Exercise 30: Create Mechanisms using Universal Planar Links

2017-02-10: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de)

Contents

- Summary of the already explained aspects of the VLFL-Toolbox
- Motivation for this tutorial
- 1. Plotting points (PL) and vertices (3D)
- 2. Plotting lines
- 3. Plotting angles
- 4. Plotting coordinate
- 8. Adding text to the drawings
- 9. Helpful generic polygons for
- Final Remarks

Summary of the already explained aspects of the VLFL-Toolbox

The previous examples of this series covered the following topics:

- Example 01: First Steps Using the VLFL-Toolbox for Solid Object Design
- Example 02: Using the VLFL-Toolbox for STL-File Export and Import
- Example 03: Closed 2D Contours and Boolean Operations in 2D
- Example 04: 2.5D Design Using Boolean Operators on Closed Polygon Lists (CPL)
- Example 05: Creation, Relative Positioning and Merging of Solid Geometries (SG)
- Example 06: Relative Positioning and Alignment of Solid Geometries (SG)
- Example 07: Rotation of Closed Polygon Lists for Solid Geometry Design
- Example 08: Slicing, Closing, Cutting and Separation of Solid Geometries
- Example 09: Boolean Operations with Solid Geometries
- Example 10: Packaging of Sets of Solid Geometries (SG)
- Example 11: Attaching Coordinates Frames to Create Kinematik Models
- Example 12: Define Robot Kinematics and Detect Collisions
- Example 13: Mounting Faces and Conversion of Blocks into Leightweight-structures
- Example 14: Manipulation Functions for Closed Polygons and Laser Cutting (SVG)
- Example 15: Create a Solid by 2 Closed Polygons
- Example 16: Create a Solid along a contour
- Example 20: Programmaticaly Interface to SimMechanics Multi-Body Toolbox
- Example 21: Programmaticaly Design of SimMechanisms Mechanisms and Simulink Signals
- Example 22: Creation of fixed nodes for the assembly of joints
- Example 23: Automatic Creation of a Missing Link and 3D Print of a Complete Model
- Example 24: Automatic Creation of a Joint Limitations
- Example 25: Automatic Creation of Video Titels, Endtitels and Textpages

- Example 26: Create Mechanisms using Universal Planar Links
- Example 27: Fourbar-Linkage: 2 Pose Syntheses and Linkage Export for 3D Printing
- Example 28: Fourbar-Linkage: 3 Pose Syntheses and Linkage Export for 3D Printing

Motivation for this tutorial

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 =

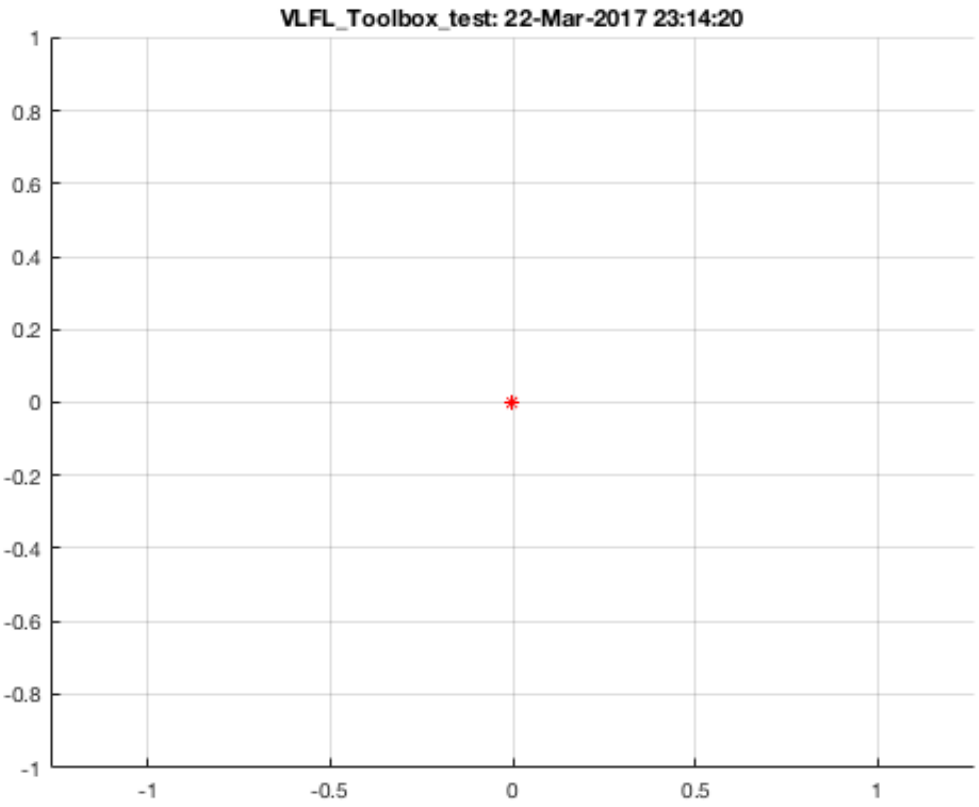
     0
     0


v =

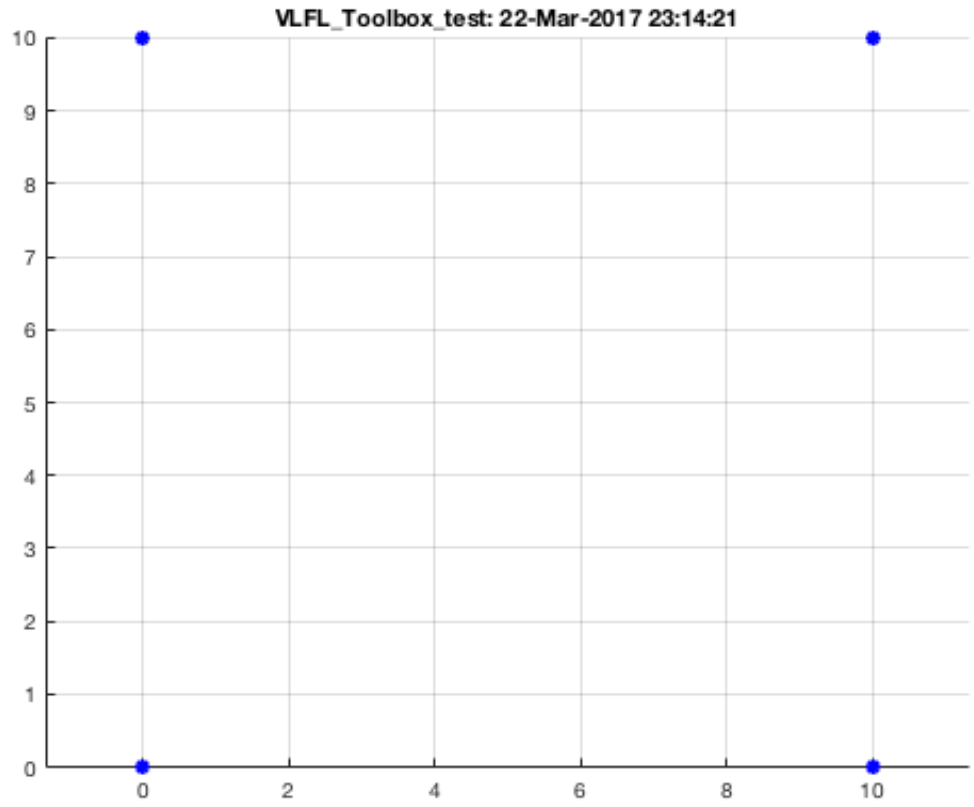
     0     0     0


v =

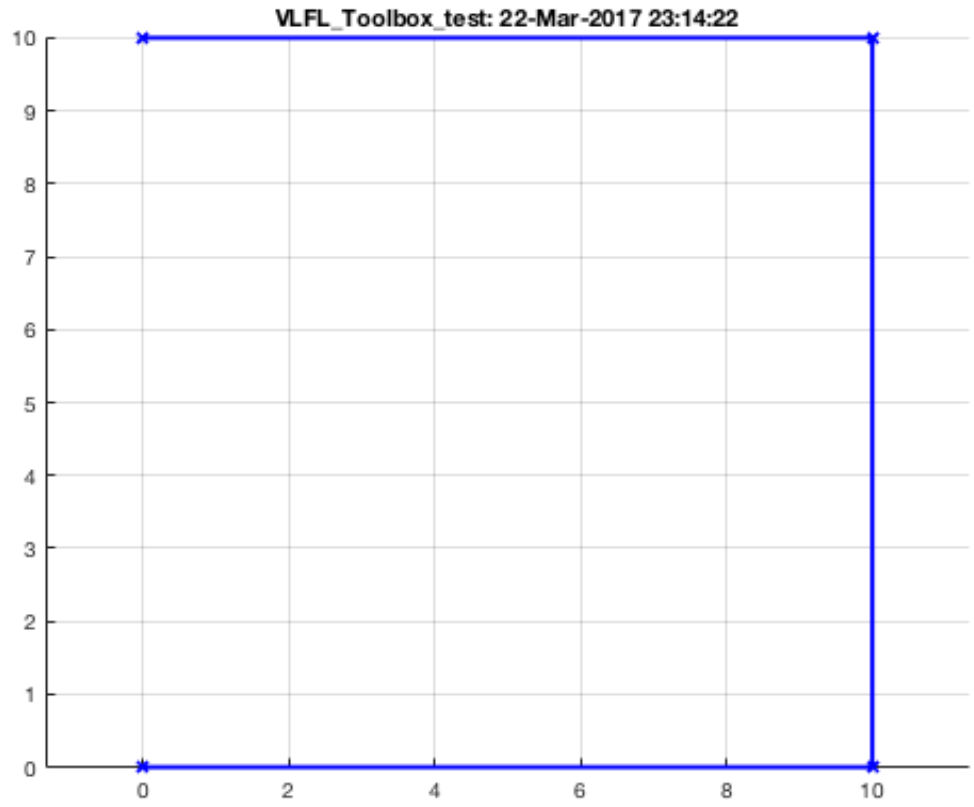
     0
     0
     0
```



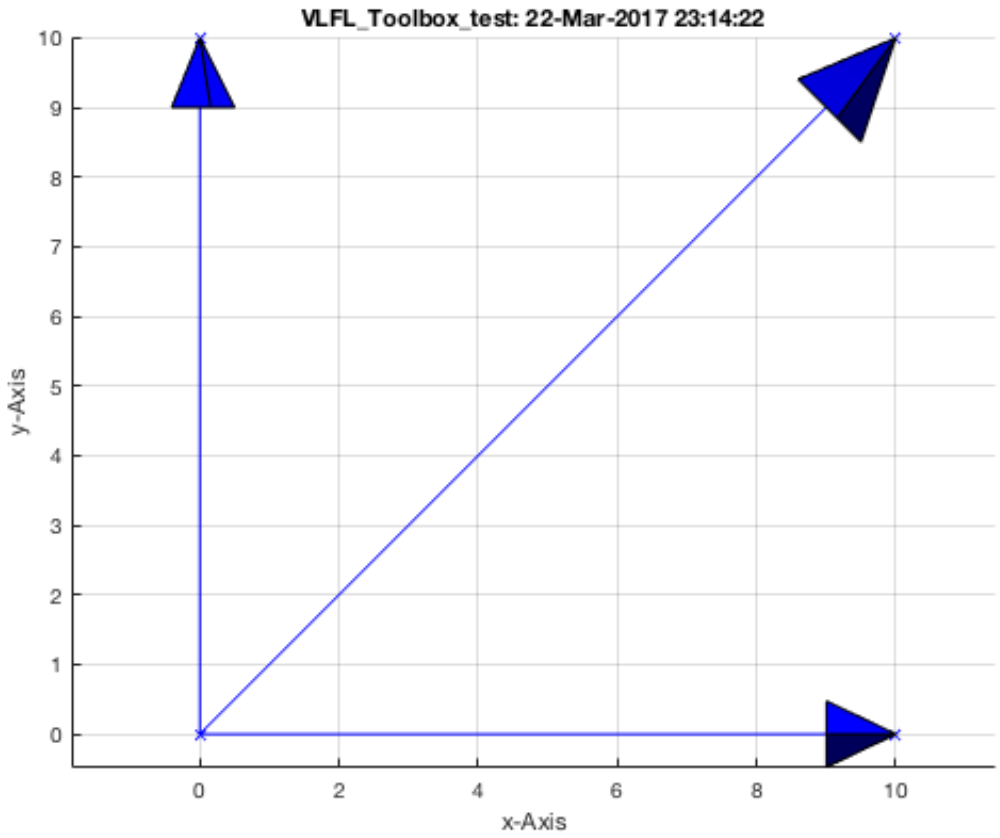

```
SGfigure; PLplot([0 0;10 0;10 10; 0 10],'b*',2);            % point plot of point list
```



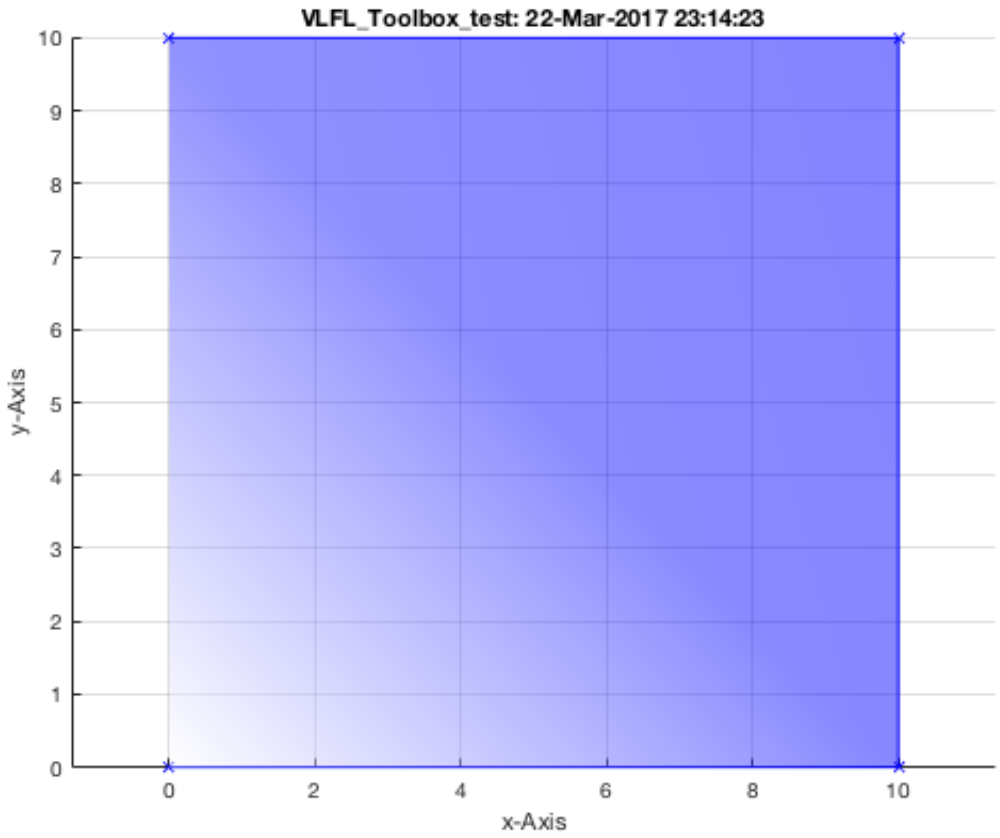

```
SGfigure; PLplot([0 0;10 0;10 10; 0 10],'bx-',2);           % Line plot of point list
```



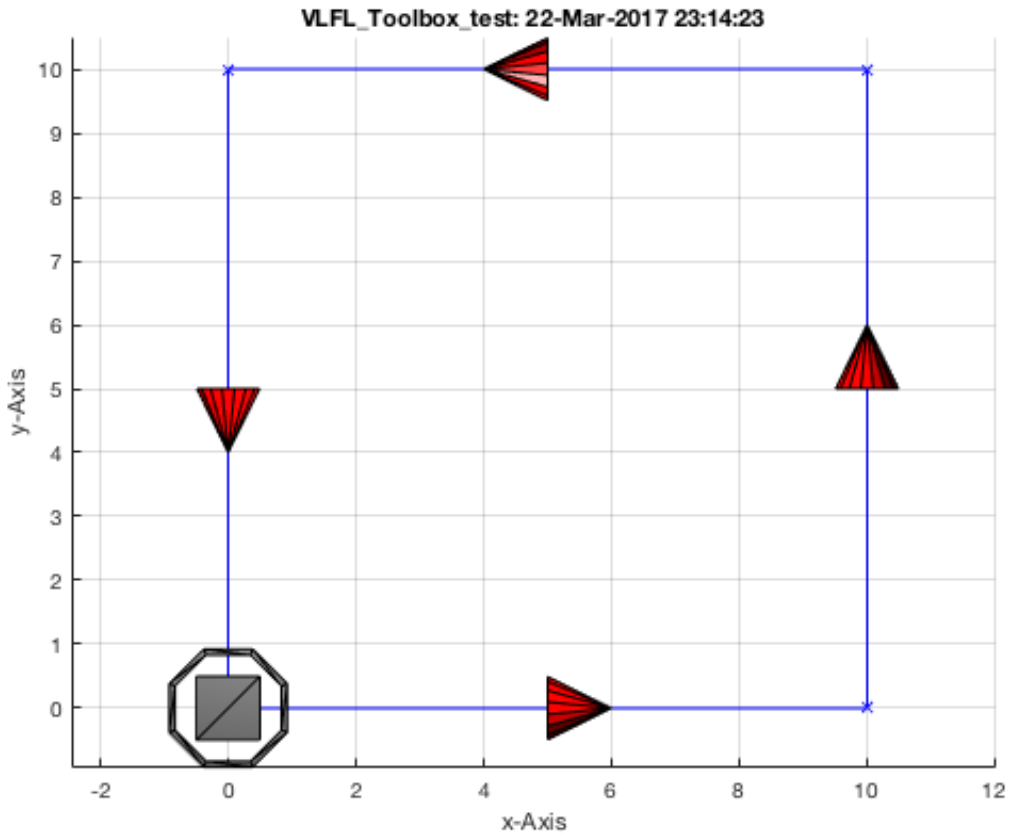

```
SGfigure; PLplot([0 0;10 0;10 10; 0 10],'bx-',1,1);          % Vector plot of point list
```



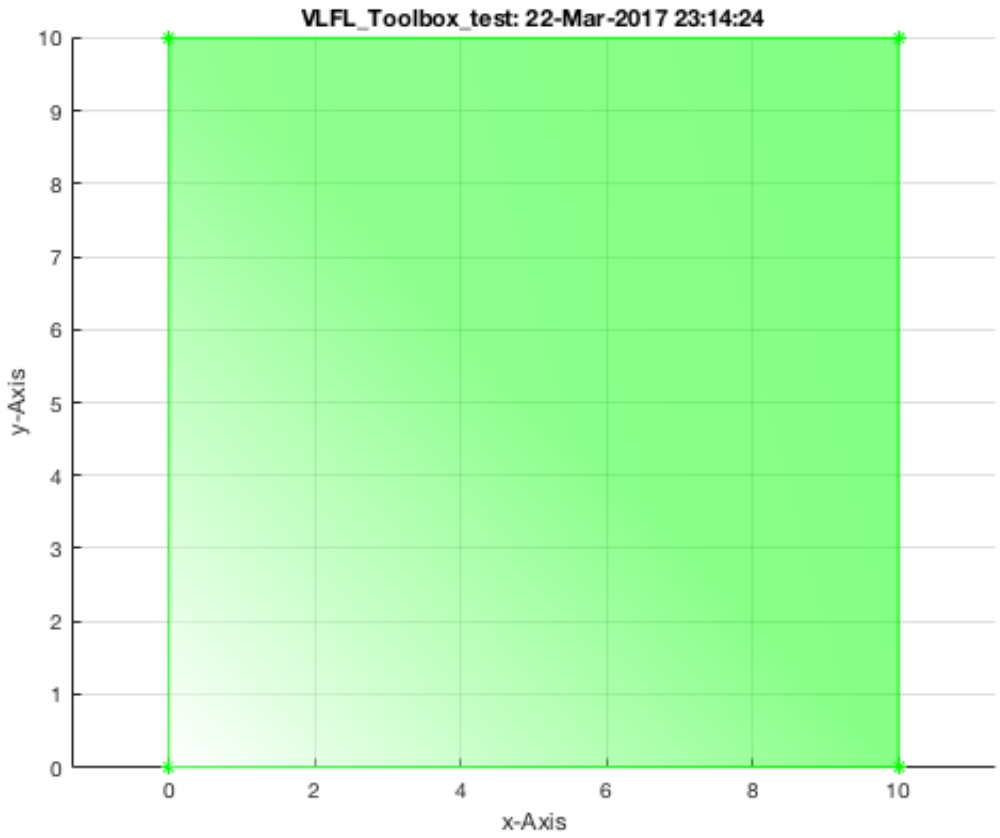

```
SGfigure; PLplot([0 0;10 0;10 10; 0 10],'bx-',1,'',0.5);    % Surface enlosed by point list
```




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

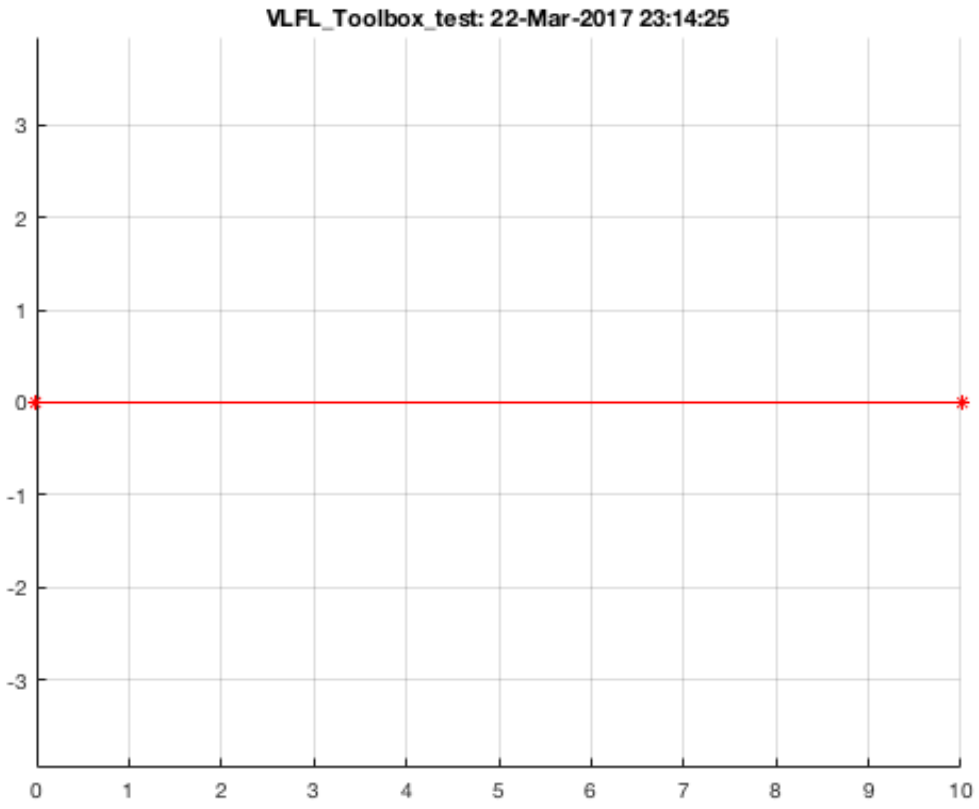



```
SGfigure; CPLfaceplot([0 0;10 0;10 10; 0 10],'g*-',1,0.5);  % Plotting closed polygon surfa
ces
```

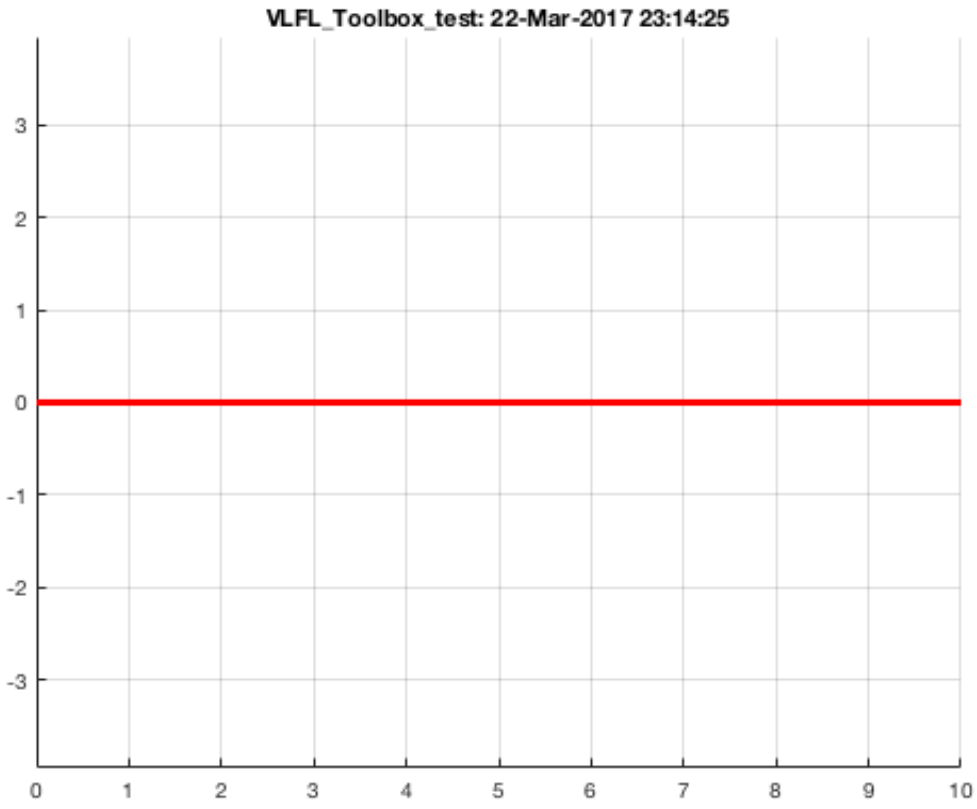


2. Plotting lines

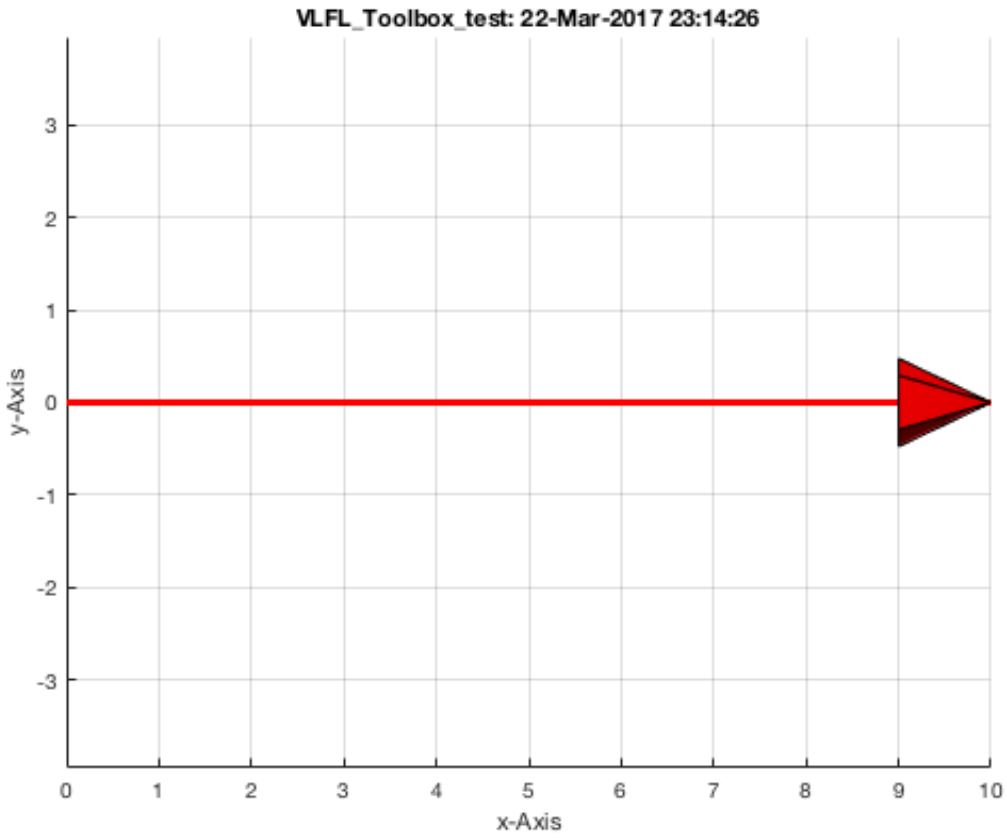
```
SGfigure; lplot([0 0],[10 0],'r*-');
```



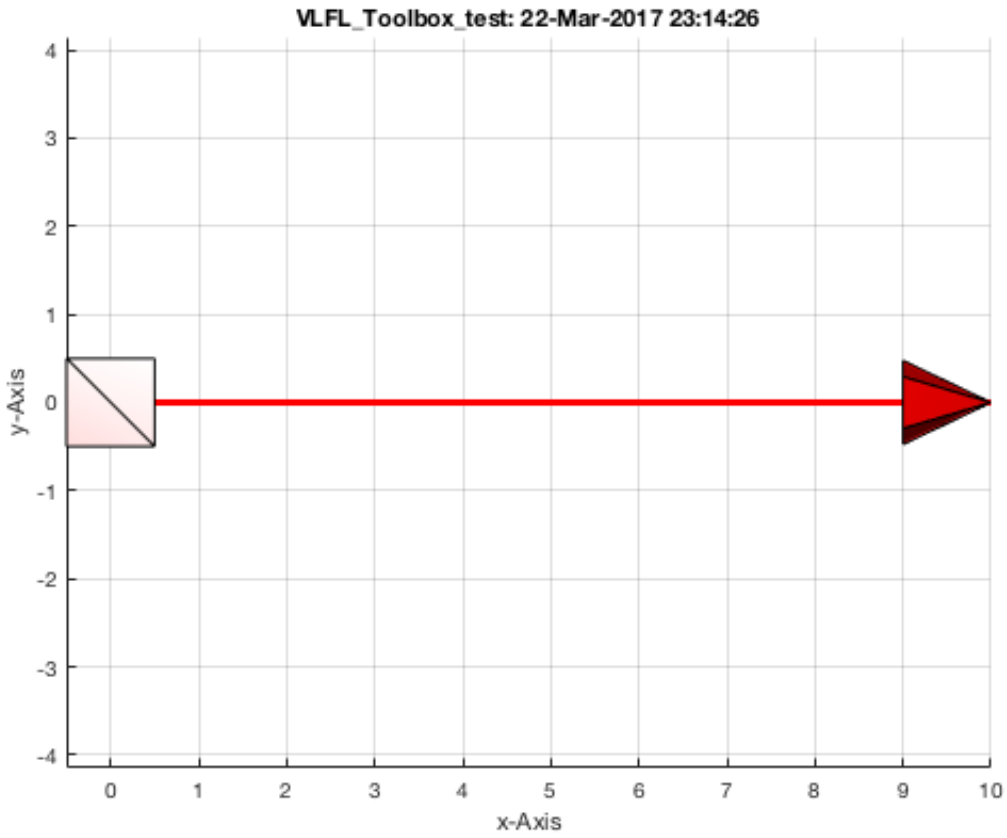

```
SGfigure; lplot([0 0],[10 0],'r-',3);
```




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




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




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

```
Error using plot3
Not enough input arguments.

Error in slplot

Error in VLFL_EXP30 (line 81)
SGfigure; slplot([0 0],[10 0],'r-',3,1,1);
```

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
```

Published with MATLAB® R2016b