Tutorial 30: Creating graphical drawings using point, lines, surfaces, frames etc.

2017-02-10: 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 3.4 required)

In this tutorial, features are introduced that allow you to create drawings for the descriptive geometry using Matlab. The goal is to display lines, planes, spaces and polygons, surfaces and surface-bounded volumes and to label them mathematically (Tex-style). some functions are based in individual points such as:

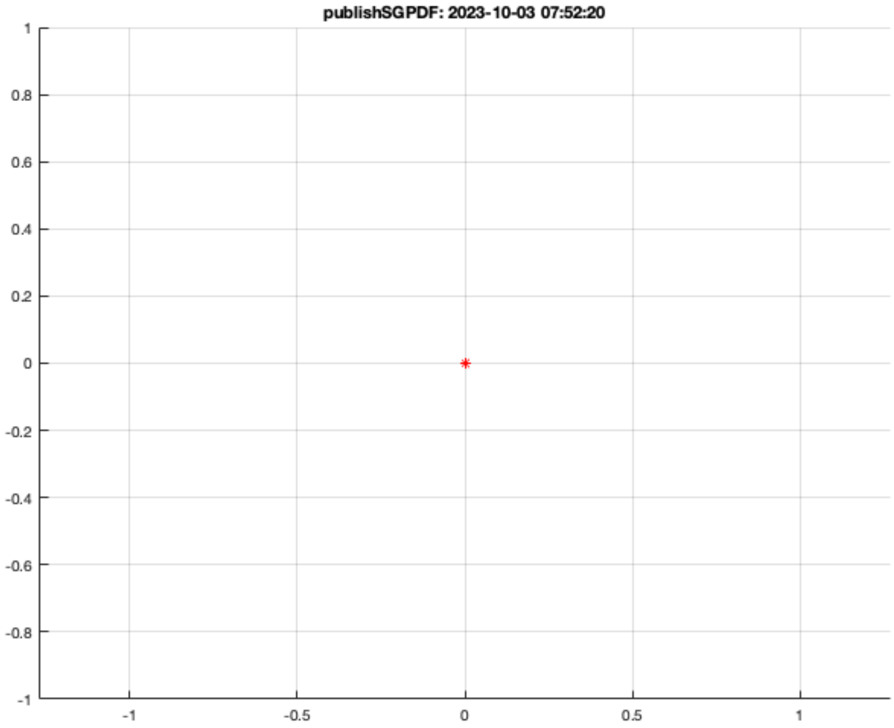
- pplot - plot a point in defined color, shape and size
- lplot - plot a line between two points with color, width, tip, start point end point
- aplot - plot an angle at a point using a line and a second line or angle
- slplot - plot a staight line using a start point and direction vector
- tplot;
- tfplot; some functions are based on point lists (PL) or vertex lists (VL), such as: PLplot, VLplot,

1. Plotting points (PL) and vertices (3D)

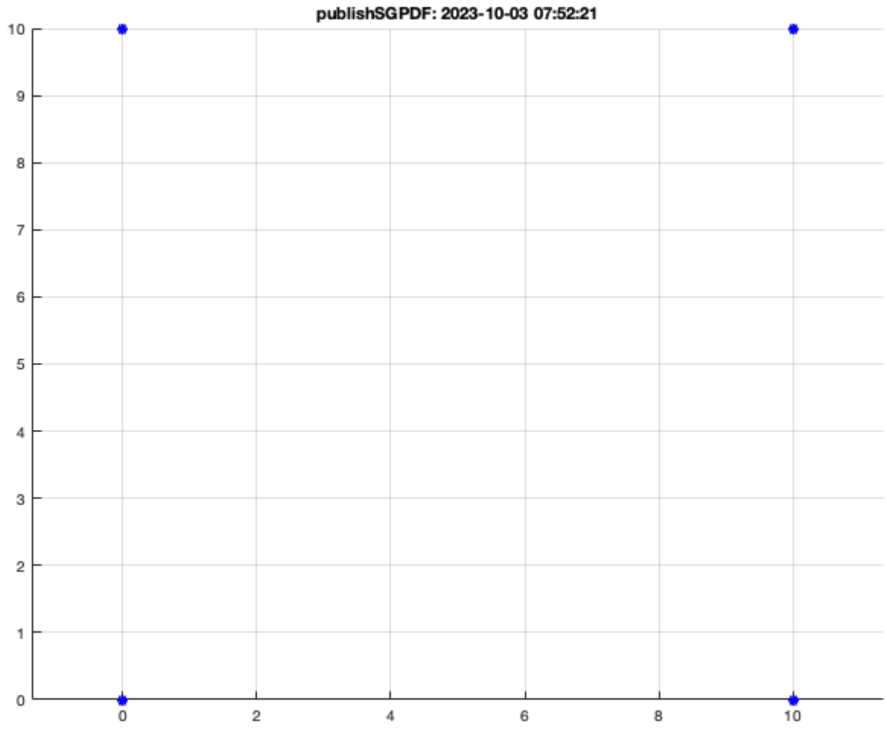
```
p=[0 0]      % row style
p=[0;0]      % column style
v=[0 0 0]  % row style
v=[0; 0; 0]  % column style


SGfigure; pplot(p);
```

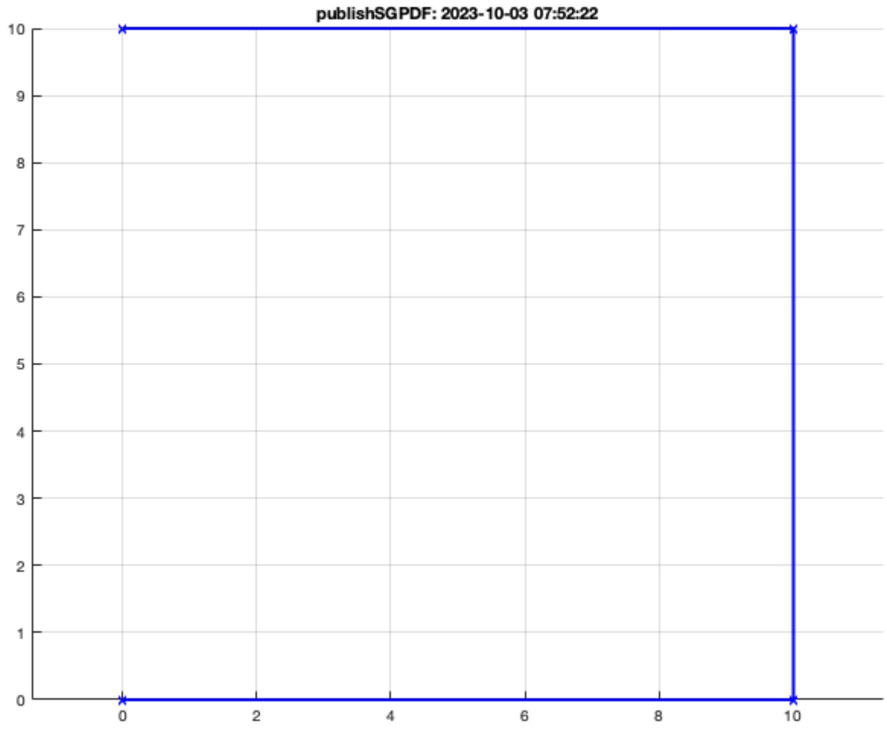
```
p =
     0     0
p =
     0
     0
v =
     0     0     0
v =
     0
     0
     0
```



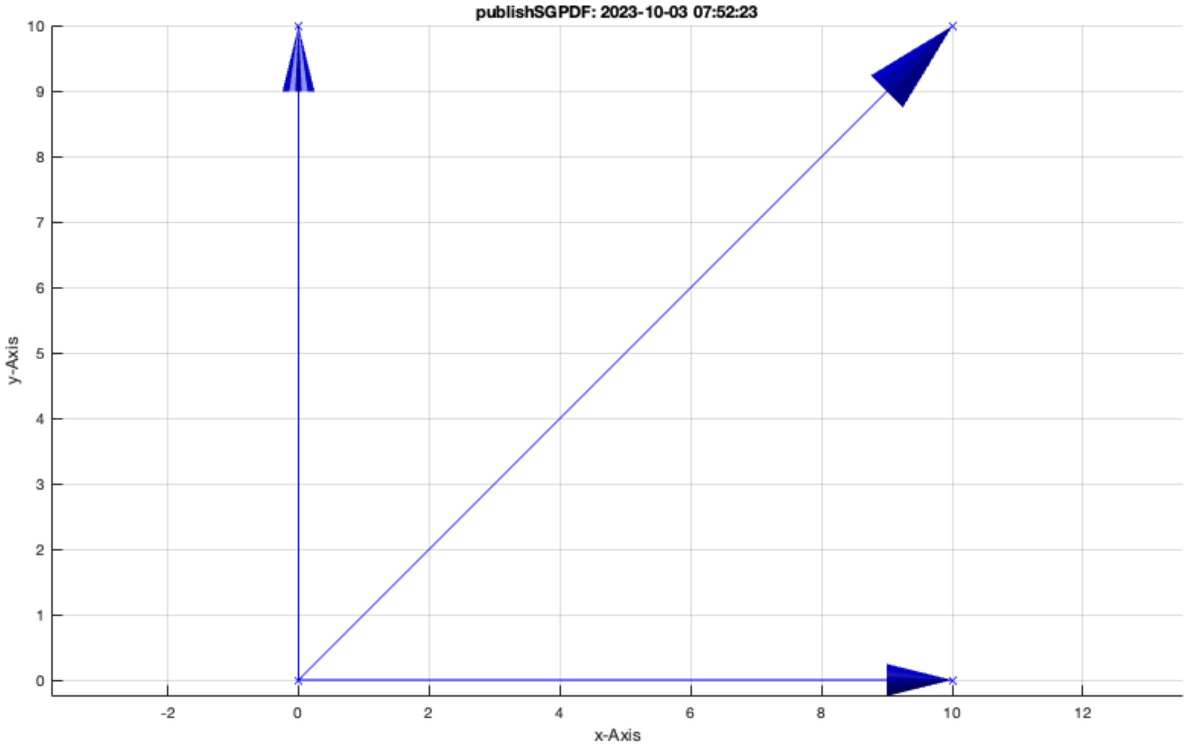
```
SGfigure; PLplot([0 0;10 0;10 10; 0 10],'b*',2);            % point plot of point list
```



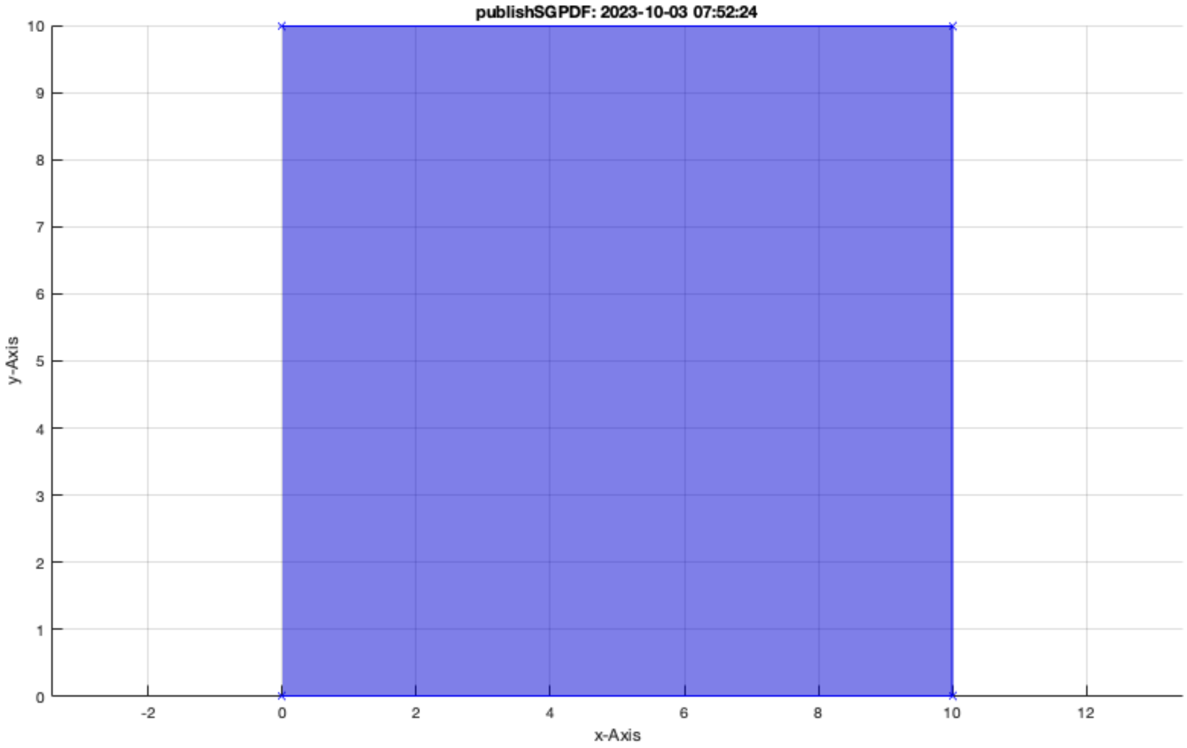
```
SGfigure; PLplot([0 0;10 0;10 10; 0 10],'bx-',2);          % Line plot of point list
```



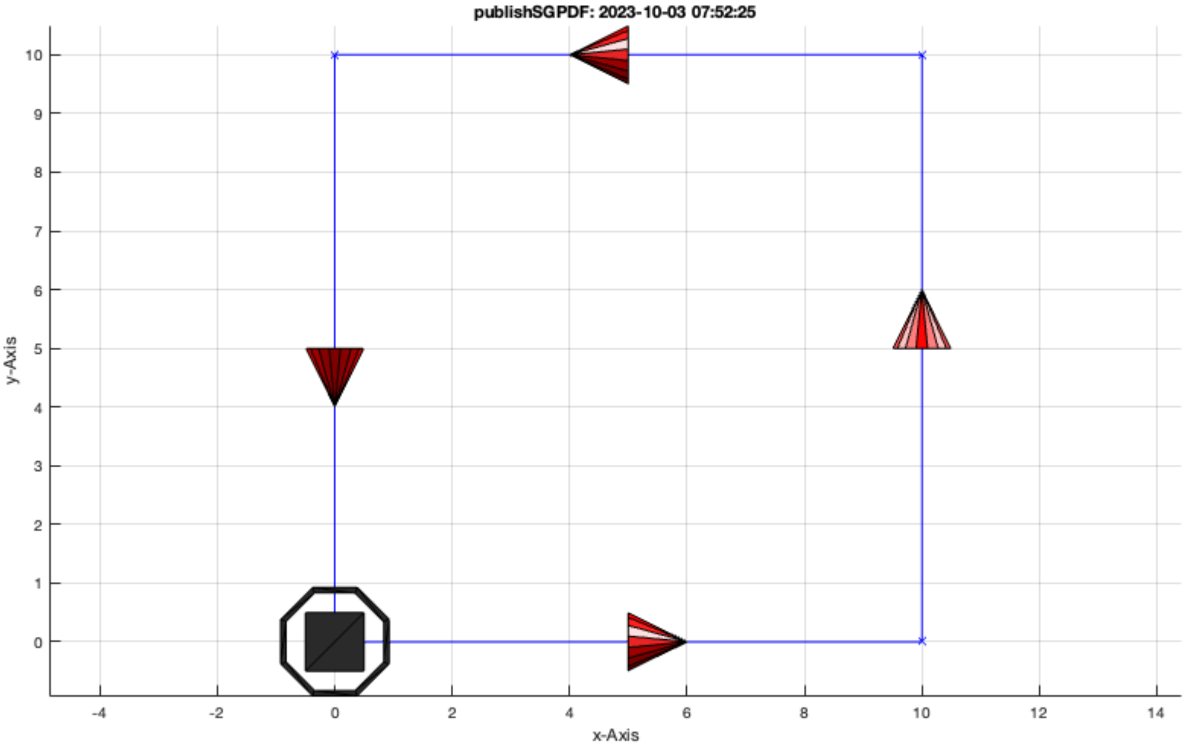
```
SGfigure; PLplot([0 0;10 0;10 10; 0 10],'bx-',1,1);          % Vector plot of point list
```



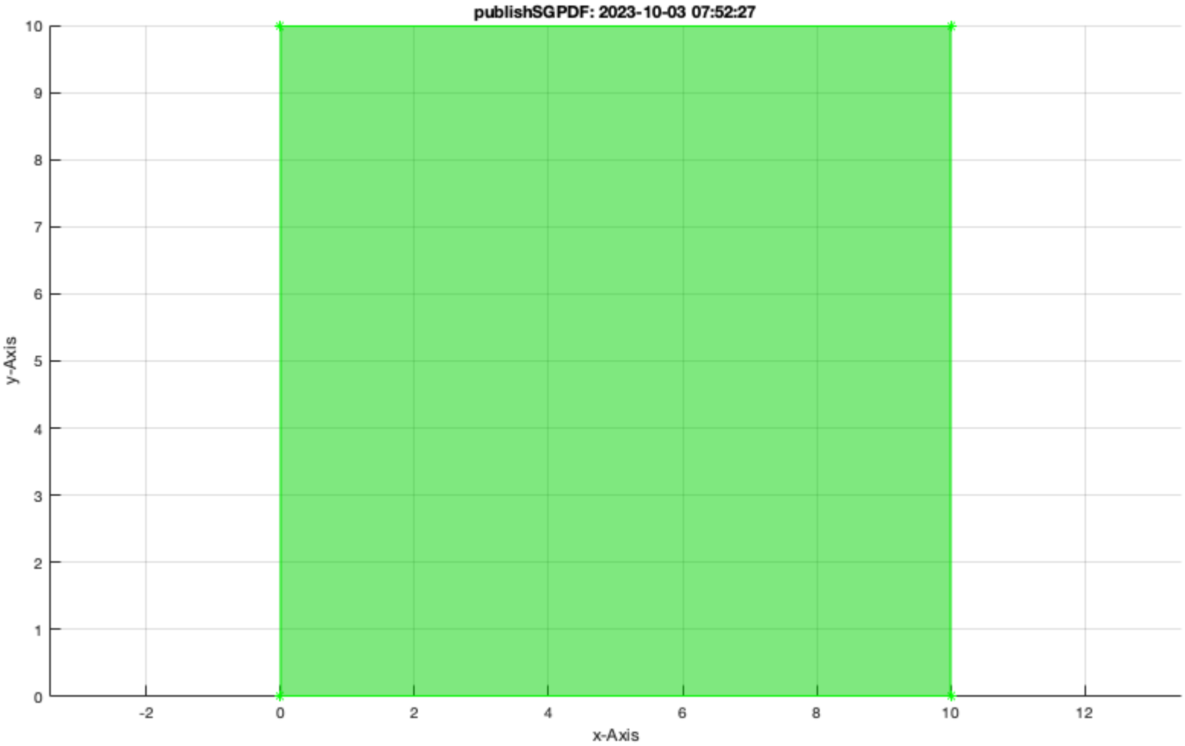
```
SGfigure; PLplot([0 0;10 0;10 10; 0 10],'bx-',1,'',0.5);    % Surface enlosed by point list
```



```
SGfigure; CPLplot([0 0;10 0;10 10; 0 10],'bx-',1,1,1,1);    % Plotting closed polygon
```

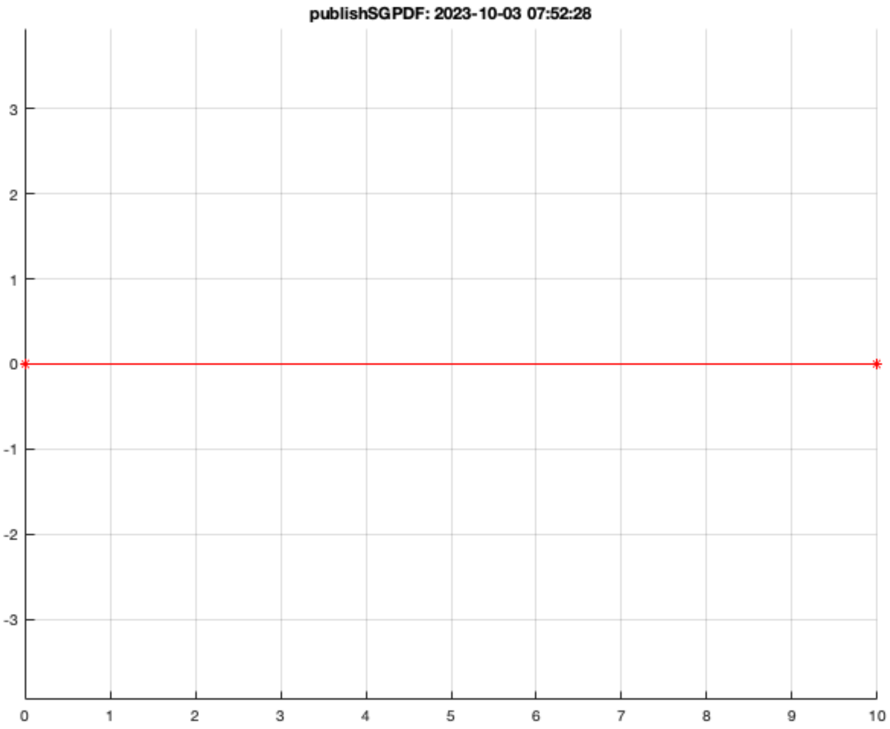


```
SGfigure; CPLfaceplot([0 0;10 0;10 10; 0 10],'g*-',1,0.5);  % Plotting closed polygon surfaces
```

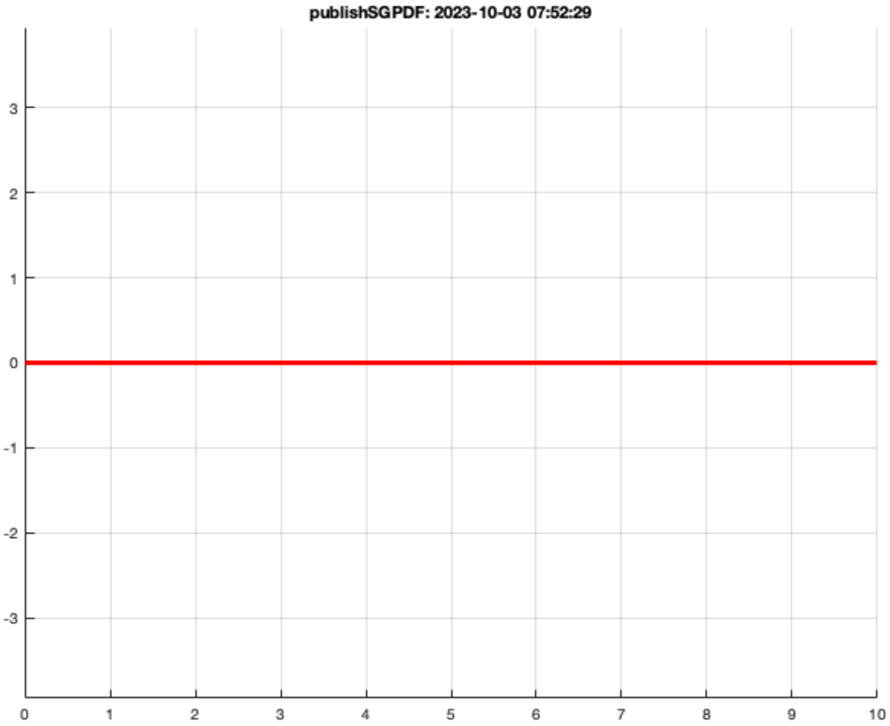


2. Plotting lines

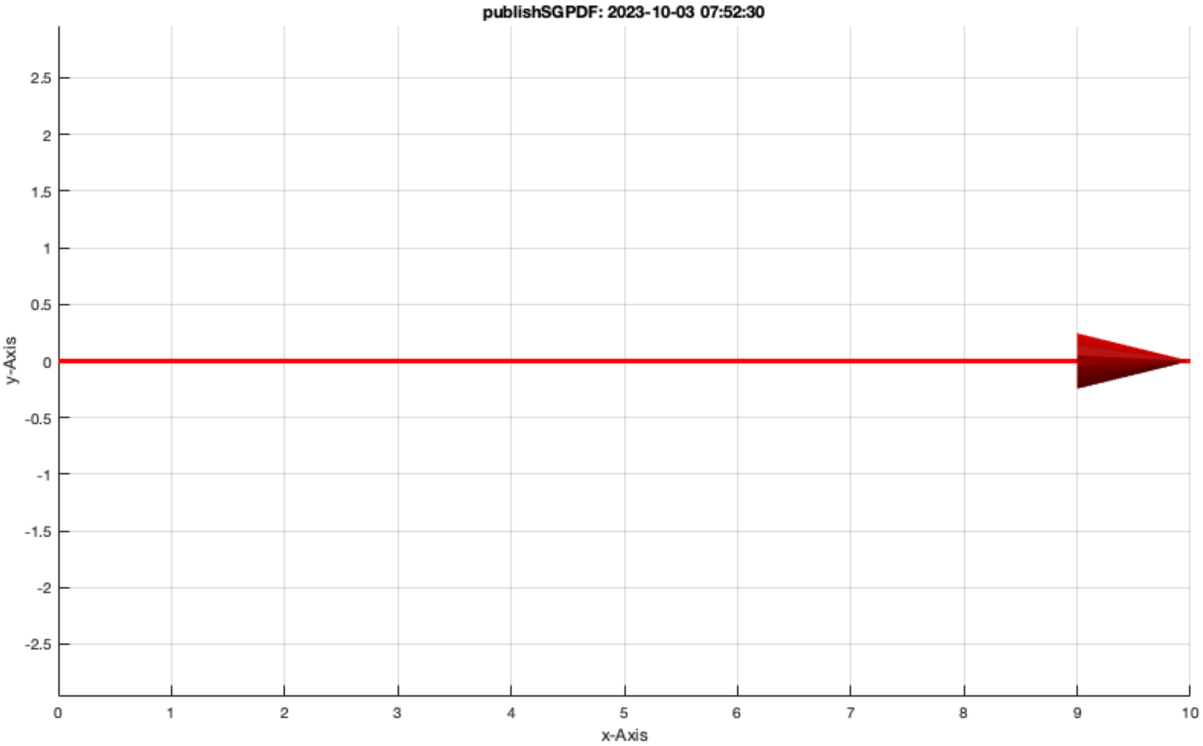
```
SGfigure; lplot([0 0],[10 0],'r*-');
```



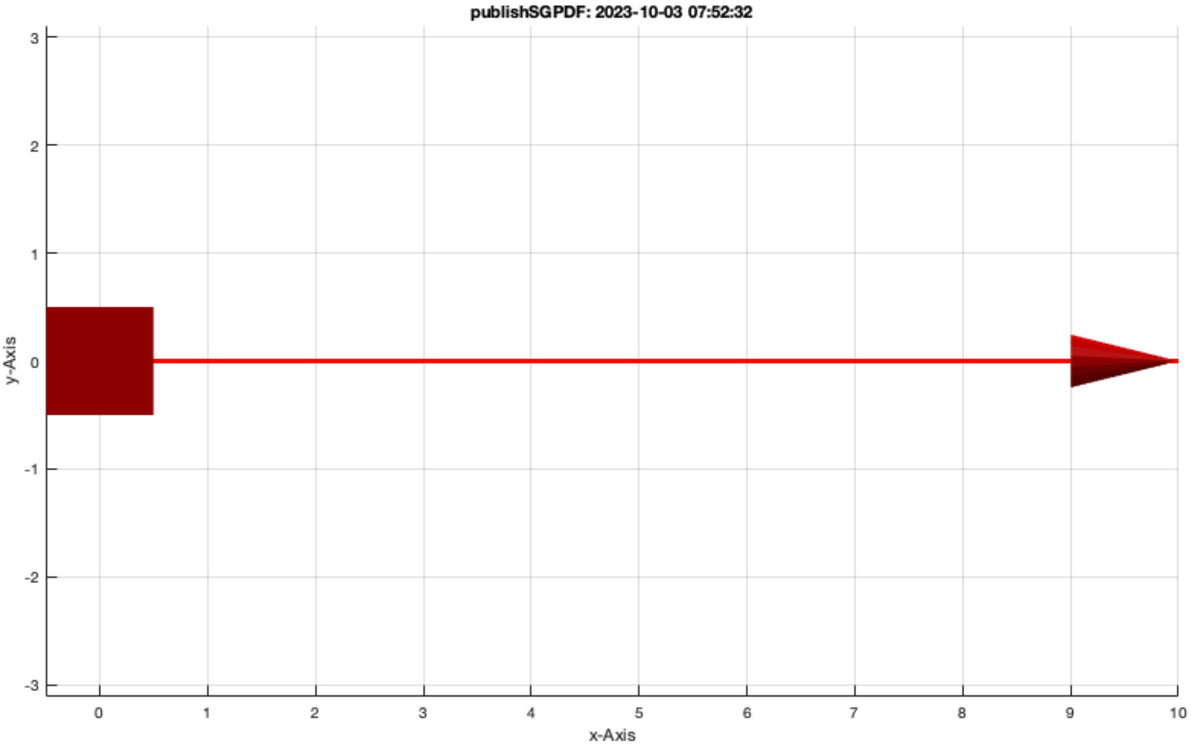
```
SGfigure; lplot([0 0],[10 0],'r-',3);
```



```
SGfigure; lplot([0 0],[10 0],'r-',3,1);
```

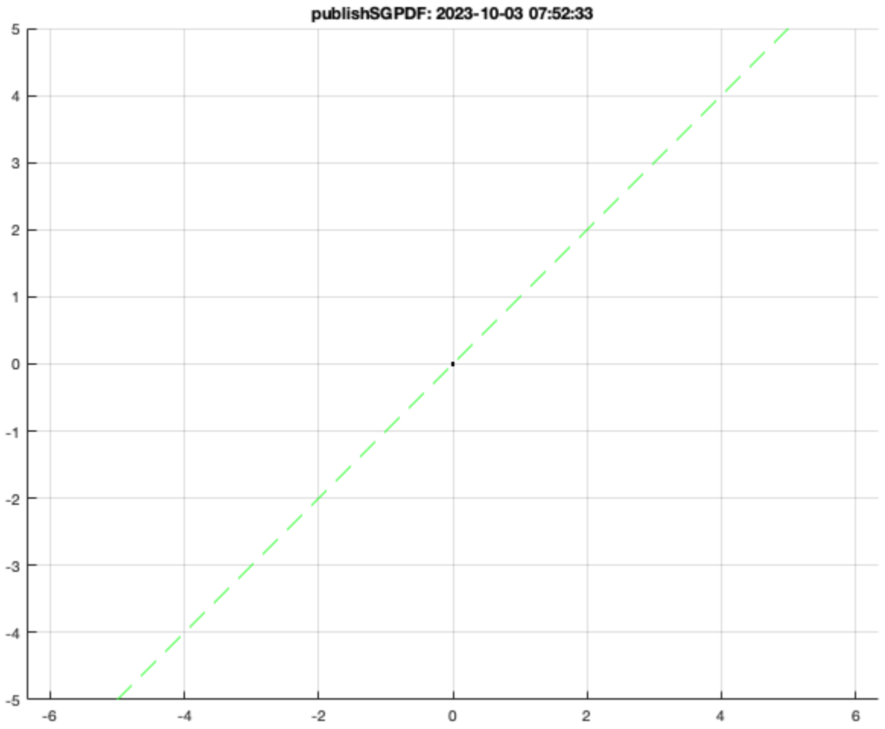


```
SGfigure; lplot([0 0],[10 0],'r-',3,1,1);
```



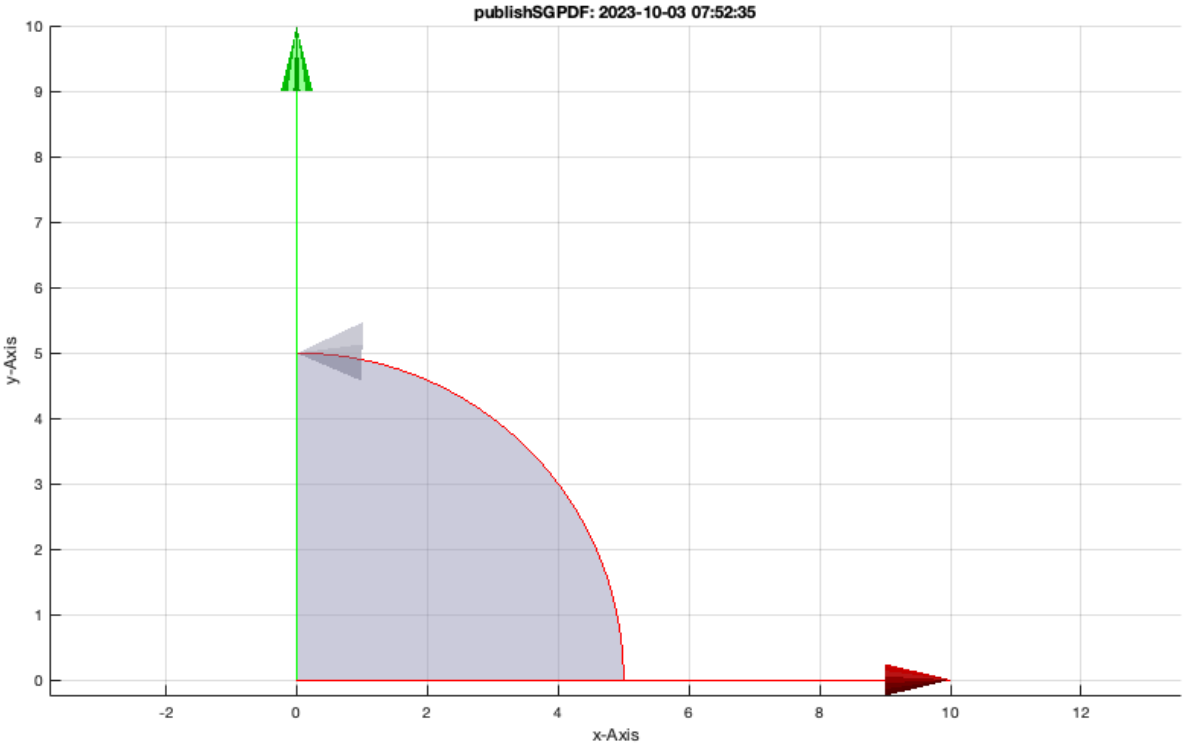
slplot([0 0],[10 0],'r-',3,1,1);

```
SGfigure; slplot([0 0 0],[1 1 1],'color','g--')
```



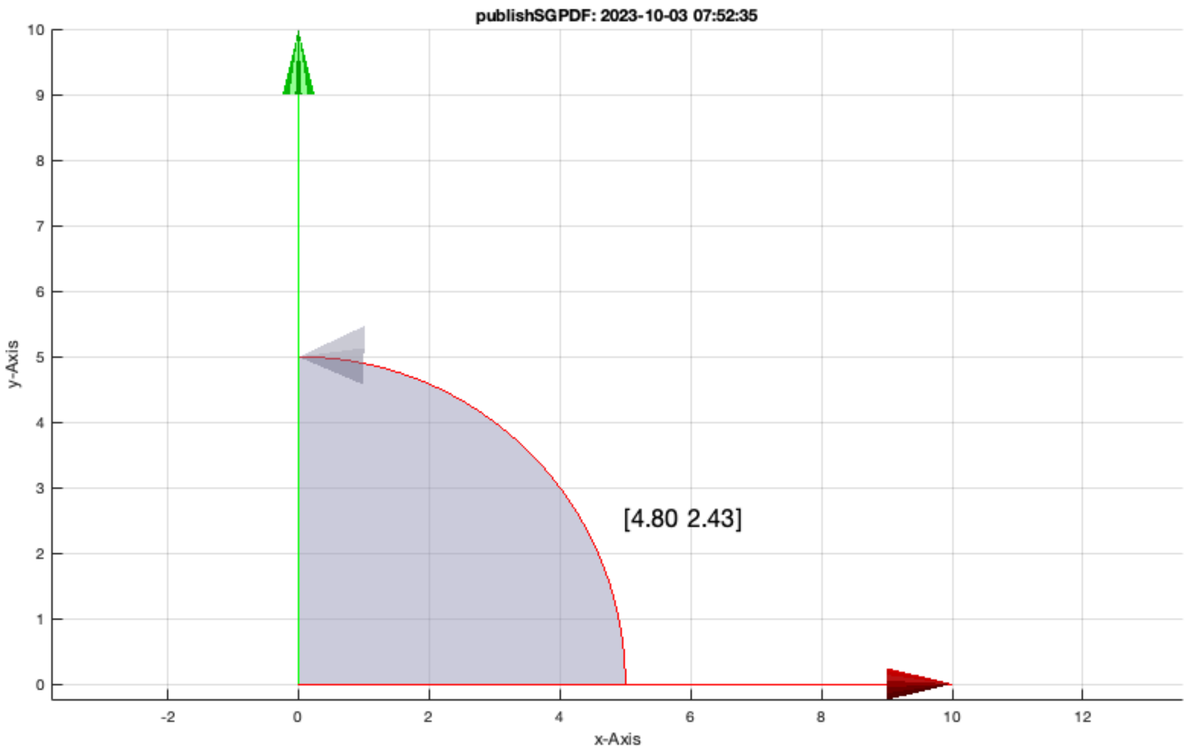
3. Plotting angles

```
SGfigure; aplot([0 0],[10 0],pi/2);
```



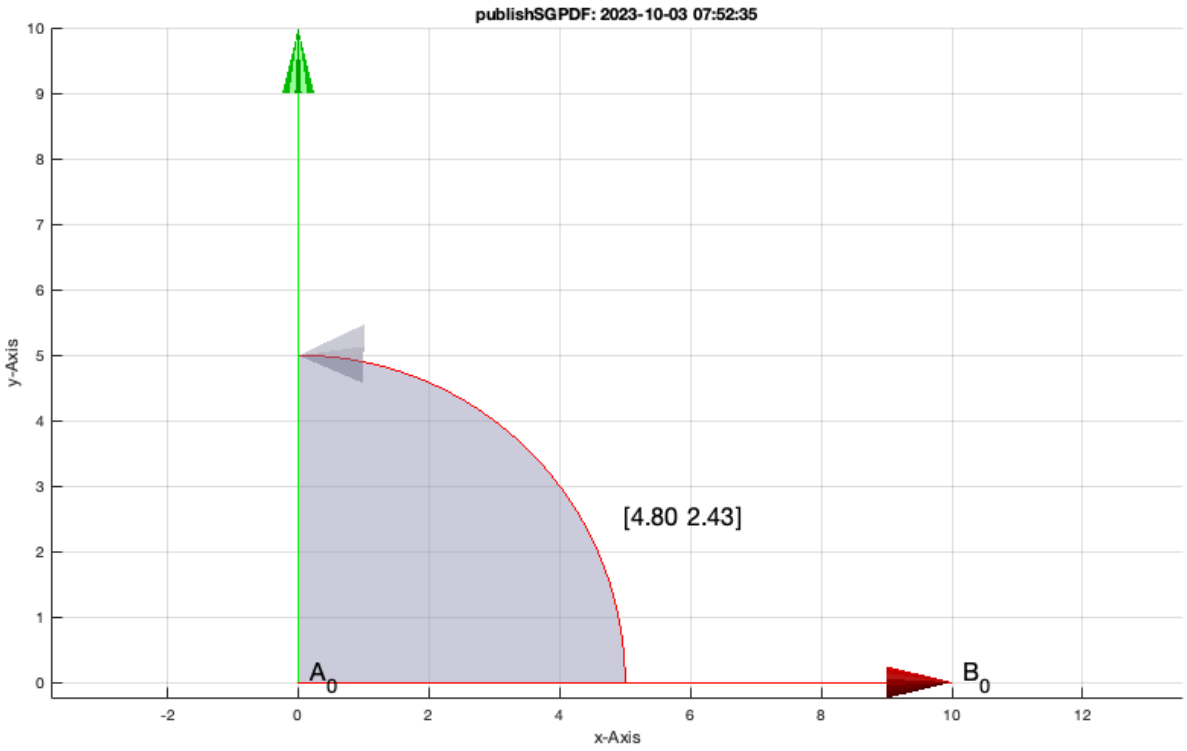
4. Plotting coordinate

```
p=ginput(1); textP (p,sprintf('[%.2f %.2f]',p));
```



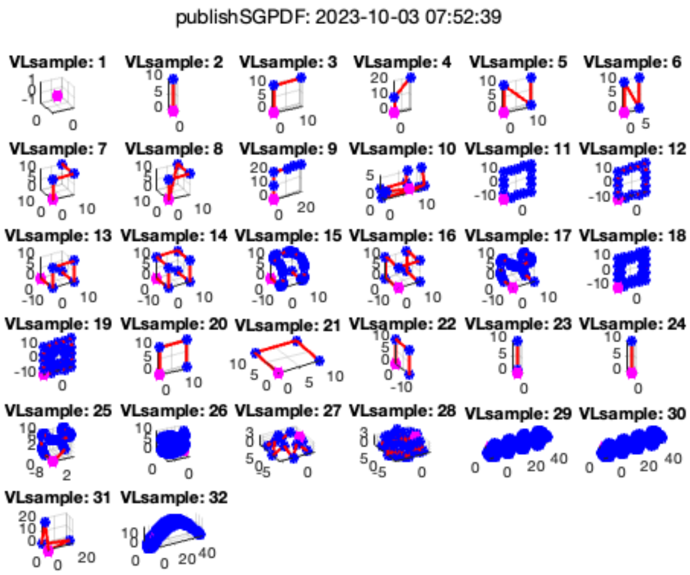
8. Adding text to the drawings

```
textP ([0 0],'A0'); textP ([10 0],'B0');
```

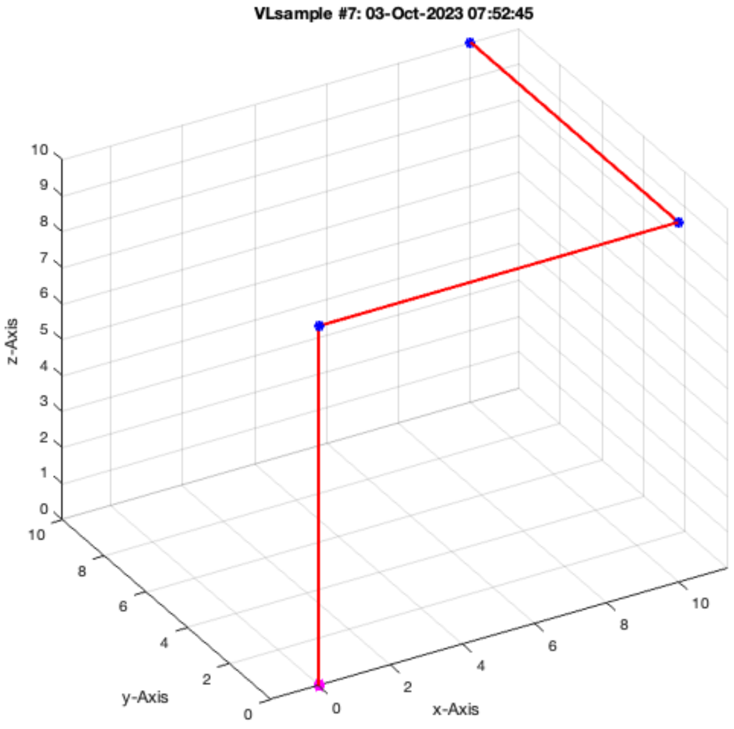


9. Helpful generic polygons for

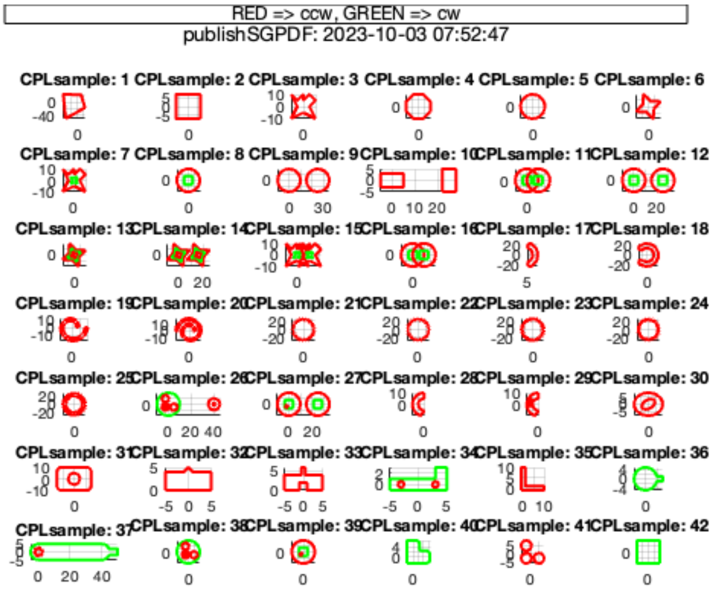
```
SGfigure; VLsample;
```



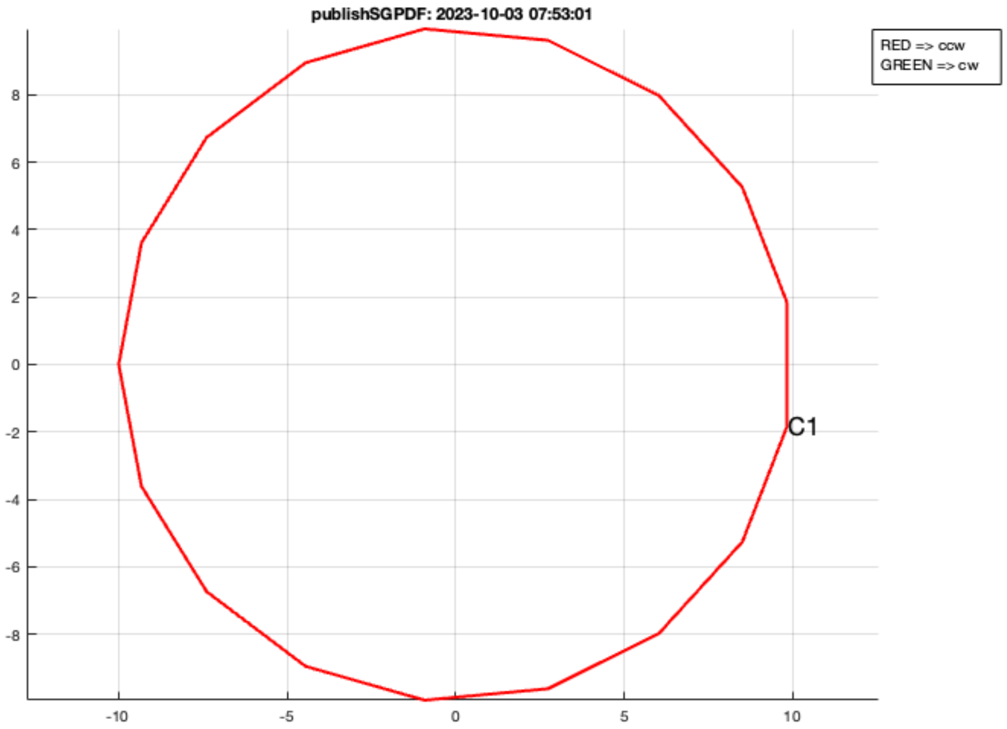
```
SGfigure; VLsample(7);
```



```
SGfigure; CPLsample;
```



```
SGfigure; CPLsample(5);
```



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 07:53:01!
Executed 03-Oct-2023 07:53:03 by 'timlueth' on a MACI64 using Mac OSX 13.6 | R2023a Update 5 | SG-Lib 5.4
======================================== Used Matlab products: ======================================
distrib_computing_toolbox
fixed_point_toolbox
map_toolbox
matlab
simmechanics
simscape
simulink
=====================================================================================================
```

Published with MATLAB® R2023a