Geogebra

Web:
http://www.geogebra.org

Quick start guide:
http://www.geogebra.org/help/geogebraquickstart_en.pdf

Advanced guide
http://www.geogebra.org/book/intro-en.pdf

On-line manual
http://wiki.geogebra.org/en/Manual:Main_Page

Geogebra

1: Think of a constructive sequence

2: Prepare list of geometric parameters

3: Introduce parameters and sequence in Geogebra

Geogebra

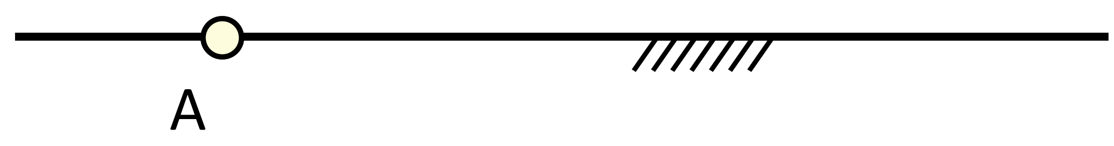
1: Think of a constructive sequence

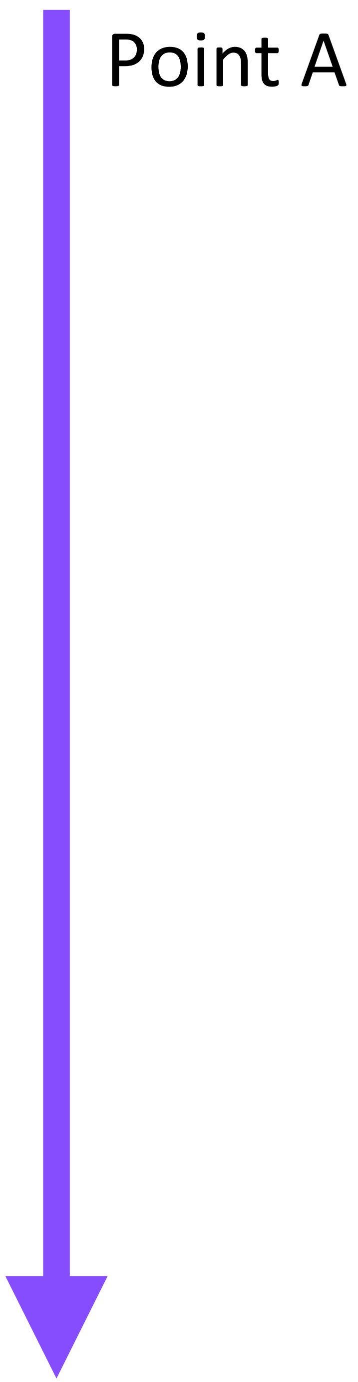
2: Prepare list of geometric parameters

3: Introduce parameters and sequence in Geogebra

Geogebra

1: Think of a constructive sequence



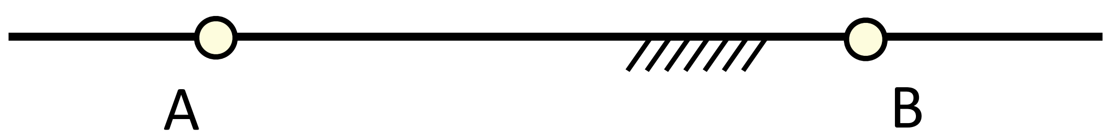


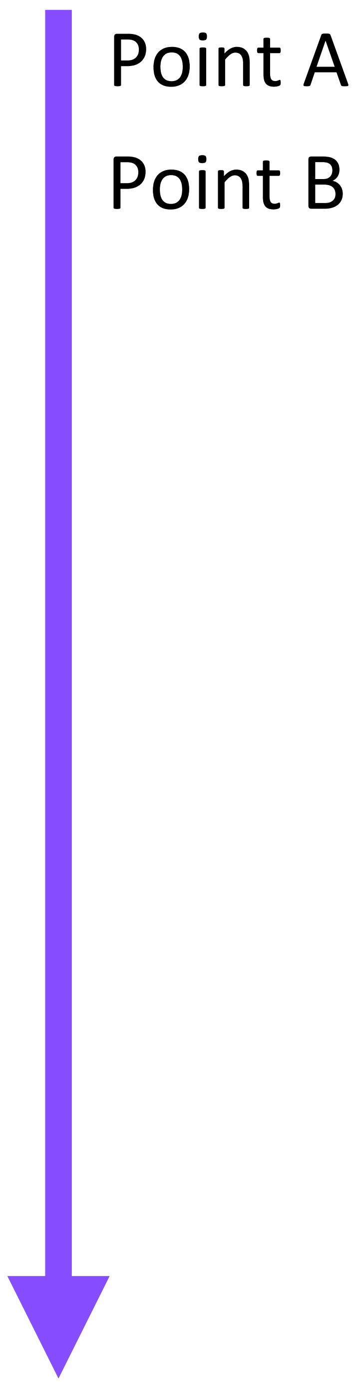
2: Prepare list of geometric parameters

3: Introduce parameters and sequence in Geogebra

Geogebra

1: Think of a constructive sequence





2: Prepare list of geometric parameters

3: Introduce parameters and sequence in Geogebra

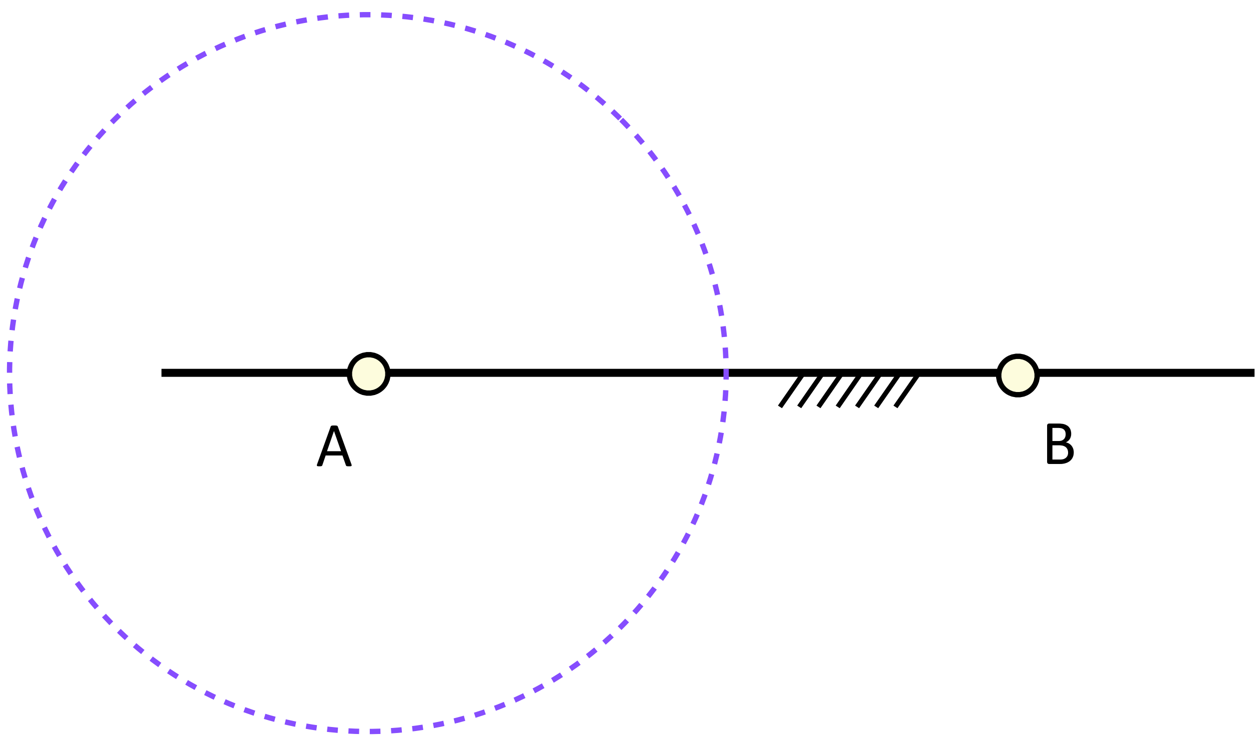
Geogebra

1: Think of a constructive sequence

Point A

Point B

Circle C_A

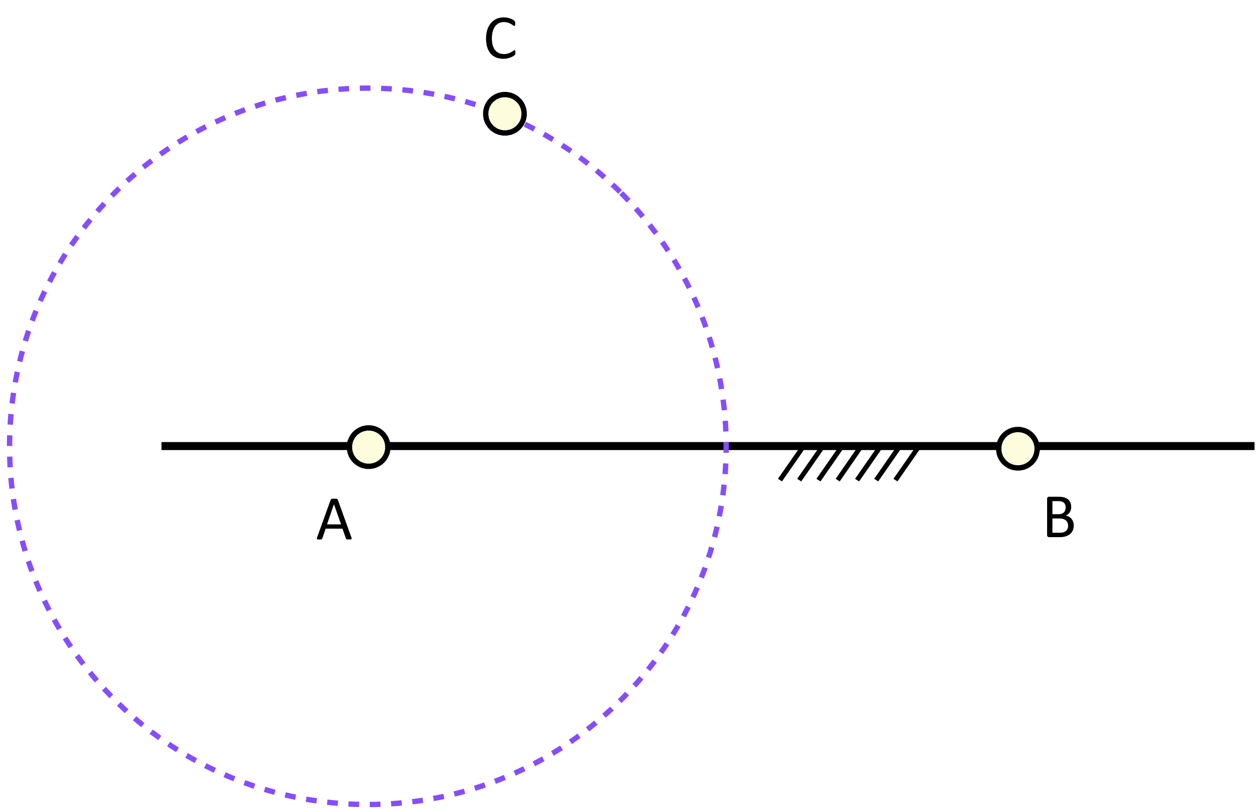


2: Prepare list of geometric parameters

3: Introduce parameters and sequence in Geogebra

Geogebra

1: Think of a constructive sequence



Point A

Point B

Circle C_A

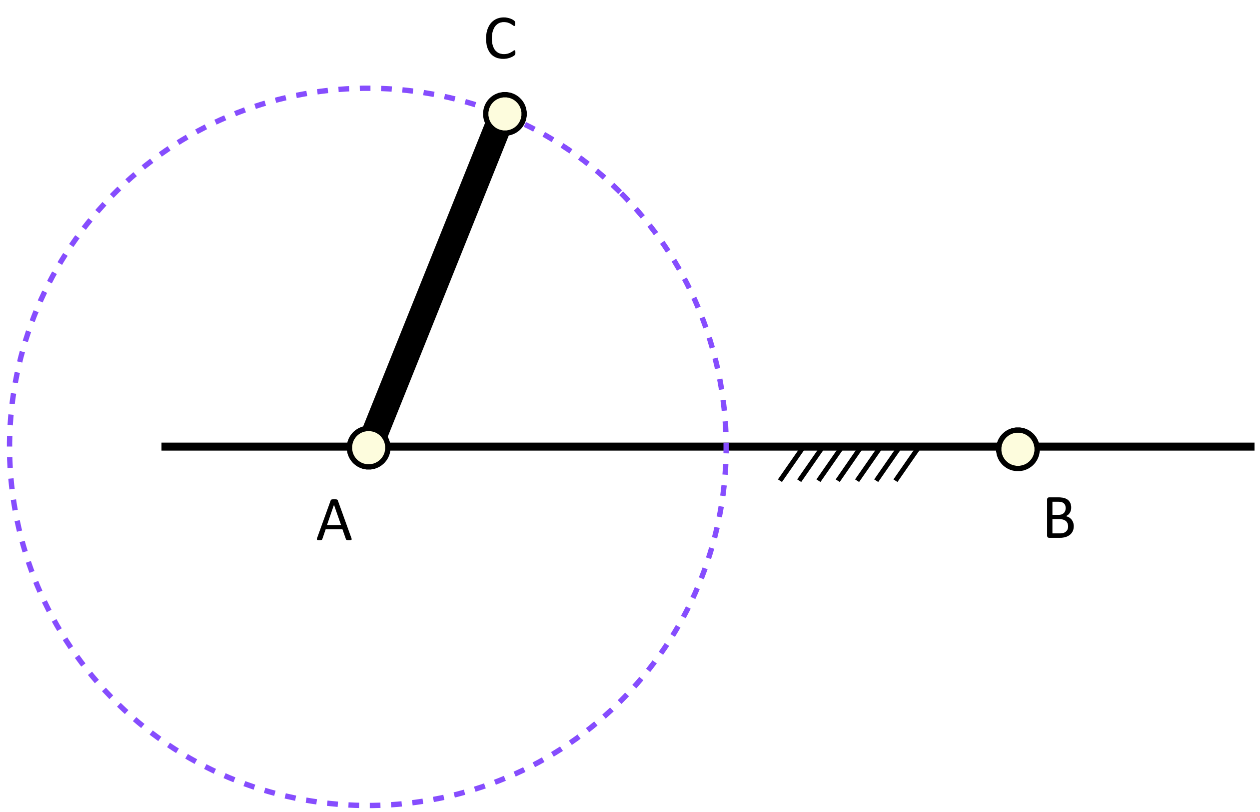
Point C on C_A

2: Prepare list of geometric parameters

3: Introduce parameters and sequence in Geogebra

Geogebra

1: Think of a constructive sequence



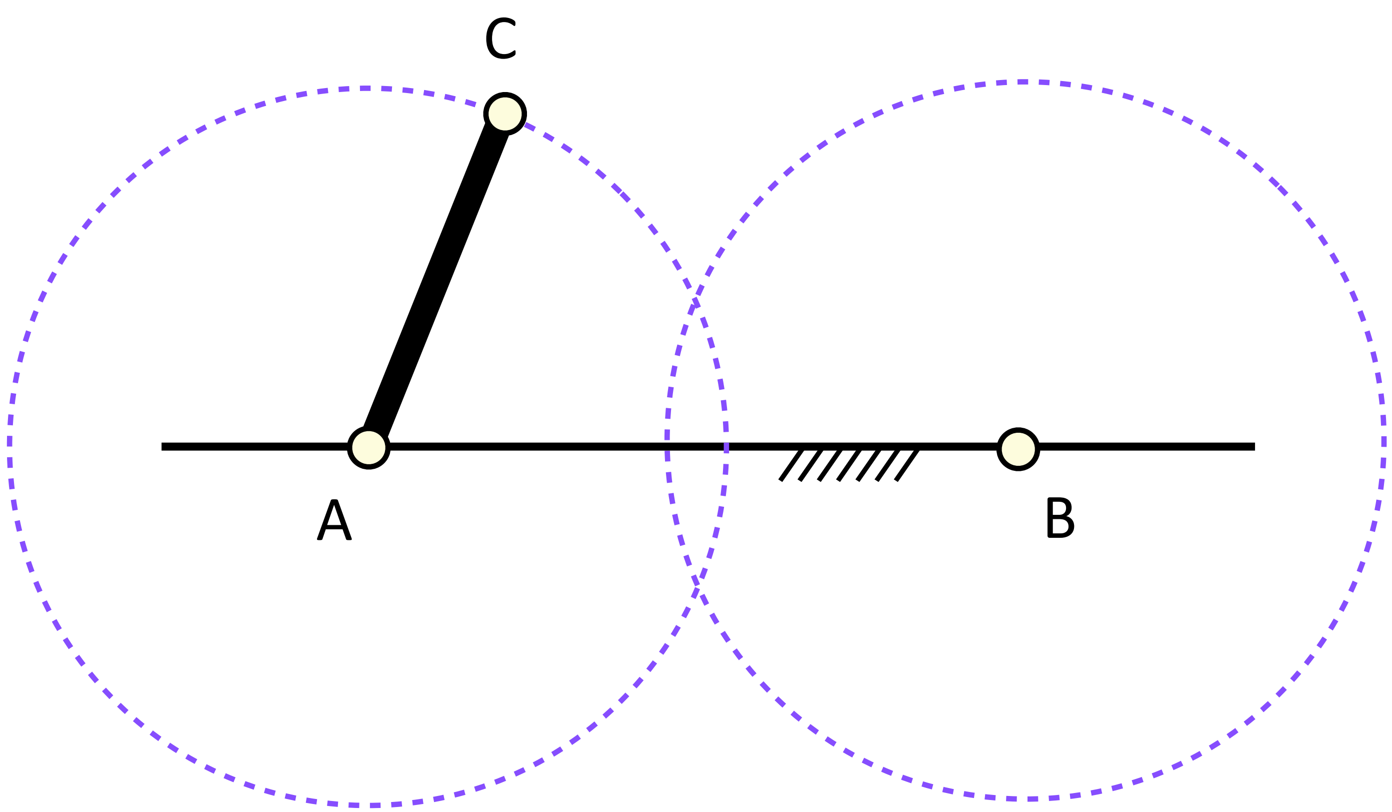
- Point A
- Point B
- Circle C_A
- Point C on C_A
- Segment AC

2: Prepare list of geometric parameters

3: Introduce parameters and sequence in Geogebra

Geogebra

1: Think of a constructive sequence



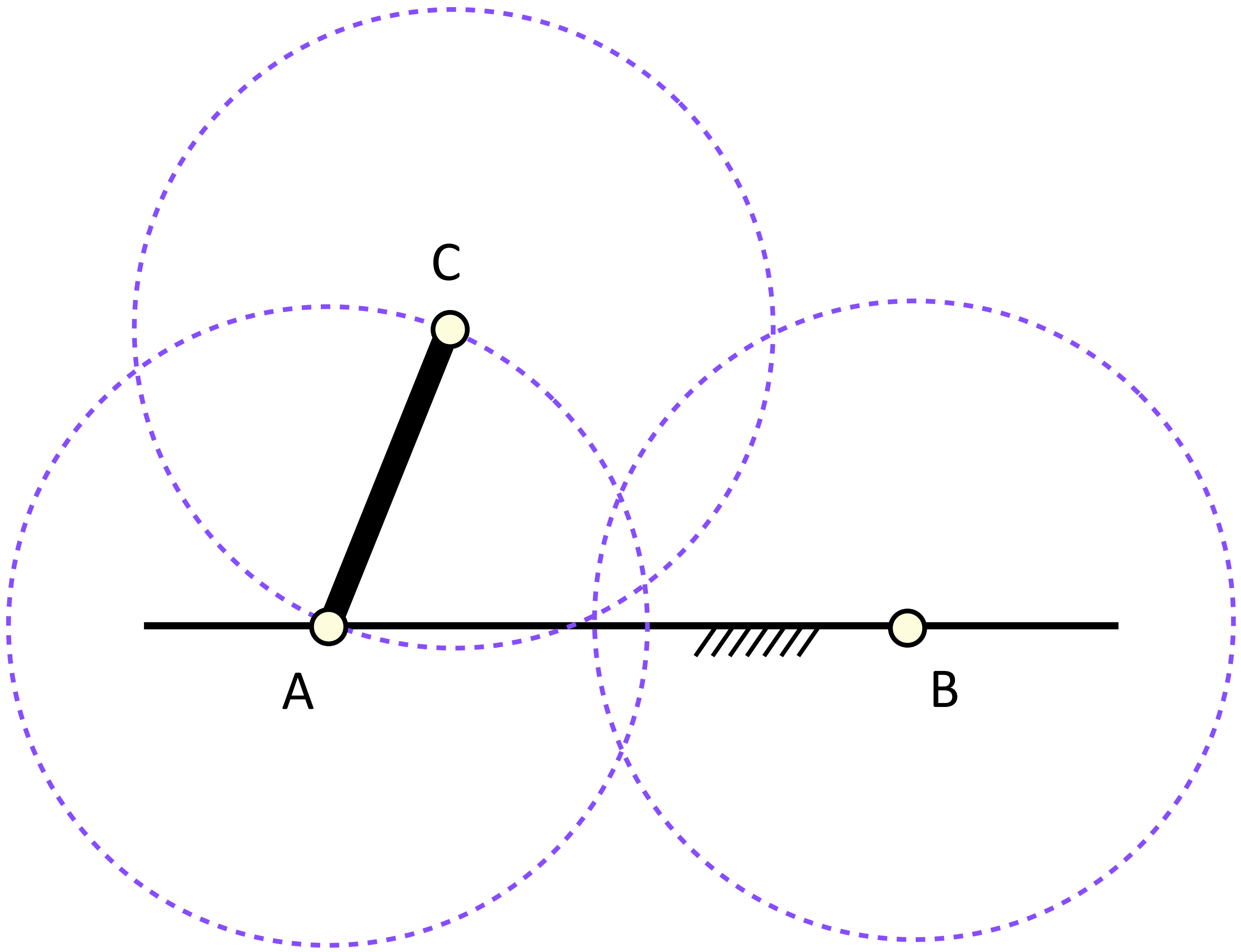
- Point A
- Point B
- Circle C_A
- Point C on C_A
- Segment AC
- Circle C_B

2: Prepare list of geometric parameters

3: Introduce parameters and sequence in Geogebra

Geogebra

1: Think of a constructive sequence



Point A

Point B

Circle C_A

Point C on C_A

Segment AC

Circle C_B

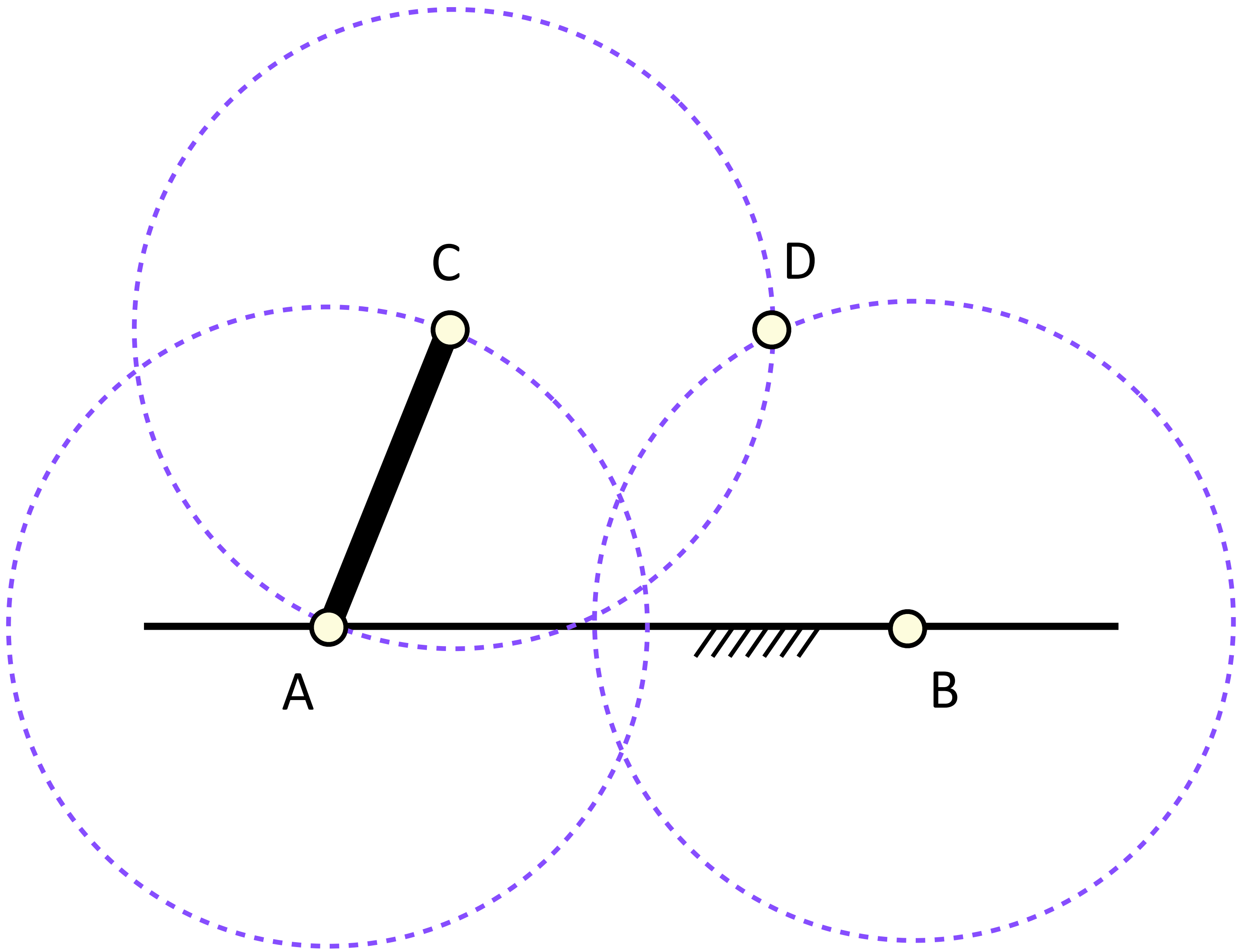
Circle C_C

2: Prepare list of geometric parameters

3: Introduce parameters and sequence in Geogebra

Geogebra

1: Think of a constructive sequence



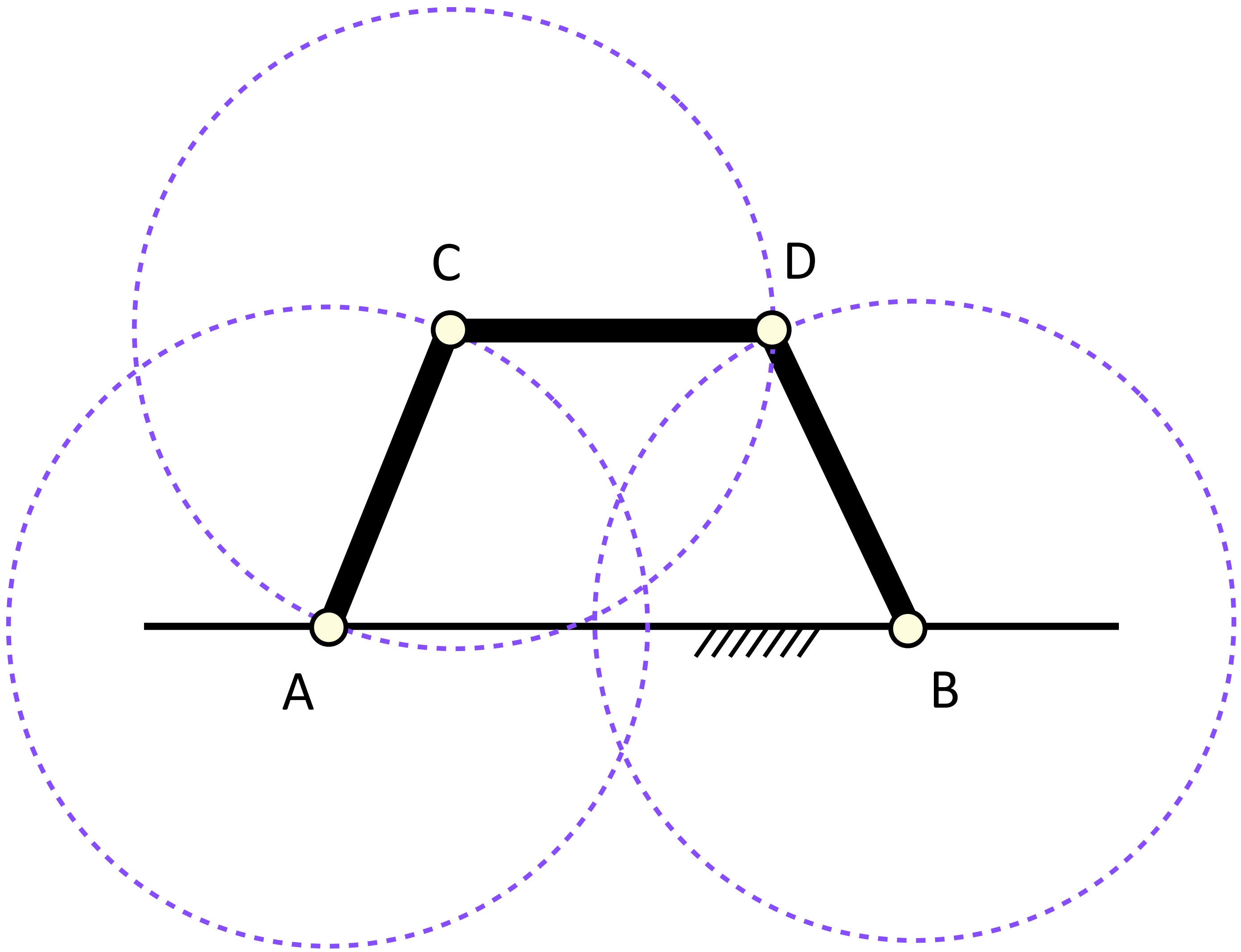
- Point A
- Point B
- Circle C_A
- Point C on C_A
- Segment AC
- Circle C_B
- Circle C_C
- Point D at $C_A \cap C_B$

2: Prepare list of geometric parameters

3: Introduce parameters and sequence in Geogebra

Geogebra

1: Think of a constructive sequence



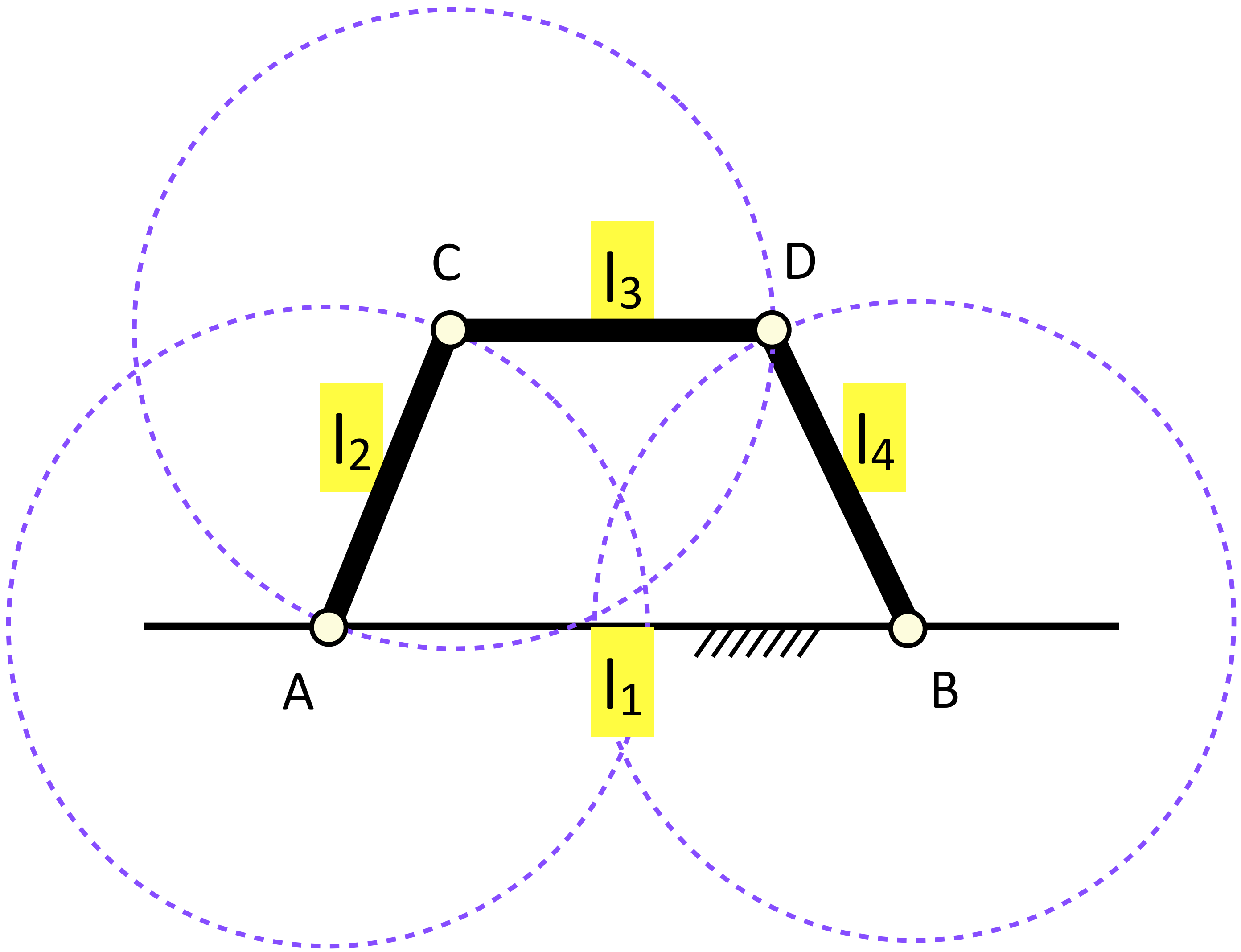
- Point A
- Point B
- Circle C_A
- Point C on C_A
- Segment AC
- Circle C_B
- Circle C_C
- Point D at $C_A \cap C_B$
- Segment BD
- Segment CD

2: Prepare list of geometric parameters

3: Introduce parameters and sequence in Geogebra

Geogebra

1: Think of a constructive sequence



- Point A
- Point B
- Circle C_A
- Point C on C_A
- Segment AC
- Circle C_B
- Circle C_C
- Point D at $C_A \cap C_B$
- Segment BD
- Segment CD

2: Prepare list of geometric parameters

$l_1 = 8$ $l_2 = 3.5$ $l_3 = 7.5$ $l_4 = 5.2$

3: Introduce parameters and sequence in Geogebra

Geogebra
syntax

```
l_1 = 8
l_2 = 3.5
l_3 = 7.5
l_4 = 5.2
A = (0,0)
B = (l_1,0)
C_A = Circle[A,l_2]
C = Point[C_A]
S_{AC} = Segment[A,C]
C_B = Circle[B,l_4]
C_C = Circle[C,l_3]
D = Intersect[C_B,C_C]
S_{BD} = Segment[B,D]
S_{CD} = Segment[C,D]
```

Geogebra
syntax

```
l_1 = 8
l_2 = 3.5
l_3 = 7.5
l_4 = 5.2
A = (0,0)
B = (l_1,0)
C_A = Circle[A,l_2]
C = Point[C_A]
S_{AC} = Segment[A,C]
C_B = Circle[B,l_4]
C_C = Circle[C,l_3]
D = Intersect[C_B,C_C]
S_{BD} = Segment[B,D]
S_{CD} = Segment[C,D]
```

Bilateration

Geogebra
syntax

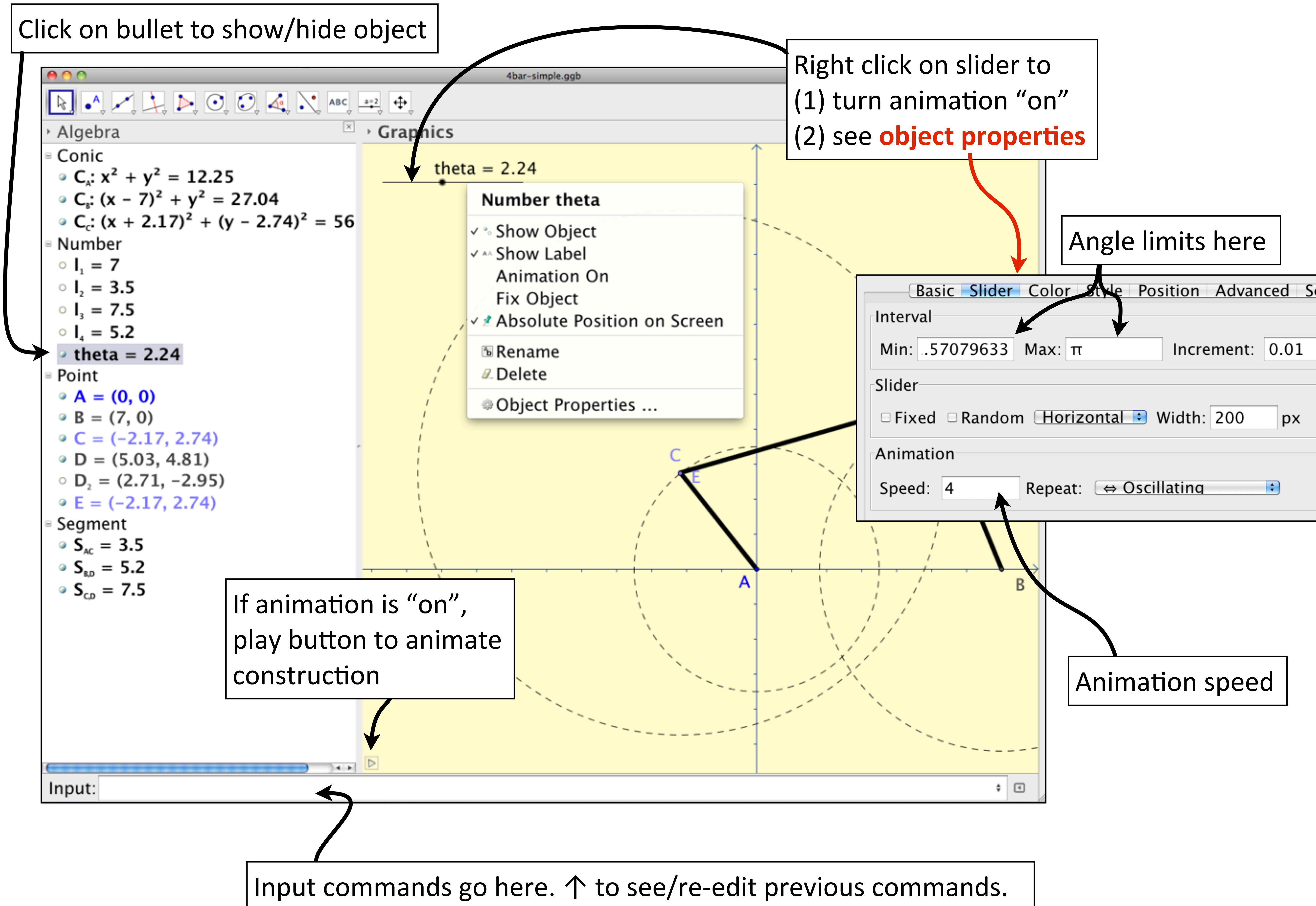
```
l_1 = 8
l_2 = 3.5
l_3 = 7.5
l_4 = 5.2
A = (0,0)
B = (l_1,0)
C_A = Circle[A,l_2]
C = Point[C_A]
S_{AC} = Segment[A,C]
C_B = Circle[B,l_4]
C_C = Circle[C,l_3]
D = Intersect[C_B,C_C]
S_{BD} = Segment[B,D]
S_{CD} = Segment[C,D]
```

Alternative

```
theta = pi/2
C = l_2*(cos(theta),sin(theta))
```

↓

Allows the setting of
(1) a control slider on theta
(2) joint angle limits in the slider


Click on bullet to show/hide object
Right click on slider to
(1) turn animation "on"
(2) see object properties
Angle limits here
Animation speed
If animation is "on", play button to animate construction
Input commands go here. ↑ to see/re-edit previous commands.
4bar-simple.ggb
Algebra
Graphics
Conic
Number
Point
Segment
theta = 2.24
Number theta
Show Object
Show Label
Animation On
Fix Object
Absolute Position on Screen
Rename
Delete
Object Properties ...
Basic
Slider
Color
Style
Position
Advanced
Interval
Min:
Max:
Increment:
0.01
Fixed
Random
Horizontal
Width:
200
px
Animation
Speed:
4
Repeat:
Oscillating
Input:

CUIK

Web

http://www.iri.upc.edu/cuik

Overview paper and start guide

http://goo.gl/Y91ach

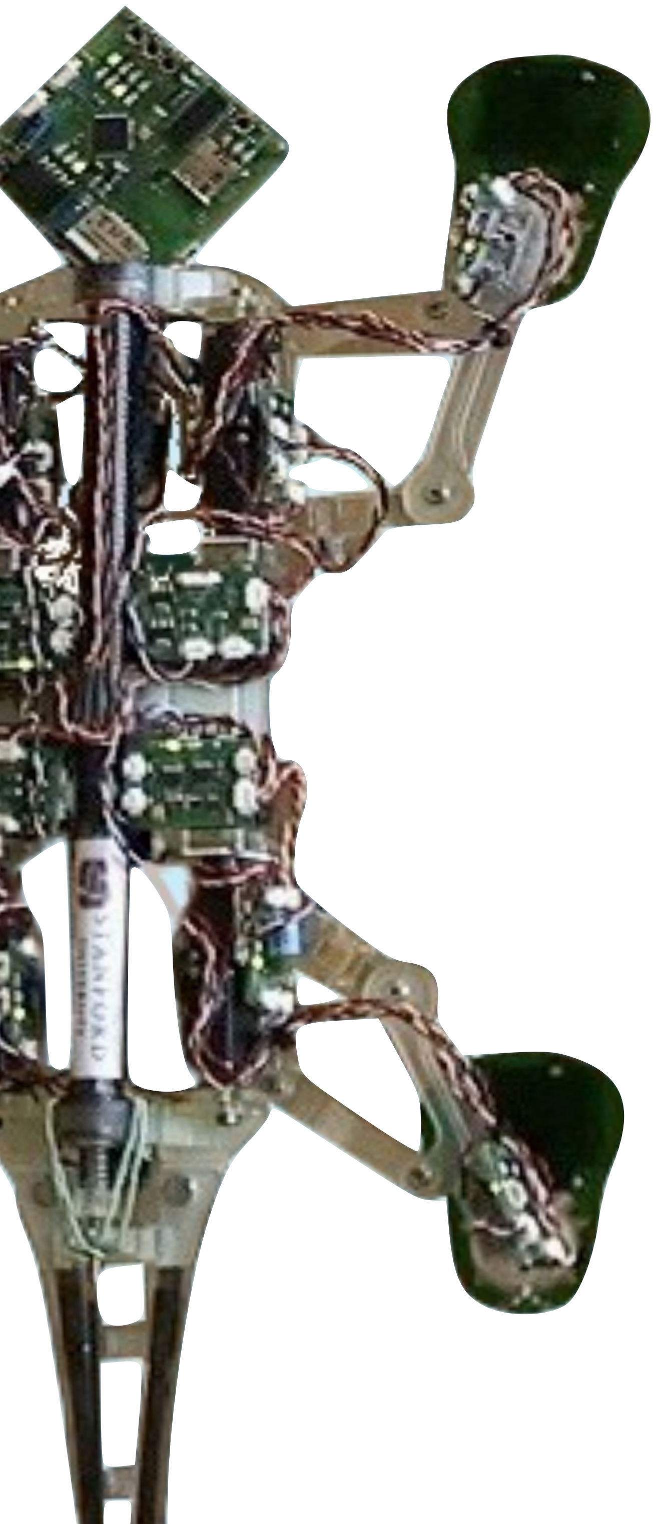
Documentation

http://goo.gl/M6jqte

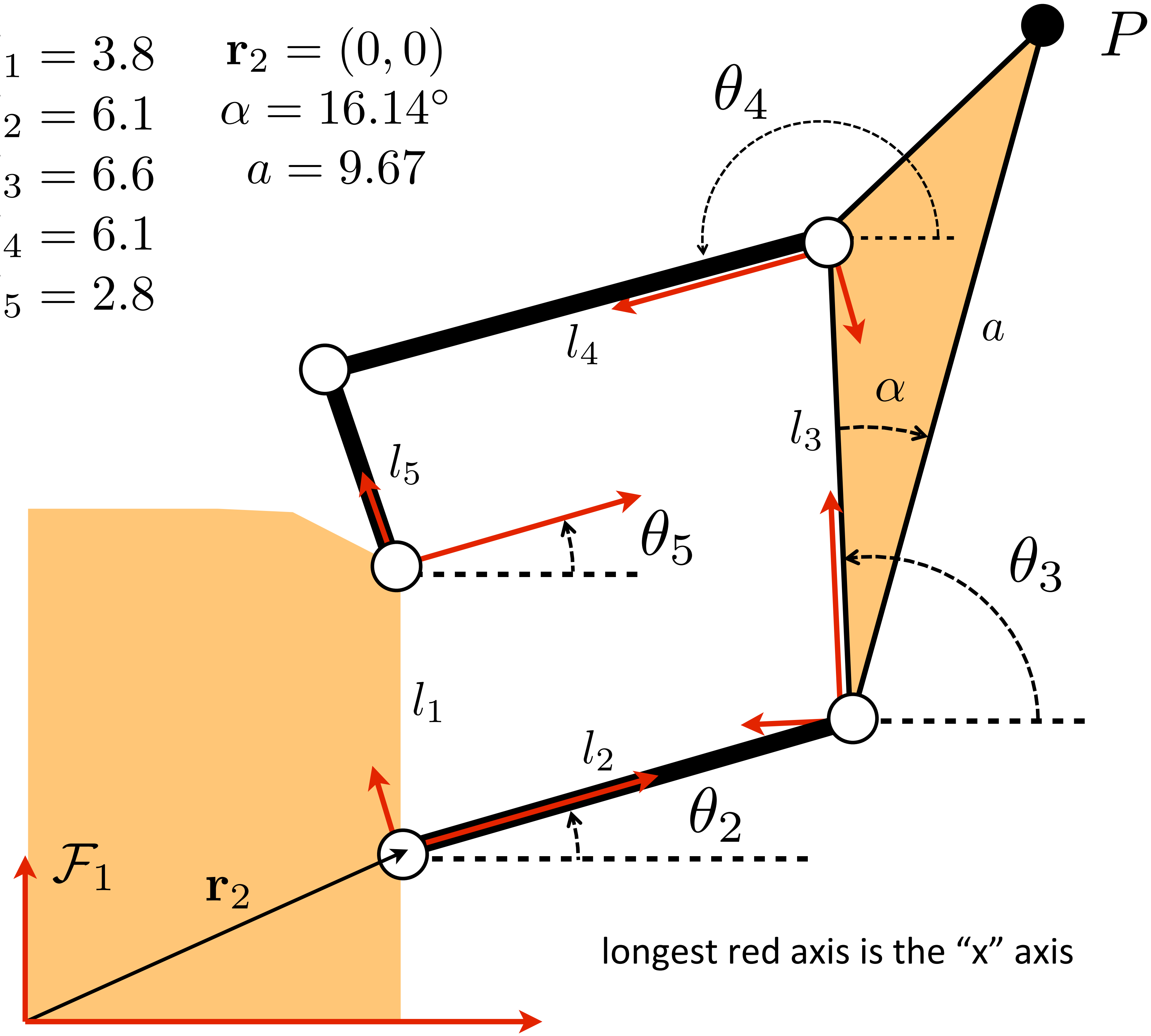
Problem 4

Write the loop equations of Stickybot's leg

Add extra equations to include the (x, y) coords of point P

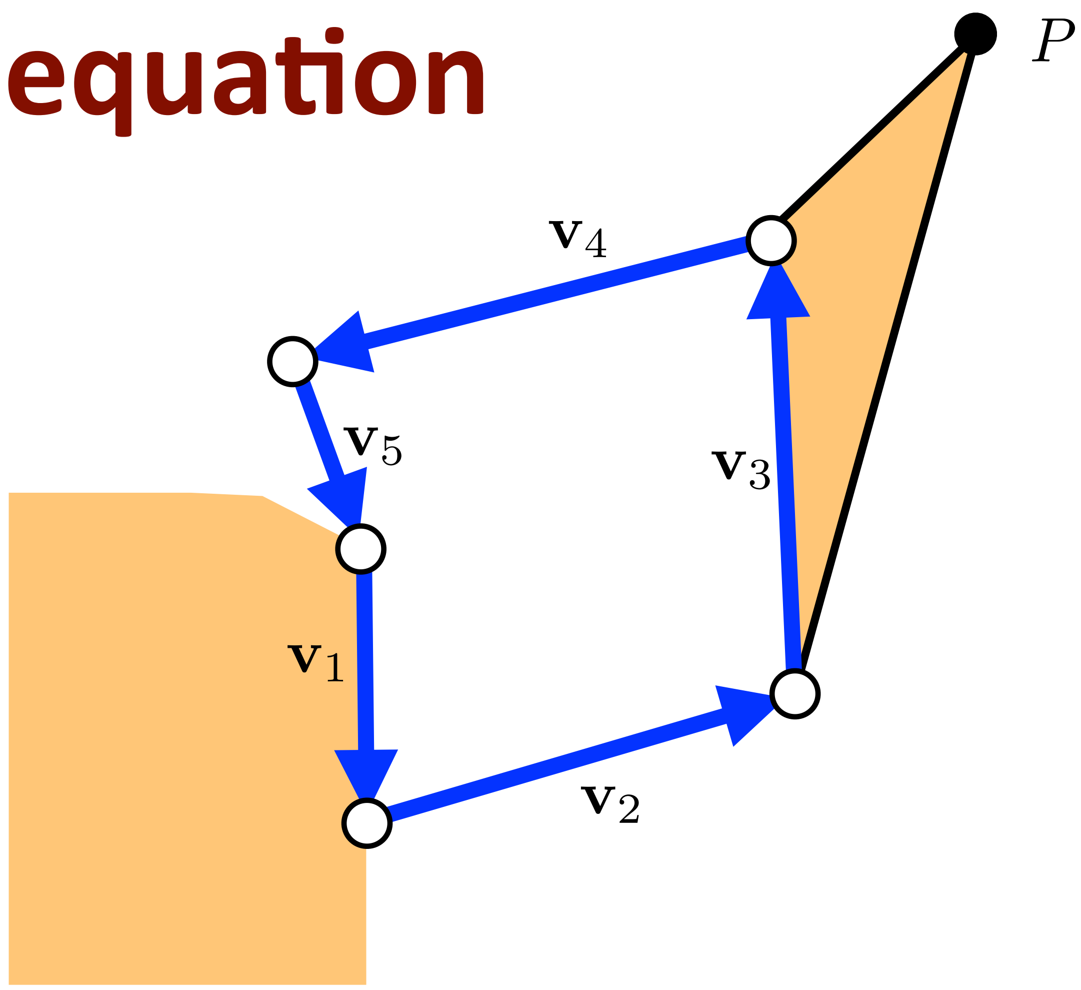


$$l_1 = 3.8 \qquad \mathbf{r}_2 = (0,0)$$
$$l_2 = 6.1 \qquad \alpha = 16.14°$$
$$l_3 = 6.6 \qquad a = 9.67$$
$$l_4 = 6.1$$
$$l_5 = 2.8$$



longest red axis is the "x" axis

Writing the loop equation



General form of the loop equation

$$\sum_{i=1}^{5} \mathbf{R}_i \cdot \mathbf{v}_i = \mathbf{0}$$

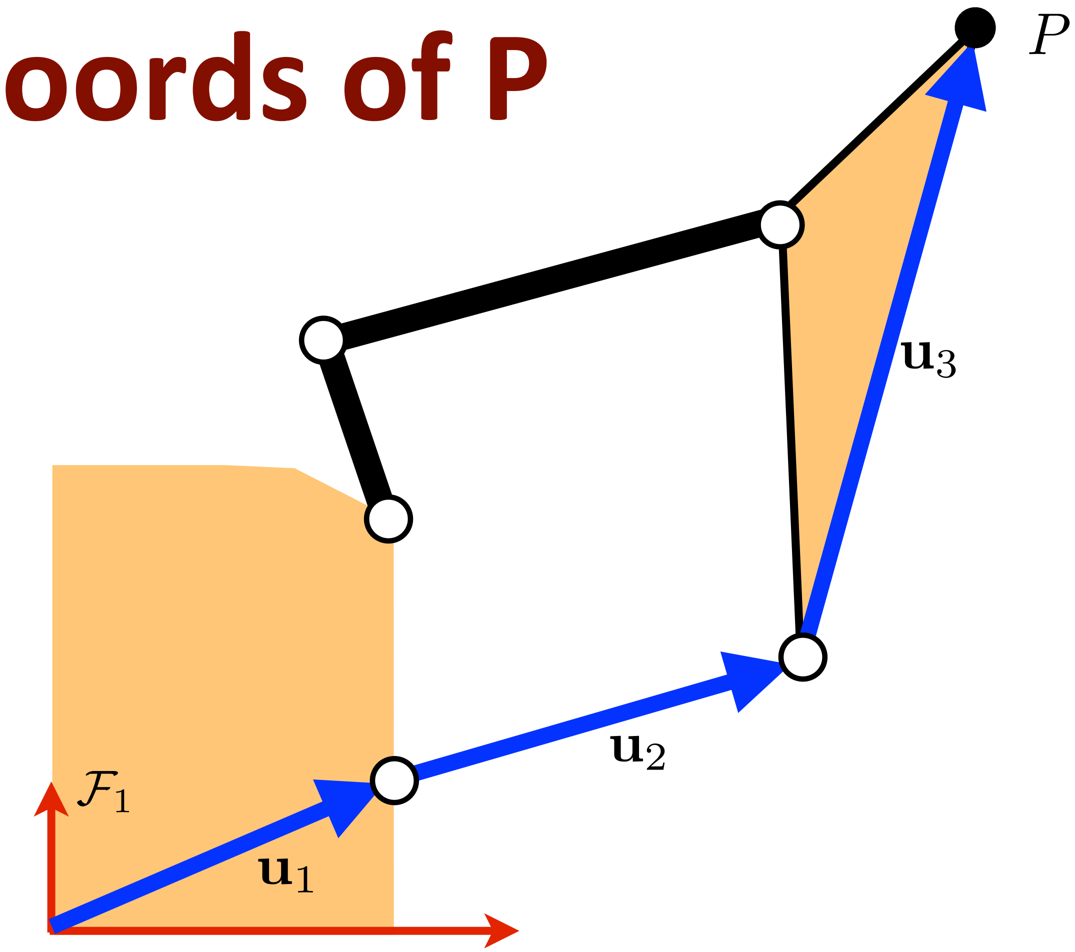
In the given reference frames:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -l_1 \end{bmatrix} + \begin{bmatrix} c_2 & -s_2 \\ s_2 & c_2 \end{bmatrix} \begin{bmatrix} l_2 \\ 0 \end{bmatrix} + \begin{bmatrix} c_3 & -s_3 \\ s_3 & c_3 \end{bmatrix} \begin{bmatrix} l_3 \\ 0 \end{bmatrix} + \begin{bmatrix} c_4 & -s_4 \\ s_4 & c_4 \end{bmatrix} \begin{bmatrix} l_4 \\ 0 \end{bmatrix} + \begin{bmatrix} c_5 & -s_5 \\ s_5 & c_5 \end{bmatrix} \begin{bmatrix} 0 \\ -l_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

In scalar form:

$$\left.\begin{aligned} l_2\, c_2 + l_3\, c_3 + l_4\, c_4 + l_5\, s_5 &= 0 \\ -l_1 + l_2\, s_2 + l_3\, s_3 + l_4\, s_4 - l_5\, c_5 &= 0 \end{aligned}\right\}$$

Introducing the coords of P



Vector sum to locate the position of P

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{u}_1 + \mathbf{u}_2 + \mathbf{u}_3$$

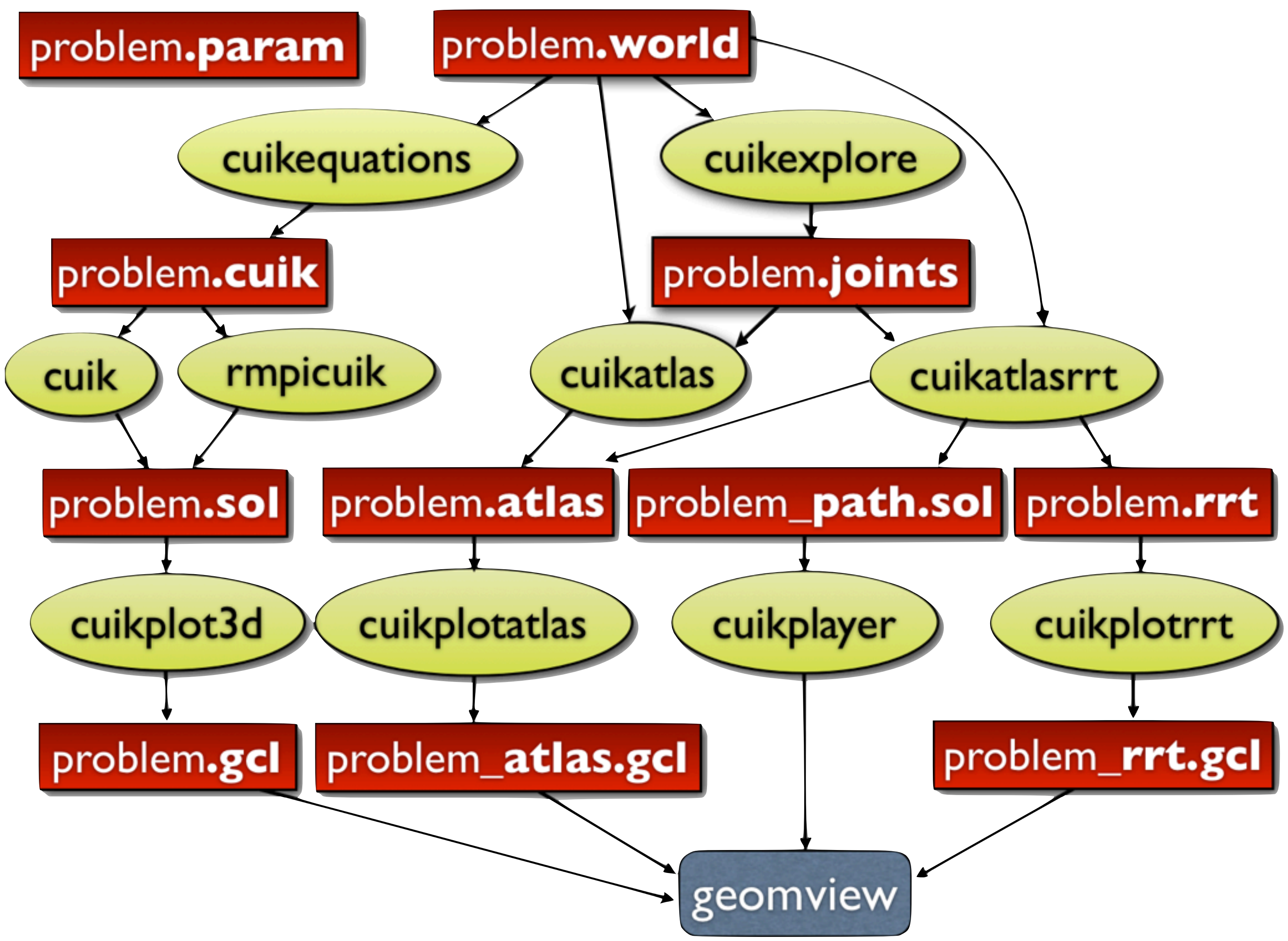
In the given reference frames:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{r}_2 + l_2 \begin{bmatrix} c_2 \\ s_2 \end{bmatrix} + \begin{bmatrix} c_3 & -s_3 \\ s_3 & c_3 \end{bmatrix} \begin{bmatrix} a\cos\alpha \\ -a\sin\alpha \end{bmatrix}$$

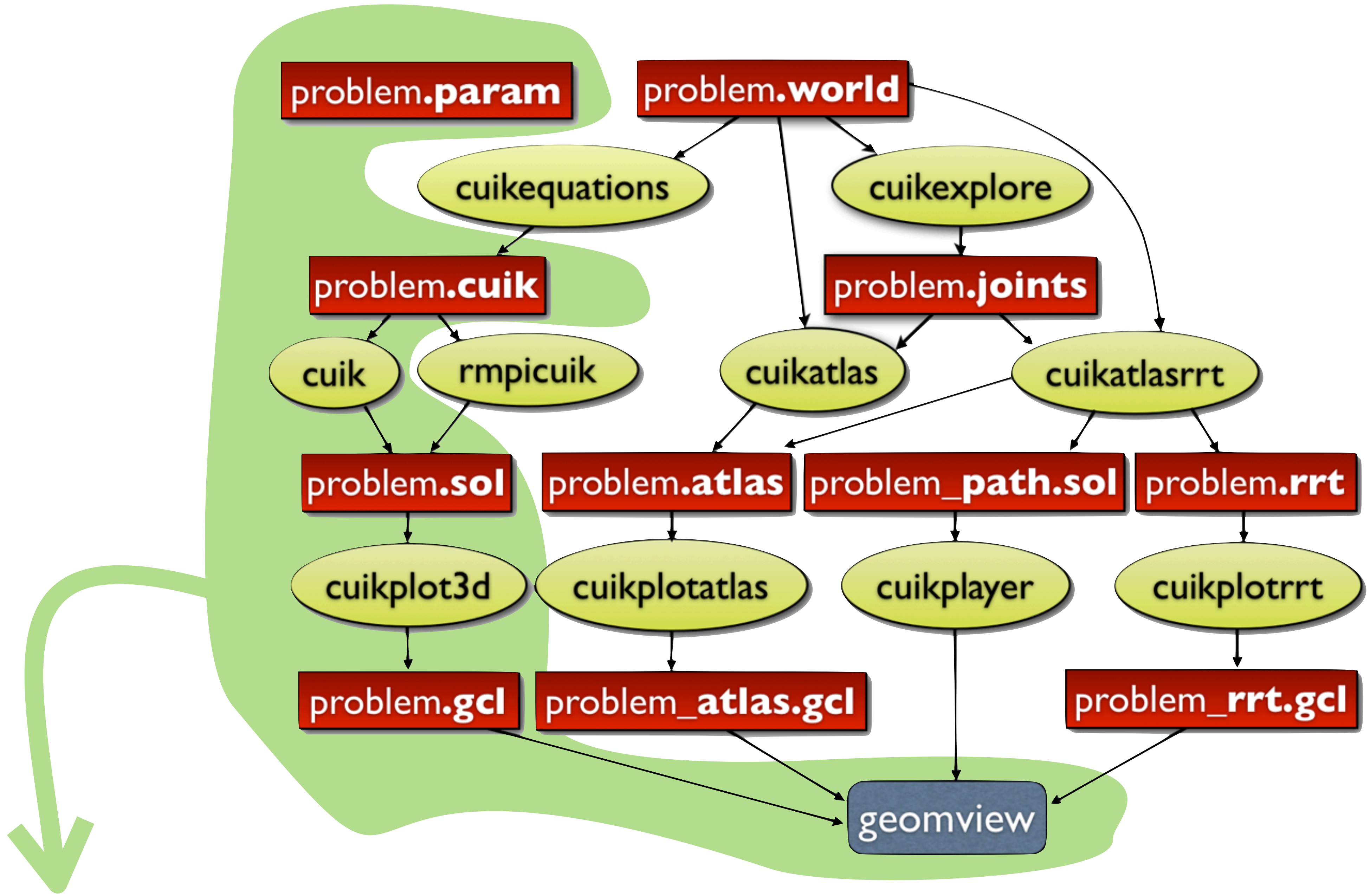
In scalar form and using $\mathbf{r}_2 = [0, 0]^T$:

$$\left. \begin{aligned} x &= l_2\, c_2 + a\cos\alpha\; c_3 + a\sin\alpha\; s_3 \\ y &= l_2\, s_2 + a\cos\alpha\; s_3 - a\sin\alpha\; c_3 \end{aligned} \right\}$$

CUIK's problem solving cycle



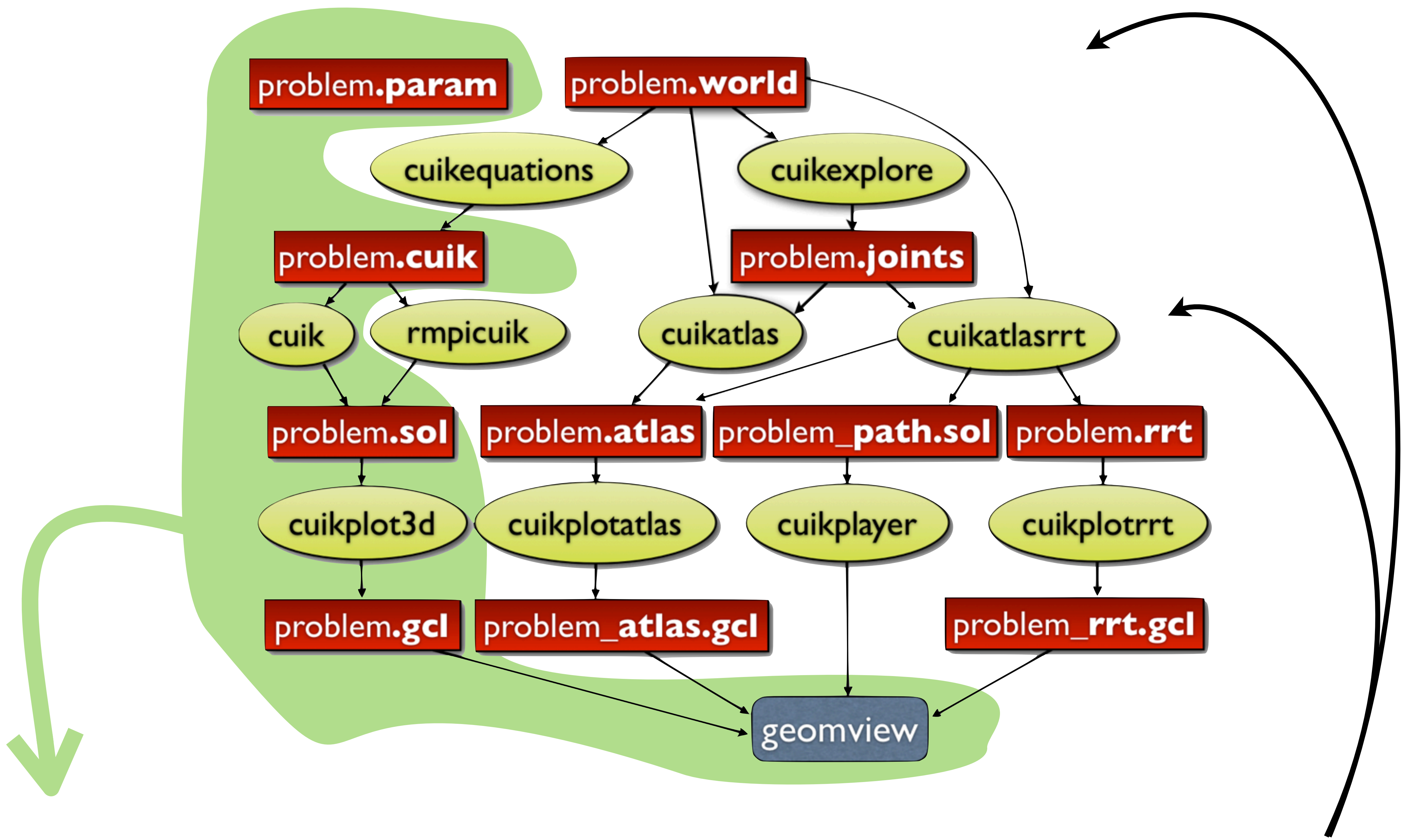
CUIK's problem solving cycle



Used in this course

(explained next)

CUIK's problem solving cycle



Used in this course
(explained next)

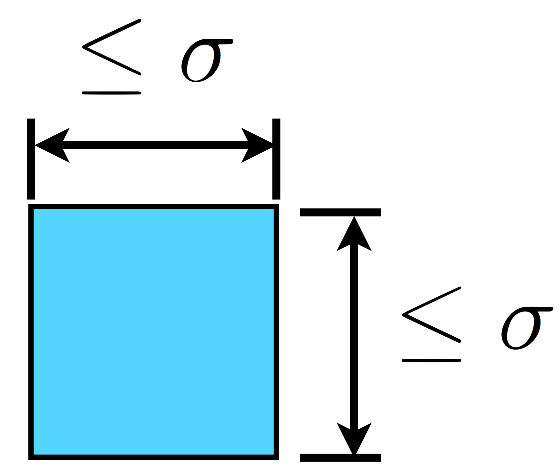
http://www.iri.upc.edu/cuik
for an overview of these

stickybot.param

```
%%%% PARAMETERS %%%%

SIGMA = 0.1

RHO = 0.95

SMALL_SIGMA = 0.001
EPSILON = 1e-6
ERROR_SPLIT = TRUE
LR2TM_Q=0.25
LR2TM_S=0.1

SIMPLIFICATION_LEVEL = 1
```

Output size threshold σ

Box is a **solution box** if all side lengths are below σ



Box reduction threshold ρ

Repeat box reduction while

$$\frac{\text{Volume}(\text{Reduced box})}{\text{Volume}(\text{Original box})} < \rho$$

Afterwards, bisect the box

stickybot.cuik

```
[CONSTANTS]

l1:=3.8
l2:=6.1
l3:=6.6
l4:=6.1
l5:=2.8

a :=9.67
alpha := 16.14 * pi/180
calpha:=cos(alpha)
salpha:=sin(alpha)

r := l2+a

[SYSTEM VARS]

x:[-r,r]
y:[-r,r]
c2:[-1,1]
s2:[-1,1]
c3:[-1,1]
s3:[-1,1]
c4:[-1,1]
s4:[-1,1]
c5:[-1,1]
s5:[-1,1]
```

```
[SYSTEM EQS]

% Loop closure equations

l2*c2 + l3*c3 + l4*c4 + l5*s5 = 0;
-l1   + l2*s2 + l3*s3 + l4*s4 - l5*c5 = 0;

% Position of P

x = l2*c2 + a*calpha*c3+a*salpha*s3;
y = l2*s2 + a*calpha*s3-a*salpha*c3;

% Circle equations

s2^2 + c2^2 = 1;
s3^2 + c3^2 = 1;
s4^2 + c4^2 = 1;
s5^2 + c5^2 = 1;

% Lock the input to solve a FKP problem

c2 = cos(0);
s2 = sin(0);

c5 = cos(0);
s5 = sin(0);
```

stickybot.sol

generated with: ~/CuikSuite/bin/cuik stickybot

```
==========================================================================
Cuik executed in 1 processors
SOLUTIONS (to input file stickybot.cuik):

   1 (err:3.69268e-06 tm:0):{ 10  x[8.78811710513,8.78811819992] y[9.28885876005,9.28886002588]    ... }
   2 (err:1.54197e-07 tm:0):{ 10  x[-2.94877518079,-2.94877346147] y[3.41007203352,3.41007354778]  ... }

STATITISTICS FOR Cuik:

Volume of the search space: 15916.3
Volume of the solution space: 0
  Volume ratio:  0.00 %
Max solution diagonal: 2.30851e-06
Number of processors: 1
User time in process: 0.000000 seg (0.000000 min)
Box level information:
  N processed boxes :      3
  Max depth         :      2
  Types of boxes:
    N solution boxes:      2 (0) (66.67%)
    N empty boxes   :      0  ( 0.00%)
    N bisected boxes:      1  (33.33%)
  Box Reductions    :      4
  N Errors          :      0
==========================================================================
```

stickybot.sol

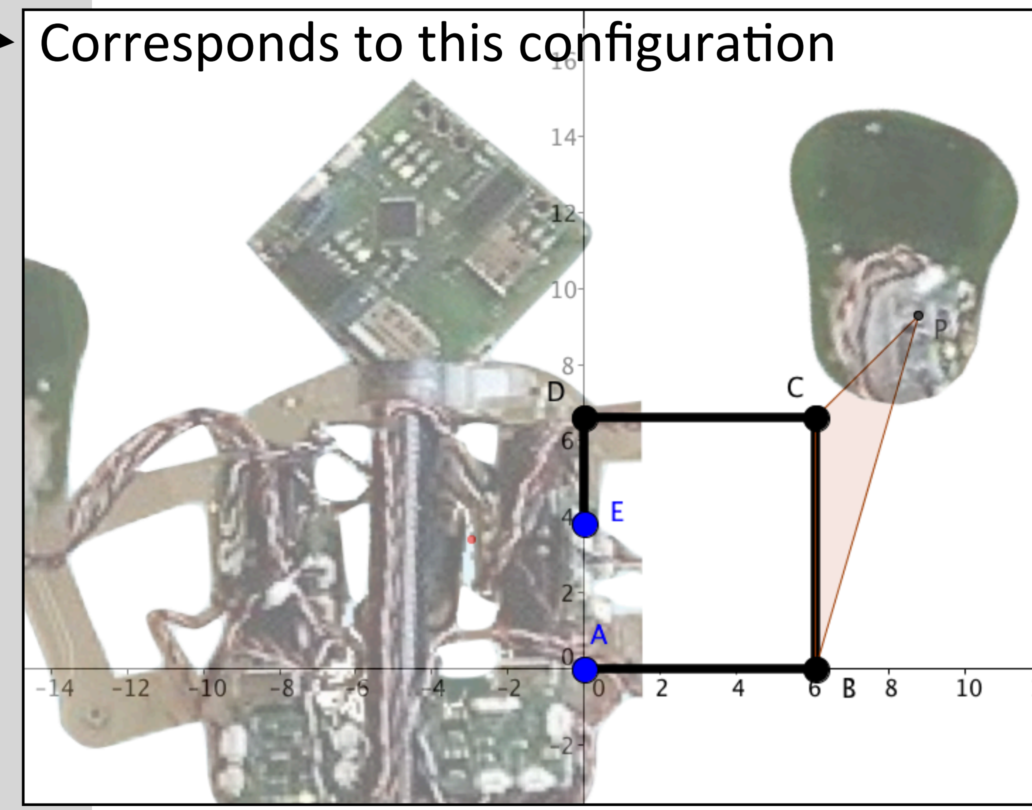
generated with: `~/CuikSuite/bin/cuik stickybot`

```
==========================================

ot.cuik):

 x[8.78811710513,8.78811819992] y[9.28885876005,9.28886002588]    ... }
 x[-2.94877518079,-2.94877346147] y[3.41007203352,3.41007354778]  ... }



6.3


06

eg (0.000000 min)




 (66.67%)
0.00%)
3.33%)


==========================================
```

stickybot.sol

generated with: ~/CuikSuite/bin/cuik stickybot

```
==========================================

ot.cuik):

 x[8.78811710513,8.78811819992] y[9.28885876005,9.28886002588]   ... }
 x[-2.94877518079,-2.94877346147] y[3.41007203352,3.41007354778]  ... }

6.3

06

eg (0.000000 min)

 (66.67%)
0.00%)
3.33%)

==========================================
```

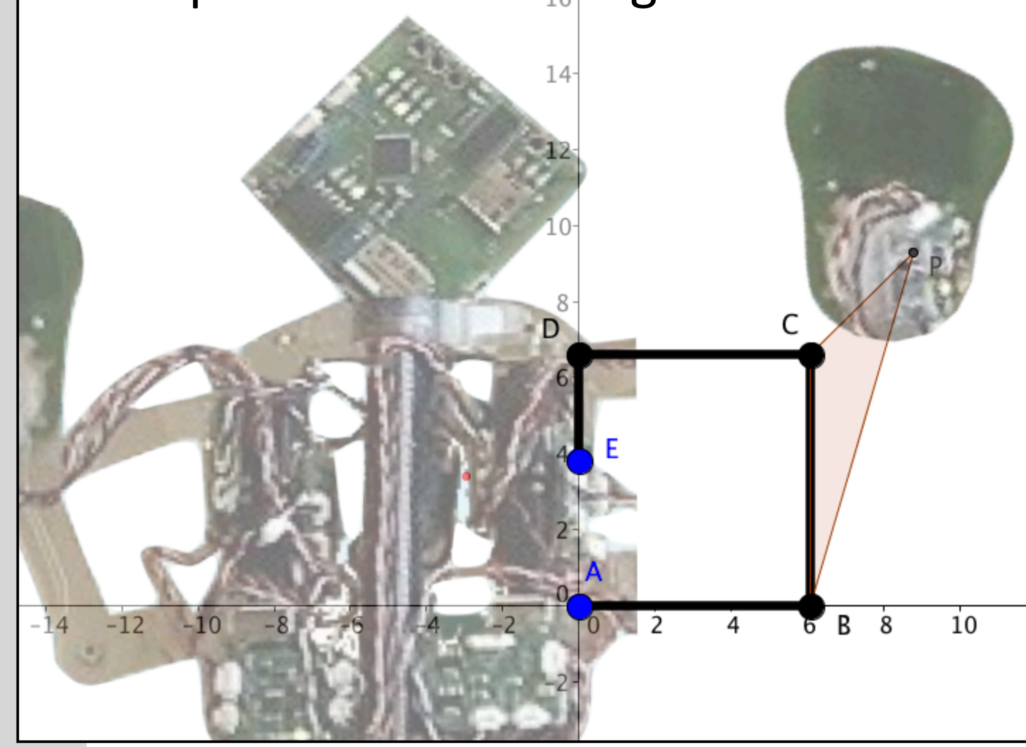


stickybot.sol

generated with: ~/CuikSuite/bin/cuik stickybot

```
==========================================

ot.cuik):

 x[8.78811710513,8.78811819992] y[9.28885876005,9.28886002588]   ... }
 x[-2.94877518079,-2.94877346147] y[3.41007203352,3.41007354778]  ... }


6.3


06

eg (0.000000 min)



 (66.67%)
0.00%)
3.33%)


==========================================
```

Corresponds to this configuration



Corresponds to
the second configuration
with triangle BCD flipped
relative to line BD

Simulate a motion path

Now edit **stickybot.cuik** and save it to **stickybot-path.cuik**

```
[CONSTANTS]

l1:=3.8
l2:=6.1
l3:=6.6
l4:=6.1
l5:=2.8

a :=9.67
alpha := 16.14 * pi/180
calpha:=cos(alpha)
salpha:=sin(alpha)

r := l2+a

[SYSTEM VARS]

x:[-r,r]
y:[-r,r]
c2:[-1,1]
s2:[-1,1]
c3:[-1,1]
s3:[-1,1]
c4:[-1,1]
s4:[-1,1]
c5:[-1,1]
s5:[-1,1]
```

```
[SYSTEM EQS]

% Loop closure equations

l2*c2 + l3*c3 + l4*c4 + l5*s5 = 0;
-l1   + l2*s2 + l3*s3 + l4*s4 - l5*c5 = 0;

% Position of P

x = l2*c2 + a*calpha*c3+a*salpha*s3;
y = l2*s2 + a*calpha*s3-a*salpha*c3;

% Circle equations

s2^2 + c2^2 = 1;
s3^2 + c3^2 = 1;
s4^2 + c4^2 = 1;
s5^2 + c5^2 = 1;

% Lock the input to solve a FKP problem

c2 = cos(0);
s2 = sin(0);

c5 = cos(0);
s5 = sin(0);
```

(The lines `c2 = cos(0);` and `s2 = sin(0);` are crossed out.)

Remove to free angle θ_2

cuikplot3d & geomview

Solve the equations

```
> cuik stickybot-path
...
<9.14993e-14 0.26329>[13]  -> <7.79572e-16 0.136254>b[7]
<3.66936e-16 0.136254>[14]  -> <1.71125e-18 0.0640773>s
<4.12636e-16 0.136254>[14]  -> <5.01475e-18 0.072153>s
<8.5034e-14 0.26329>[13]  -> <5.61213e-16 0.126925>b[6]
<2.61673e-16 0.126925>[14]  -> <1.21288e-18 0.0586089>s
<2.9954e-16 0.126925>[14]  -> <3.65218e-18 0.0682478>s
```

Many solutions appear

cuikplot3d & geomview

Solve the equations

```
> cuik stickybot-path
...
<9.14993e-14 0.26329>[13]  -> <7.79572e-16 0.136254>b[7]
<3.66936e-16 0.136254>[14]  -> <1.71125e-18 0.0640773>s
<4.12636e-16 0.136254>[14]  -> <5.01475e-18 0.072153>s
<8.5034e-14 0.26329>[13]  -> <5.61213e-16 0.126925>b[6]
<2.61673e-16 0.126925>[14]  -> <1.21288e-18 0.0586089>s
<2.9954e-16 0.126925>[14]  -> <3.65218e-18 0.0682478>s
```

Many solutions appear

View the solutions

```
> cuikplot3d stickybot-path 1 2 3 0 stickybot-path.gcl
Reading solution file: stickybot-path.sol
Creating plot file: stickybot-path.gcl
```

```
> geomview stickybot-path.gcl
```

cuikplot3d & geomview

Solve the equations

```
> cuik stickybot-path
...
<9.14993e-14 0.26329>[13]  -> <7.79572e-16 0.136254>b[7]
<3.66936e-16 0.136254>[14]  -> <1.71125e-18 0.0640773>s
<4.12636e-16 0.136254>[14]  -> <5.01475e-18 0.072153>s
<8.5034e-14 0.26329>[13]  -> <5.61213e-16 0.126925>b[6]
<2.61673e-16 0.126925>[14]  -> <1.21288e-18 0.0586089>s
<2.9954e-16 0.126925>[14]  -> <3.65218e-18 0.0682478>s
```

Many solutions appear

View the solutions

```
> cuikplot3d stickybot-path 1 2 3 0 stickybot-path.gcl
Reading solution file: stickybot-path.sol
Creating plot file: stickybot-path.gcl
```

Vars 1st 2nd and 3rd of stickybot.cuik (x,y,c2)

```
> geomview stickybot-path.gcl
```

cuikplot3d & geomview

Solve the equations

```
> cuik stickybot-path
...
<9.14993e-14 0.26329>[13]  -> <7.79572e-16 0.136254>b[7]
<3.66936e-16 0.136254>[14]  -> <1.71125e-18 0.0640773>s
<4.12636e-16 0.136254>[14]  -> <5.01475e-18 0.072153>s
<8.5034e-14 0.26329>[13]  -> <5.61213e-16 0.126925>b[6]
<2.61673e-16 0.126925>[14]  -> <1.21288e-18 0.0586089>s
<2.9954e-16 0.126925>[14]  -> <3.65218e-18 0.0682478>s
```

Many solutions appear

View the solutions

```
> cuikplot3d stickybot-path 1 2 3 0 stickybot-path.gcl
Reading solution file: stickybot-path.sol
Creating plot file: stickybot-path.gcl
```

Vars 1st 2nd and 3rd of stickybot.cuik (x,y,c2)

Do not magnify the boxes

```
> geomview stickybot-path.gcl
```

cuikplot3d & geomview

Solve the equations

```
> cuik stickybot-path
...
<9.14993e-14 0.26329>[13]  -> <7.79572e-16 0.136254>b[7]
<3.66936e-16 0.136254>[14]  -> <1.71125e-18 0.0640773>s
<4.12636e-16 0.136254>[14]  -> <5.01475e-18 0.072153>s
<8.5034e-14 0.26329>[13]  -> <5.61213e-16 0.126925>b[6]
<2.61673e-16 0.126925>[14]  -> <1.21288e-18 0.0586089>s
<2.9954e-16 0.126925>[14]  -> <3.65218e-18 0.0682478>s
```

Many solutions appear

View the solutions

```
> cuikplot3d stickybot-path 1 2 3 0 stickybot-path.gcl
Reading solution file: stickybot-path.sol
Creating plot file: stickybot-path.gcl

> geomview stickybot-path.gcl
```

Vars 1st 2nd and 3rd of stickybot.cuik (x,y,c2)

Do not magnify the boxes

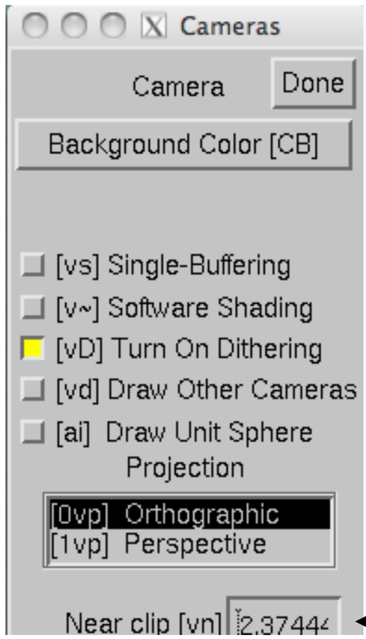
Type a command with no arguments for help

```
> cuikplot3d
    Use:
      cuikplot3d <filename> dimx dimy dimz min_size <plotname>
        <filename>     => the input .sol file
        dimx dimy dimz => the three dimensions to be plotted (numbered from 1)
        min_size       => minimum size for the boxes
        <plotname>     => the output .gcl file
```

1

Inspect→cameras →orthographic

to view the plot in orthogonal (rather than perspective) projection

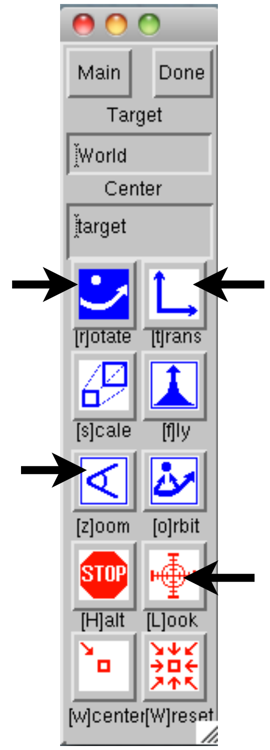


← Set -1000 + enter in Near clip

2

Zoom/translate/rotate

for a global view of the solution boxes

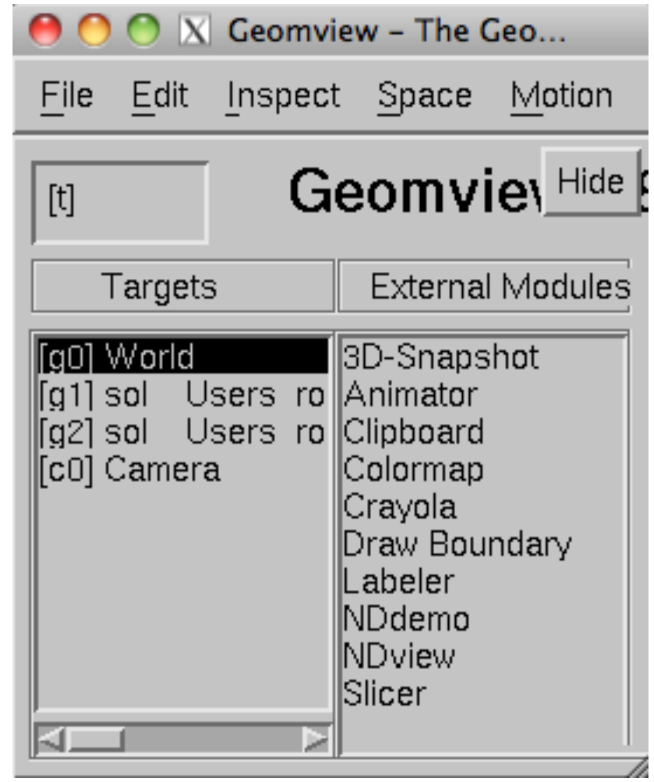


Chooses a good view

3

Maintain "World" selected

To zoom/translate/rotate the whole world

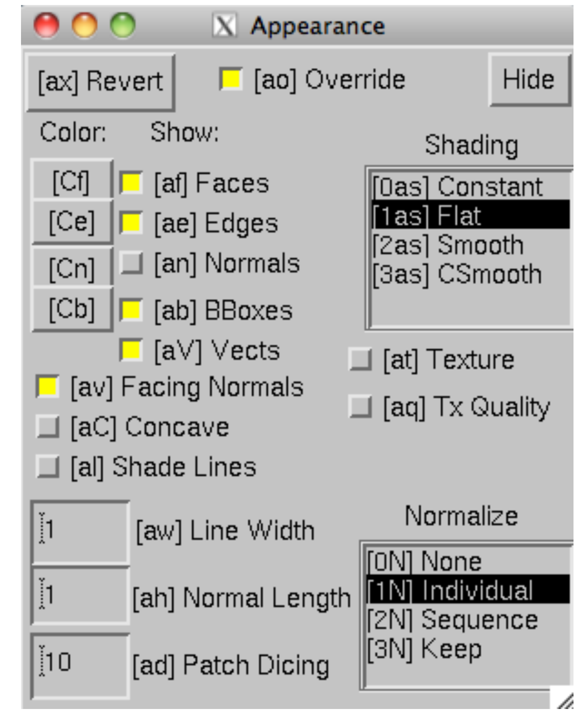


4

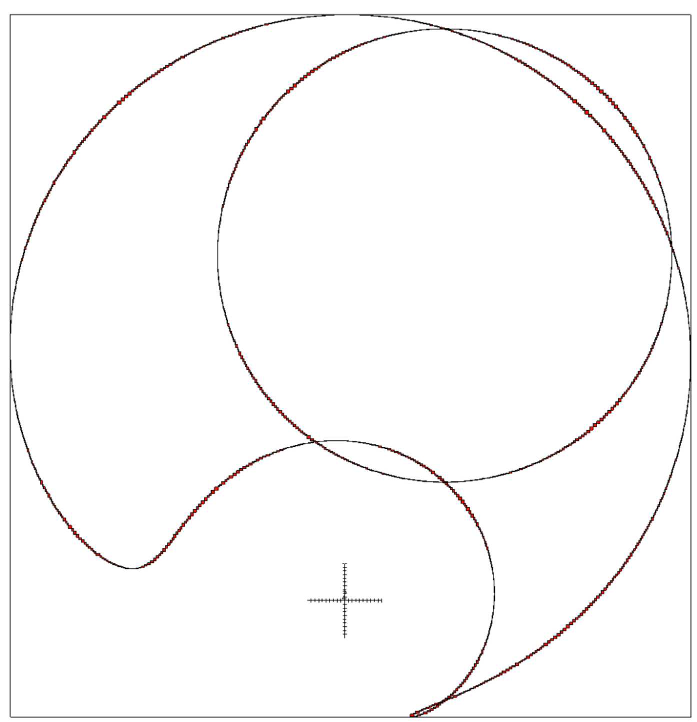
Inspect→Appearance→edges

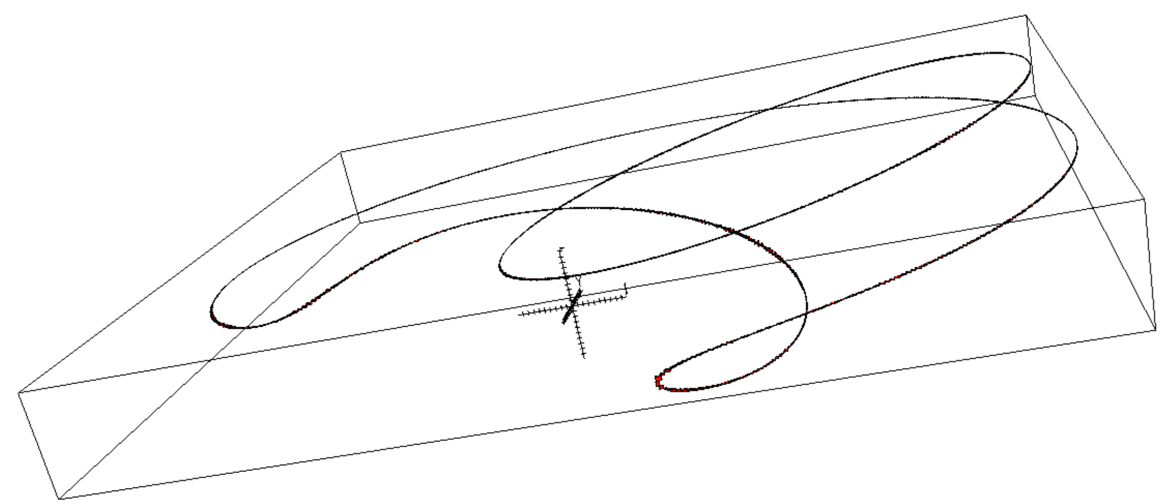
to view the box edges

[Cf] and [Ce] change the face and edge colors



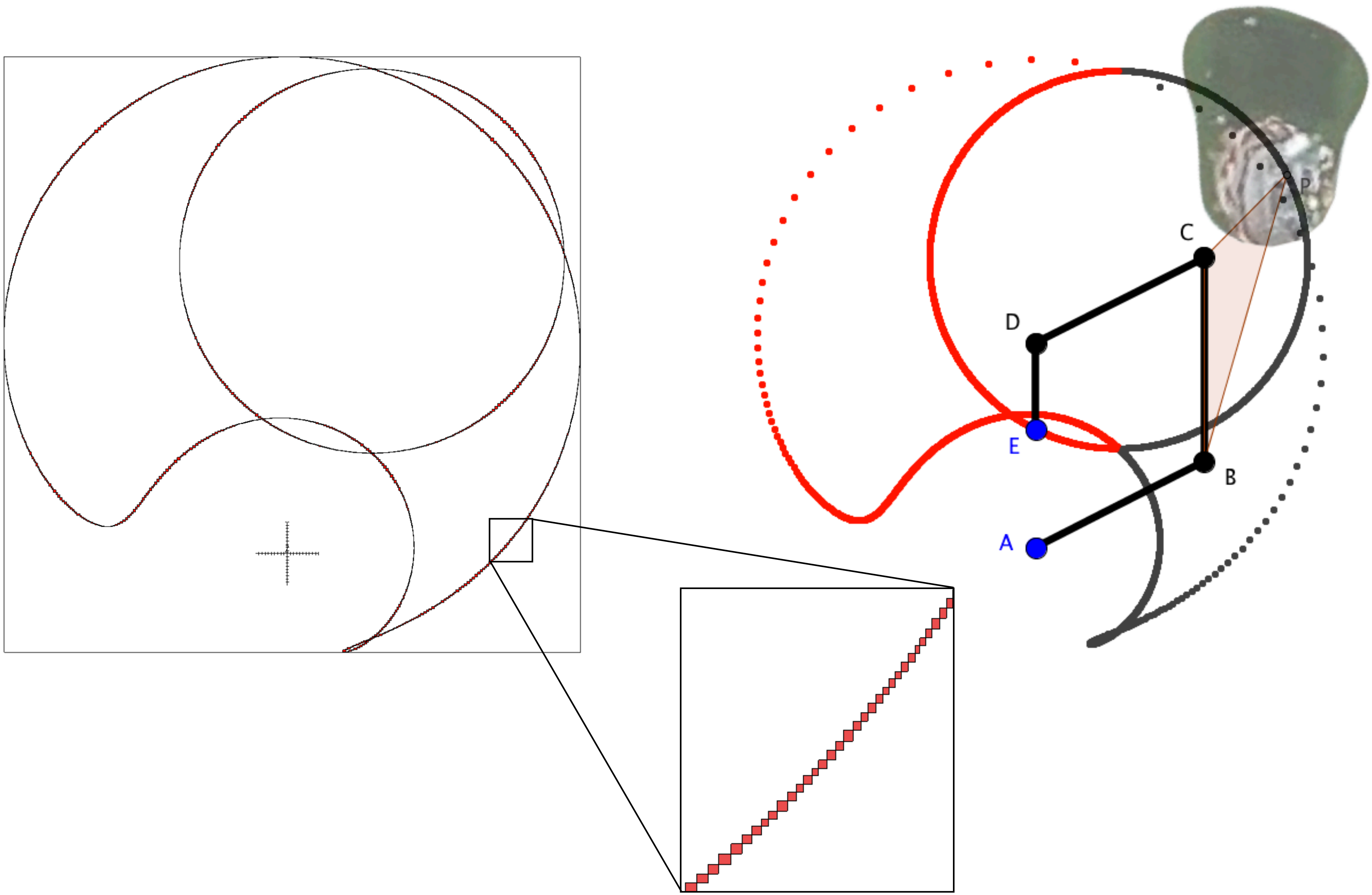
Try to obtain similar views to these ones





Select "BBoxes" in Appearance panel to display the bounding boxes

Compare the path with that of Geogebra's animation



Compute a workspace

Edit **`stickybot.cuik`** and save it to **`stickybot-ws.cuik`**

```
[CONSTANTS]

l1:=3.8
l2:=6.1
l3:=6.6
l4:=6.1
l5:=2.8

a :=9.67
alpha := 16.14 * pi/180
calpha:=cos(alpha)
salpha:=sin(alpha)

r := l2+a

[SYSTEM VARS]

x:[-r,r]
y:[-r,r]
c2:[cos(pi/4),1]
s2:[-1,1]
c3:[-1,1]
s3:[-1,1]
c4:[-1,1]
s4:[-1,1]
c5:[cos(pi/4),1]
s5:[-1,1]
```

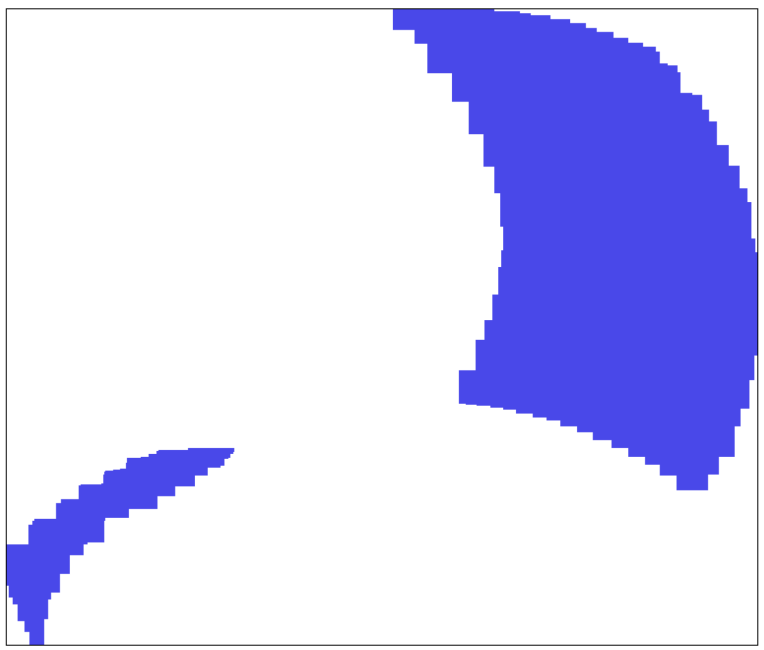
Limit θ_2 and θ_5 to $[-\pi/4, \pi/4]$ to simulate joint limits

```
[SYSTEM EQS]

% Loop closure equations

l2*c2 + l3*c3 + l4*c4 + l5*s5 = 0;
-l1   + l2*s2 + l3*s3 + l4*s4 - l5*c5 = 0;

% Position of P

x = l2*c2 + a*calpha*c3+a*salpha*s3;
y = l2*s2 + a*calpha*s3-a*salpha*c3;

% Circle equations

s2^2 + c2^2 = 1;
s3^2 + c3^2 = 1;
s4^2 + c4^2 = 1;
s5^2 + c5^2 = 1;

% Lock the input to solve a FKP problem

c2 = cos(0);
s2 = sin(0);

c5 = cos(0);
s5 = sin(0);
```

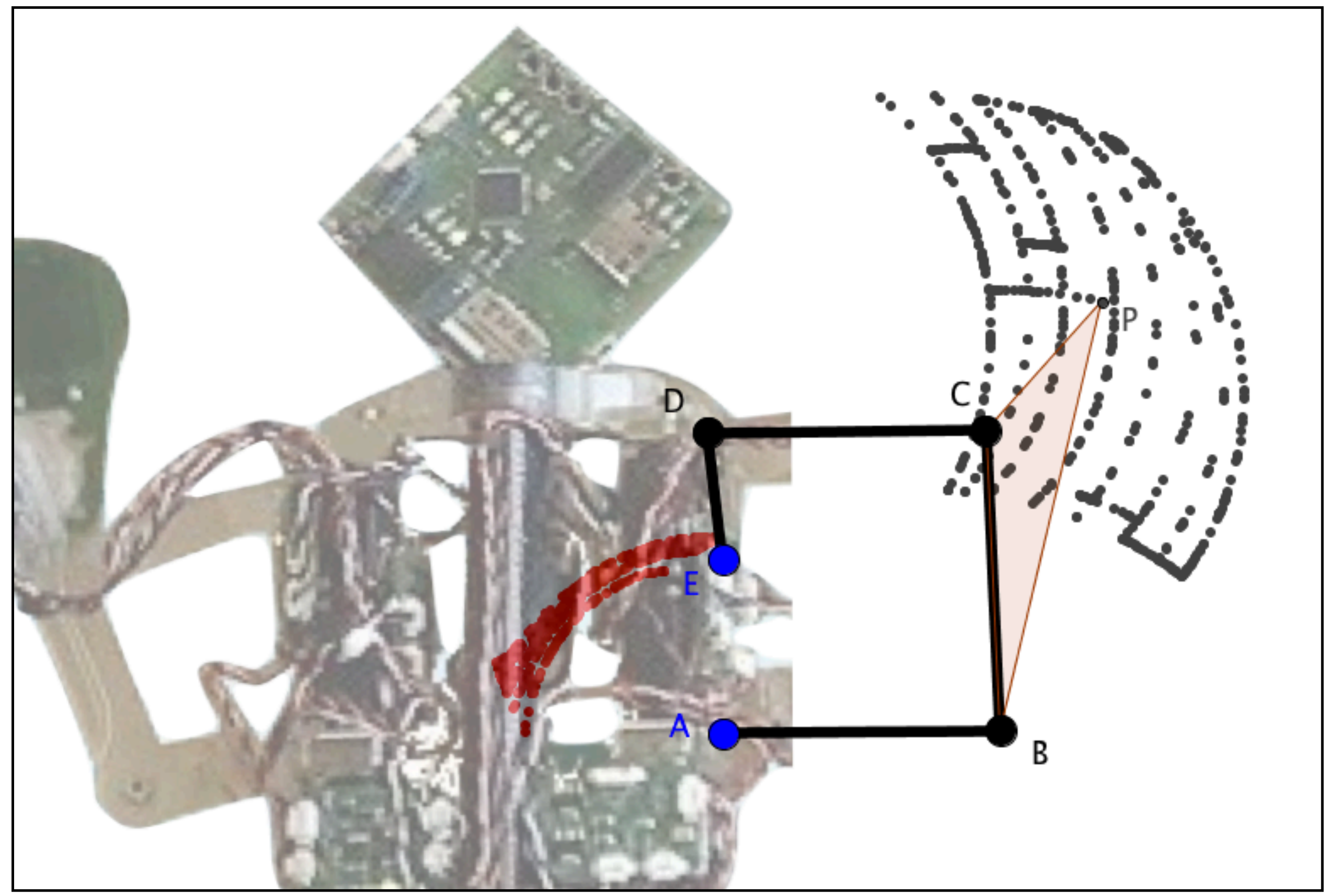
Unconstrain to obtain the **position workspace** of P

Position workspace of P

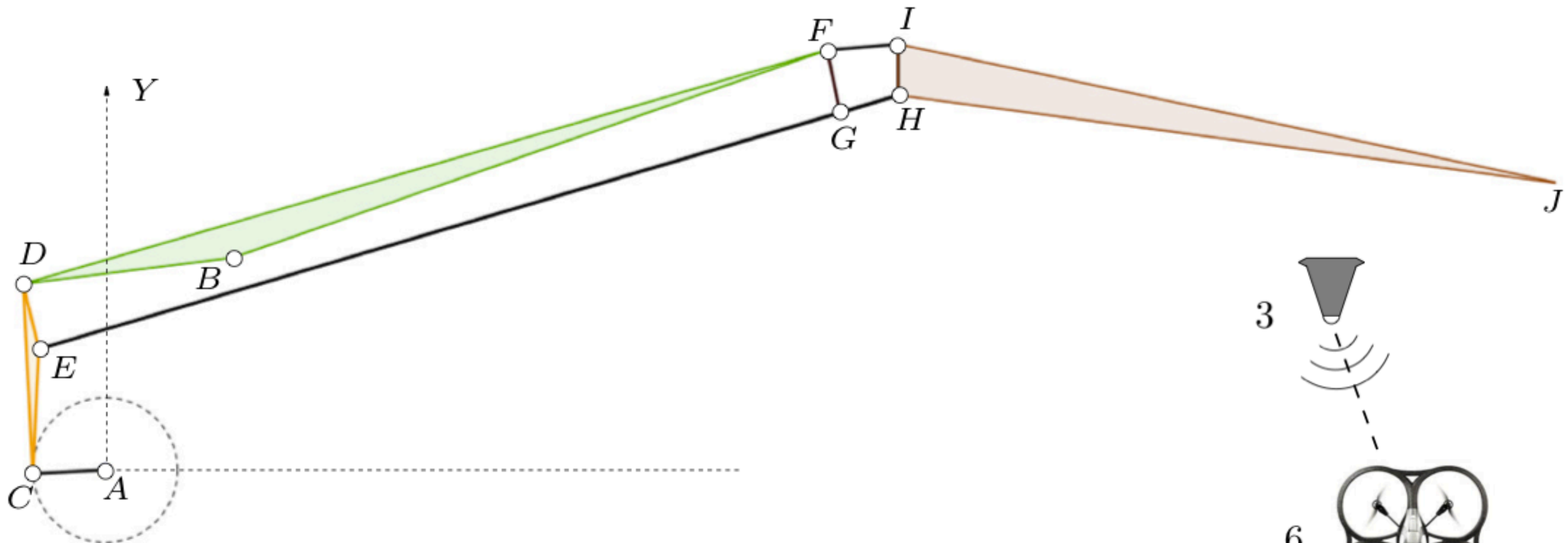
CUIK's vs Geogebra's



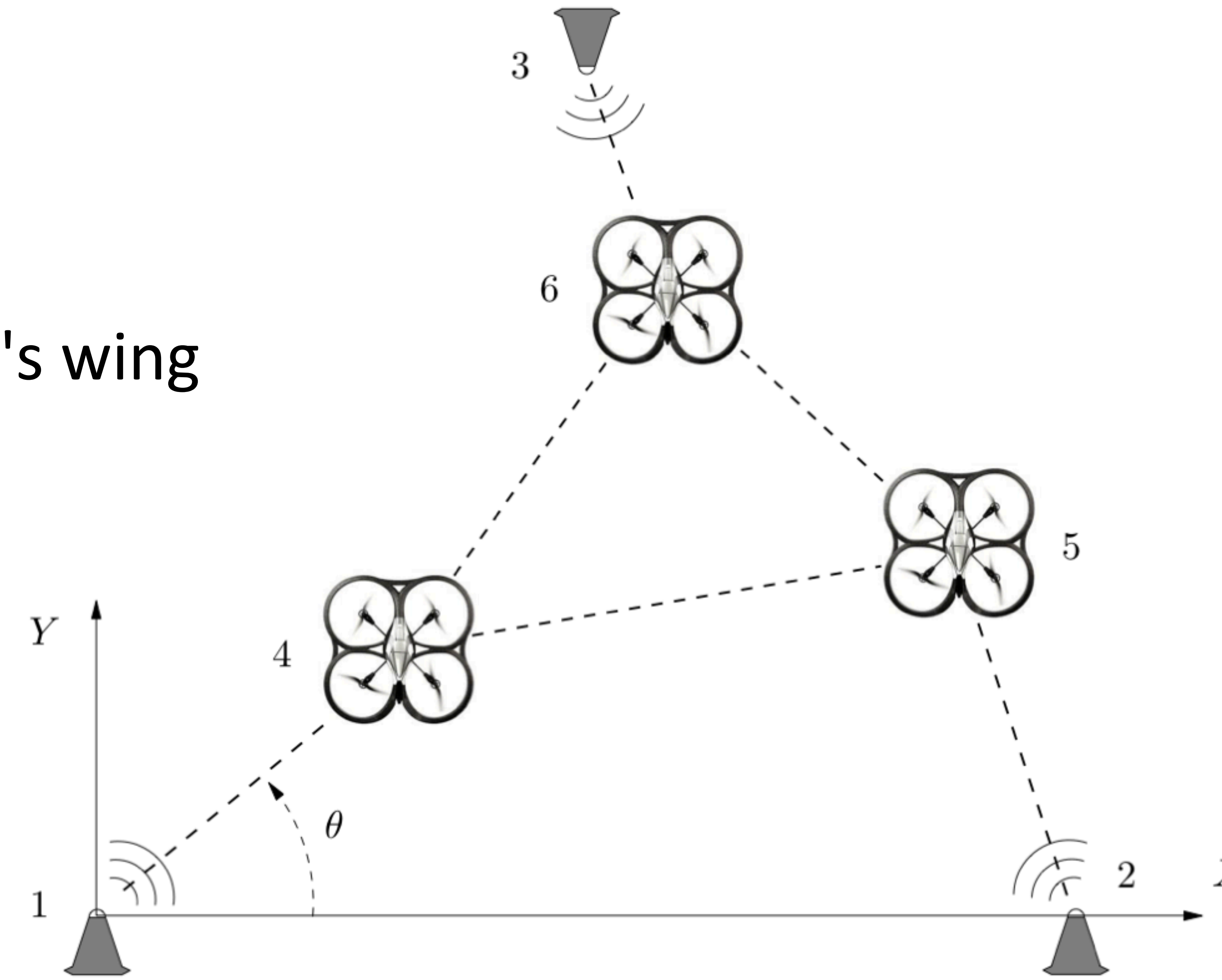
Obtained with σ=1



You can solve Miniproject 1



Simulation of Smartbird's wing



Quadrotor localization problem

Appendices

Interpreting CUIK's stdout/stderr output

- Not much informative at a user's level
- Type `cuik &> logfile` to send such output to a logfile
- Used to see CUIK's progression or to debug
- A line for each box processed:

Initial box volume | Max side length | Search tree level | Final box volume | Max side length

Reduced box is a solution ("S" if it is also "verified")

```
<9.12922e-16 0.164551>[8]  -> <4.0709e-18 0.0773365>s

<2.63837e-09 0.97423>[6]  -> <1.6872e-11 0.478704>b[3]

<247.785 12.345>[6]  -> e
```

Reduced box is bisected through var # 3

Reduced box is empty

Further details on *.sol files

Max equation error at the box center

Time at which the box is found

Number of problem variables

Variable name

Variable interval

```
==========================================================================
Cuik executed in 1 processors
SOLUTIONS (to input file stickybot.cuik):

   1 (err:3.69268e-06 tm:0):{ 10  x[8.78811710513,8.78811819992] y[9.28885876005,9.28886002588]    ... }
   2 (err:1.54197e-07 tm:0):{ 10  x[-2.94877518079,-2.94877346147] y[3.41007203352,3.41007354778]  ... }

STATITISTICS FOR Cuik:

Volume of the search space: 15916.3
Volume of the solution space: 0
  Volume ratio:  0.00 %
Max solution diagonal: 2.30851e-06
Number of processors: 1
User time in process: 0.000000 seg (0.000000 min)
Box level information:
  N processed boxes :      3
  Max depth         :      2
  Types of boxes:
    N solution boxes:      2 (0) (66.67%)
    N empty boxes   :      0  ( 0.00%)
    N bisected boxes:      1  (33.33%)
  Box Reductions    :      4
  N Errors          :      0
==========================================================================
```