Tutorial 45: Creation of Solids using the SG-Coder - SGofCPLcommand

2018-11-25: Tim C. Lueth and Yilun Sun, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de) - Last Change: 2018-11-25

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: Creation of Kinematic Chains and Robot Structures

Motivation for this tutorial: (Originally SolidGeometry 4.4 required)

The function SGofCPLcommand has been written for the fast generation of solids via a formal language. With this function, solids can be generated from short character strings in a geometric language and reverse polish notation. This tutorial shows how this language can be used.

List of function introduced in this tutorial

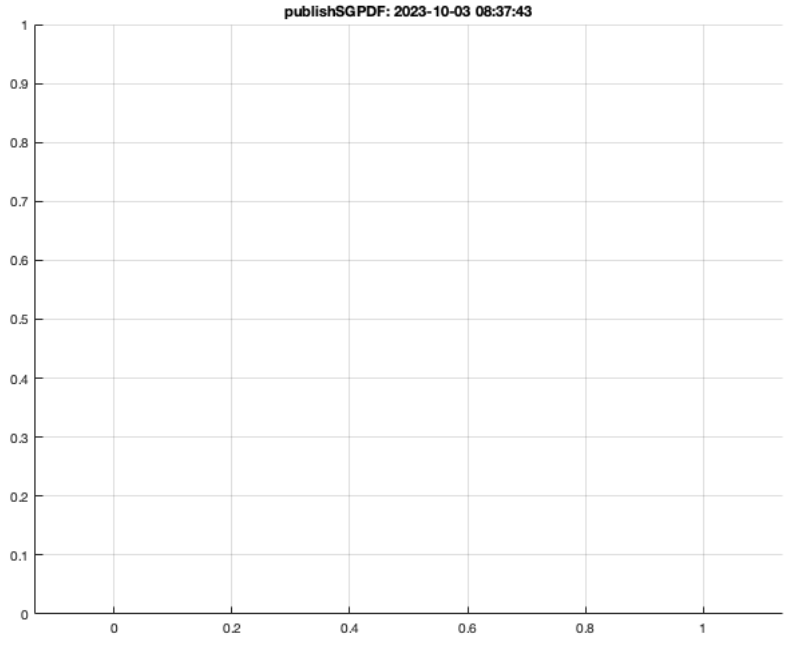
- SGofCPLcommand - Formal language (RPN/Forth) to create solid volumes *

1.1 First use of 2.5 D design using simple contour by extrusion and rotation

```
SGofCPLcommand('help');
```

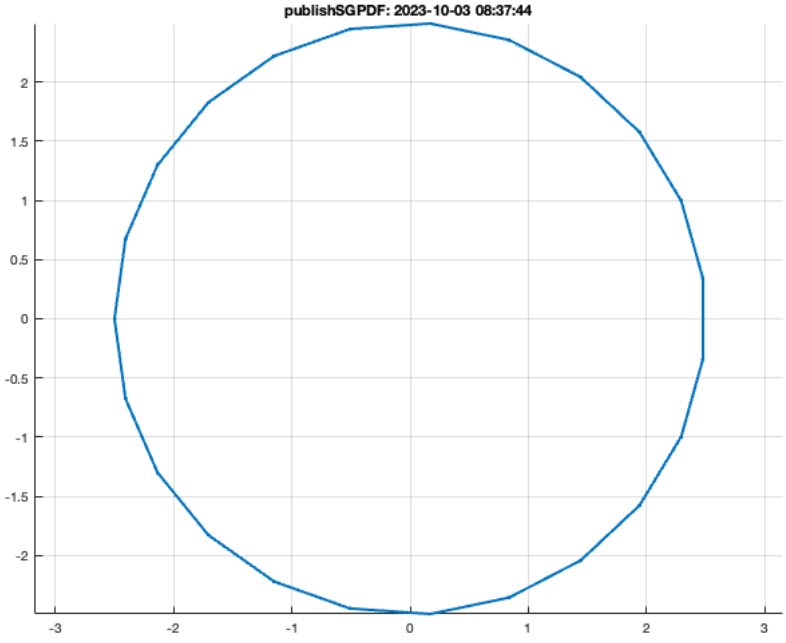
```
SGofCPLcommands supports the following commands, separated by commas:
See publishable tutorial VLFL_EXP45.m


===  CPL shape commands  ================================
b x-size y-size [d]                     => Box as rectangle or displace trapaze
c diameter diameter edges               => Cylinder or ellipse as polygon
cs phi r-outer r-inner offset           => Cylinder segment with angle
co r-outer length r-inner               => Cylinder oval segment with optional holes
d diameter x-coord y-coord n-faces      => Drilling hole at x/y with n edges
g diameter teeth-nr turn                => Gear that it turned (f.i. 0.5 teeth)
ms diameter-1 diameter-2                => Motor shaft contour

===  CPL manipulation commands  =========================
move x-coord y-coord                    => Move the CPL relatively
cp x-coord y-coord                      => Center point change
ch distance radius                      => Convex hull in distance with optional radius
dupc x-copies y copies distance         => Duplicates contour in as x y pattern
dupr radius number offset               => Duplicates contour radial in n copies
rad radius                              => Radial edges (+r) or cuts (-r) at each corner
rot degree x y                          => Rotate a contour ccw by degree on centerpoint x y, default is center of CPL
mirr x1 y1 x2 y2                               => mirror CPL at straigth line through p1 p2
fitt char                               => create fitting buffer 'c', 't', 'i' for all holes in CPL

===  CPL stack commands  ================================
enter                   => current CPL is shifted to the stack
dup                     => current CPL is duplicated to the stack
swap                    => current CPL is swaped with CPL on stack
+/add                   => current CPL is added to stack CPL
-/sub                   => current CPL is substracted from stack CPL
rem                     => stack CPL ist substracted from current CPL
&/isec                  => current CPL is intersected with stack CPL
hs height               => stack connection using height

===  CPL to solid commands  =============================
h height z-displacement                 => Height of the extruded solid
hc height z-displacement                => Height of extrusion with smoothed edges
r angle pitch                           => Extrusion by rotation (degree) and optional pitch

===  SG element commands  ===============================
scr mm length diameter                  => Screw/Cutter/Nut threat of diameter and length
sph diameter  end-angle                 => Sphere
text string size-x size-z               => Text string without space

===  SG manipulation commands  ==========================
move x-coord y-coord                    => Move the solid relatively
cham rar upper-dist lower-dist          => Chamfer the edges of a 2.5 solid
rotx degree                             => rotate around the x-axis
roty degree                             => rotate around the y-axis
rotz degree                             => rotate around the z-axis
mirr x-degree y-degree z-displacement   => mirror at the xy plane rotated around x and y-axis shifted in z
grow offset                             => Grow/shrink the current solid with an offset
melt                                    => Boolean addition of all solid elements
textstamp string                        => Add a string to the largest surface
dupr number                             => Duplicates solids radial in n copies
dups nx ny nz d                         => Duplicates solid spatial in x y z with distance d
dupg nx ny nz d                         => Duplicates solid on a grid in x y z with distance d
hollow wall                             => Creates a hollow solid
shell wall distance                     => Creates a shell for the solid
cutx x1 x2                              => Cuts the solid at x1 and x2
cuty y1 y2                              => Cuts the solid at y1 and y2
```

```
cutz z1 z2                              => Cuts the solid at z1 and z2

===  SG save and restore commands  ================================
save name                               => saves the solid into name.SG
load name                               => loads a solid from name.SG
write name                              => saves the solid into File name.STL
read name                               => reads the solid from File name.STL
mag fact                                => magnifies the current solid
col color                               => colors the faces of the current solid

===  SG stack commands  ===========================================
rel command parameter    => current SG is move relatively to stack
enter                    => current SG is shifted to the stack
dup                      => current SG is duplicated to the stack
swap                     => current SG is swaped with SG on stack
clear nr                 => Clear stack content 1..n
+/add                    => current SG is added to stack SG
-/sub                    => current SG is substracted from stack SG
rem                      => stack SG is substracted from current SG
&/isec                   => current SG is intersected with stack SG

===  Frame commands  ==============================================
fset name ident rotation feature         => attach a frame to the current SG
falign Stack-Frame SG-Frame degree       => align current SG-frame with stack frame and rotate

===  Macro commands  ==============================================
$cmd: <string>:          => Define macro using $1 $2 $3 $4 as parameters
$cmd $1 $2 $3 $4         => Use macro as command with $1 $2 $3 $4 as parameters

Try: SGofCPLcommand('g 3 22, h 5, c 5, h 20, roty -90, move 10 10')
Try: SGofCPLcommand('$cob: co $1 $2 $3, h $3, fset B 1 0 R1, fset F 2 0 R2:, $cob 10 50 5, dup, falign F B 45')
Try: SGofCPLcommand('b 30 10, h 10, enter, $hole: c 3, h 10, rotx 90,move $1 $2 $3 :,$hole 10 5 5,-,enter, $hole 0 5 5,-,enter, $hole -10 5 5,-') % by

SGofCPLcommand: SGofCPLcommand("help")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("help");');
```



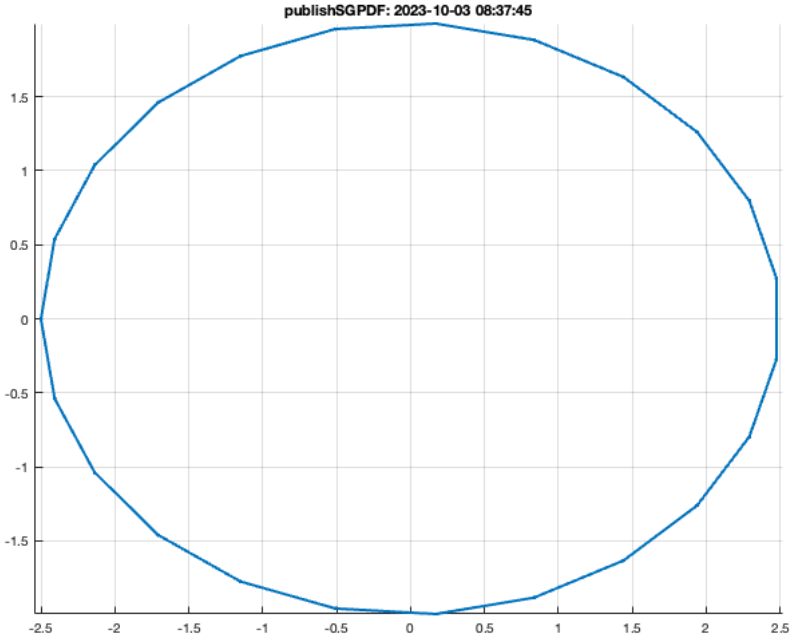

```
SGofCPLcommand('c 5');
```

```
SGofCPLcommand: SGofCPLcommand("c 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5");');
```



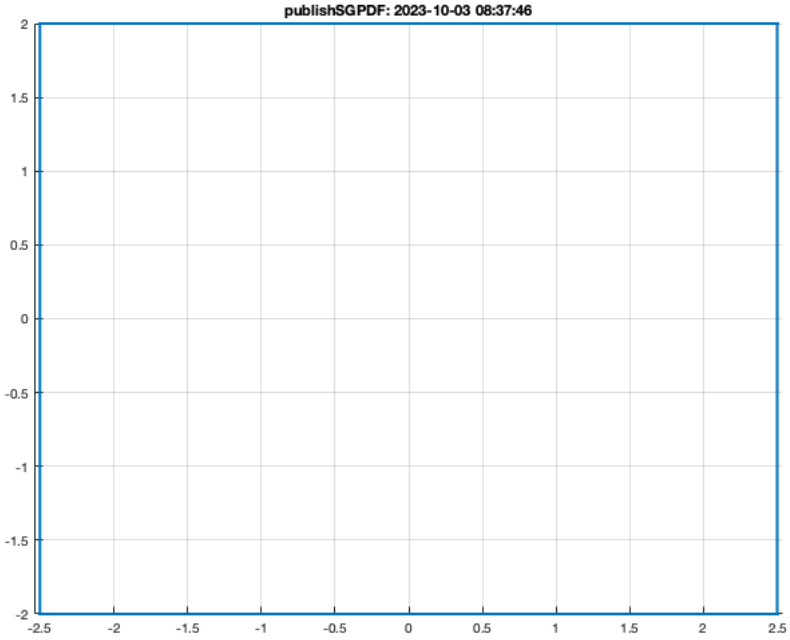

```
SGofCPLcommand('c 5 4');
```

```
SGofCPLcommand: SGofCPLcommand("c 5 4")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5 4");');
```



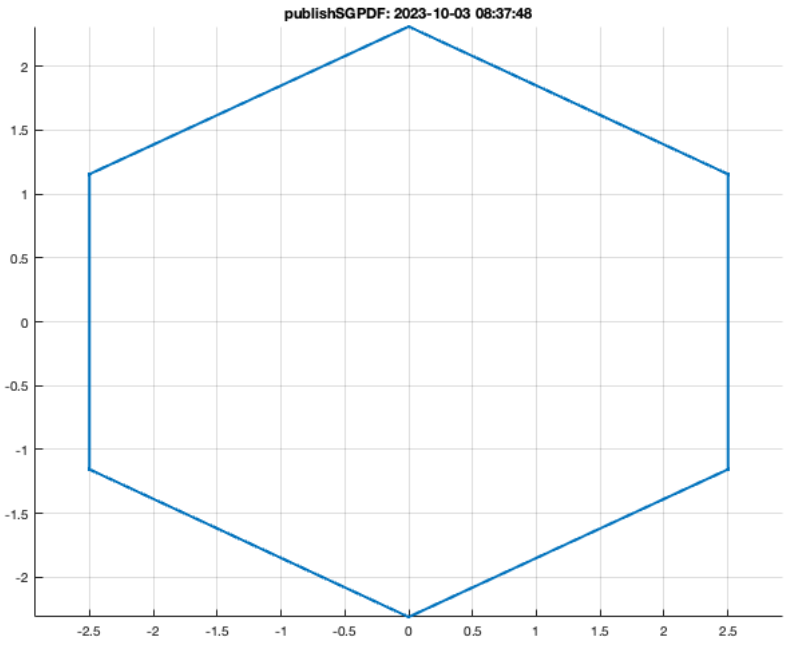

```
SGofCPLcommand('c 5 4 4');
```

```
SGofCPLcommand: SGofCPLcommand("c 5 4 4")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5 4 4");');
```



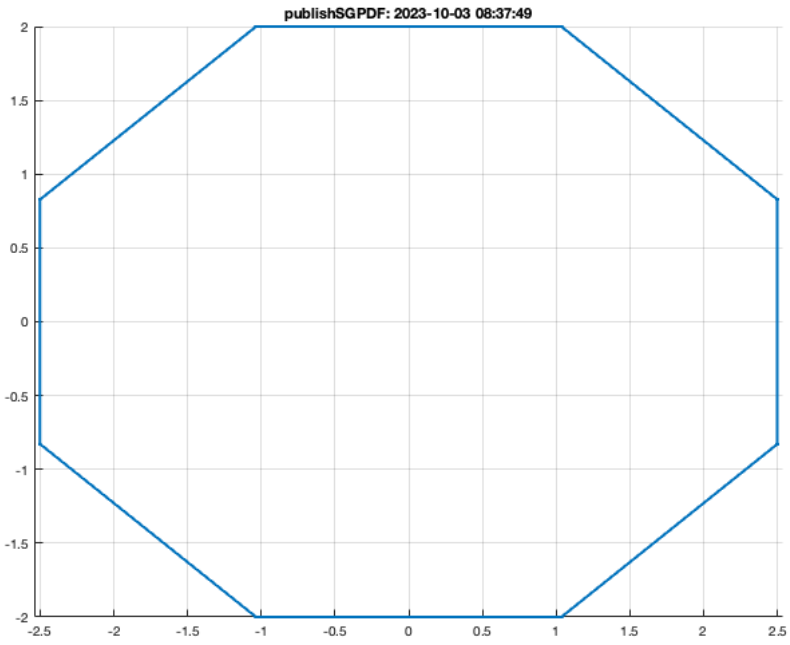

```
SGofCPLcommand('c 5 4 6');
```

```
SGofCPLcommand: SGofCPLcommand("c 5 4 6")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5 4 6");');
```



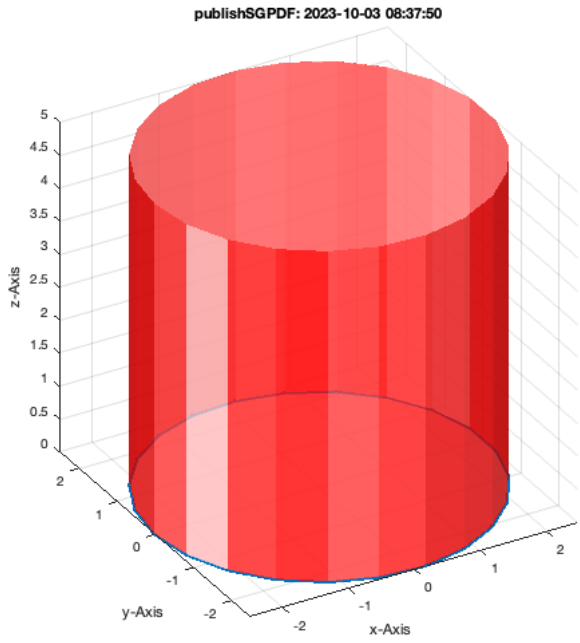

```
SGofCPLcommand('c 5 4 8');
```

```
SGofCPLcommand: SGofCPLcommand("c 5 4 8")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5 4 8");');
```



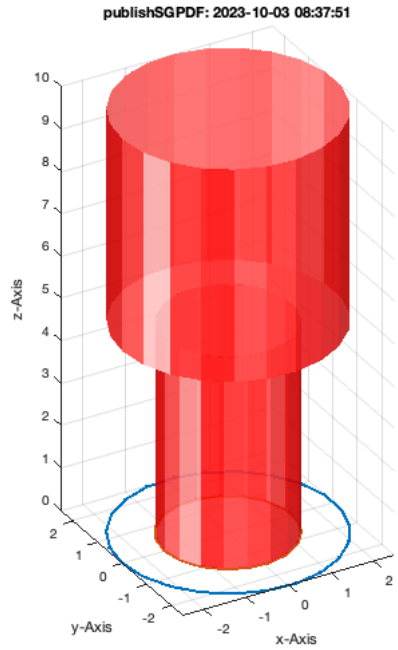

```
SGofCPLcommand('c 5,h 5');
```

```
SGofCPLcommand: SGofCPLcommand("c 5,h 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5,h 5");');
```



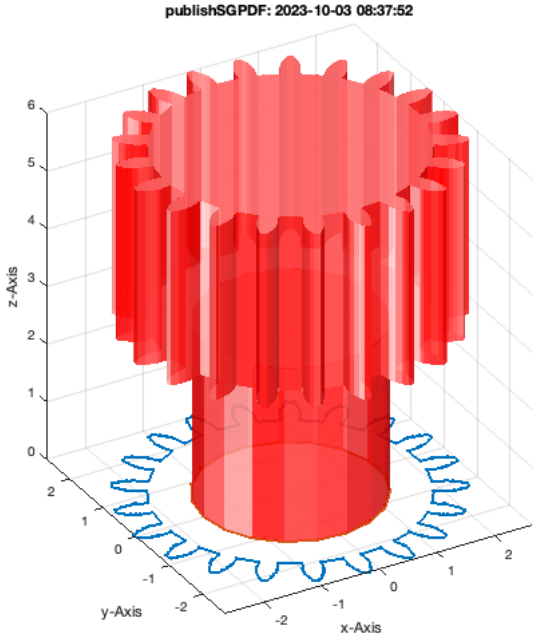

```
SGofCPLcommand('c 5,h 5, c 3, h 5');
```

```
SGofCPLcommand: SGofCPLcommand("c 5,h 5, c 3, h 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5,h 5, c 3, h 5");');
```



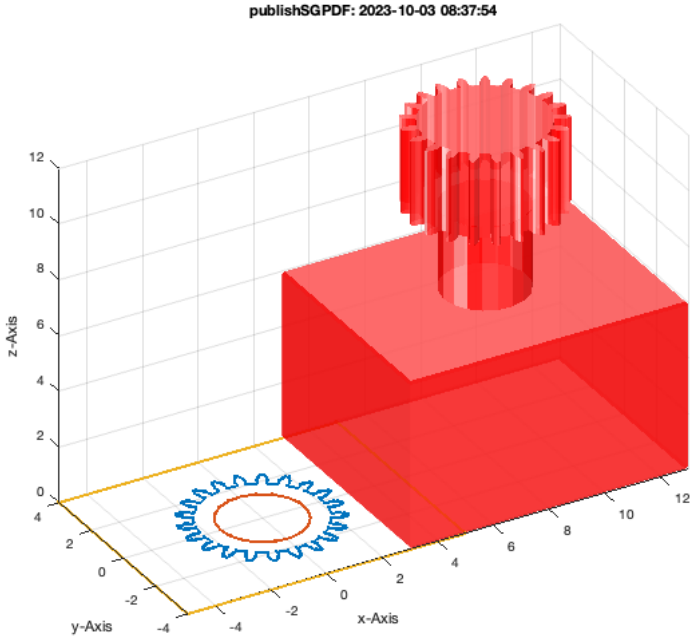

```
SGofCPLcommand('g 5 21,h 3, c 3, h 3');
```

```
SGofCPLcommand: SGofCPLcommand("g 5 21,h 3, c 3, h 3")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("g 5 21,h 3, c 3, h 3");');
```



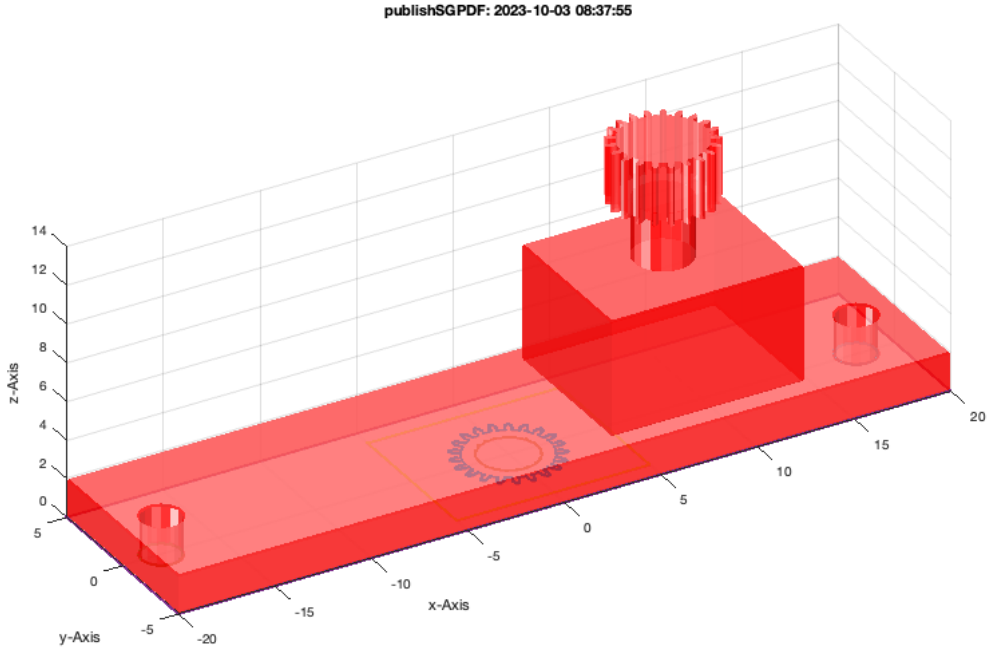

```
SGofCPLcommand('g 5 21,h 3, c 3, h 3, b 10 8, hc 6, move 8 0');
```

```
SGofCPLcommand: SGofCPLcommand("g 5 21,h 3, c 3, h 3, b 10 8, hc 6, move 8 0")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("g 5 21,h 3, c 3, h 3, b 10 8, hc 6, move 8 0");');
```



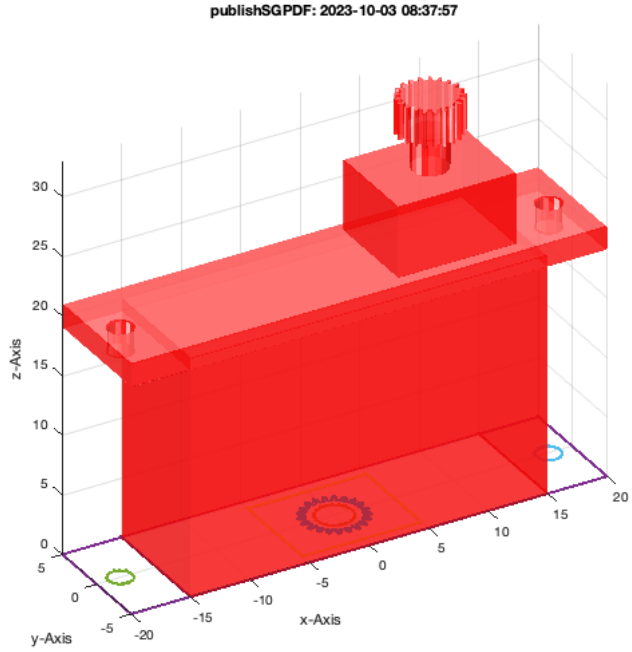

```
SGofCPLcommand('g 5 21,h 3, c 3, h 3, b 10 8, hc 6, move 8 0,b 40 10, d 2 -18 0, d 2 18 0, hc 2 ');
```

```
SGofCPLcommand: SGofCPLcommand("g 5 21,h 3, c 3, h 3, b 10 8, hc 6, move 8 0,b 40 10, d 2 -18 0, d 2 18 0, hc 2")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("g 5 21,h 3, c 3, h 3, b 10 8, hc 6, move 8 0,b 40 10, d 2 -18 0, d 2 18 0, hc 2");');
```



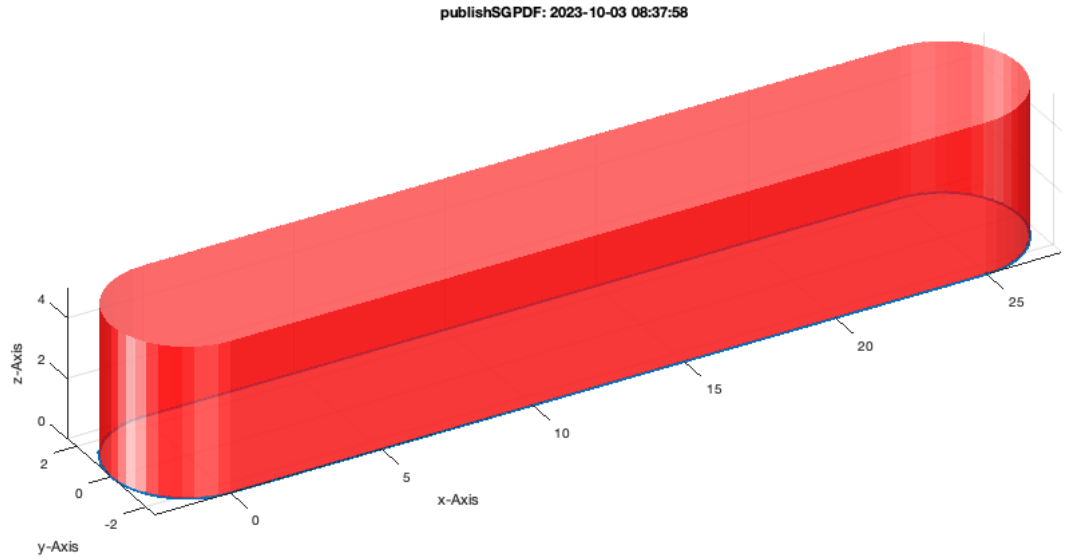

```
SGofCPLcommand('g 5 21,h 3, c 3, h 3, b 10 8, hc 6, move 8 0,b 40 10, d 2 -18 0, d 2 18 0, hc 2, b 30 10, hc 20 -1 ');
```

```
SGofCPLcommand: SGofCPLcommand("g 5 21,h 3, c 3, h 3, b 10 8, hc 6, move 8 0,b 40 10, d 2 -18 0, d 2 18 0, hc 2, b 30 10, hc 20 -1")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("g 5 21,h 3, c 3, h 3, b 10 8, hc 6, move 8 0,b 40 10, d 2 -18 0, d 2 18 0, hc 2, b 30 10, hc 20 -1");');
```



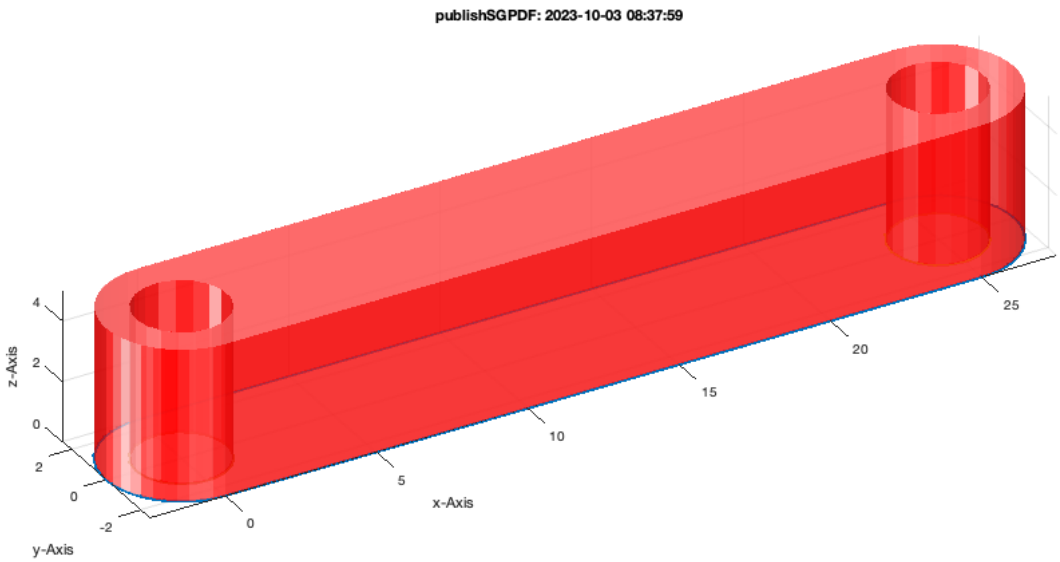

```
SGofCPLcommand('co 5 30, h 5');
```

```
SGofCPLcommand: SGofCPLcommand("co 5 30, h 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("co 5 30, h 5");');
```



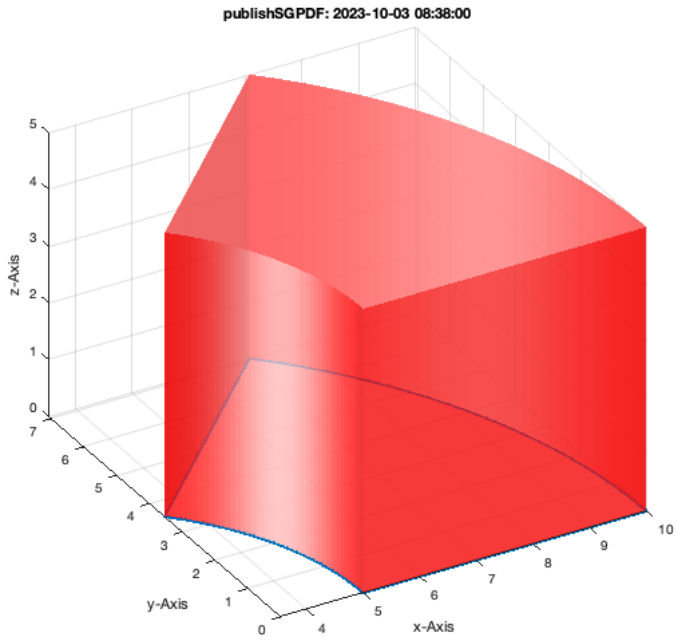

```
SGofCPLcommand('co 5 30 3, h 5');
```

```
SGofCPLcommand: SGofCPLcommand("co 5 30 3, h 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("co 5 30 3, h 5");');
```



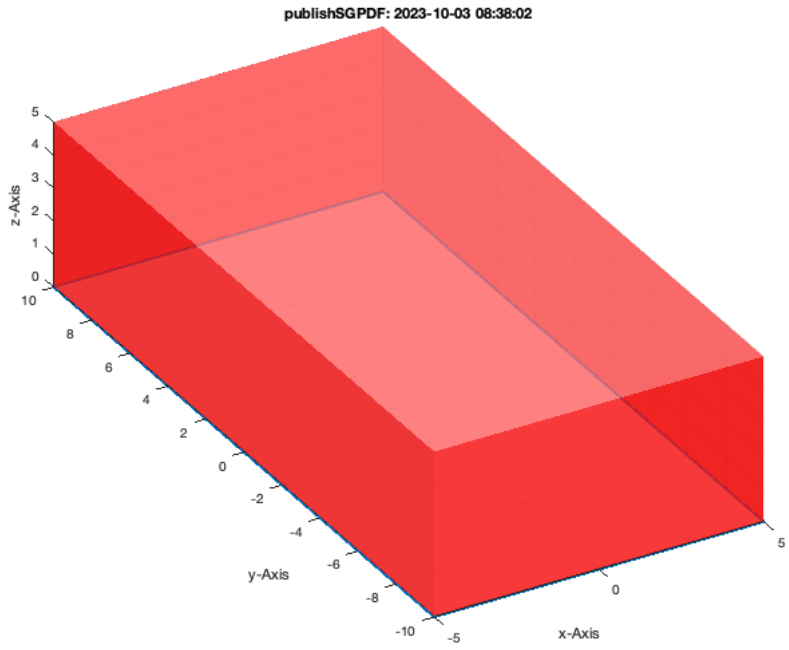

```
SGofCPLcommand('cs 45 10 5, h 5');
```

```
SGofCPLcommand: SGofCPLcommand("cs 45 10 5, h 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("cs 45 10 5, h 5");');
```



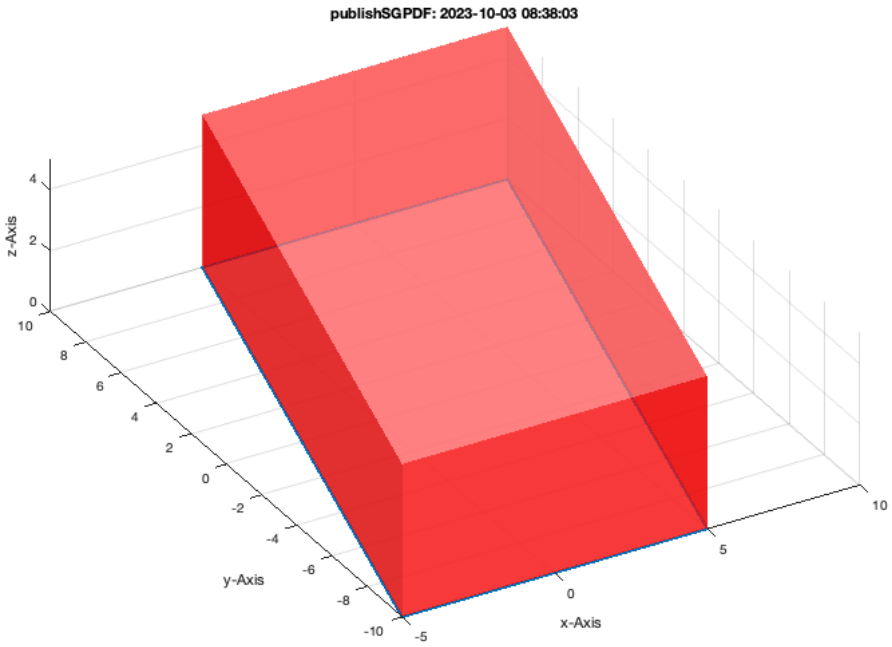

```
SGofCPLcommand('b 10 20, h 5');
```

```
SGofCPLcommand: SGofCPLcommand("b 10 20, h 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 20, h 5");');
```



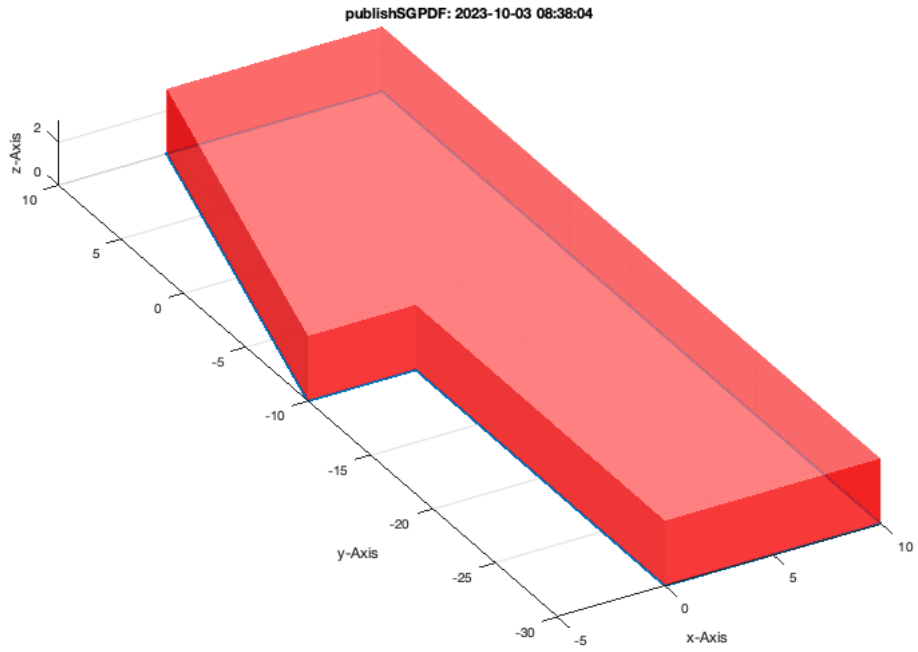

```
SGofCPLcommand('b 10 20 5, h 5');
```

```
SGofCPLcommand: SGofCPLcommand("b 10 20 5, h 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 20 5, h 5");');
```



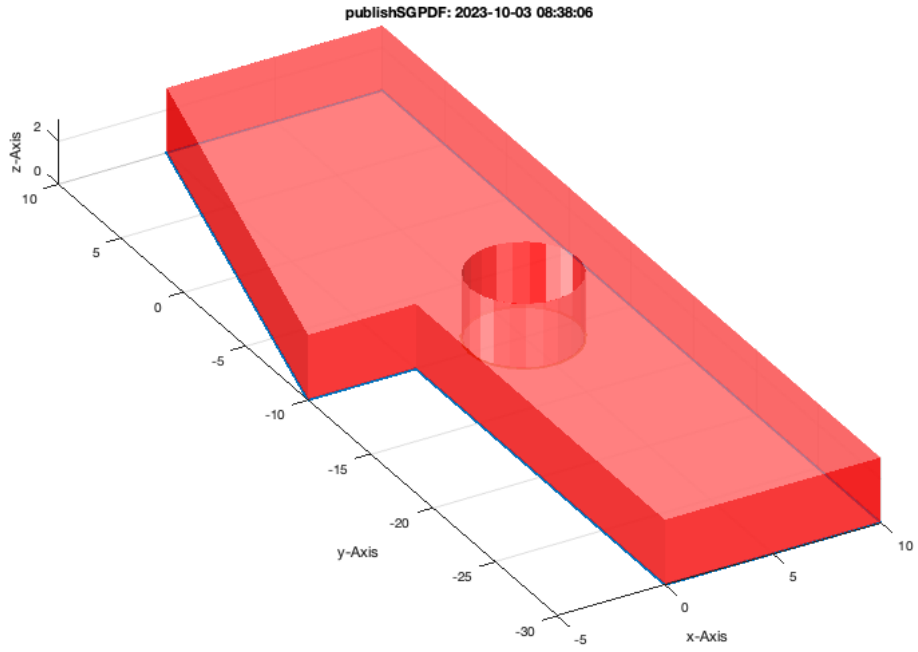

```
SGofCPLcommand('b 10 40, move 5 -10, b 10 20 5, h 3');
```

```
SGofCPLcommand: SGofCPLcommand("b 10 40, move 5 -10, b 10 20 5, h 3")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 40, move 5 -10, b 10 20 5, h 3");');
```



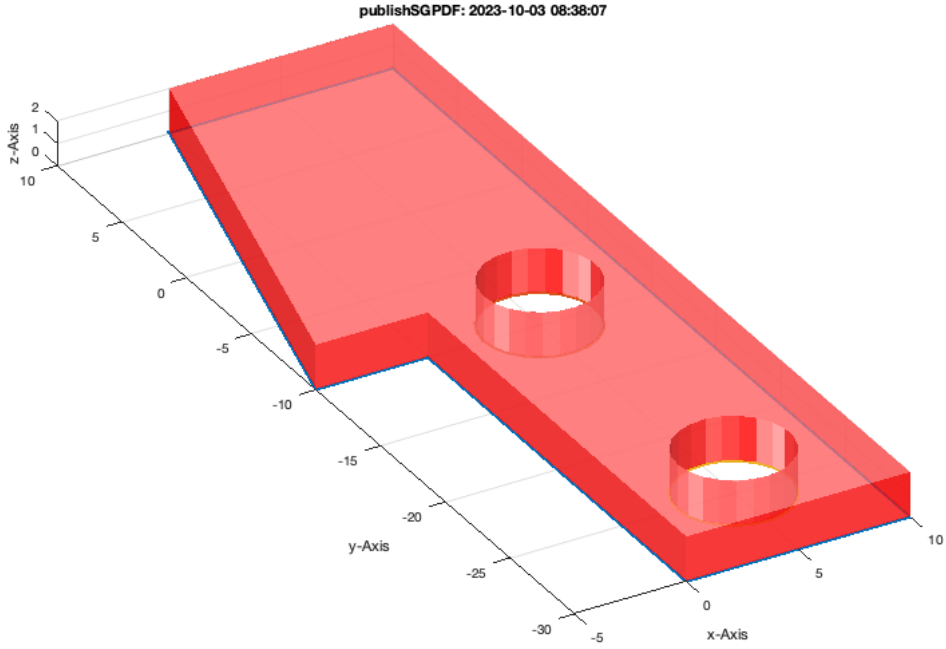

```
SGofCPLcommand('b 10 40, c 5, move 5 -10, b 10 20 5, h 3');
```

```
SGofCPLcommand: SGofCPLcommand("b 10 40, c 5, move 5 -10, b 10 20 5, h 3")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 40, c 5, move 5 -10, b 10 20 5, h 3");');
```



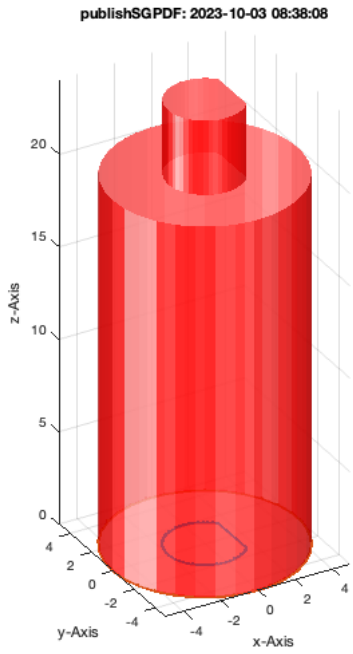

```
SGofCPLcommand('b 10 40, c 5, move 5 -10, b 10 20 5, d 5 5 -25, h 2');
```

```
SGofCPLcommand: SGofCPLcommand("b 10 40, c 5, move 5 -10, b 10 20 5, d 5 5 -25, h 2")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 40, c 5, move 5 -10, b 10 20 5, d 5 5 -25, h 2");');
```



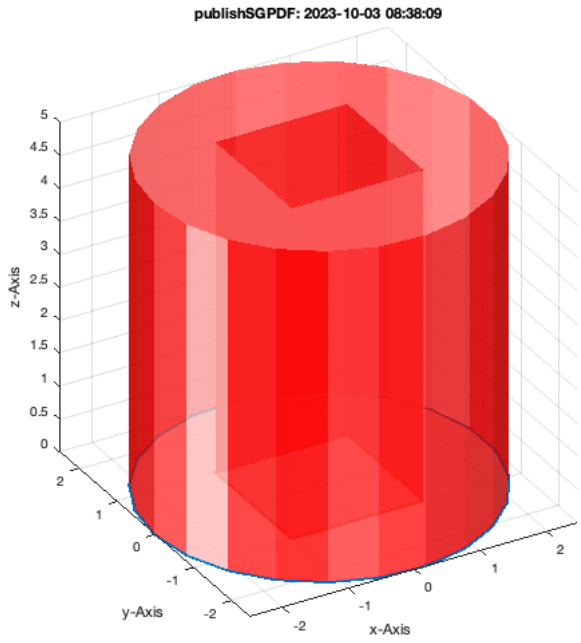

```
SGofCPLcommand('ms 4 3,h 4, c 10, h 20'); % Motorshaft
```

```
SGofCPLcommand: SGofCPLcommand("ms 4 3,h 4, c 10, h 20")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("ms 4 3,h 4, c 10, h 20");');
```



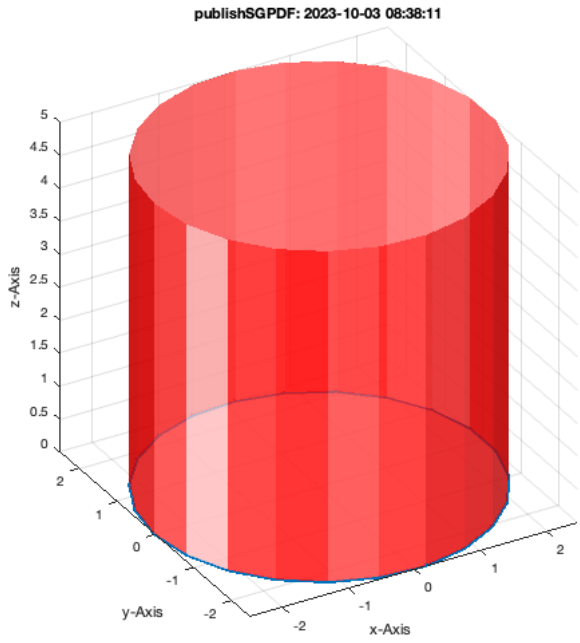

```
SGofCPLcommand('c 5, b 2 2, h 5');
```

```
SGofCPLcommand: SGofCPLcommand("c 5, b 2 2, h 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5, b 2 2, h 5");');
```



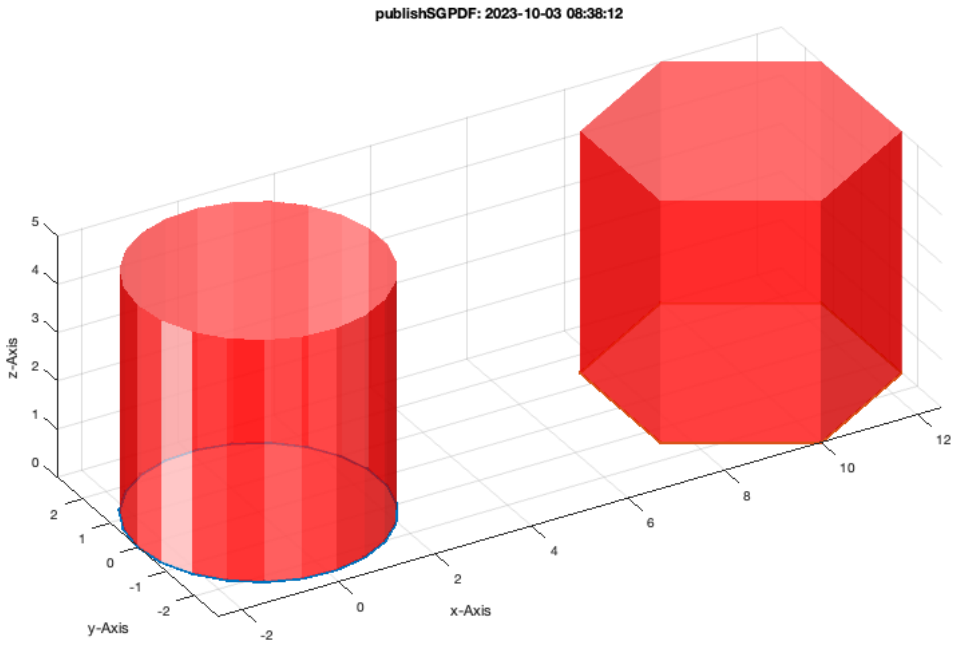

```
SGofCPLcommand('d 5 0 0, h 5');               % Drilling hole at 0 0 with enough edges
```

```
SGofCPLcommand: SGofCPLcommand("d 5 0 0, h 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("d 5 0 0, h 5");');
```



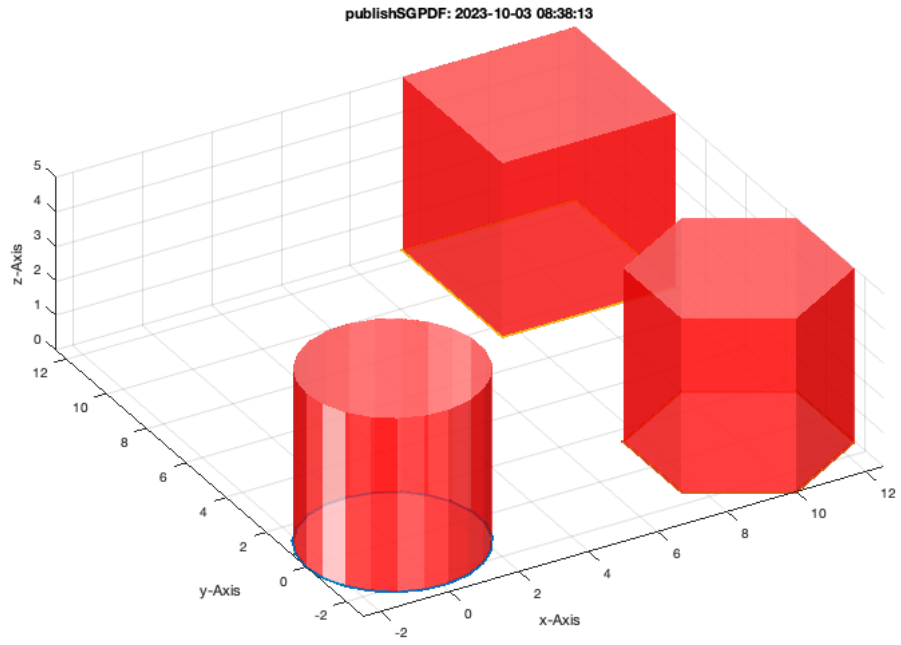

```
SGofCPLcommand('d 5 0 0, d 5 10 0 6, h 5');    % Drilling hole at 10 0 with 6 edges
```

```
SGofCPLcommand: SGofCPLcommand("d 5 0 0, d 5 10 0 6, h 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("d 5 0 0, d 5 10 0 6, h 5");');
```



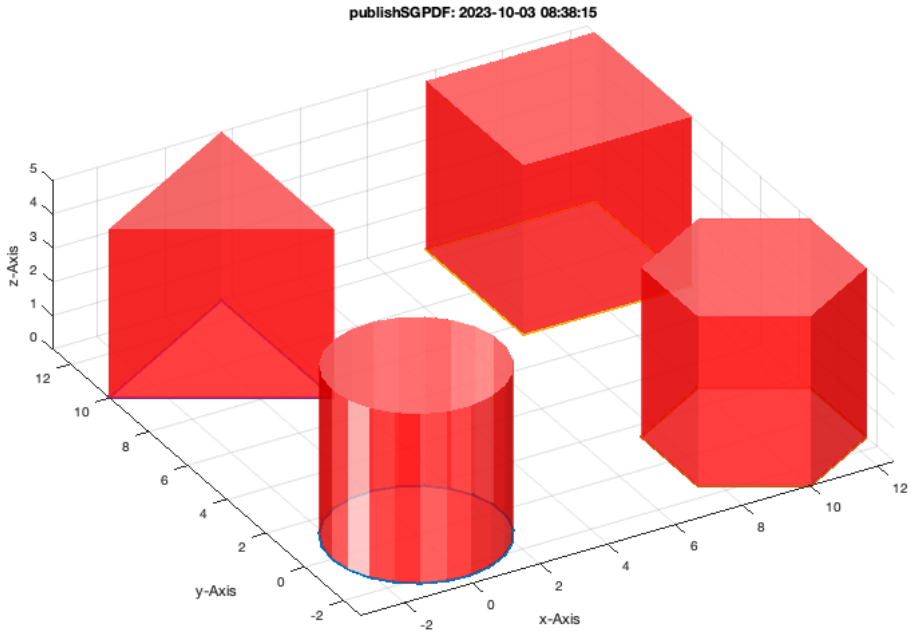

```
SGofCPLcommand('d 5 0 0, d 5 10 0 6, d 5 10 10 4, h 5'); % Drilling hole at 10 10 with 4 edges
```

```
SGofCPLcommand: SGofCPLcommand("d 5 0 0, d 5 10 0 6, d 5 10 10 4, h 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("d 5 0 0, d 5 10 0 6, d 5 10 10 4, h 5");');
```



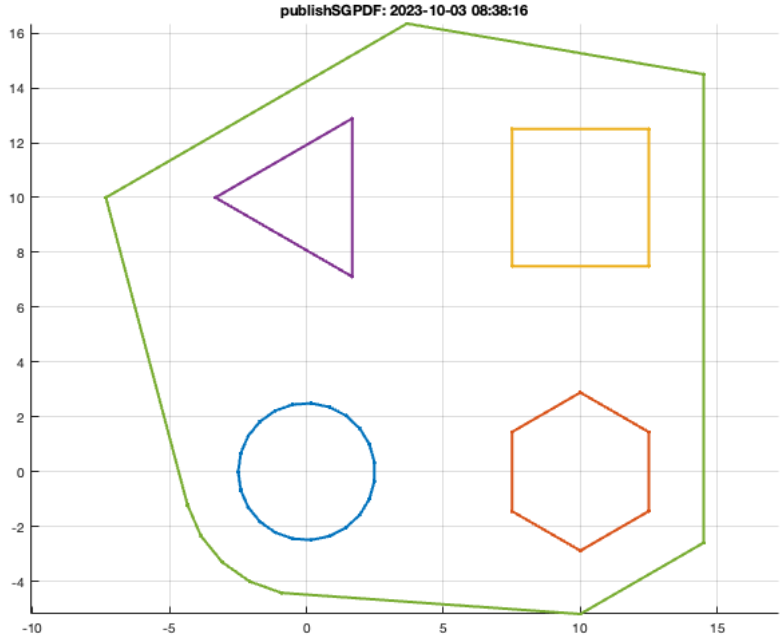

```
SGofCPLcommand('d 5 0 0, d 5 10 0 6, d 5 10 10 4, d 5 0 10 3, h 5');
```

```
SGofCPLcommand: SGofCPLcommand("d 5 0 0, d 5 10 0 6, d 5 10 10 4, d 5 0 10 3, h 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("d 5 0 0, d 5 10 0 6, d 5 10 10 4, d 5 0 10 3, h 5");');
```



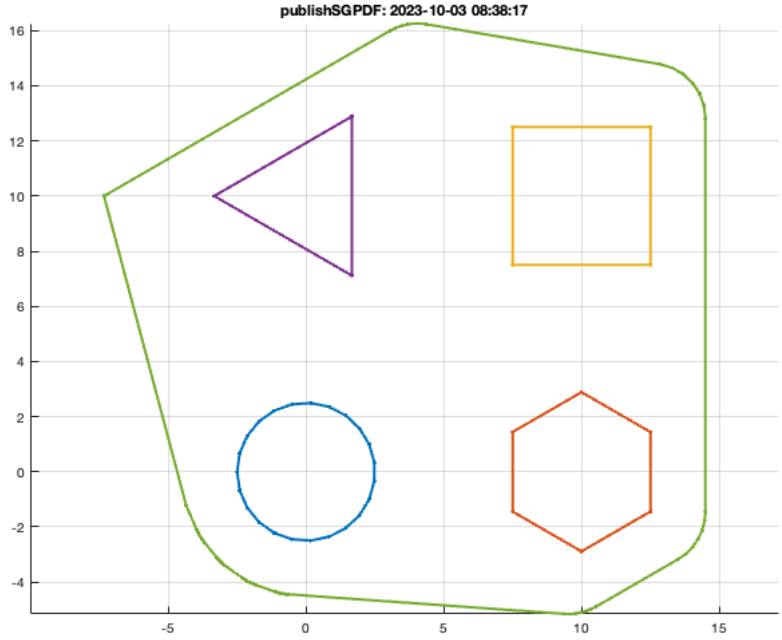

```
SGofCPLcommand('d 5 0 0, d 5 10 0 6, d 5 10 10 4, d 5 0 10 3, ch 2'); % Convex hull around shape
```

```
SGofCPLcommand: SGofCPLcommand("d 5 0 0, d 5 10 0 6, d 5 10 10 4, d 5 0 10 3, ch 2")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("d 5 0 0, d 5 10 0 6, d 5 10 10 4, d 5 0 10 3, ch 2");');
```



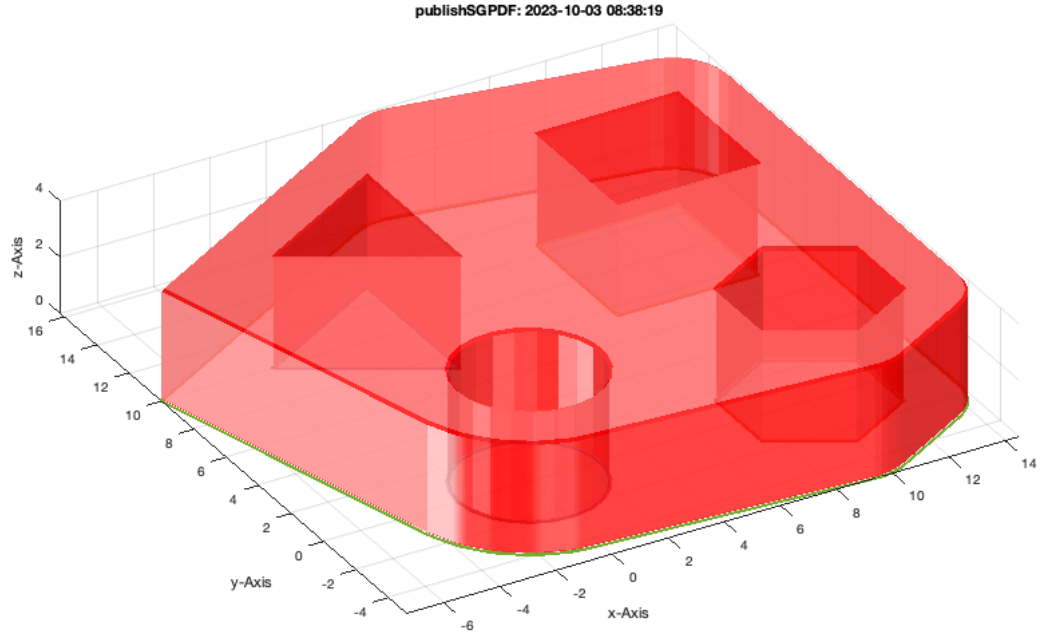

```
SGofCPLcommand('d 5 0 0, d 5 10 0 6, d 5 10 10 4, d 5 0 10 3, ch 2 2'); % Convex hull around shape with radial corners
```

```
SGofCPLcommand: SGofCPLcommand("d 5 0 0, d 5 10 0 6, d 5 10 10 4, d 5 0 10 3, ch 2 2")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("d 5 0 0, d 5 10 0 6, d 5 10 10 4, d 5 0 10 3, ch 2 2");');
```




```
SGofCPLcommand('d 5 0 0, d 5 10 0 6, d 5 10 10 4, d 5 0 10 3, ch 2 2, hc 4'); % Convex hull extruded
```

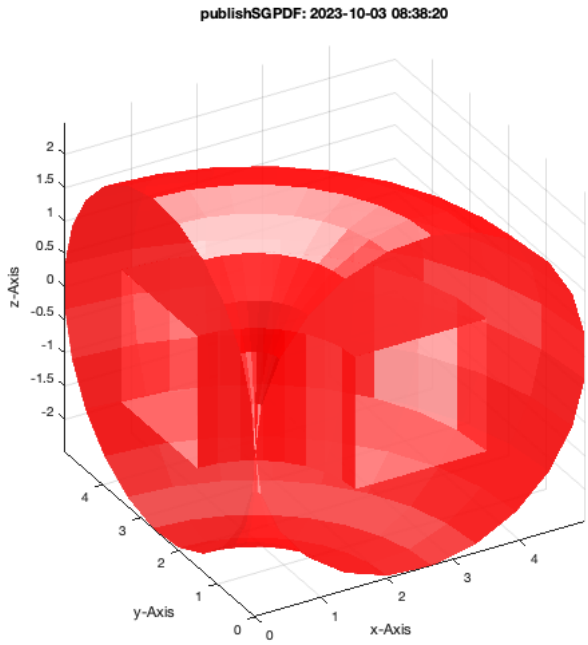
```
SGofCPLcommand: SGofCPLcommand("d 5 0 0, d 5 10 0 6, d 5 10 10 4, d 5 0 10 3, ch 2 2, hc 4")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("d 5 0 0, d 5 10 0 6, d 5 10 10 4, d 5 0 10 3, ch 2 2, hc 4");');
```



1.2. First use of 2.5 D design using simple contour by rotation

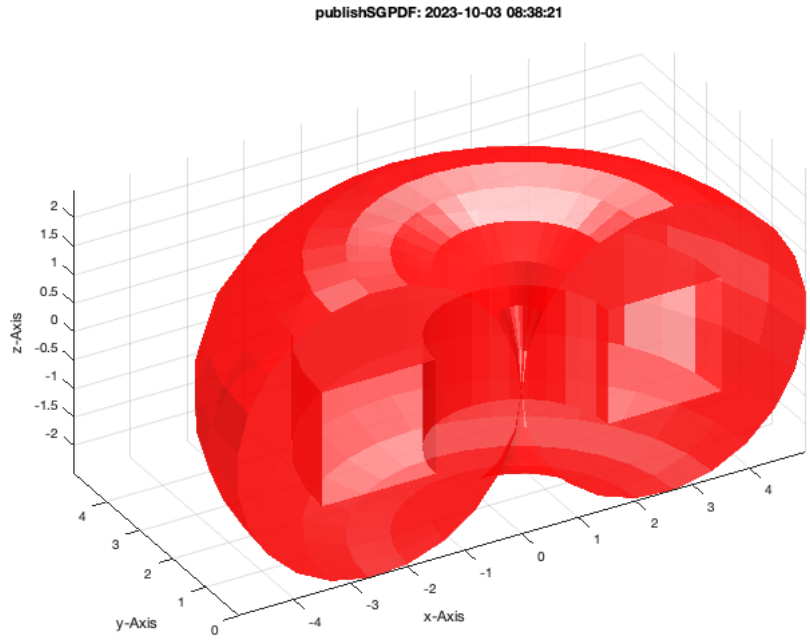
```
SGofCPLcommand('c 5, b 2 2, r 90');
```

```
SGofCPLcommand: SGofCPLcommand("c 5, b 2 2, r 90")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5, b 2 2, r 90");');
```



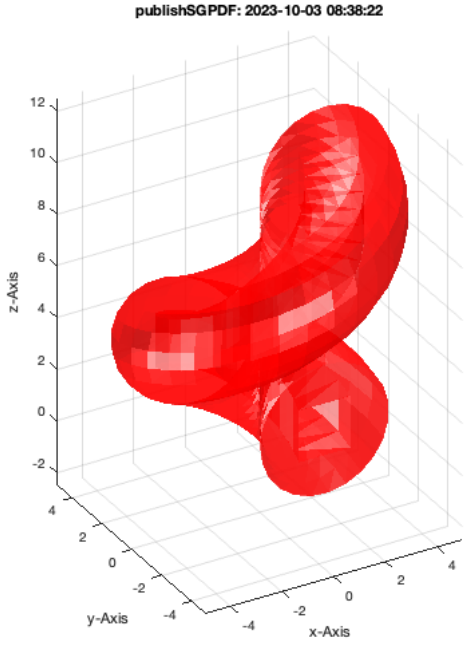

```
SGofCPLcommand('c 5, b 2 2, r 180');
```

```
SGofCPLcommand: SGofCPLcommand("c 5, b 2 2, r 180")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5, b 2 2, r 180");');
```



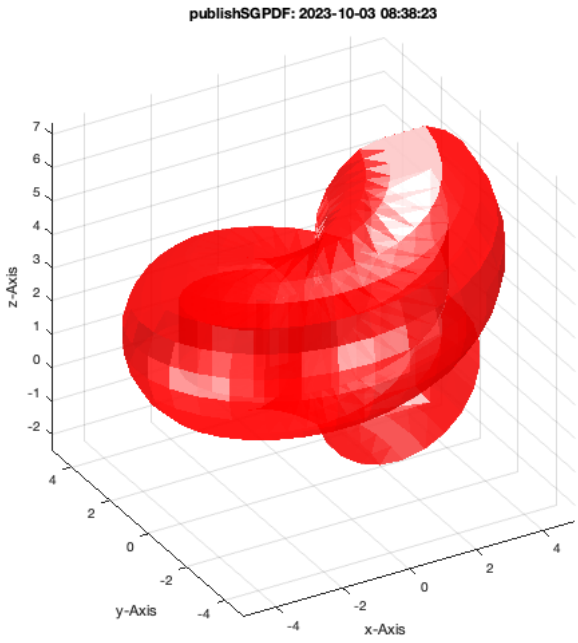

```
SGofCPLcommand('c 5, b 2 2, r 360 10');
```

```
SGofCPLcommand: SGofCPLcommand("c 5, b 2 2, r 360 10")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5, b 2 2, r 360 10");');
```



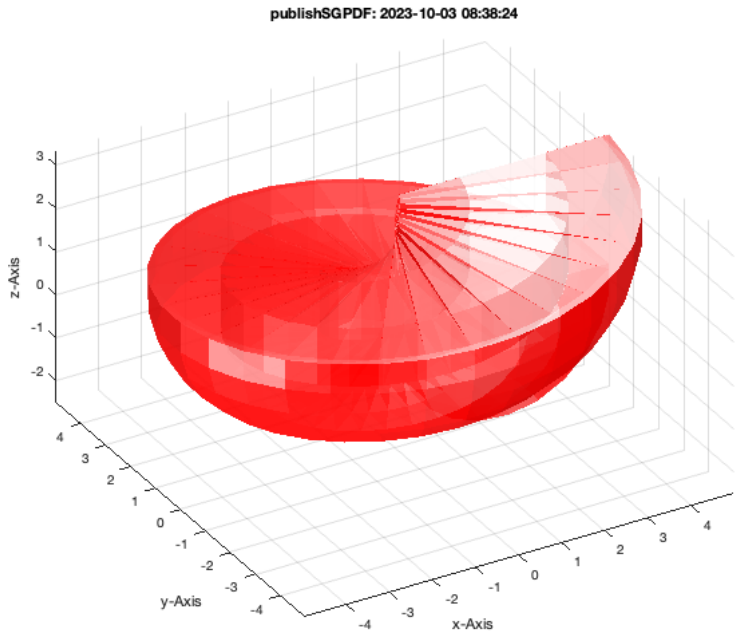

```
SGofCPLcommand('c 5, b 2 2, r 360 5');
```

```
SGofCPLcommand: SGofCPLcommand("c 5, b 2 2, r 360 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5, b 2 2, r 360 5");');
```



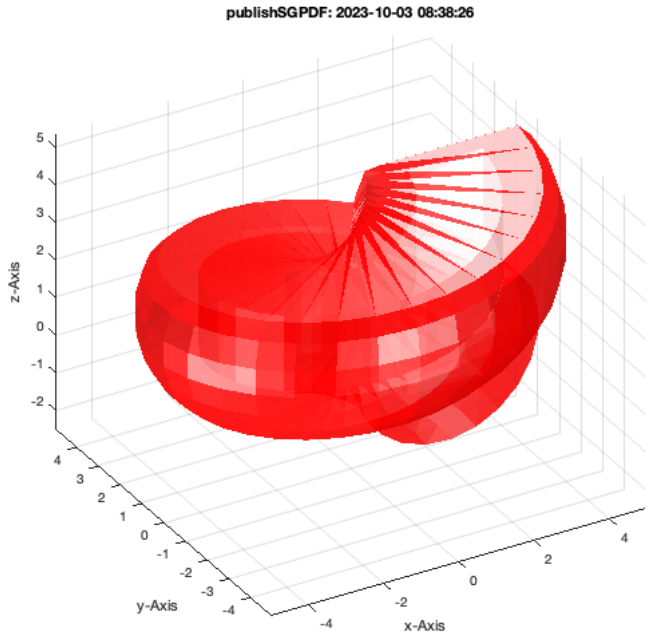

```
SGofCPLcommand('c 5, b 2 2, r 360 3');
```

```
SGofCPLcommand: SGofCPLcommand("c 5, b 2 2, r 360 3")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5, b 2 2, r 360 3");');
```




```
SGofCPLcommand('c 5, b 2 2, r 360 4');
```

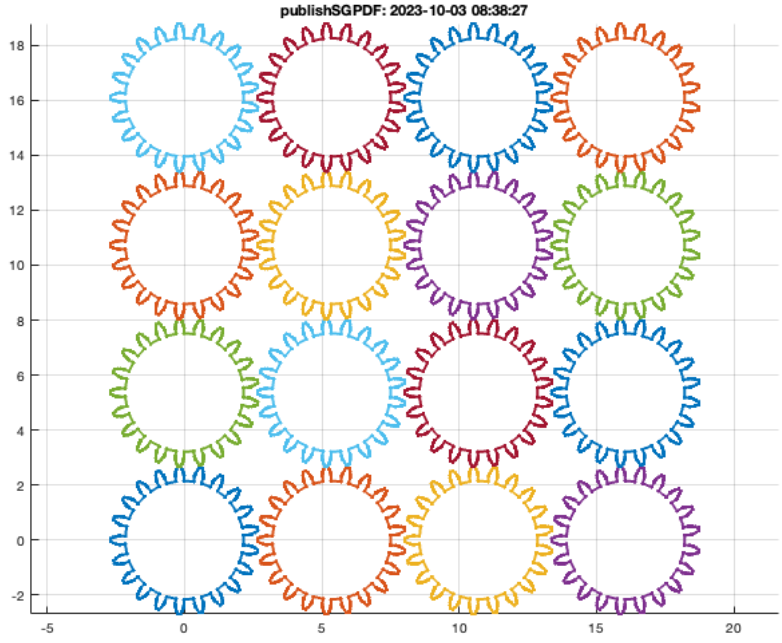
```
SGofCPLcommand: SGofCPLcommand("c 5, b 2 2, r 360 4")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5, b 2 2, r 360 4");');
```



1.3. Contour duplication cartesion and rotation based

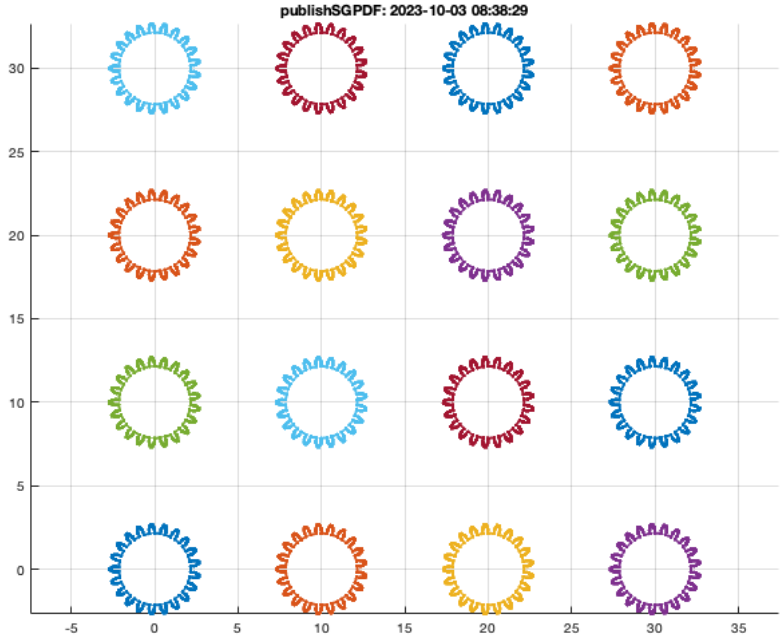
```
SGofCPLcommand('g 5 21, dupc 4 4');
```

```
SGofCPLcommand: SGofCPLcommand("g 5 21, dupc 4 4")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("g 5 21, dupc 4 4");');
```



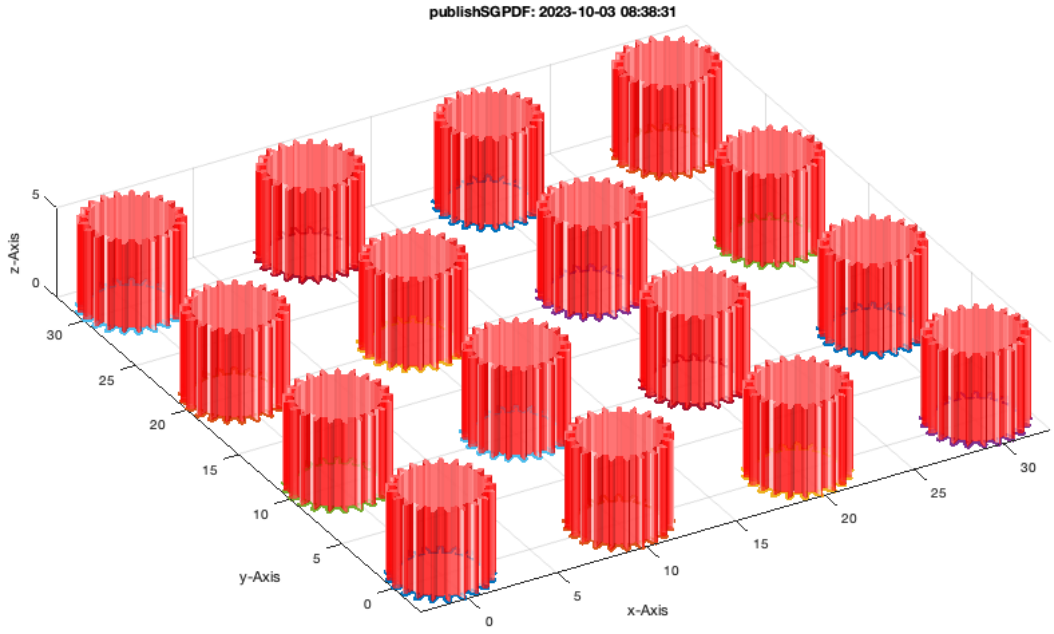

```
SGofCPLcommand('g 5 21, dupc 4 4 10');
```

```
SGofCPLcommand: SGofCPLcommand("g 5 21, dupc 4 4 10")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("g 5 21, dupc 4 4 10");');
```



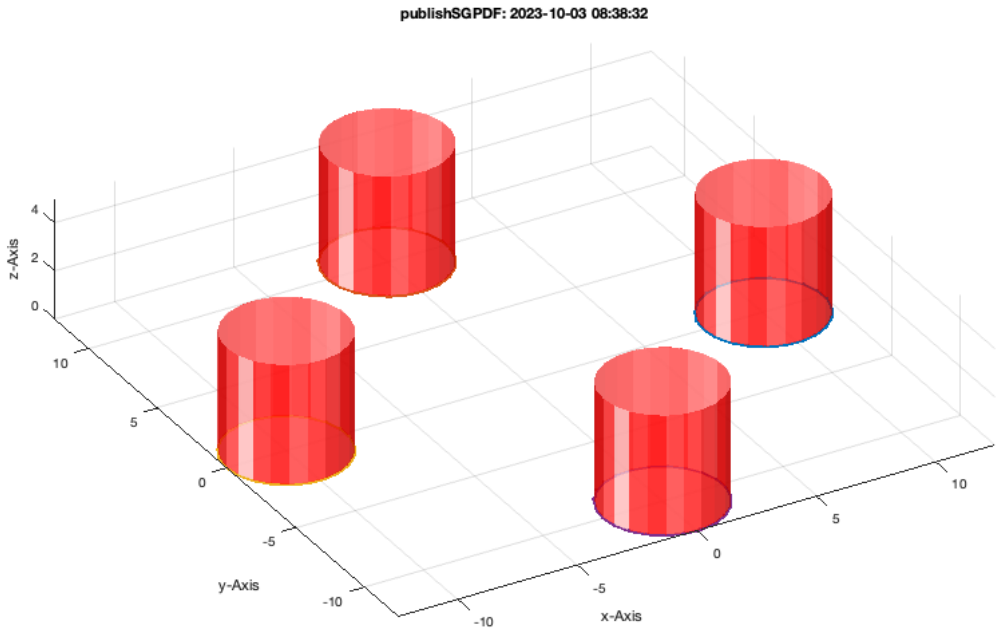

```
SGofCPLcommand('g 5 21, dupc 4 4 10, h 5');
```

```
SGofCPLcommand: SGofCPLcommand("g 5 21, dupc 4 4 10, h 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("g 5 21, dupc 4 4 10, h 5");');
```



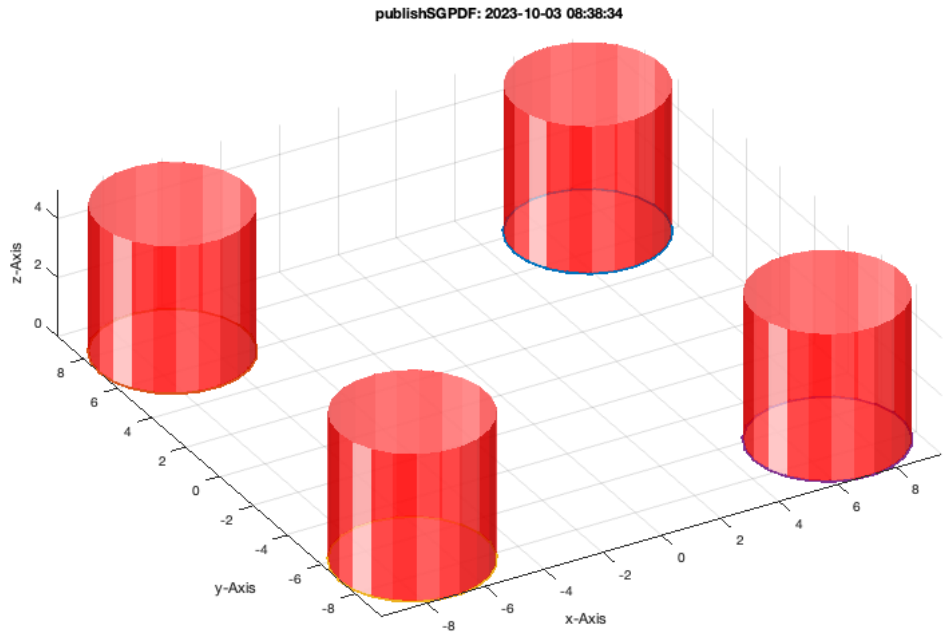

```
SGofCPLcommand('c 5, dupr 10 4, h 5');
```

```
SGofCPLcommand: SGofCPLcommand("c 5, dupr 10 4, h 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5, dupr 10 4, h 5");');
```




```
SGofCPLcommand('c 5,dupr 10 4 45, h 5');
```

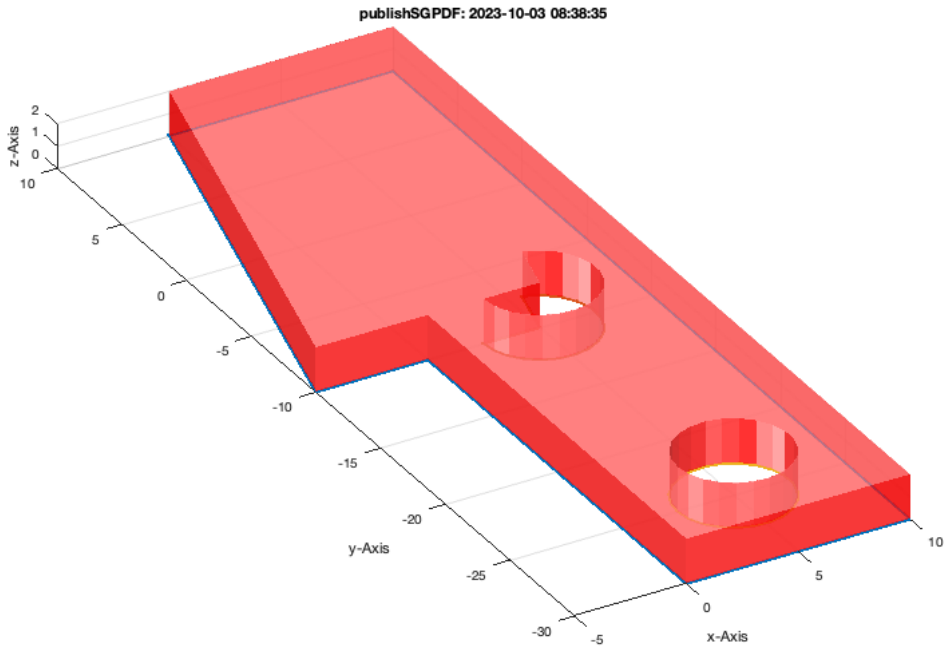
```
SGofCPLcommand: SGofCPLcommand("c 5,dupr 10 4 45, h 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 5,dupr 10 4 45, h 5");');
```



1.4. Contour Stack and boolean design use of 2.5 D Solids

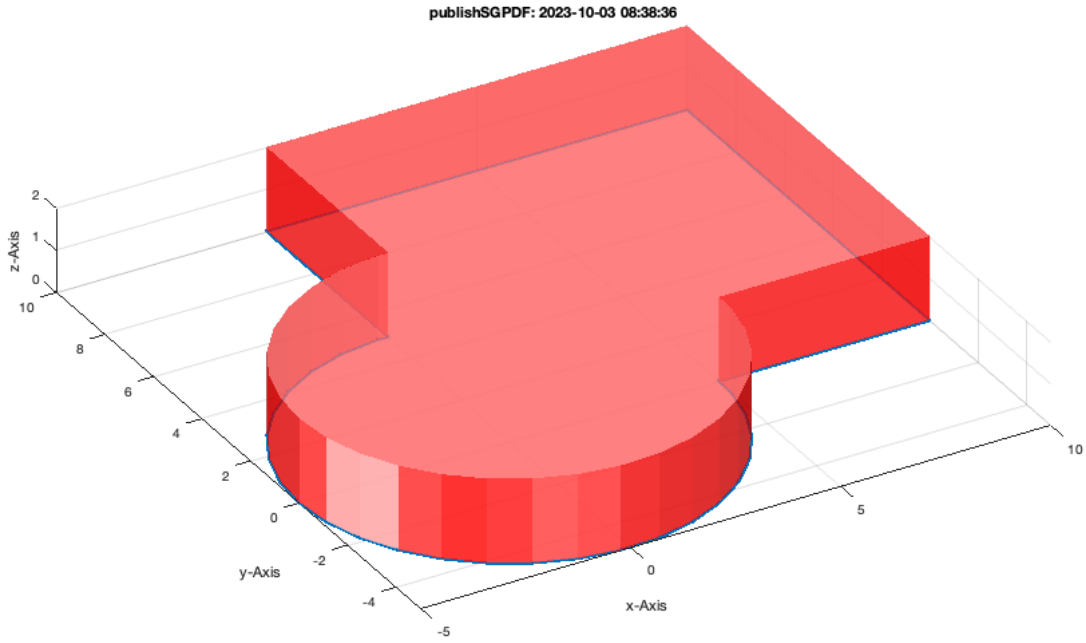
```
SGofCPLcommand('b 10 40, c 5, move 5 -10, enter, b 10 20 5, add, d 5 5 -25, h 2');
```

```
SGofCPLcommand: SGofCPLcommand("b 10 40, c 5, move 5 -10, enter, b 10 20 5, add, d 5 5 -25, h 2")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 40, c 5, move 5 -10, enter, b 10 20 5, add, d 5 5 -25, h 2");');
```



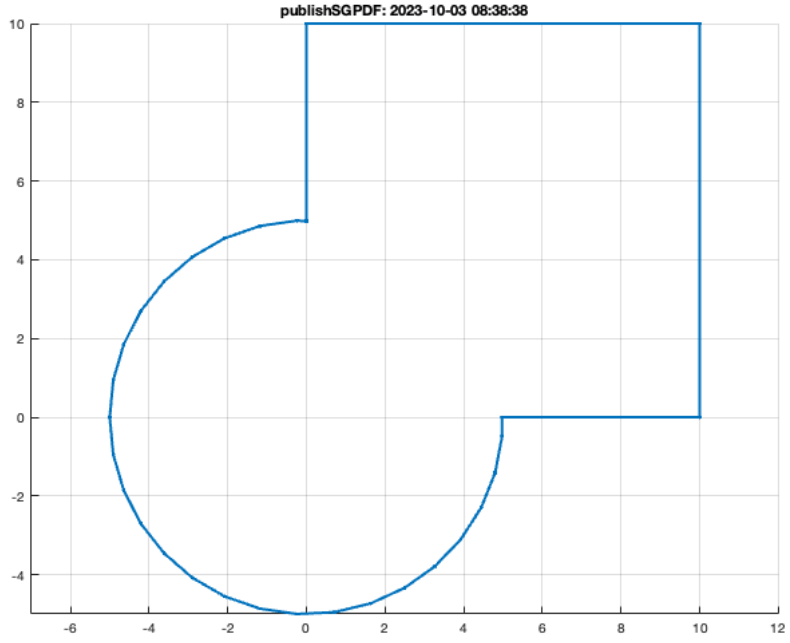

```
SGofCPLcommand('b 10 10, move 5 5, c 10, h 2');
```

```
SGofCPLcommand: SGofCPLcommand("b 10 10, move 5 5, c 10, h 2")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 10, move 5 5, c 10, h 2");');
```



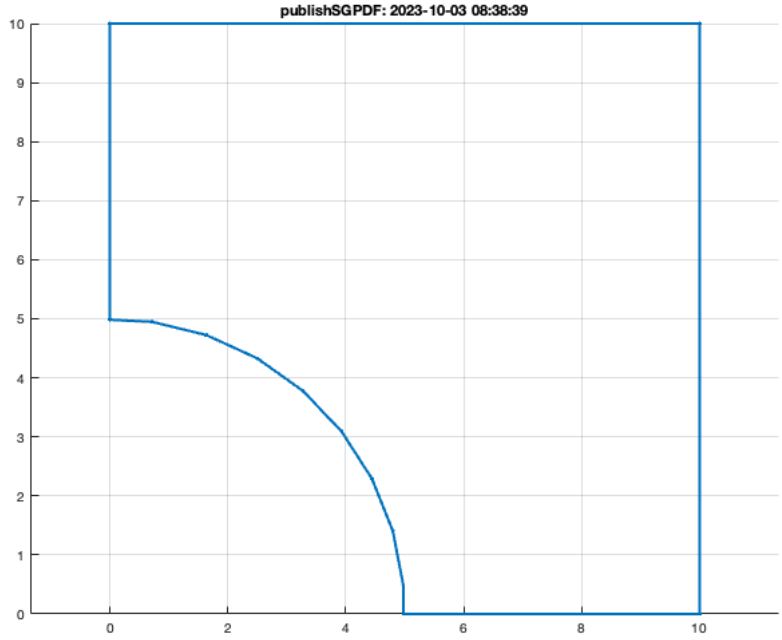

```
SGofCPLcommand('b 10 10, move 5 5, enter, c 10, add');
```

```
SGofCPLcommand: SGofCPLcommand("b 10 10, move 5 5, enter, c 10, add")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 10, move 5 5, enter, c 10, add");');
```



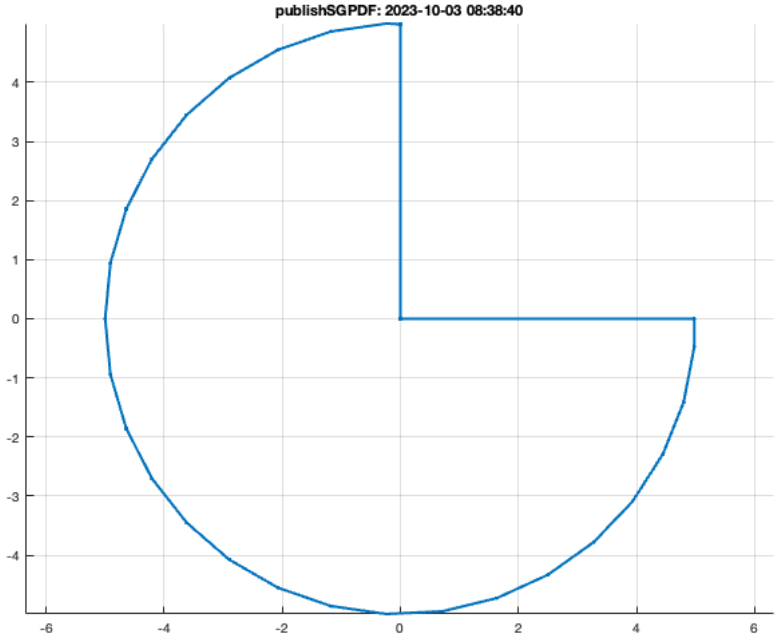

```
SGofCPLcommand('b 10 10, move 5 5, enter, c 10, sub');
```

```
SGofCPLcommand: SGofCPLcommand("b 10 10, move 5 5, enter, c 10, sub")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 10, move 5 5, enter, c 10, sub");');
```



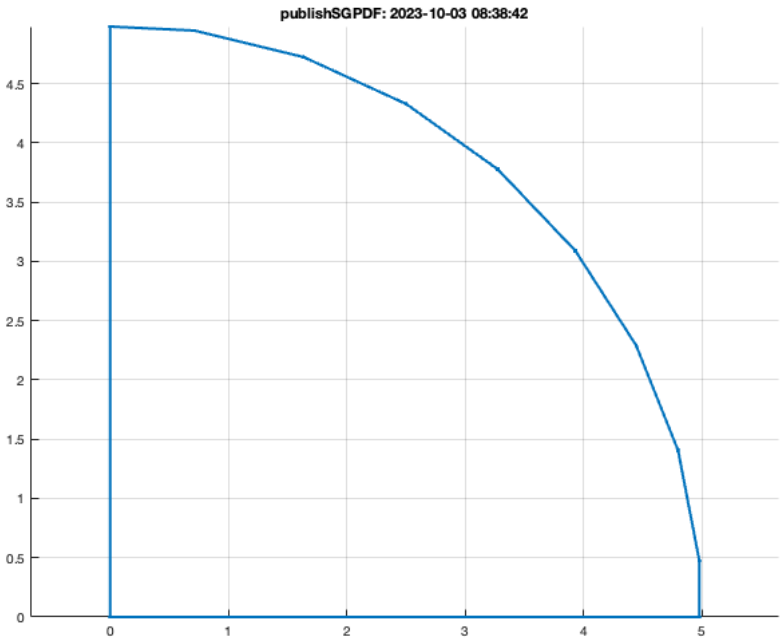

```
SGofCPLcommand('b 10 10, move 5 5, enter, c 10, rem');
```

```
SGofCPLcommand: SGofCPLcommand("b 10 10, move 5 5, enter, c 10, rem")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 10, move 5 5, enter, c 10, rem");');
```



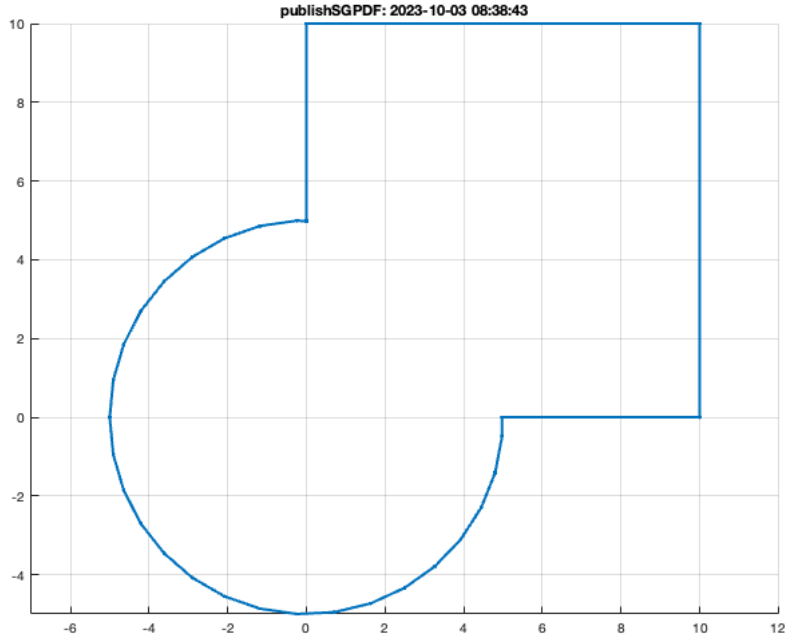

```
SGofCPLcommand('b 10 10, move 5 5, enter, c 10, isec');
```

```
SGofCPLcommand: SGofCPLcommand("b 10 10, move 5 5, enter, c 10, isec")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 10, move 5 5, enter, c 10, isec");');
```



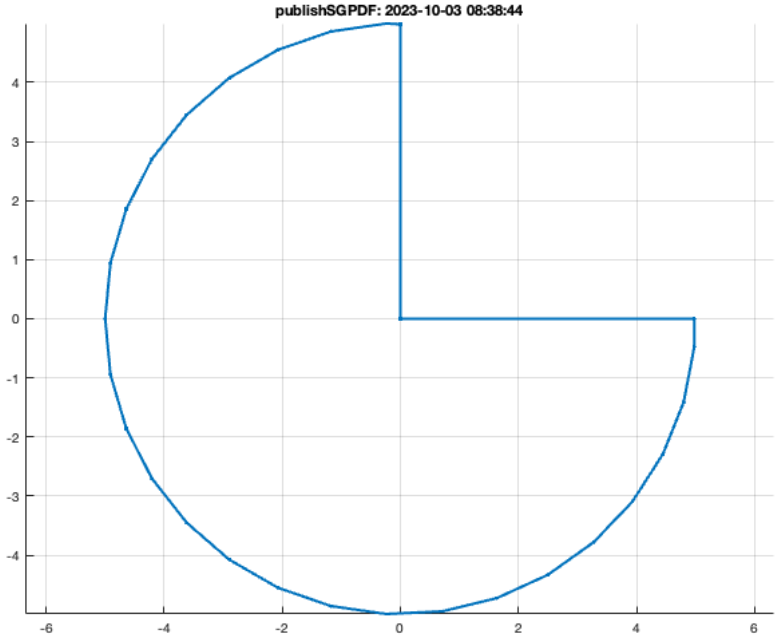

```
SGofCPLcommand('b 10 10, move 5 5, enter, c 10, swap, add');
```

```
SGofCPLcommand: SGofCPLcommand("b 10 10, move 5 5, enter, c 10, swap, add")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 10, move 5 5, enter, c 10, swap, add");');
```



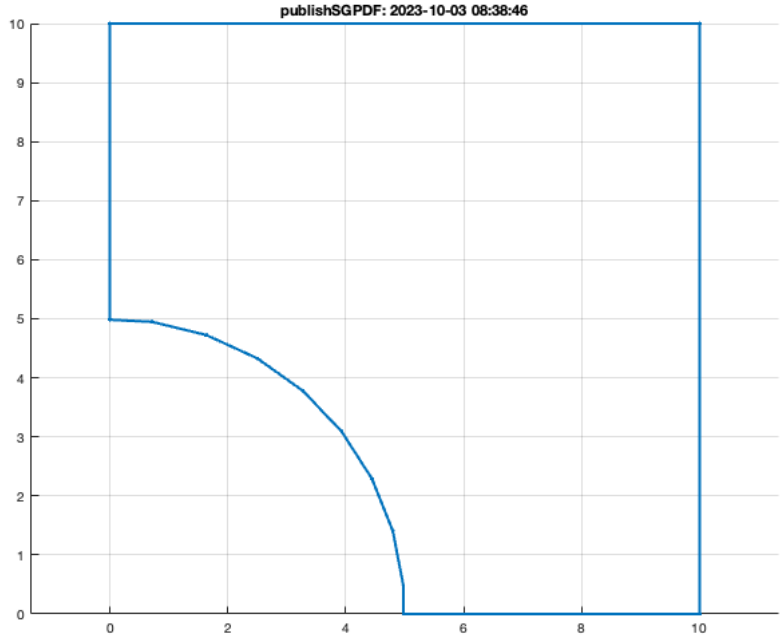

```
SGofCPLcommand('b 10 10, move 5 5, enter, c 10, swap, sub');
```

```
SGofCPLcommand: SGofCPLcommand("b 10 10, move 5 5, enter, c 10, swap, sub")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 10, move 5 5, enter, c 10, swap, sub");');
```



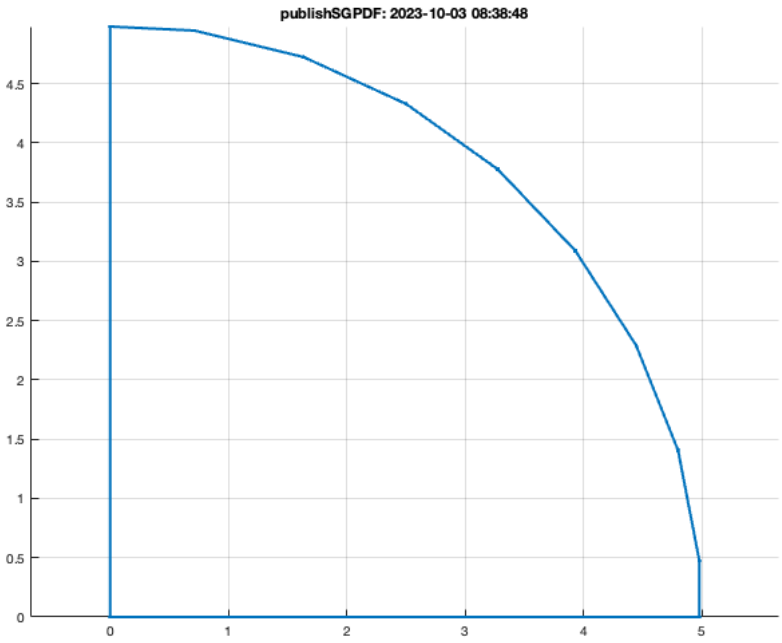

```
SGofCPLcommand('b 10 10, move 5 5, enter, c 10, swap, rem');
```

```
SGofCPLcommand: SGofCPLcommand("b 10 10, move 5 5, enter, c 10, swap, rem")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 10, move 5 5, enter, c 10, swap, rem");');
```



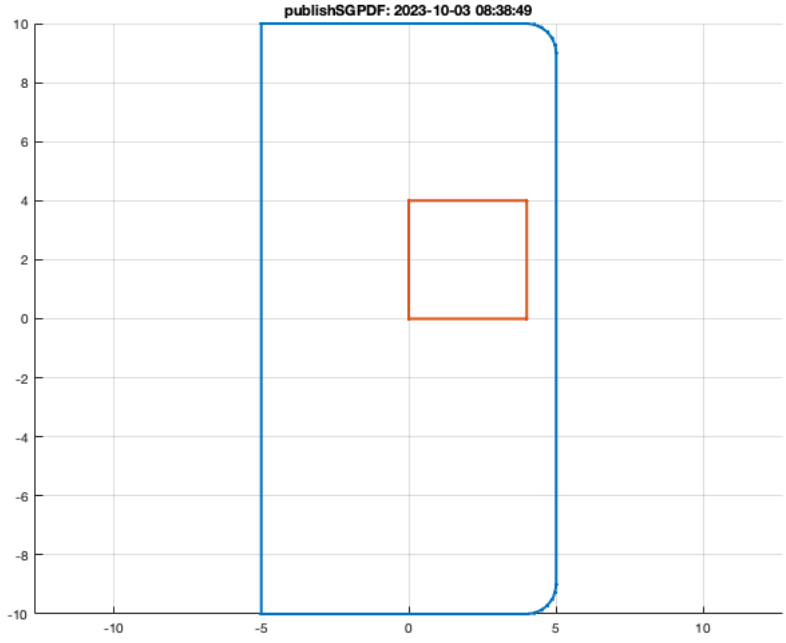

```
SGofCPLcommand('b 10 10, move 5 5, enter, c 10, swap, isec');
```

```
SGofCPLcommand: SGofCPLcommand("b 10 10, move 5 5, enter, c 10, swap, isec")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 10, move 5 5, enter, c 10, swap, isec");');
```



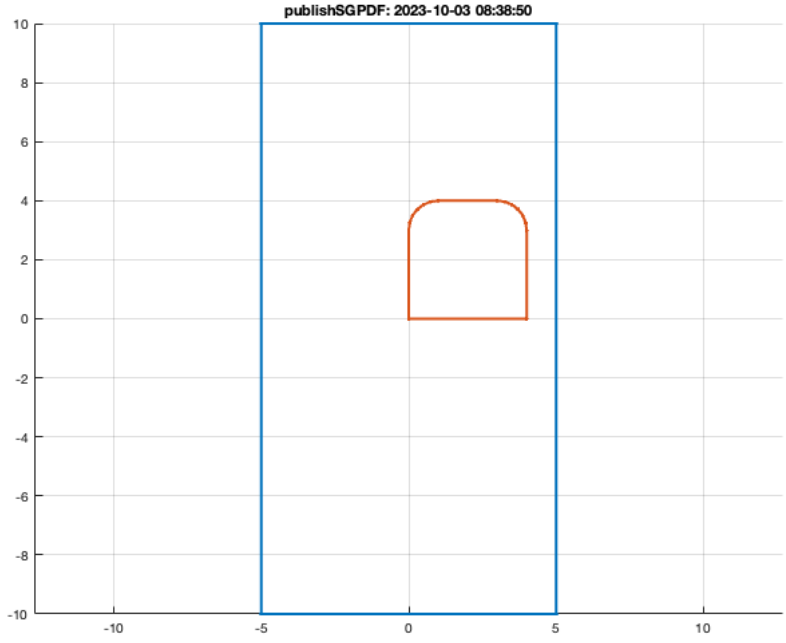

```
SGofCPLcommand('b 10 20, rad 1, d 4 2 2 4')
```

```
SGofCPLcommand: SGofCPLcommand("b 10 20, rad 1, d 4 2 2 4")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 20, rad 1, d 4 2 2 4");');
ans =
     []
```




```
SGofCPLcommand('b 10 20, enter, d 4 2 2 4, rad 1, sub')
```

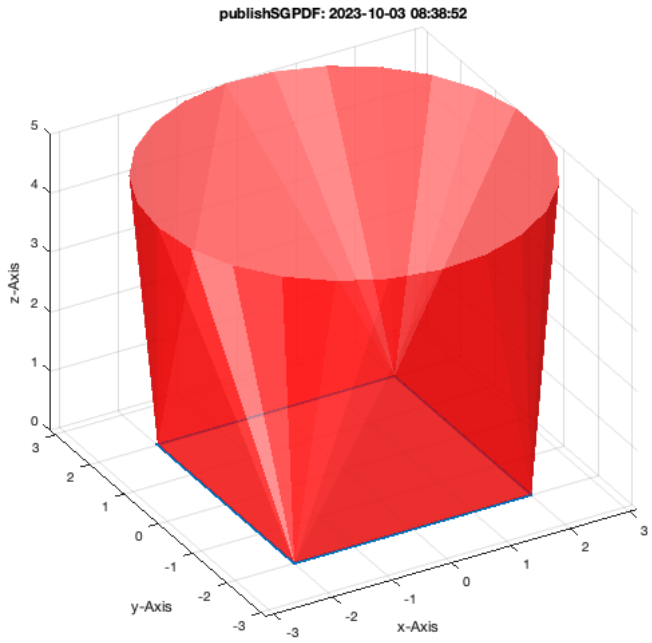
```
SGofCPLcommand: SGofCPLcommand("b 10 20, enter, d 4 2 2 4, rad 1, sub")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 10 20, enter, d 4 2 2 4, rad 1, sub");');
ans =
     []
```



1.5 Contour stack and connection two CPL on stack

```
SGofCPLcommand('c 6.3, enter, b 4 4, hs 5')
```

```
SGofCPLcommand: SGofCPLcommand("c 6.3, enter, b 4 4, hs 5")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 6.3, enter, b 4 4, hs 5");');
ans =
  struct with fields:

    VL: [29×3 double]
    FL: [54×3 double]
```

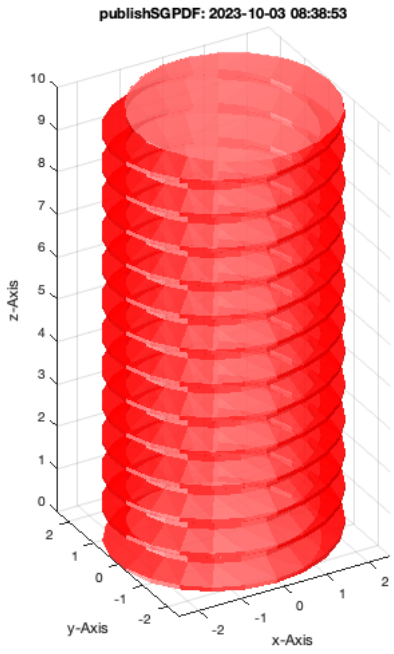


2.1. Solid Geometry Creation Commands Scre, thread tap, nut

2.1.1 Screw

```
SGofCPLcommand('scr 5 10');
```

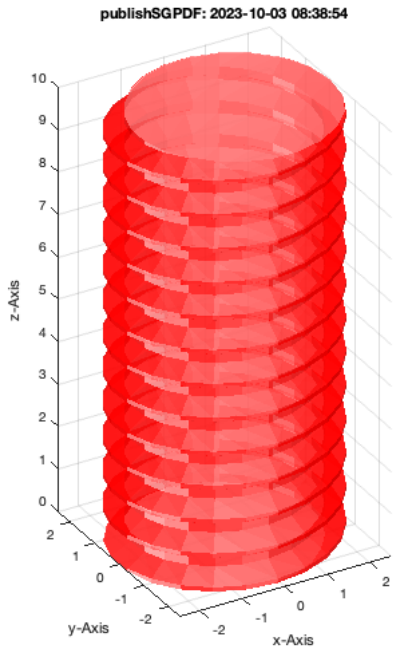
```
SGofCPLcommand: SGofCPLcommand("scr 5 10")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("scr 5 10");');
```



2.1.2 Thread tap

```
SGofCPLcommand('scr -5 10');
```

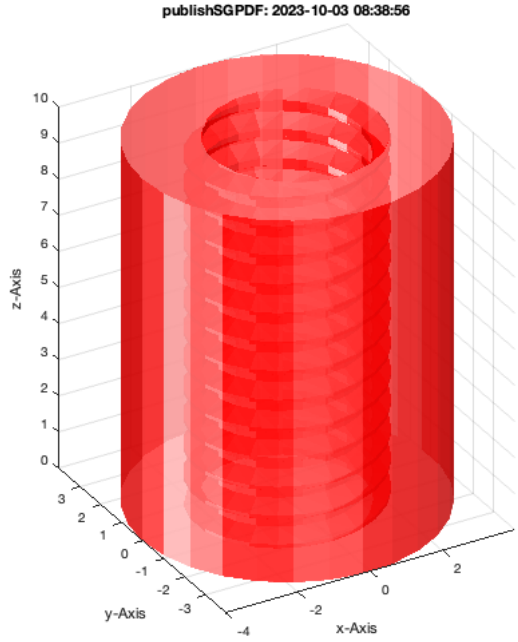
```
SGofCPLcommand: SGofCPLcommand("scr -5 10")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("scr -5 10");');
```



2.1.3 Nut

```
SGofCPLcommand('scr -5 10 8');
```

```
SGofCPLcommand: SGofCPLcommand("scr -5 10 8")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("scr -5 10 8");');
```



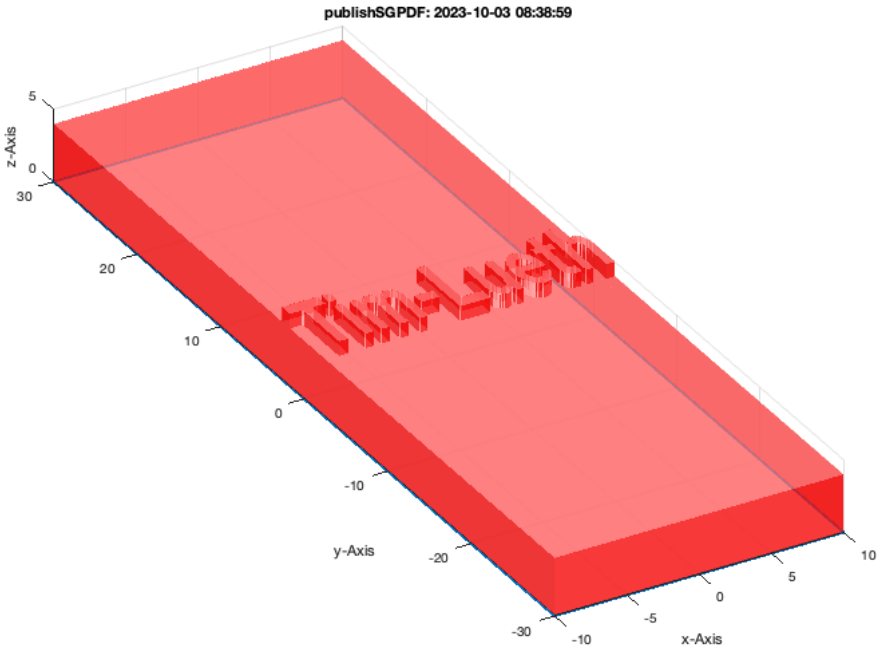
2.2. Solid Geometry Text

Simple text

```
SGofCPLcommand('b 20 60, h 4, enter, text Tim-Lueth 20 2, rel incenter, rel ontop -1, add')
```

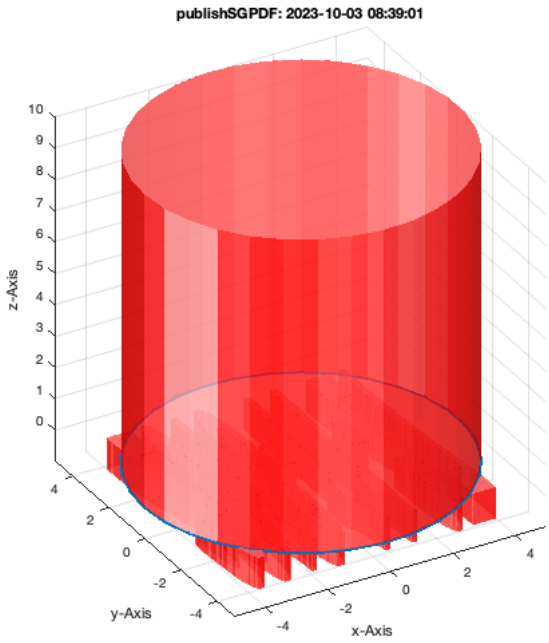
```
SGofCPLcommand: SGofCPLcommand("b 20 60, h 4, enter, text Tim-Lueth 20 2, rel incenter, rel ontop -1, add")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 20 60, h 4, enter, text Tim-Lueth 20 2, rel incenter, rel ontop -1, add");');
ans =
```

```
  struct with fields:

    VL: [1191×3 double]
    FL: [2378×3 double]
    FC: [2378×3 double]
```



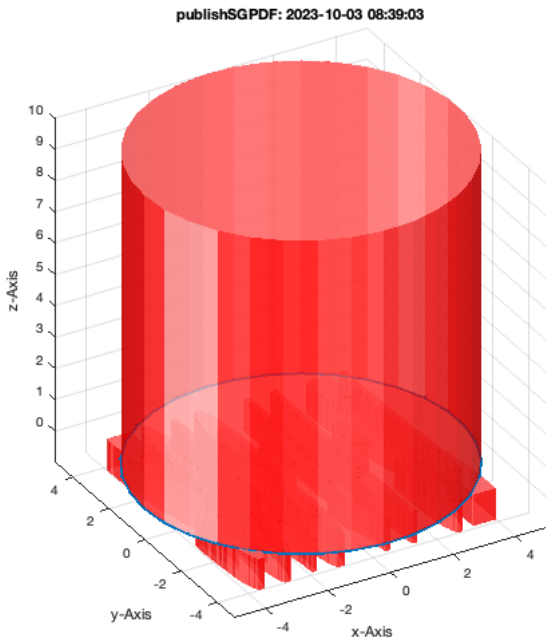

```
SGofCPLcommand('c 10, h 10, textstamp 2018-11-27')
```

```
SGofCPLcommand: SGofCPLcommand("c 10, h 10, textstamp 2018-11-27")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 10, h 10, textstamp 2018-11-27");');
ans =
  struct with fields:

    VL: [1458×3 double]
    FL: [2884×3 double]
    FC: [2884×3 double]
```



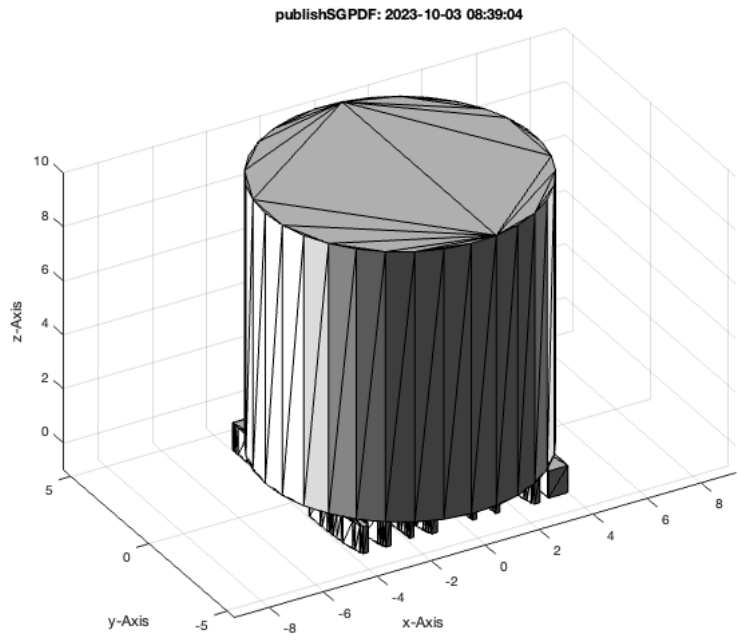

```
SGofCPLcommand('c 10, h 10, textstamp 2018-11-27, save AAAA')

AAAA
```

```
SGofCPLcommand: SGofCPLcommand("c 10, h 10, textstamp 2018-11-27, save AAAA")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 10, h 10, textstamp 2018-11-27, save AAAA");');
ans =
  struct with fields:

    VL: [1458×3 double]
    FL: [2884×3 double]
    FC: [2884×3 double]
AAAA =
  struct with fields:

     SG: [1×1 struct]
    CPL: []
```



```
SGfigure; SGplot(AAAA.SG,'y',0.2); view(-30,30);
CPLplot(AAAA.CPL,'r-',4)
```

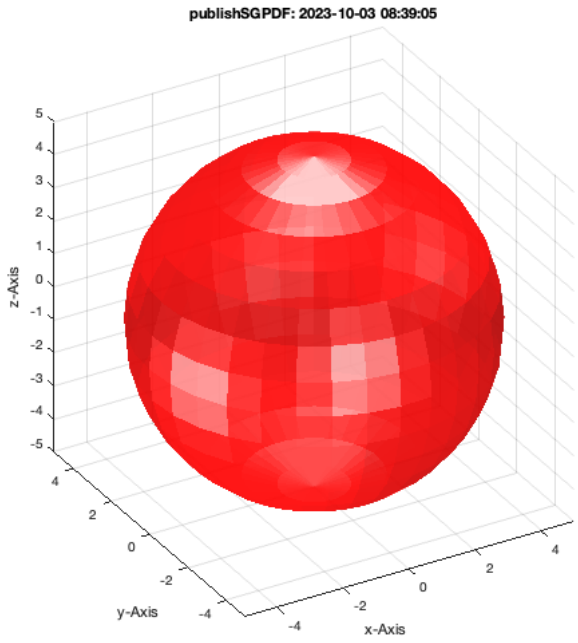
```
ans =
     []
```



2.3. Solid Geometry Spheres

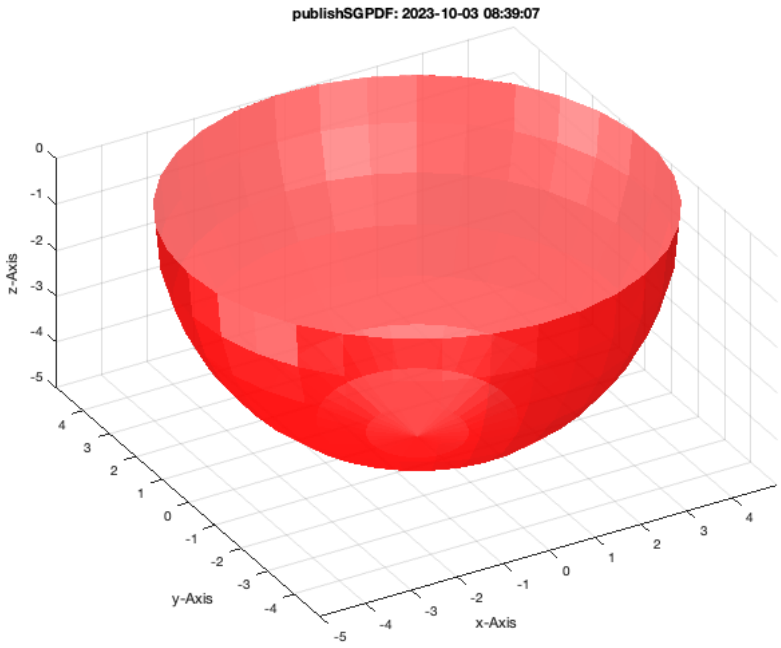
```
SGofCPLcommand('sph 10');
```

```
SGofCPLcommand: SGofCPLcommand("sph 10")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("sph 10");');
```



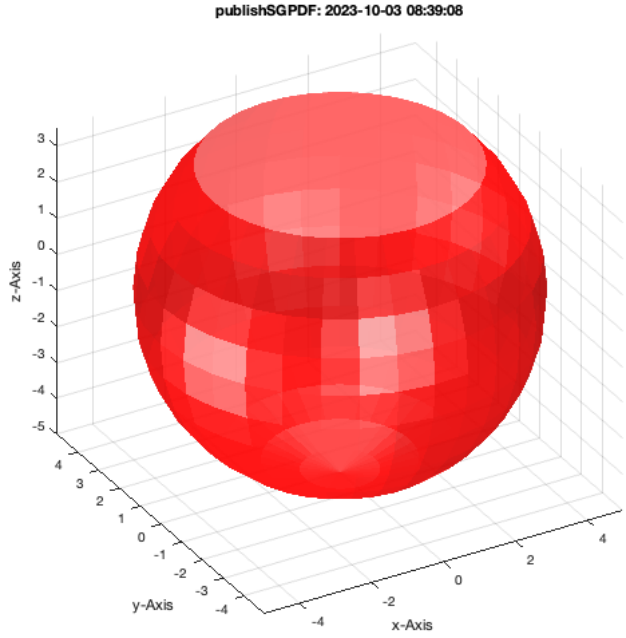

```
SGofCPLcommand('sph 10 0');
```

```
SGofCPLcommand: SGofCPLcommand("sph 10 0")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("sph 10 0");');
```



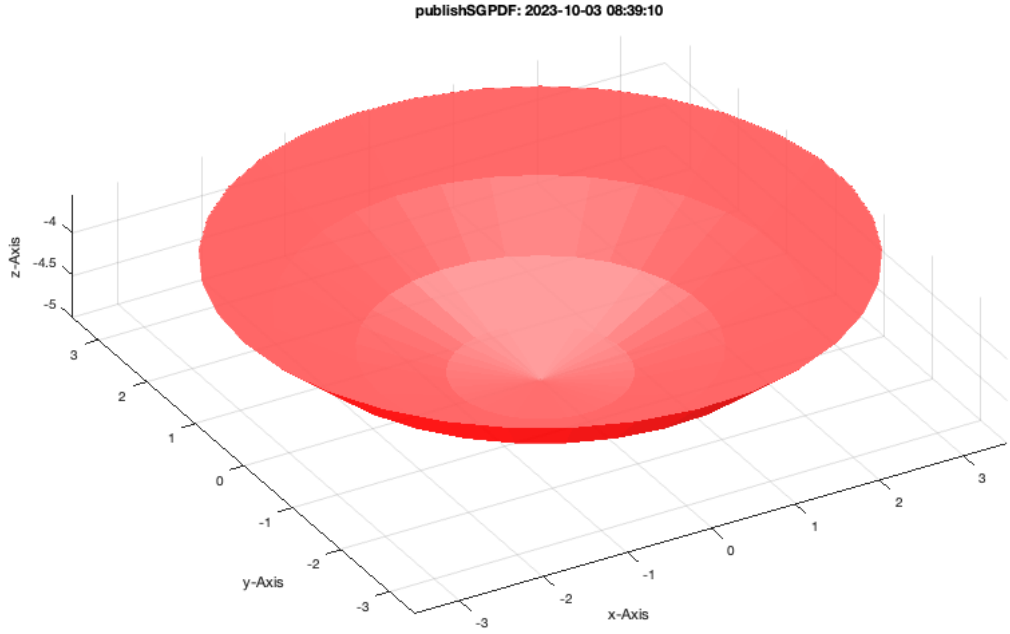

```
SGofCPLcommand('sph 10 45');
```

```
SGofCPLcommand: SGofCPLcommand("sph 10 45")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("sph 10 45");');
```




```
SGofCPLcommand('sph 10 -45');
```

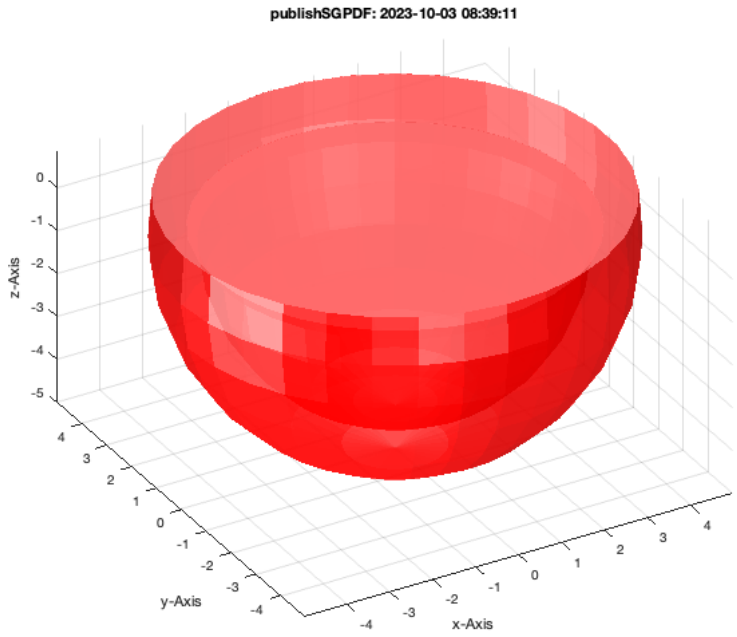
```
SGofCPLcommand: SGofCPLcommand("sph 10 -45")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("sph 10 -45");');
```



3.1 Hollowing and Shell Creation and Solid Separation by Cut

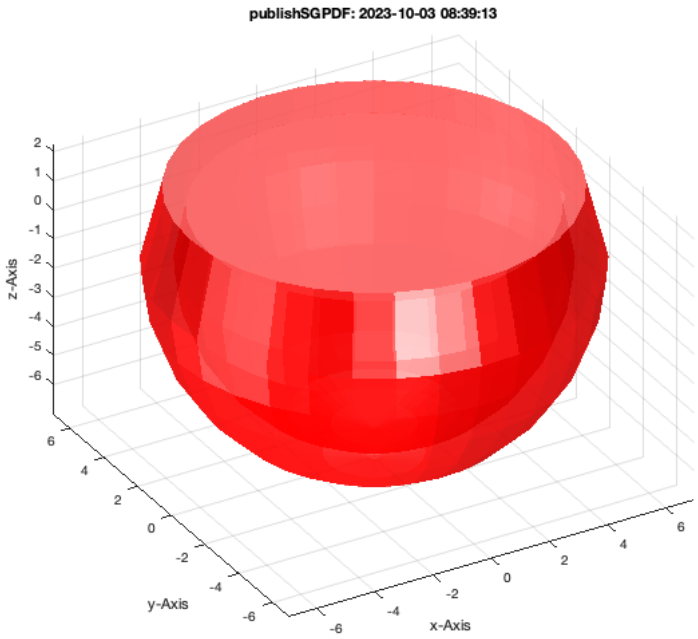
```
SGofCPLcommand('sph 10 +10, hollow -1');
```

```
SGofCPLcommand: SGofCPLcommand("sph 10 +10, hollow -1")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("sph 10 +10, hollow -1");');
```



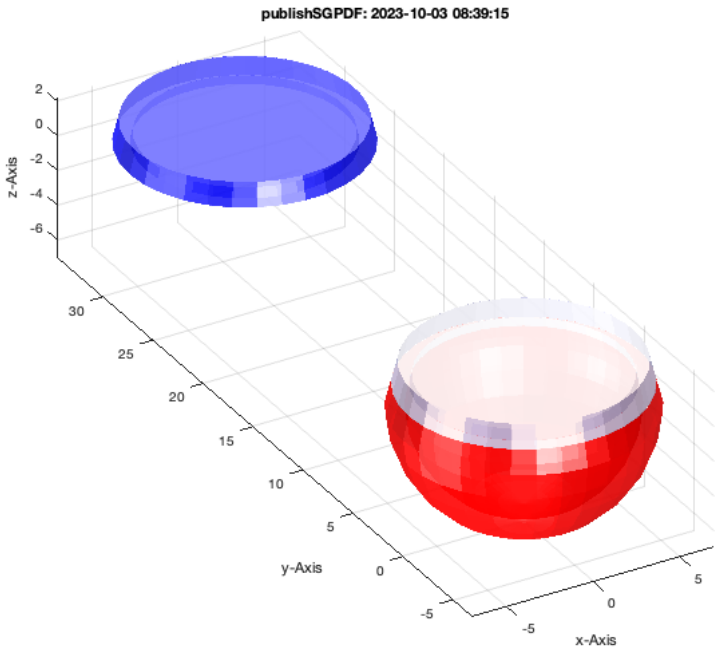

```
SGofCPLcommand('sph 10 +10, shell');
```

```
SGofCPLcommand: SGofCPLcommand("sph 10 +10, shell")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("sph 10 +10, shell");');
```



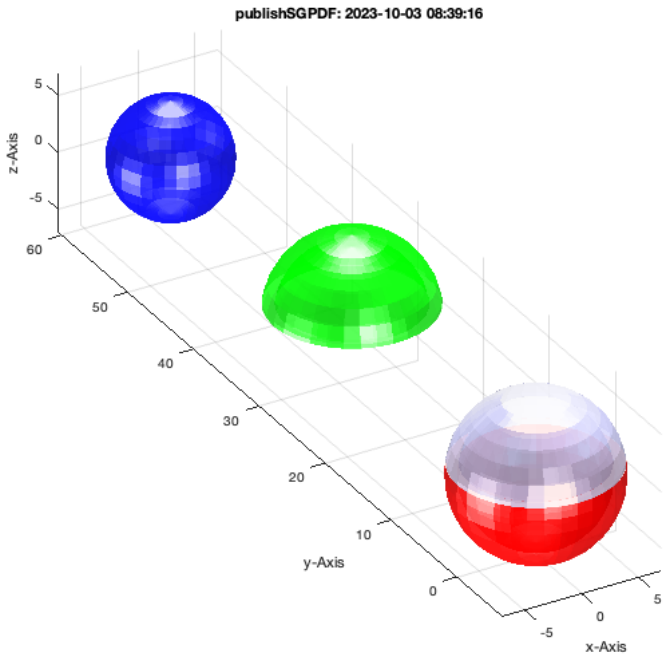

```
SGofCPLcommand('sph 10 +10, shell, cutz 1');
```

```
SGofCPLcommand: SGofCPLcommand("sph 10 +10, shell, cutz 1")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("sph 10 +10, shell, cutz 1");');
```



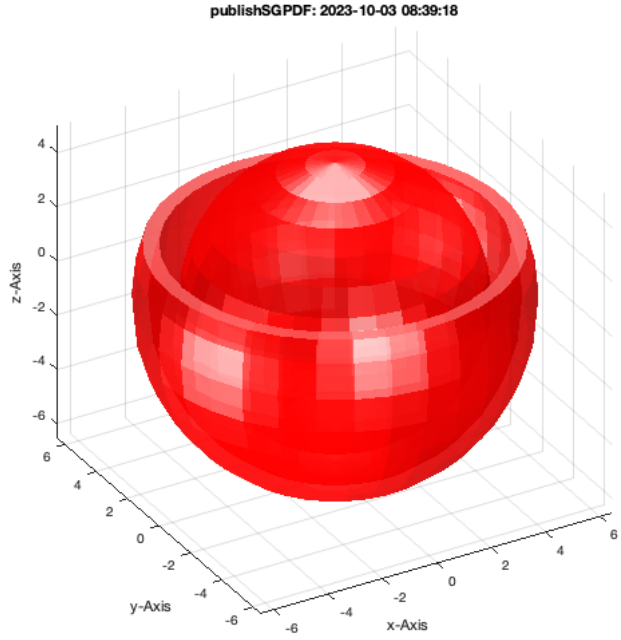

```
SGofCPLcommand('sph 10 , dup, shell, cutz 1');
```

```
SGofCPLcommand: SGofCPLcommand("sph 10 , dup, shell, cutz 1")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("sph 10 , dup, shell, cutz 1");');
```




```
SGofCPLcommand('sph 10 , dup, shell .5 1, cutz 2, clear 1, add');
```

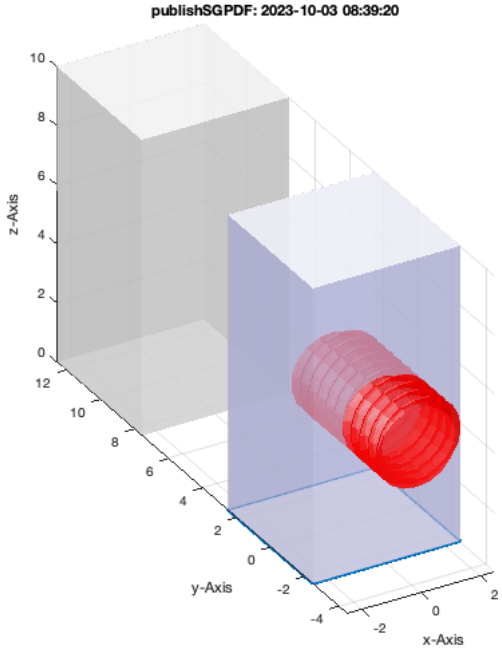
```
SGofCPLcommand: SGofCPLcommand("sph 10 , dup, shell .5 1, cutz 2, clear 1, add")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("sph 10 , dup, shell .5 1, cutz 2, clear 1, add");');
```



4. Solid Geometry Stack commands for Beooleand operations and relative movemets

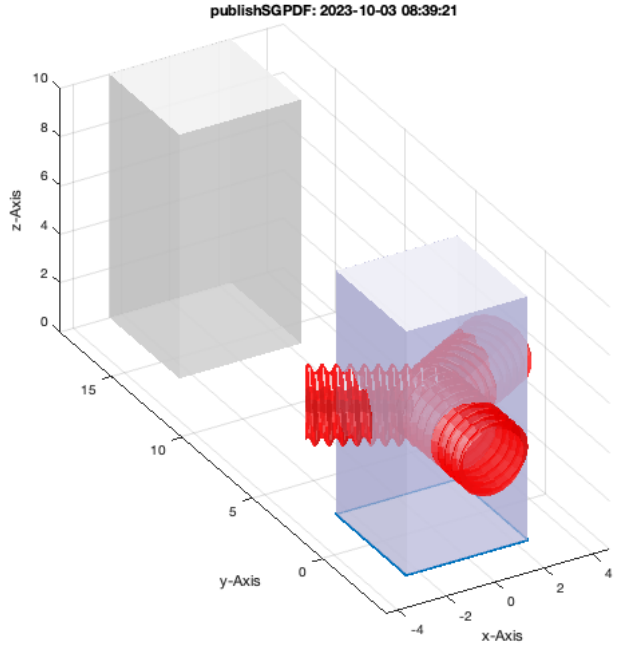
```
SGofCPLcommand('b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -3');
```

```
SGofCPLcommand: SGofCPLcommand("b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -3")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -3");');
```



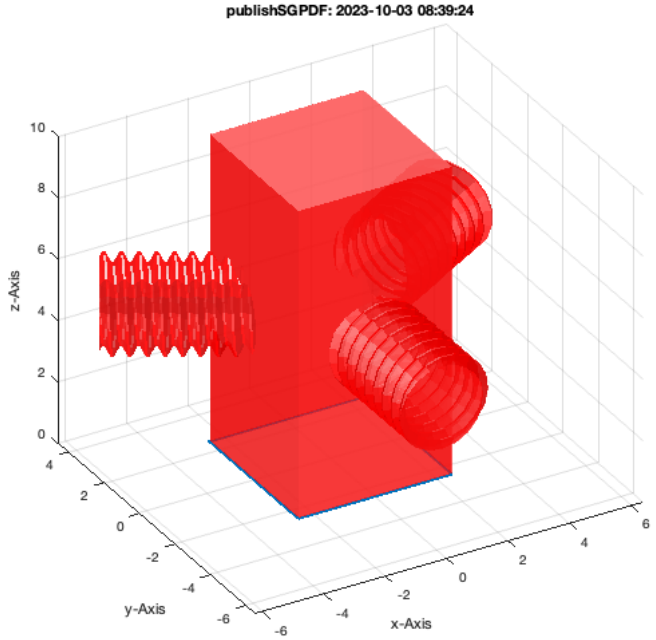

```
SGofCPLcommand('b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -3, dupr 3');
```

```
SGofCPLcommand: SGofCPLcommand("b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -3, dupr 3")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -3, dupr 3");');
```



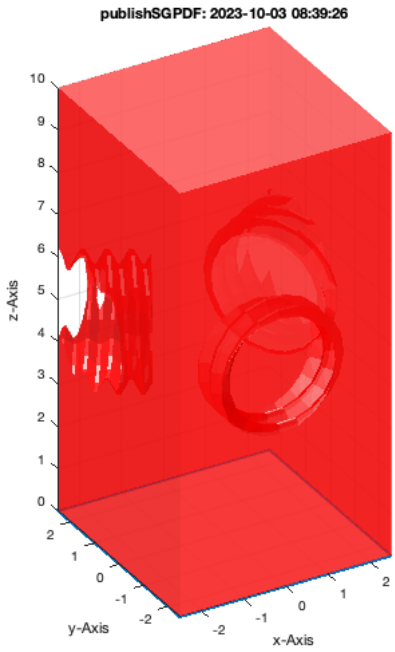

```
SGofCPLcommand('b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -1, dupr 3, add');
```

```
SGofCPLcommand: SGofCPLcommand("b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -1, dupr 3, add")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -1, dupr 3, add");');
```



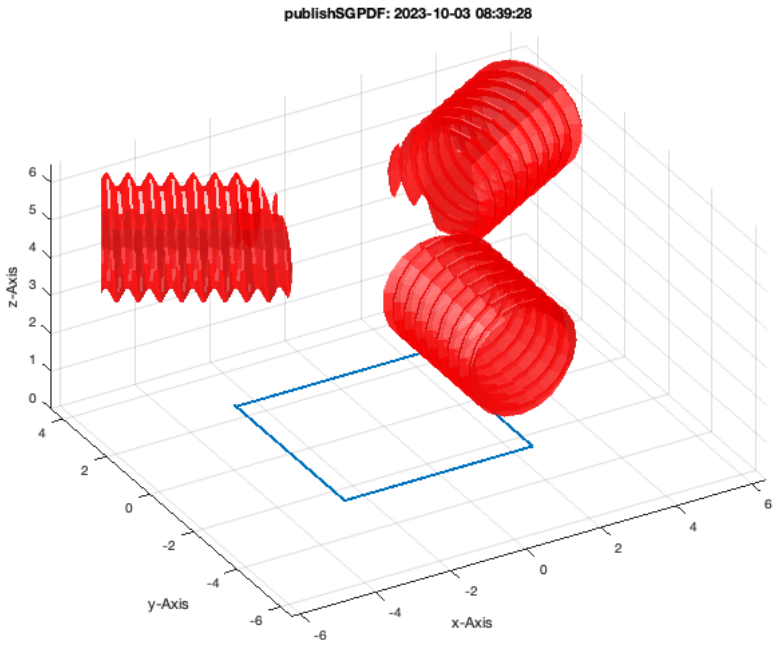

```
SGofCPLcommand('b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -1, dupr 3, sub');
```

```
SGofCPLcommand: SGofCPLcommand("b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -1, dupr 3, sub")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -1, dupr 3, sub");');
```



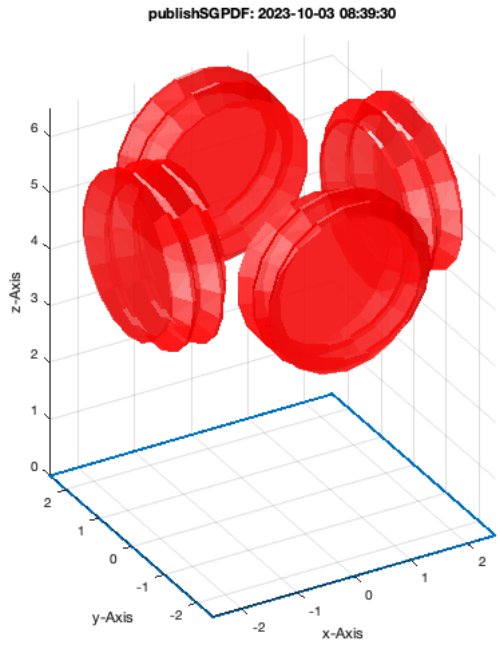

```
SGofCPLcommand('b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -1, dupr 3, rem');
```

```
SGofCPLcommand: SGofCPLcommand("b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -1, dupr 3, rem")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -1, dupr 3, rem");');
```




```
SGofCPLcommand('b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -1, dupr 4, isec');
```

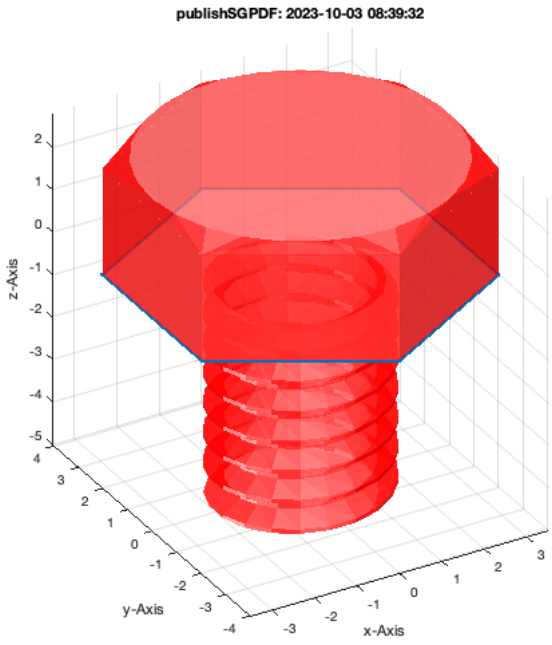
```
SGofCPLcommand: SGofCPLcommand("b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -1, dupr 4, isec")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("b 5 5, h 10, enter, scr 3 5, rotx, rel incenter, rel infront -1, dupr 4, isec");');
```



M4 x 5 Screw

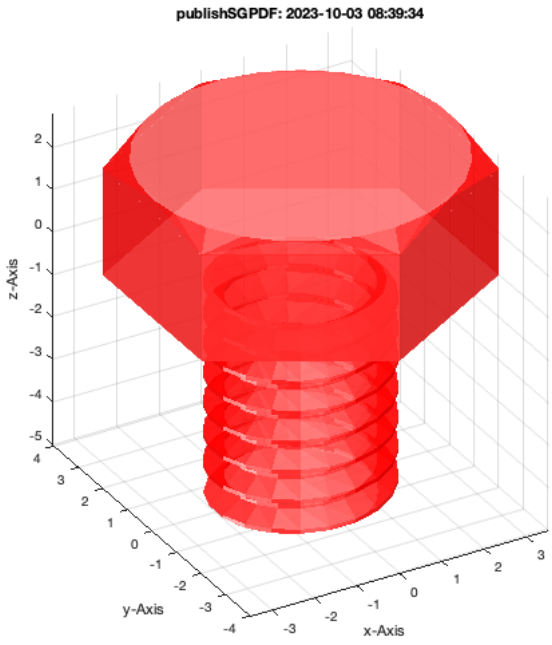
```
SGofCPLcommand('c 7, enter, d 7 0 0 6, hs 0.3, h 2.5, melt, enter, scr 4 5, rel under, add, save M4');
```

```
SGofCPLcommand: SGofCPLcommand("c 7, enter, d 7 0 0 6, hs 0.3, h 2.5, melt, enter, scr 4 5, rel under, add, save M4")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("c 7, enter, d 7 0 0 6, hs 0.3, h 2.5, melt, enter, scr 4 5, rel under, add, save M4");');
```



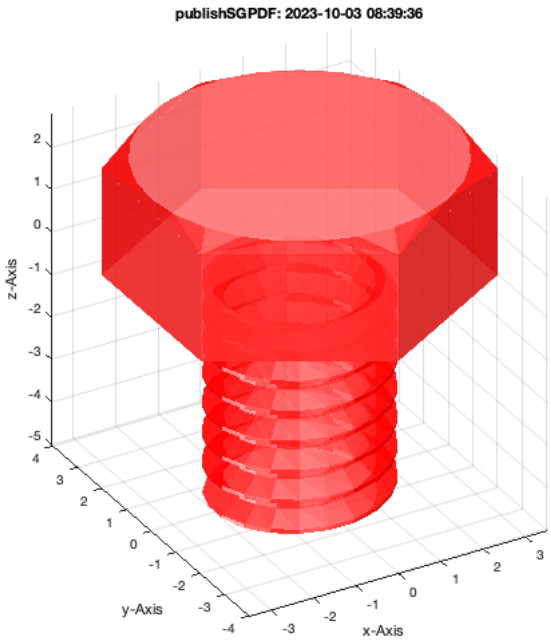

```
SGofCPLcommand('load M4');
```

```
SGofCPLcommand: SGofCPLcommand("load M4")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("load M4");');
```



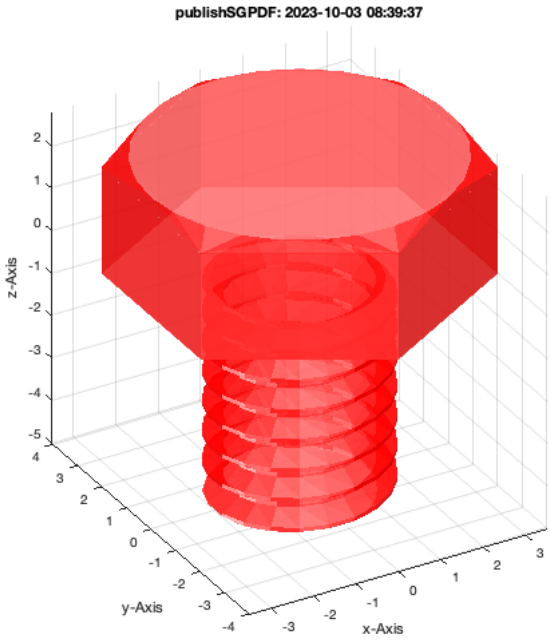

```
SGofCPLcommand('load M4, write screw_M4');
```

```
publishSGPDF:<a href = "matlab: openbydoubleclick ('/Users/timlueth/Desktop')">/Users/timlueth/Desktop/</a><a href = "matlab: openbydoubleclick ('/User
ans =
    '/Users/timlueth/Desktop/screw_M4.STL'
SGofCPLcommand: SGofCPLcommand("load M4, write screw_M4")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("load M4, write screw_M4");');
```




```
SGofCPLcommand('read screw_M4');
```

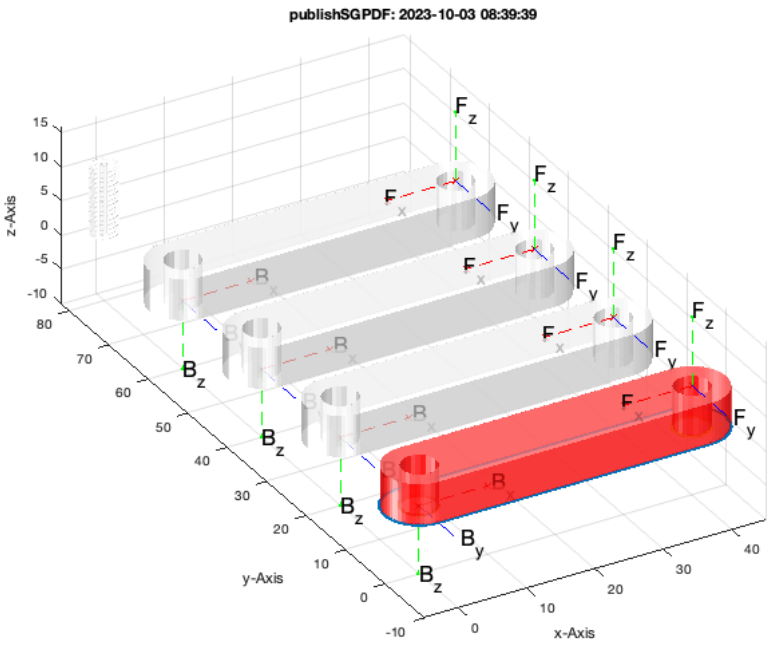
```
nam =
    '/Users/timlueth/Desktop/screw_M4.STL'
LOADING BINARY STL-File: /Users/timlueth/Desktop/screw_M4.STL
Binary Header: COLOR=RGBA,MATERIAL=AAAABBBBCCCCDDDD;SOLID "screw_M4 by timlueth" 03-Oct-2023 08
Color of solid defined as: "k"
Alpha of solid defined as: 65.00
Number of facets: 1926
Number of vertices: 965

SGofCPLcommand: SGofCPLcommand("read screw_M4")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("read screw_M4");');
```



5. FRAME CONCEPTS

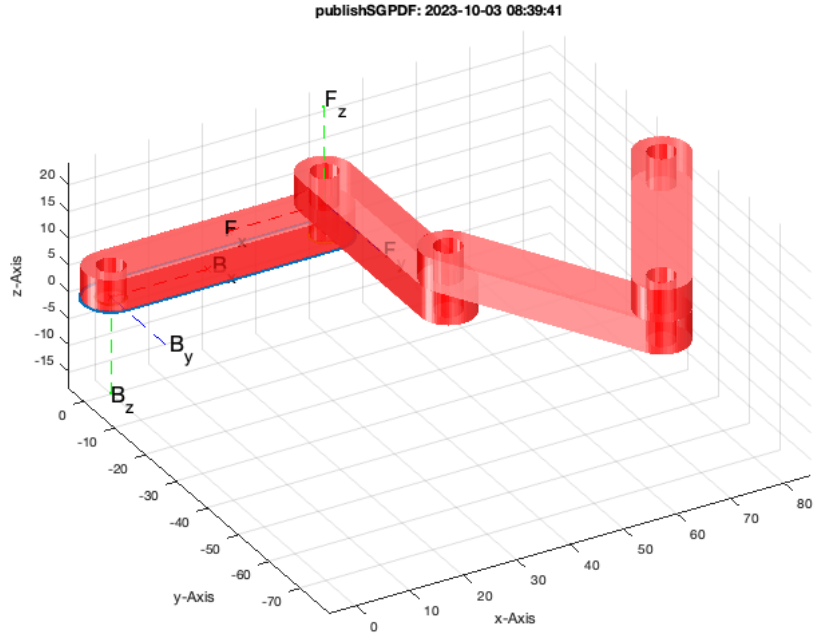
```
SGofCPLcommand('co 10 50 5, h 6, fset B 1 0 R1, fset F 2 0 R2, enter, scr 4 10 1 1, swap, dup, dup, dup ');
```

```
SGofCPLcommand: SGofCPLcommand("co 10 50 5, h 6, fset B 1 0 R1, fset F 2 0 R2, enter, scr 4 10 1 1, swap, dup, dup, dup")
SGofCPLcommand: SGfigureeval('SGofCPLcommand("co 10 50 5, h 6, fset B 1 0 R1, fset F 2 0 R2, enter, scr 4 10 1 1, swap, dup, dup, dup");');
```



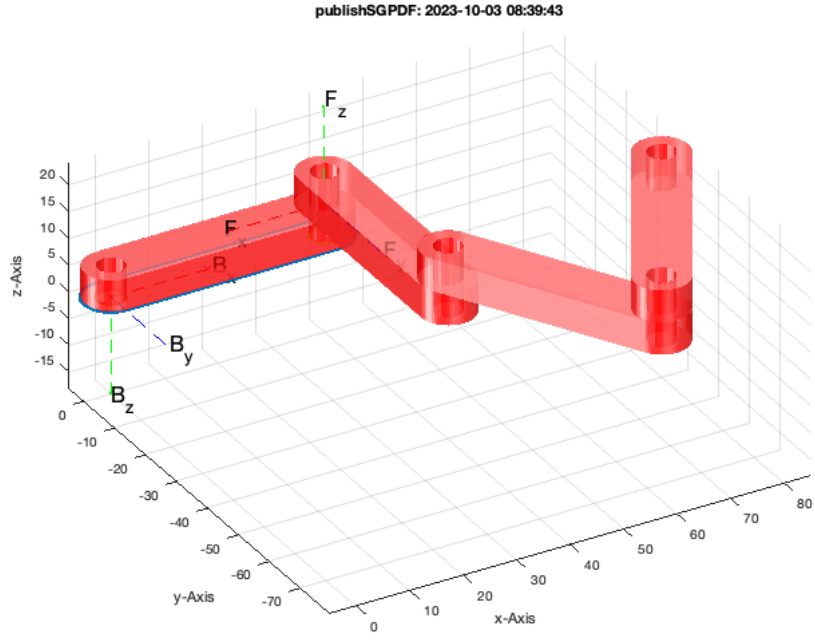

```
SGofCPLcommand('co 10 50 5, h 6, fset B 1 0 R1, fset F 2 0 R2, dup, col m, falign F B -90, dup, col b, falign F B 30, dup, col g, falign F B 120, cat,
```

```
SGofCPLcommand: SGofCPLcommand("co 10 50 5, h 6, fset B 1 0 R1, fset F 2 0 R2, dup, col m, falign F B -90, dup, col b, falign F B 30, dup, col g, falig
SGofCPLcommand: SGfigureeval('SGofCPLcommand("co 10 50 5, h 6, fset B 1 0 R1, fset F 2 0 R2, dup, col m, falign F B -90, dup, col b, falign F B 30, dup
```




```
SGofCPLcommand('co 10 50 5, h 6, fset B 1 0 R1, fset F 2 0 R2, dup, col m, falign F B -90, dup, col b, falign F B 30, dup, col g, falign F B 120, cat,
```

```
publishSGPDF:<a href = "matlab: openbydoubleclick ('/Users/timlueth/Desktop')">/Users/timlueth/Desktop/</a><a href = "matlab: openbydoubleclick ('/User
ans =
    '/Users/timlueth/Desktop/open_chain.STL'
SGofCPLcommand: SGofCPLcommand("co 10 50 5, h 6, fset B 1 0 R1, fset F 2 0 R2, dup, col m, falign F B -90, dup, col b, falign F B 30, dup, col g, falig
SGofCPLcommand: SGfigureeval('SGofCPLcommand("co 10 50 5, h 6, fset B 1 0 R1, fset F 2 0 R2, dup, col m, falign F B -90, dup, col b, falign F B 30, dup
```



Final Remarks

```
close all
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 08:39:44!
Executed 03-Oct-2023 08:39:46 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
pde_toolbox
simmechanics
simscape
simulink
================================================================================================
```

Published with MATLAB® R2023a