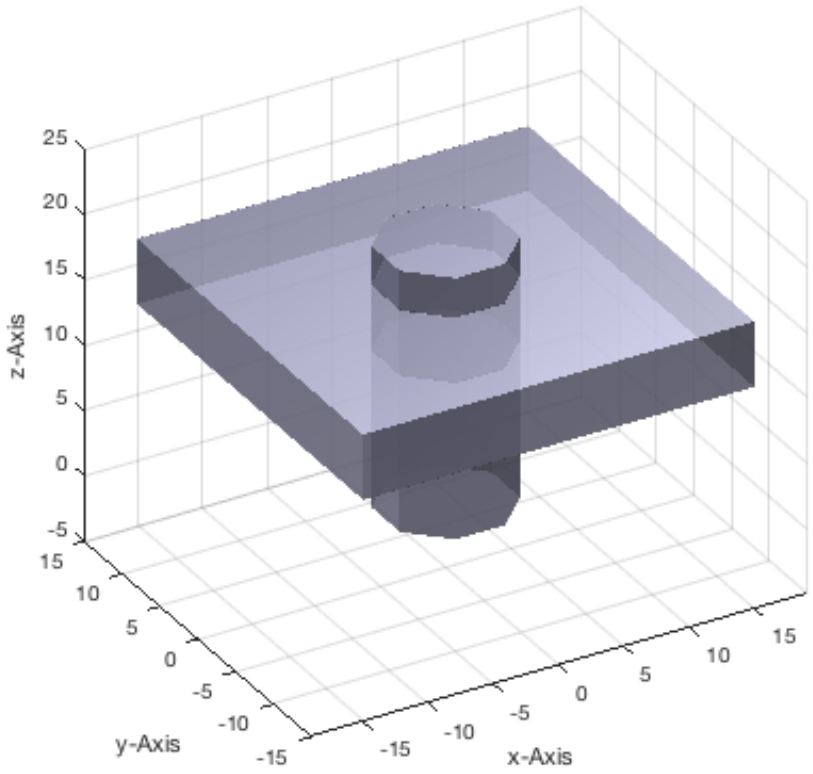
SGontop positions a solid geometry 'A' on top of solid geometry 'B'

```
close;
SG=SGcat(SGontop(A,B),B);
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



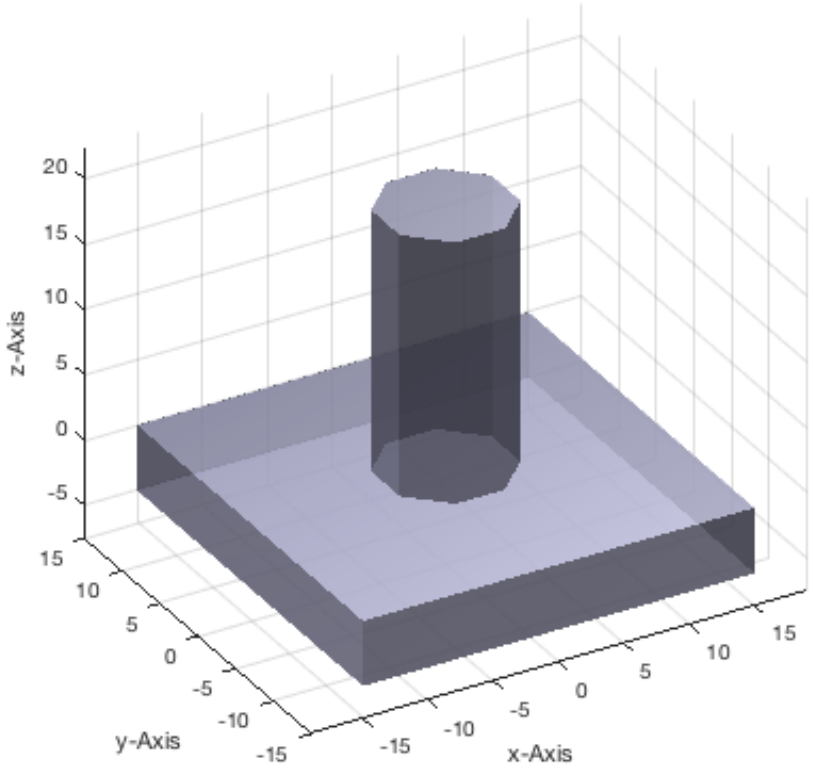
SGontop positions a solid geometry 'A' on top of solid geometry 'B' with a gap of -8

```
close all;
SG=SGcat(SGontop(A,B,-8),B);
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



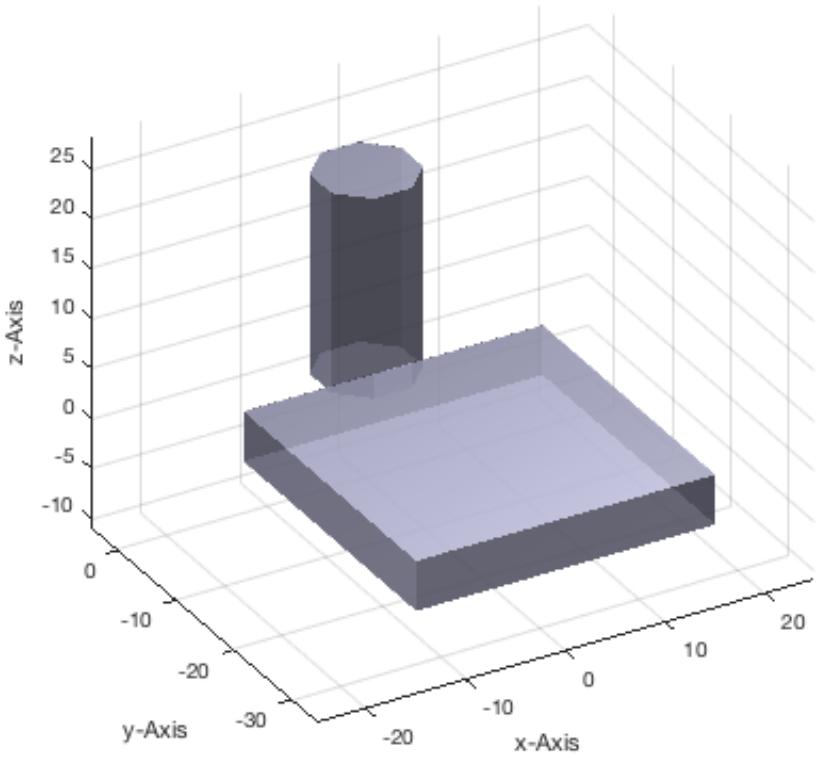
SGunder positions a solid geometry 'A' under of solid geometry 'B'

```
close all;
SG=SGcat(SGunder(A,B),B);
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



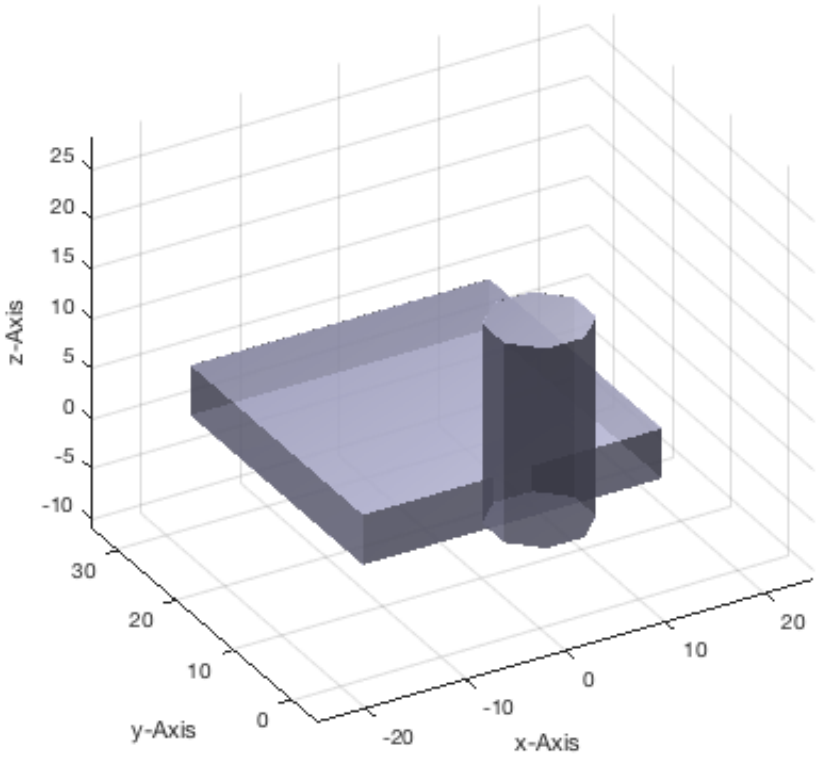
SGinfront positions a solid geometry 'A' in front of solid geometry 'B'

```
close all;
SG=SGcat(SGinfront (A,B),B);
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



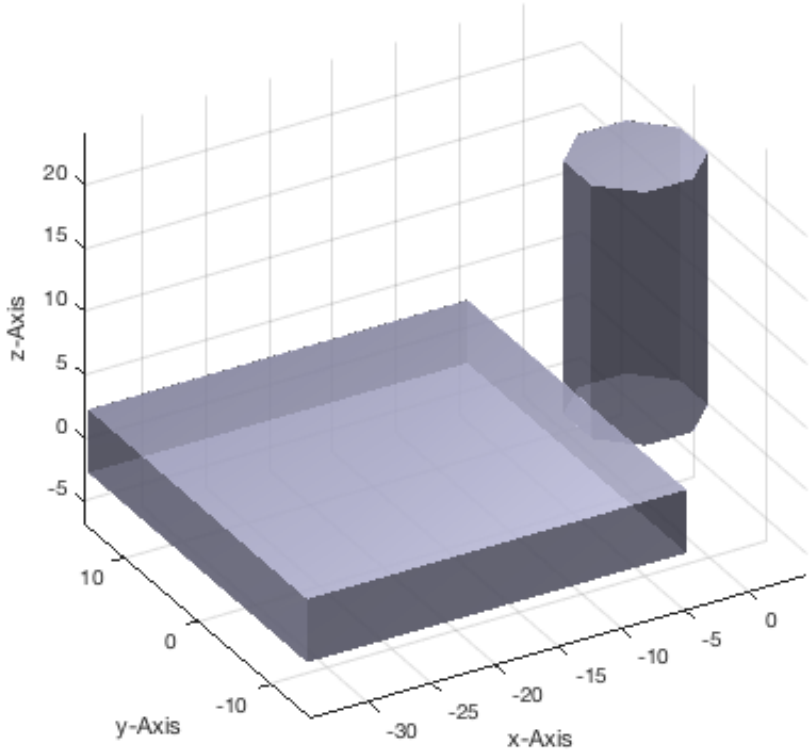
SGbehind positions a solid geometry 'A' behind of solid geometry 'B'

```
close all;
SG=SGcat(SGbehind (A,B),B);
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



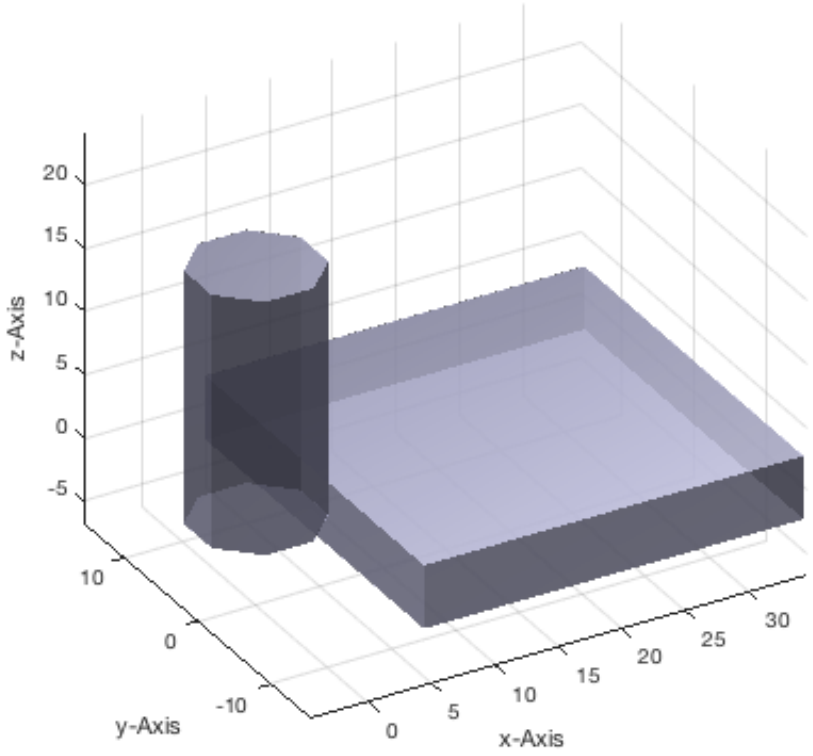
SGleft positions a solid geometry 'A' left of solid geometry 'B'

```
close all;
SG=SGcat(SGleft (A,B),B);
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



SGright positions a solid geometry 'A' right of solid geometry 'B'

```
close all;
SG=SGcat(SGright (A,B),B);
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



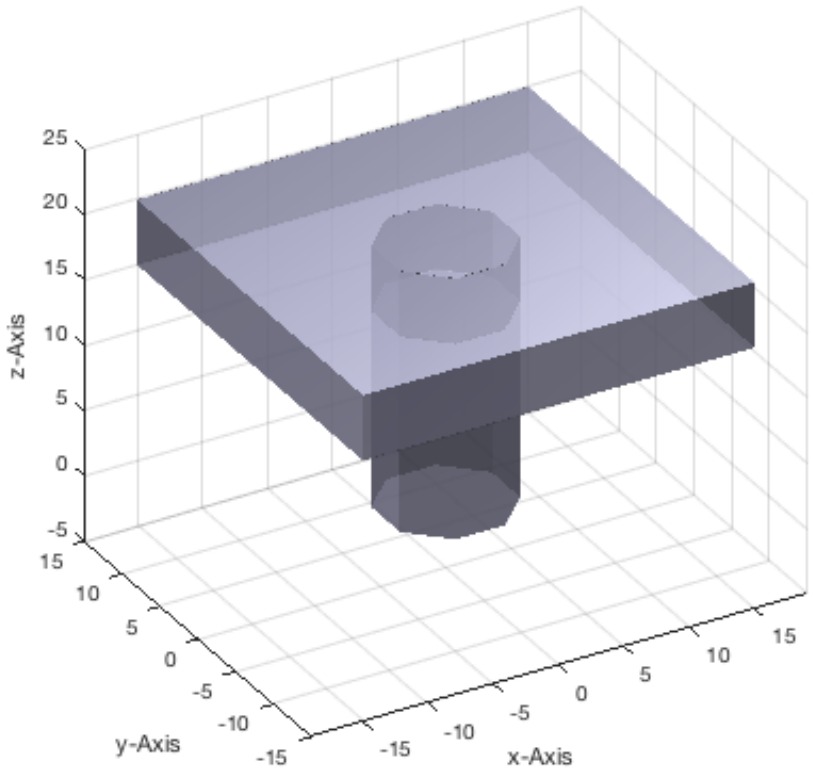
3. Relative spatial alignment of solid geometries (SG) using bounding boxes

Similar to the relative positioning, also the spatial alignment is helpful. For example solid A is aligned with solid B to achive the same 'top', 'bottom' (modifies the z-coordinates), 'front', 'back' (modifies the y-coordinates), 'left side' or 'right side' (modifies the x-coordinates).

- **SGaligntop** aligns the top of solid A with the top of solid B
- **SGalignbottom** aligns the bottom of solid A with the bottom of solid B
- **SGalignfront** aligns the front of solid A with the front of solid B
- **SGalignback** aligns the back of solid A with the back of solid B
- **SGalignleft** aligns the left side of solid A with the left side of solid B
- **SGalignright** aligns the right side of solid A with the right side of solid B

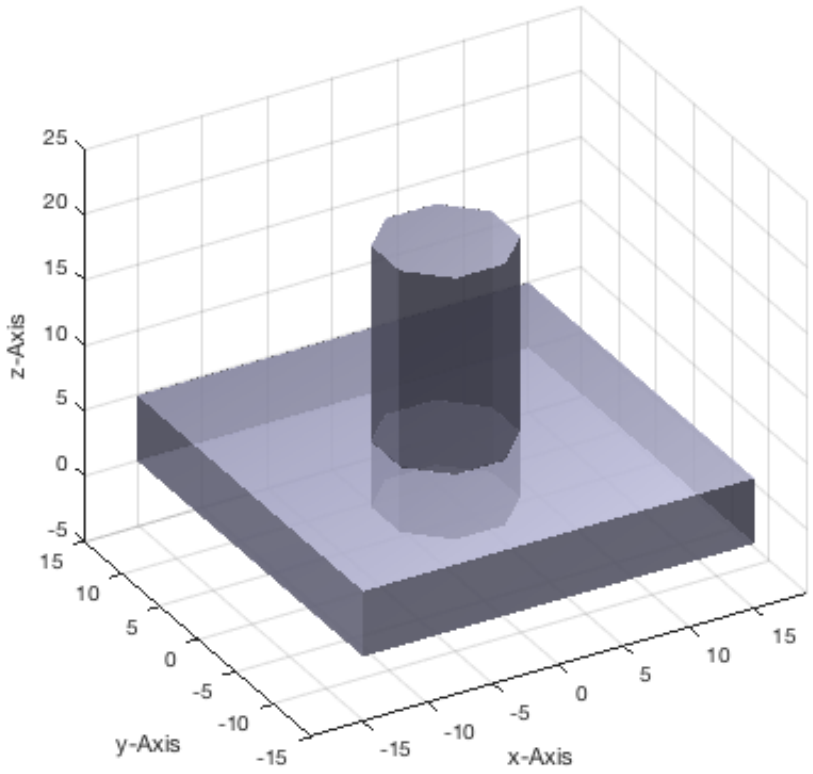
SGaligntop aligns the top of solid A with the top of solid B

```
close all;
SG=SGcat({SGaligntop(A,B),B});
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



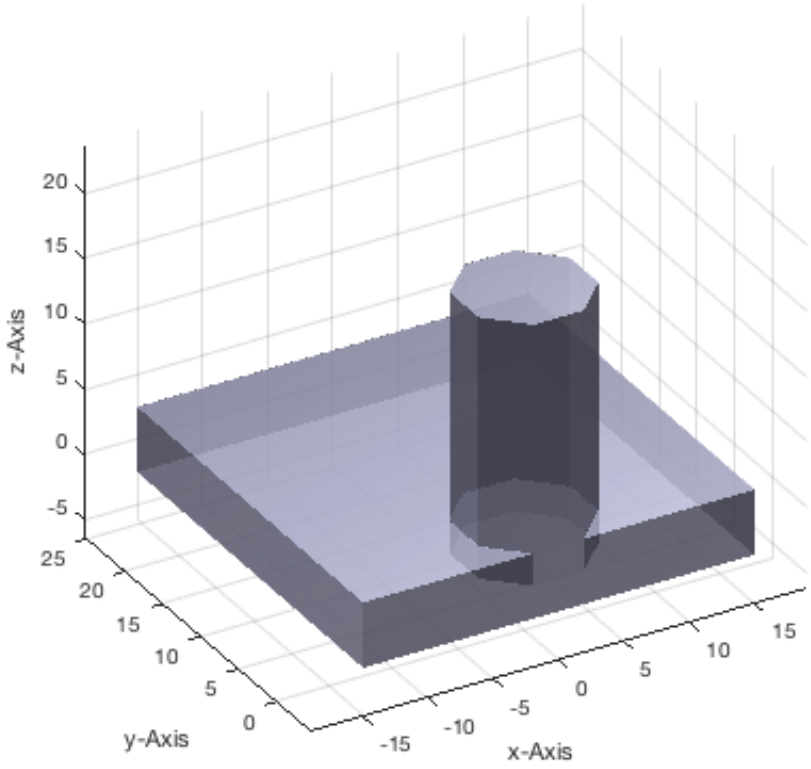
SGalignbottom aligns the bottom of solid A with the bottom of solid B

```
close all;
SG=SGcat({SGalignbottom(A,B),B});
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



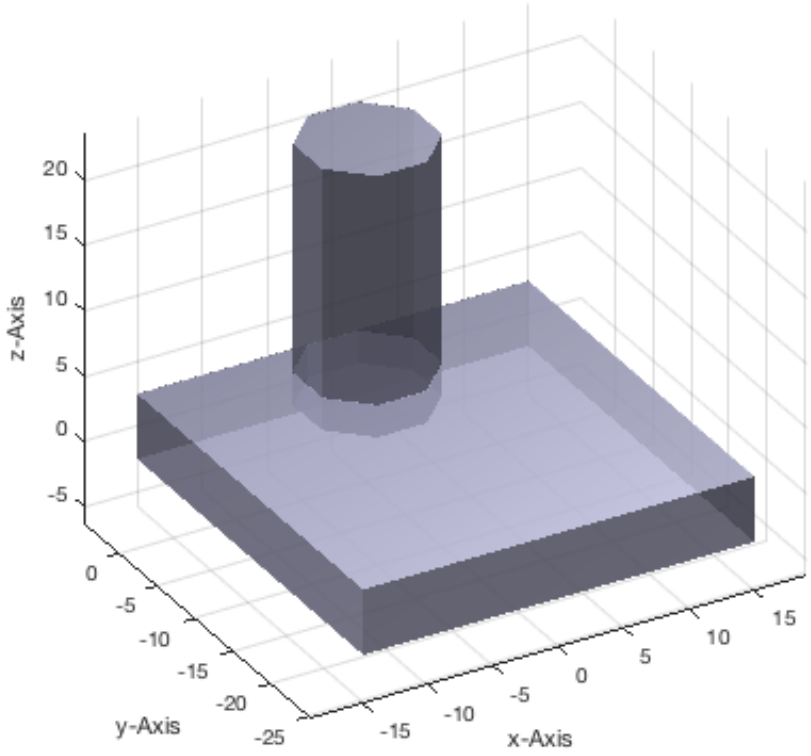
SGalignfront aligns the front of solid A with the front of solid B

```
close all;
SG=SGcat({SGalignfront(A,B),B});
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



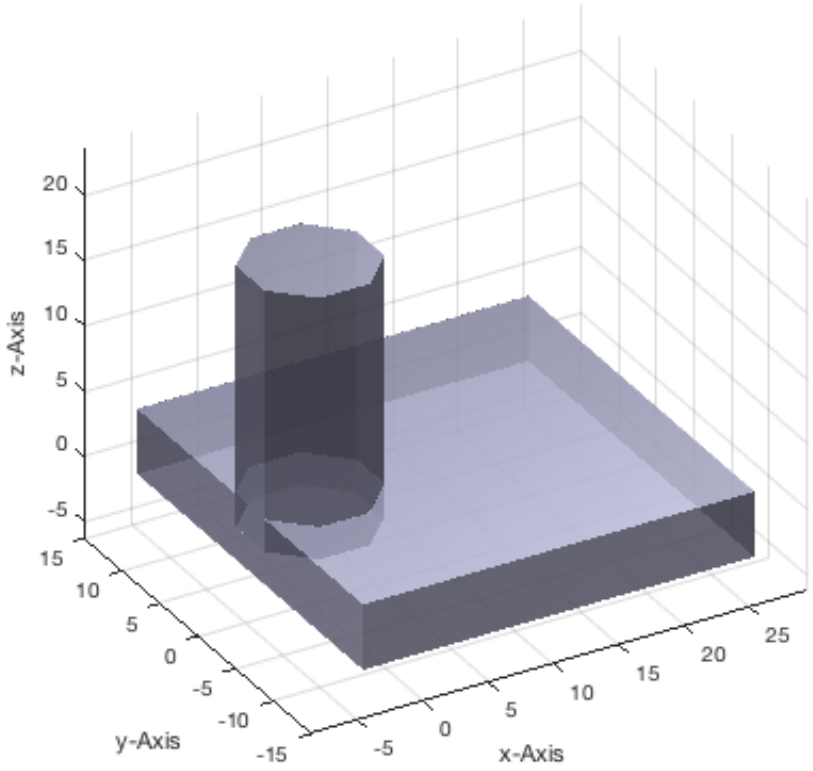
SGalignback aligns the back of solid A with the back of solid B

```
close all;
SG=SGcat({SGalignback(A,B),B});
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



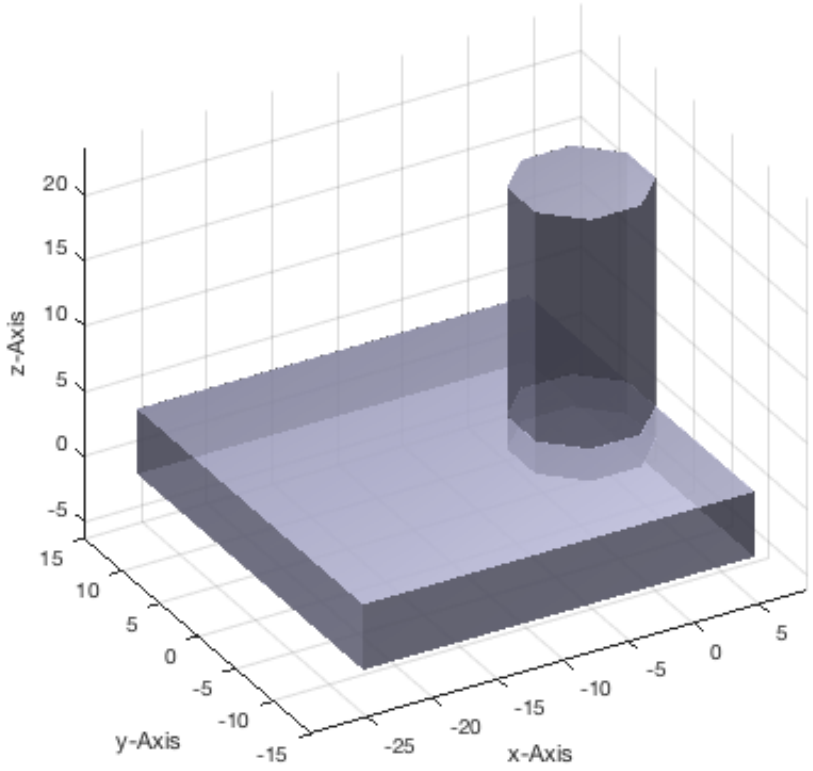
SGalignleft aligns the left side of solid A with the left side of solid B

```
close all;
SG=SGcat({SGalignleft(A,B),B});
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



SGalignright aligns the right side of solid A with the right side of solid B

```
close all;
SG=SGcat({SGalignright(A,B),B});
SGplot (SG,'w'); VLFLplotlight(1,0.7); view (-30,30);
```



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 22:58:41 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 2014-11-25*
- *Mattias Traeger, executed and published on 64 Bit PC using Windows with Matlab 2014b on YYYY-MMM-DD*

Published with MATLAB® R2016b