

Tutorial 09: Boolean Operations with Solid Geometries

2014-11-30: 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](#)
- [Motivation for this tutorial: \(Originally SolidGeometry 2.0 required\)](#)
- [3. Creating two solids for showing the boolean operations](#)
- [4. Boolean operator: Substraction A-B or A\B](#)
- [5. Boolean operator: Substraction A+B](#)
- [6. Boolean operator: Substraction B\A](#)
- [7. Boolean operator: A xor B](#)
- [8. Analyzing the results and comparision with additive design.](#)
- [Final remarks on toolbox version and execution date](#)

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 2.0 required)

The implementation of the boolean operations based on STL geometries took the author several years. The reason was not only the complexity of the numerous special cases but also the numerical accuracy or resolution of the required geometrical calculations. So even if a normal position is calculated with 12 digits accuracy, the cross product often has just an accuracy of 6 digits. Unfortunately crossing triangles with position errors of 6 digits can easily lead to phantom triangles, phantom edges which either do not really exist or are just doubles of already existing lines. Since normally all edges must have a second edge with the opposite direction, doubled lines/edges with the same direction make trouble. So to be successful with the boolean operations you should make sure that

1. No facet should be in the same plane as or overlap another facet or cross with almost parallel edges to the plane of another facet. This is always valid for one solid, but in case of a second solid for boolean operations, it is quite difficult to guarantee this.
2. It is fact that, the more boolean operations took place, to create a new solid, the more vertices and facets were created. The removal of dispensable vertices and facets is possible but is a boring non productive piece of source code. So the motivation to program such a procedure is not high.
3. No edge of a triangle should be in the sample plane or crossing but almost parallel to a plane of a facet.
4. It is fact that a normal user just want to use the boolean operator without thinking about those problems. The normal user will just be disappointed if the way to design a physical solid object finally fails because of the limitations of the crossing
5. Make definitely sure that after all boolean operations you use SGchecker to analyze the solid geometry do detect errors immediately.
6. Maybe the only solution is to use a fixed coordinate grid during all calculations to make sure that two vertices are either definitely separated or definitely the same. #

3. Creating two solids for showing the boolean operations

```
function VLFL_EXP09
```

```
VLFLfigure; view(-30,30); grid on;

A=SGofCPLz(PLcircle(10,4),10); A=SGtrans0(A);
B=SGofCPLz(PLcircle(5,10),30); B=SGtrans0(B); B=SGtransR(B,rotdeg(45,5,0));

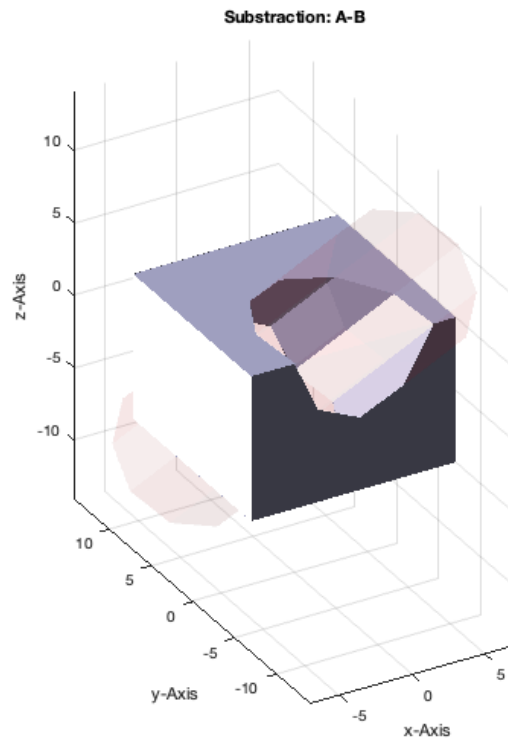
SGplot(A,'b');
SGplot(B,'r');
VLFLplotlight(0,0.9);
A=SGstripfields(A)
B=SGstripfields(B)

SGbool ('-' ,A,B);
```

```
A =
  struct with fields:

    VL: [8×3 double]
    FL: [12×3 double]
B =
  struct with fields:

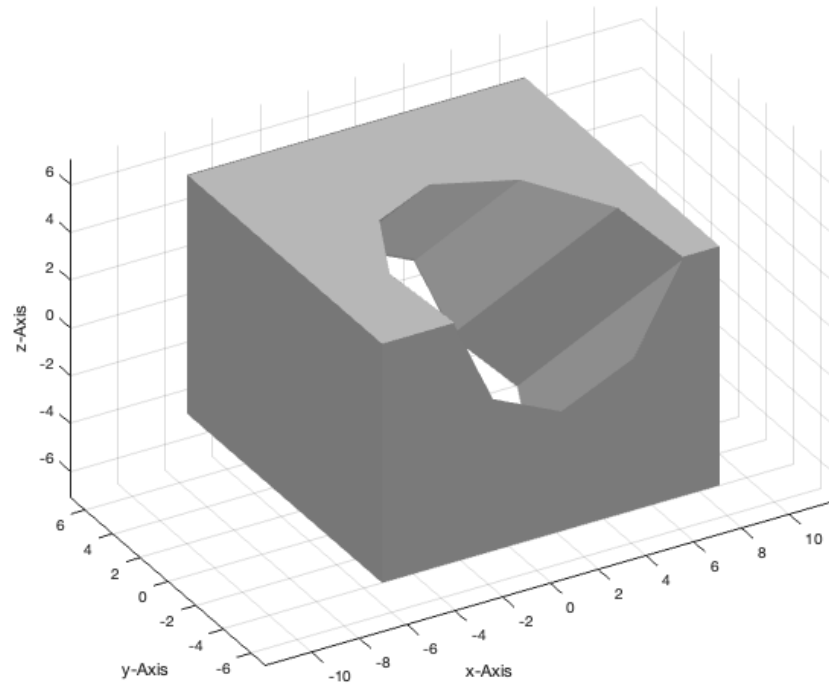
    VL: [20×3 double]
    FL: [36×3 double]
SGchecker "A-B":
```



4. Boolean operator: Substraction A-B or A\B

```
X=SGbool ('A',A,B);  
SGfigure(X); view(-30,30);
```

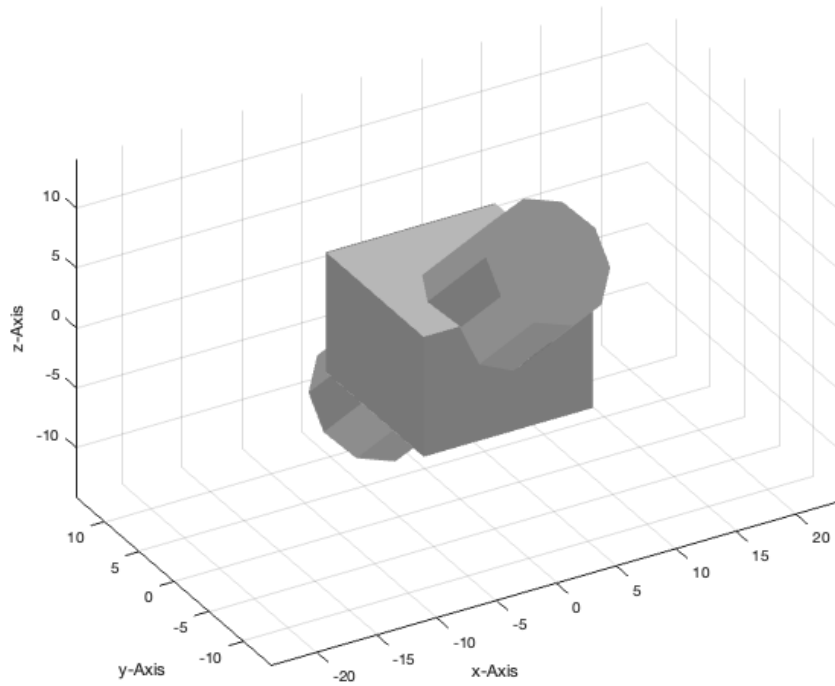
publishSGPDF: 2023-10-03 07:12:15



5. Boolean operator: Substraction A+B

```
X=SGbool ('+',A,B);  
SGfigure(X); view(-30,30);
```

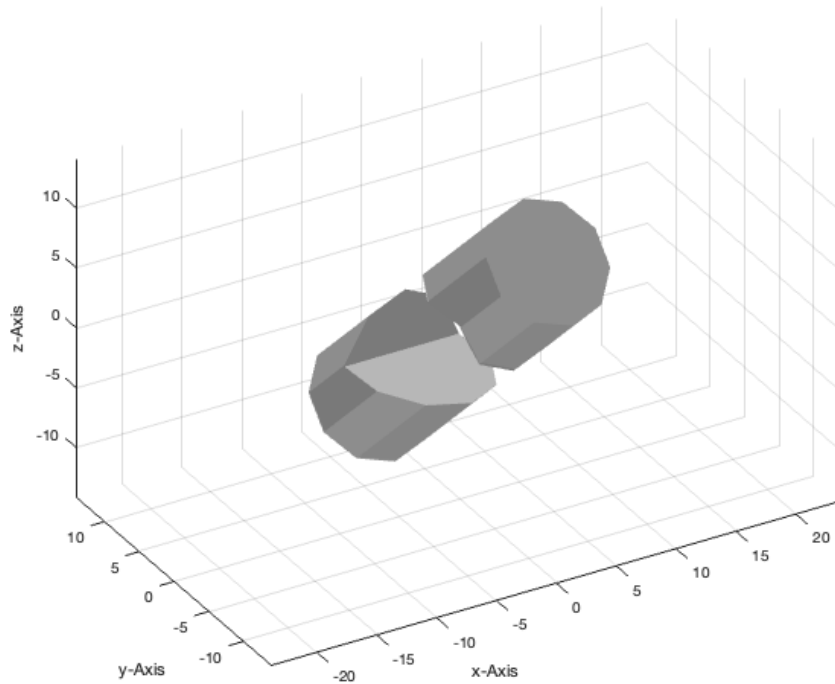
publishSGPDF: 2023-10-03 07:12:17



6. Boolean operator: Substraction B/A

```
X=SGbool ('B',A,B);  
SGfigure(X); view(-30,30);
```

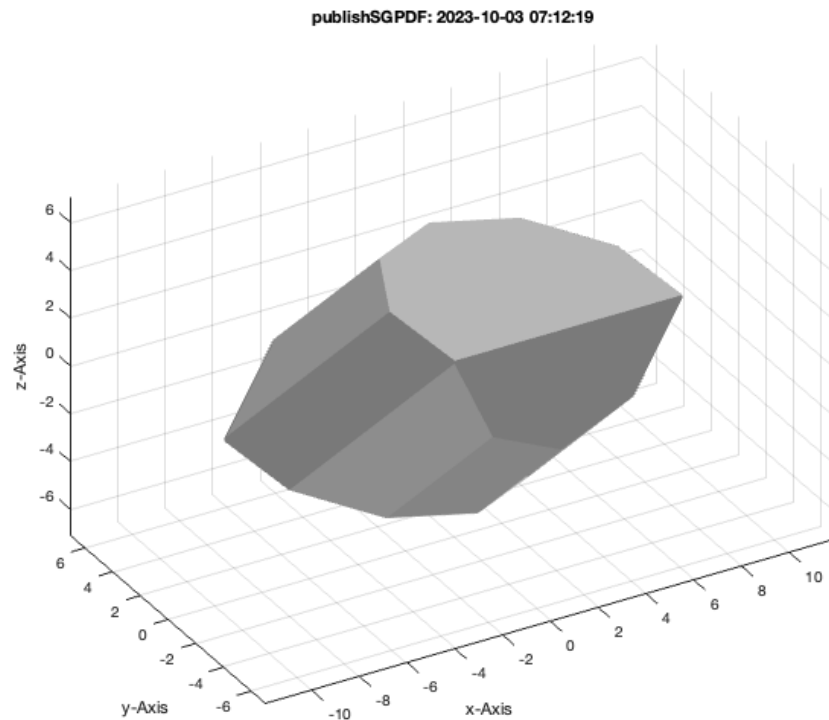
publishSGPDF: 2023-10-03 07:12:18



7. Boolean operator: A xor B

```
X=SGbool ('x',A,B);
```

```
SGfigure(X); view(-30,30);
```



8. Analyzing the results and comparison with additive design.

Analyzing the number of vertices and facets of the results of a boolean operation shows clearly that there are much more vertices and facets than the sum of the vertices and facets. In general it makes for STL more sense to add simple solids to a more complex by attaching them to another by simply pushing them into another. In this case the final number of vertices and facets is the sum of the individual facets and vertices.

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 07:12:19!

Executed 03-Oct-2023 07:12:21 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 2015-06-07*
- *Christina Friedrich, executed and published on 64 Bit PC using Windows with Matlab 2015a on 2015-06-17*

Published with MATLAB® R2023a