Tutorial 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) - 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 1.8 required)

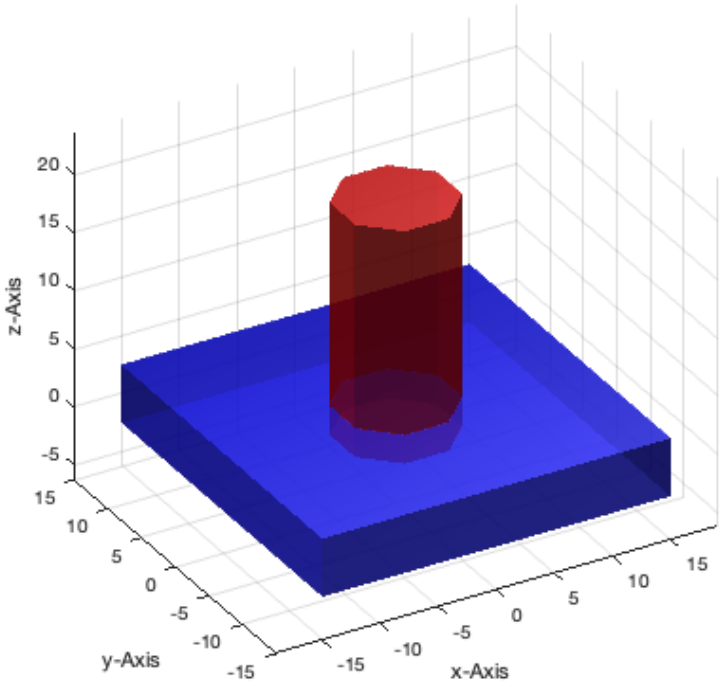
2. Relative spatial positioning of solid geometries (SG) using bounding boxes

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'

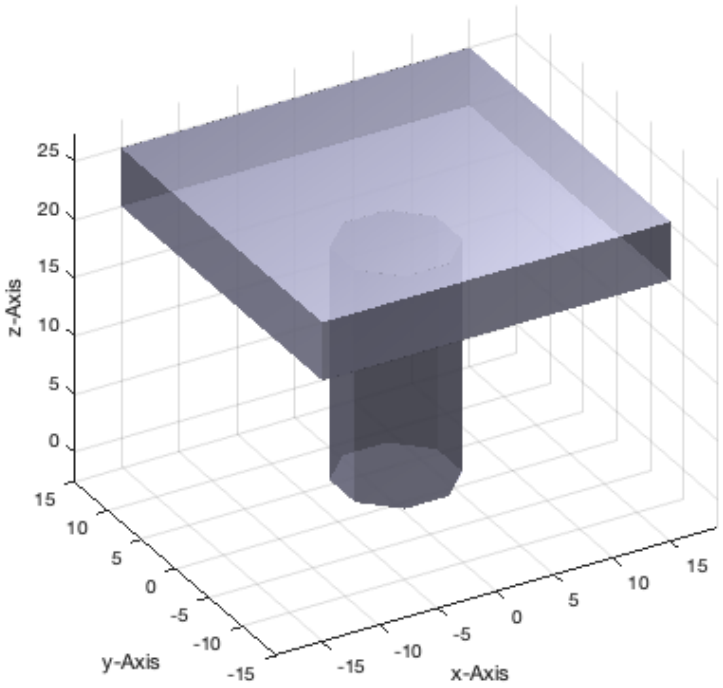
Define two solid geometrys 'A' and 'B'

```
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);
```



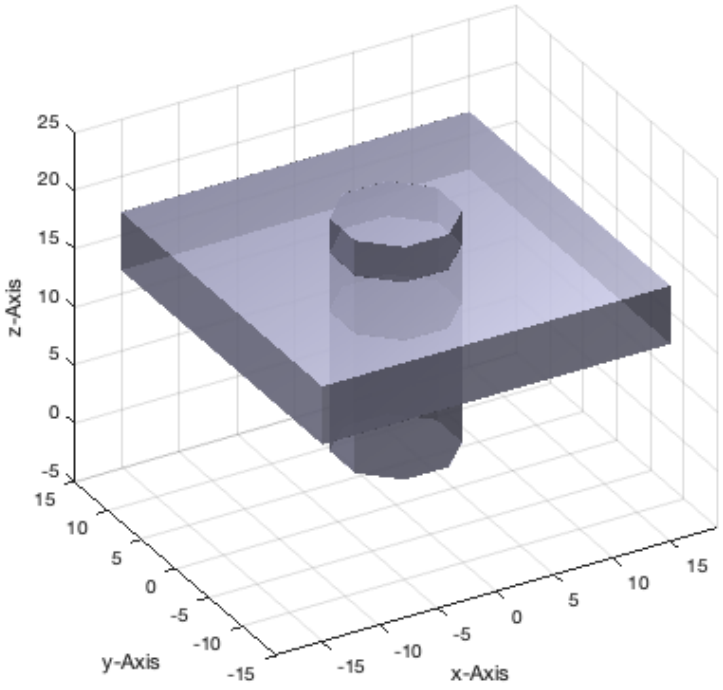
SGontop positions a solid geometry 'A' on top of solid geometry 'B'

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



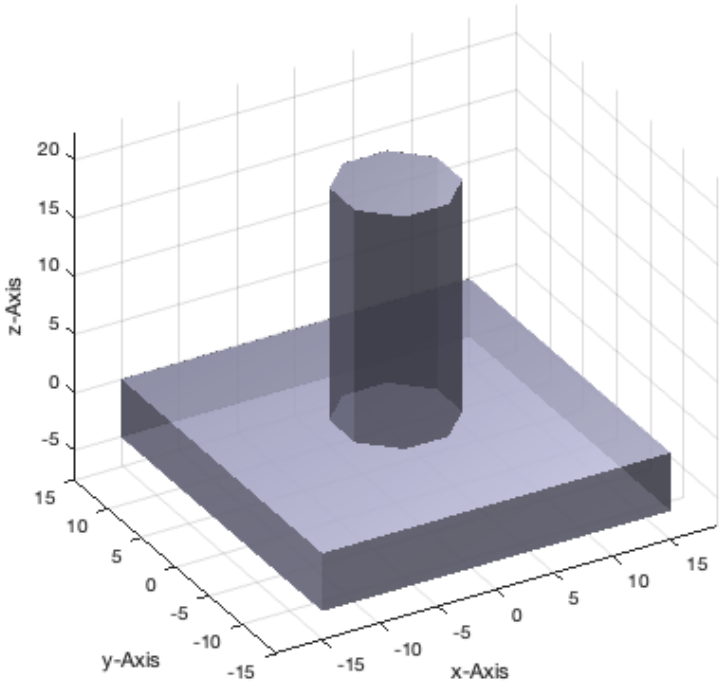
SGontop positions a solid geometry 'A' on top of solid geometry 'B' with a gap of -8

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



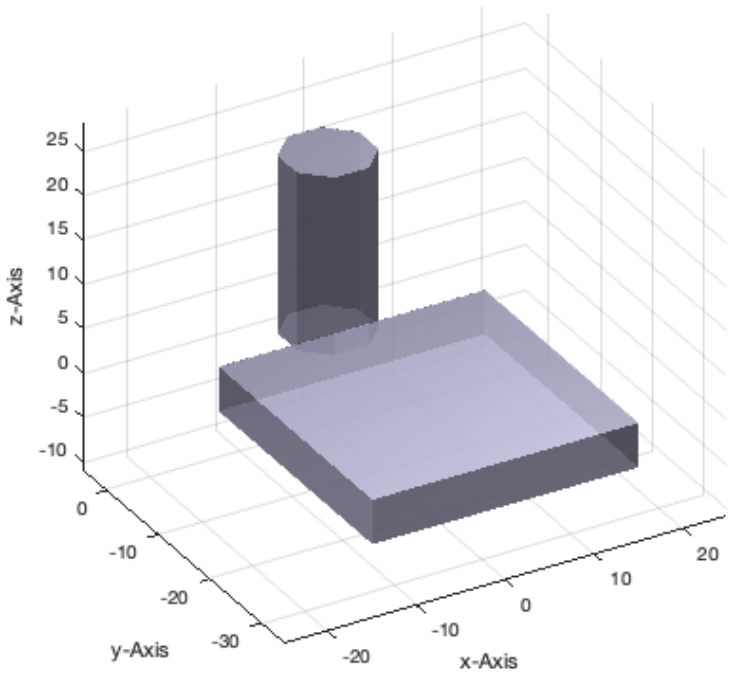
SGunder positions a solid geometry 'A' under of solid geometry 'B'

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



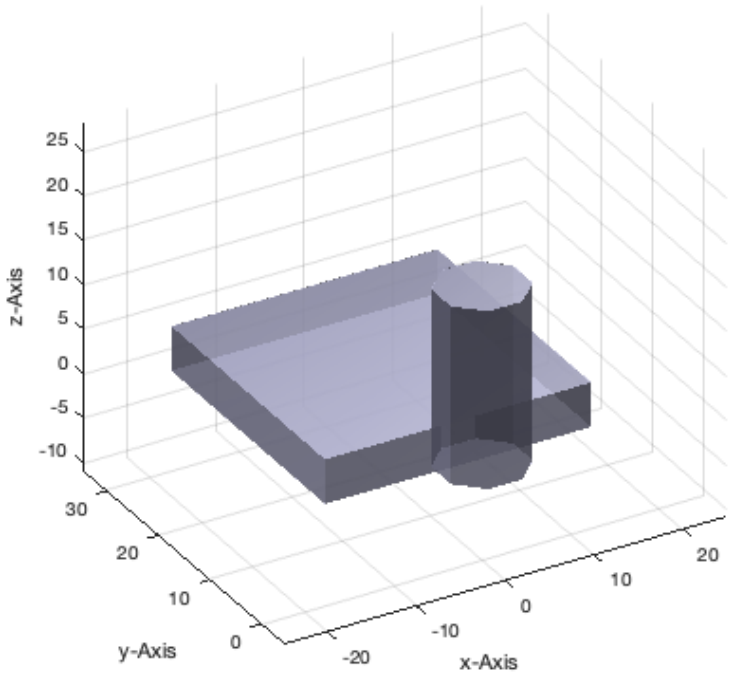
SGinfront positions a solid geometry 'A' in front of solid geometry 'B'

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



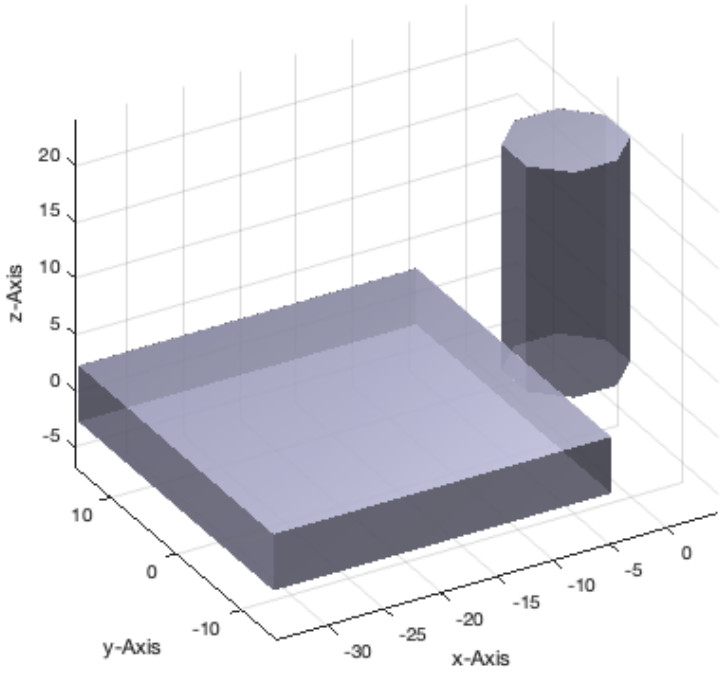
SGbehind positions a solid geometry 'A' behind of solid geometry 'B'

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



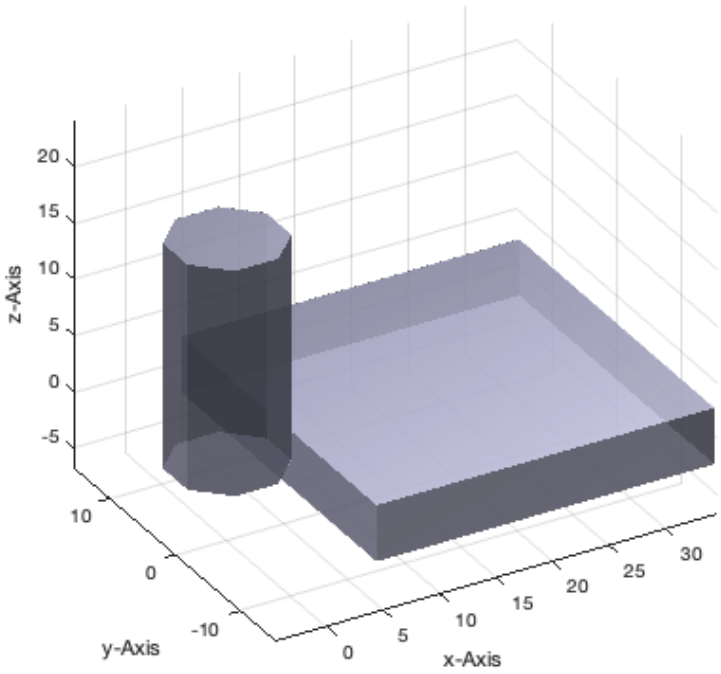
SGleft positions a solid geometry 'A' left of solid geometry 'B'

```
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);
```



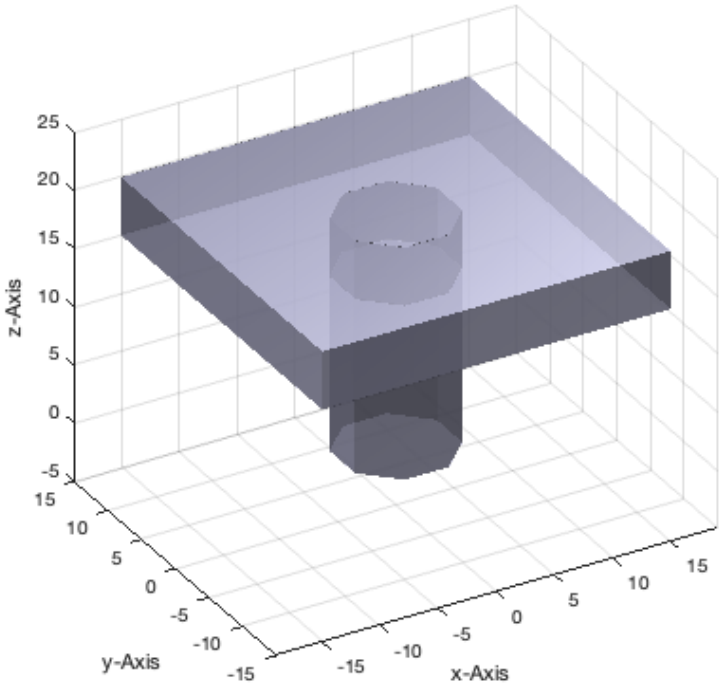
3. Relative spatial alignment of solid geometries (SG) using bounding boxes

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

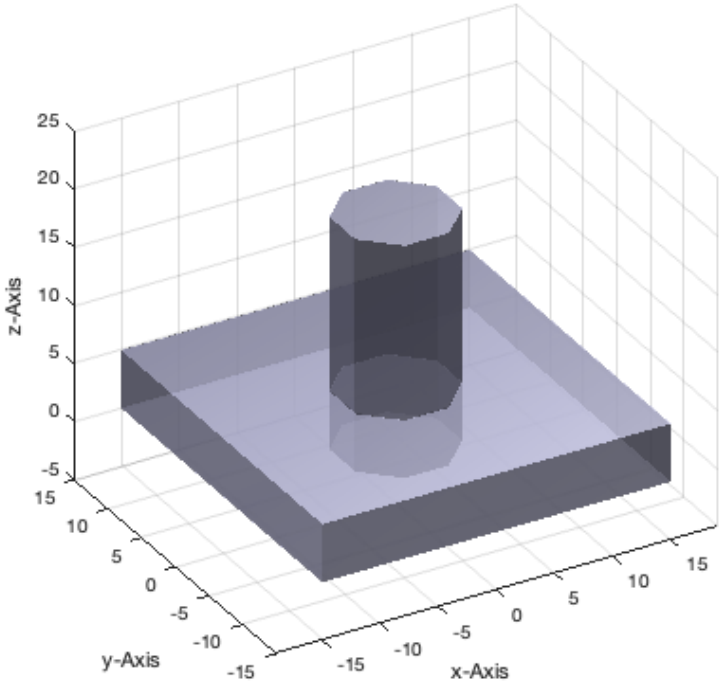
SGaligntop aligns the top of solid A with the top of solid B

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



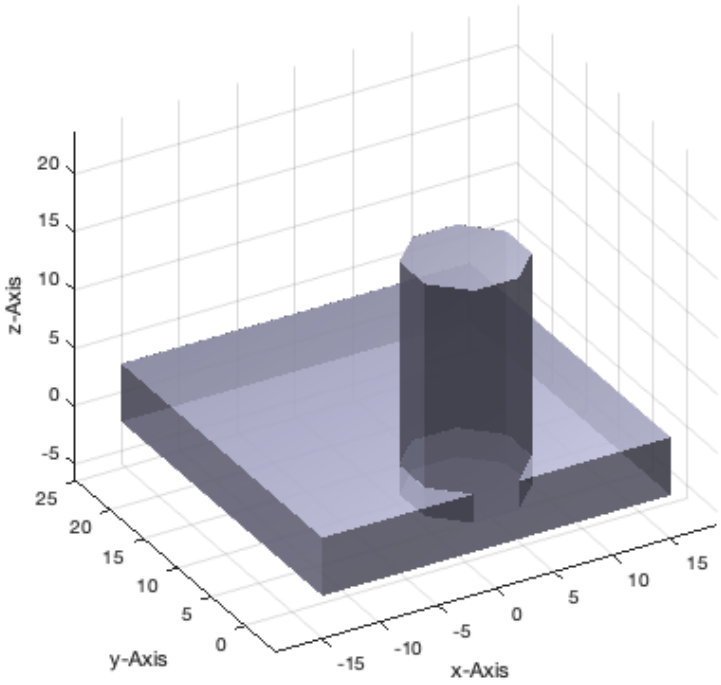
SGalignbottom aligns the bottom of solid A with the bottom of solid B

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



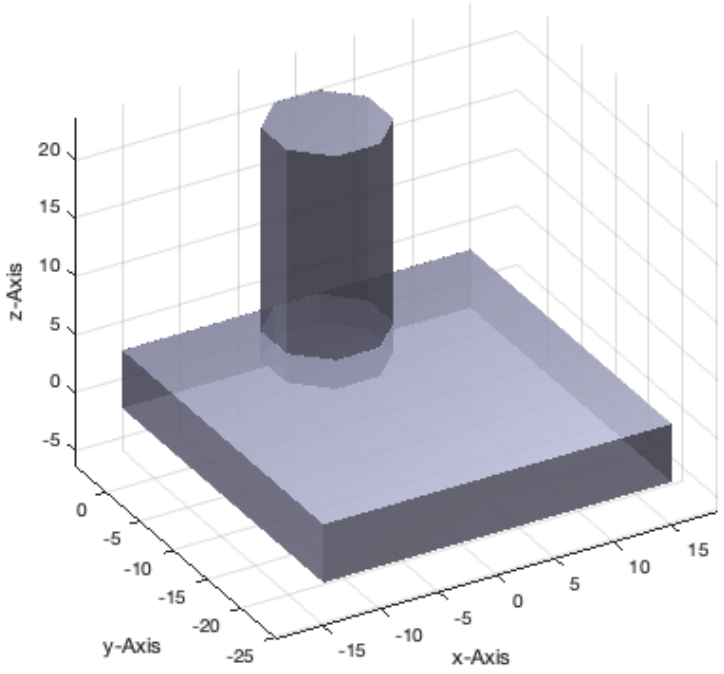
SGalignfront aligns the front of solid A with the front of solid B

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



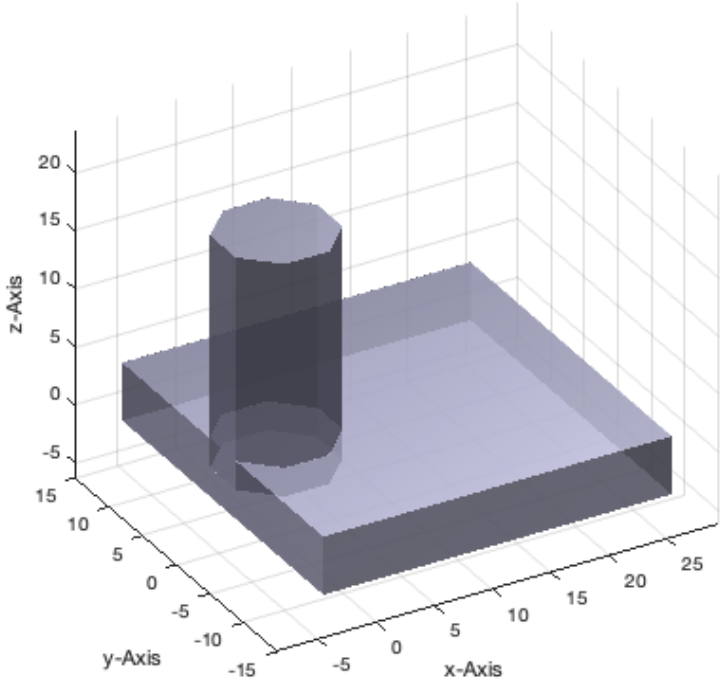
SGalignback aligns the back of solid A with the back of solid B

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



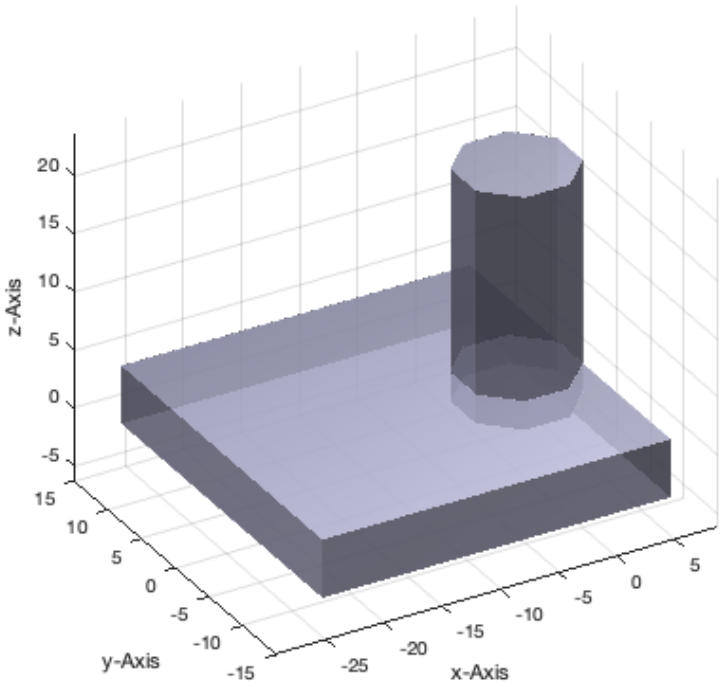
SGalignleft aligns the left side of solid A with the left side of solid B

```
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.  (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 07:11:06!
Executed 03-Oct-2023 07:11:08 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 2014-11-25*

- *Mattias Traeger, executed and published on 64 Bit PC using Windows with Matlab 2014b on YYYY-MMM-DD*

Published with MATLAB® R2023a