Engineering Sketch Pad (ESP)

Training Session 7
Sketcher Fundamentals

John F. Dannenhoffer, III
jfdannen@syr.edu
Syracuse University

Bob Haimes
haimes@mit.edu
Massachusetts Institute of Technology

updated for v1.18

Overview

- Purpose of Sketches
- Sketching Segments
- Sketching methods
  - programmatically
  - interactively
- Homework Exercises

Purpose of Sketches

- Method for generating a SheetBody, WireBody, or NodeBody
- Sketches are used a basis of grown Bodys
  - `EXTRUDE`, `REVOLVE`, `RULE`, and `BLEND`

Sketching Segments

- `LINSEG` — straight line segment
- `CIRARC` — circular arc
- `ARC` — alternative way of specifying a circular arc
- `BEZIER` — Bezier curve
- `SPLINE` — cubic spline

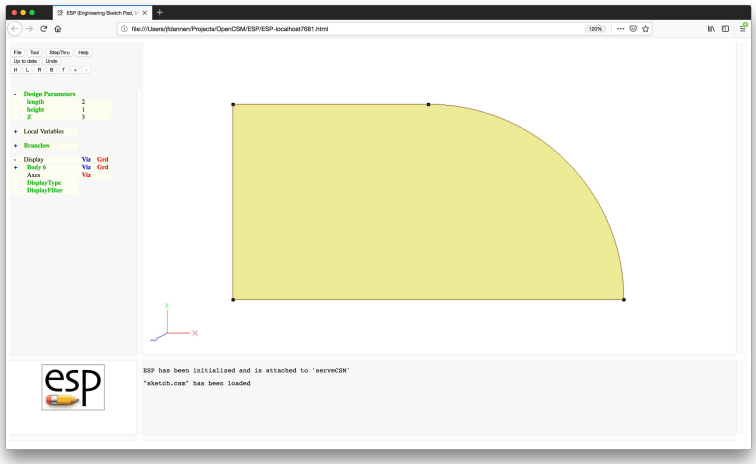
Sketching Methods

- Programmatically
  - can generate Sketch in 3D
  - user does all required math
  - is very robust
- Interactively
  - can generate Sketch only in 2D
  - required math is done by solving constraints
  - is somewhat fragile

Sketching Programmatically

- Begin with a `SKBEG` statement, which provides an initial point
- Add `LINSEG`, `CIRARC`, `BEZIER`, or `SPLINE` Segments
  - for the `BEZIER` and `SPLINE` statements, one curve is created from the point before these statement, using all the `BEZIER` or `SPLINE` statements
  - an `SSLOPE` statement before the first and/or after the last `SPLINE` statement can be used to specify the slope at the beginning or end
  - to have two adjacent curves, put a zero-length `LINSEG` between them
- Ends with a `SKEND` statement
  - if there are no Segments, a NodeBody is created
  - if the last Segment does not end at the point specified in the `SKBEG` statement, a WireBody is created
  - if the Sketch is closed, a SheetBody is created (unless the `wireonly` flag on the `SKEND` statement is non-zero)

Programmatic Sketch Example



```
# sketch

DESPMTR   L          2.0
DESPMTR   H          1.0
DESPMTR   Z          3.0

SET       s2         1/sqrt(2)

SKBEG     1.0              2.0        Z
   LINSEG 1.0+L            2.0        Z
   CIRARC 1.0+L-(1-s2)*H   2.0+s2*H   Z \
          1.0+L-H          2.0+H      Z
   LINSEG 1.0              2.0+H      Z
   LINSEG 1.0              2.0        Z
SKEND

END
```

Steps to Creating a Sketch Interactively

1. Define the Design Parameters
2. Create an empty Sketch
3. Draw the Segments
4. Constrain the Sketch
5. Solve the Sketch

Creating a Sketch

Step 1: Define the Design Parameters

- Press **Design Parameters** in the Tree window to create each of the Design Parameters
- Most Design Parameters are a scalar, so that they have only 1 row and 1 column
- Enter nominal value(s) in the box(es) that appears
- Press **OK** to proceed
- Repeat as needed

Creating a Sketch

Step 2: Create an Empty Sketch

- Press **Branches** in the Tree window to create a `SKBEG` Branch
  - coordinates should be specified at one point on the boundary of the Sketch
  - coordinates can be defined in terms of a Design Parameter
- A `SKEND` is automatically created for you
- The Sketcher is entered automatically

Creating a Sketch

Step 3: Draw the Segments (1)

- Start drawing the Sketch at the point defined in the `SKBEG` Branch
  - **X** and **Y** Constraints are automatically generated at the initial point
  - these constraints cannot be deleted
- Draw the Segments by proceeding counter-clockwise around the Sketch (which is consistent with the right-hand rule pointing out of the screen)
- Line between previous point and cursor shows proposed position of next Segment
  - blue is default color
  - if drawn in orange, a vertical (V) or horizontal constraint (H) will be added automatically

Creating a Sketch

Step 3: Draw the Segments (2)

- Supported Segment types include:
  - (straight) line Segment
    - **l** or **L** or mouse click
  - (circular) arc Segment
    - **c** or **C**
    - Segment turns red until you press the mouse button to set its approximate radius
  - cubic spline
    - **s** or **S**
    - cubic splines are shown only as straight line Segments in the Sketcher
  - Bezier curve control points
    - **b** or **B**
  - ...

Creating a Sketch

Step 3: Draw the Segments (3)

- Supported Segment types include:
  - zero-length Segment
    - **z** or **Z**
    - constraints automatically set
  - leave Sketch open (and switch mode to "Constraining...")
    - **o** or **O**
- When Sketch is closed, its interior is filled with gray (and the mode is switched to "Constraining...")
- Pressing the **Undo** button will remove the last Segment

Creating a Sketch

Step 4: Constrain the Sketch (1)

- As many constraints (**ncon**) must be defined as there are degrees of freedom (**ndof**) in the Sketch
  - these values are listed in the Key window
  - the fill turns to light green when they match (**ncon=ndof**)
  - having them match is necessary, but not sufficient, for a Sketch to be properly constrained

Creating a Sketch

Step 4: Constrain the Sketch (2)

- Constraints that can be applied to Segments:
  - set the Segment's length
    - **l** or **L**
  - make the Segment horizontal ($y_{\text{beg}} = y_{\text{end}}$)
    - **h** or **H**
    - might be automatically created if Segment was orange when created
  - make the Segment vertical ($x_{\text{beg}} = x_{\text{end}}$)
    - **v** or **V**
    - might be automatically created if Segment was orange when created
  - set the inclination in degrees (measured counter-clockwise from the right horizontal)
    - **i** or **I**

Creating a Sketch

Step 4: Constrain the Sketch (3)

- Constraints that can be applied to circular arcs:
  - acute radius (positive if convex when drawing counter-clockwise)
    - **r** or **R**
  - X-coordinate at arc center
    - **x** or **X**
  - Y-coordinate at arc center
    - **y** or **Y**
  - sweep angle in degrees (positive if convex when drawing counter-clockwise)
    - **s** or **S**

Creating a Sketch

Step 4: Constrain the Sketch (4)

- Constraints that can be applied to points:
  - specify X-coordinate
    - **x** or **X**
  - specify Y-coordinate
    - **y** or **Y**
  - adjacent Segments are perpendicular
    - **p** or **P**
  - adjacent Segment are tangent (parallel)
    - **t** or **T**
  - turning angle between adjacent Segments in degrees (positive if turning to the left)
    - **a** or **A**

Creating a Sketch

Step 4: Constrain the Sketch (5)

- Constraints that can be applied to a pair of points:
  - specify width ($x_{\text{end}} - x_{\text{beg}}$)between two points
    - **w** or **W**
    - if first point is toward the left, a positive value should be specified
    - if first point is toward the right, a negative value should be specified
  - specify depth ($y_{\text{end}} - y_{\text{beg}}$) between two points
    - **d** of **D**
    - if first point is toward the bottom, a positive value should be specified
    - if first point is toward the top, a negative value should be specified

Creating a Sketch

Step 4: Constrain the Sketch (6)

- Other options:
  - remove Constraints
    - <
    - if more than one constraint is present, you are asked which constraint to remove
  - inquire about constraints at current point or Segment
    - **?**
- Pressing the **Undo** button will remove/restore the last constraint

Creating a Sketch

Step 4: Constrain the Sketch (7)

- Special shortcuts
  - `::L[i]` is the length of the Segment `i`
  - `::I[i]` is the inclination of Segment `i` (in degrees)
  - `::R[i]` is the radius of CIRARC Segment `i`
  - `::S[i]` is the sweep of CIRARC Segment `i` (in degrees)
- Segment numbers can be determined by pressing **?** near the center of a Segment

Creating a Sketch

Step 4: Constrain the Sketch (8)

- If you need help during the constraint process
  - Press the yellow **Constraining...** button
- Redundant constraints are shown in red
  - Use the < key to remove a redundant constraint
- Suggested new constraints are shown in green
  - Add the constraint using a key that matches the hint

Creating a Sketch

Step 5: Solve the Sketch

- Press **Press to Solve**
  - if successful, Sketch will change on screen
  - if unsuccessful, read about error in Messages window to help you diagnose the problem
- Press **Sketch→Save** to return to normal (non-Sketching) mode
- Press **Press to Re-build** to see the completed Sketch

Editing an Existing Sketch

- Select one of the Branches between the `SKBEG` and `SKEND` Branches (inclusive) and press **Enter Sketcher**
- Follow directions given above

Deleting an Existing Sketch

- Select each of the Branches between the `SKBEG` and `SKEND` and press **Delete Branch** for each. Then delete the `SKEND` and `SKBEG` Branches.
- Select the `SKBEG` Branch and press **Delete Branch** (to delete whole sketch at once)

Image Manipulation in the Sketcher

- Recenter Sketch
  - **Ctrl-h** key or **H** button
- Move the Sketch to the left
  - **Ctrl-l** key or **L** button or ← key
- Move the Sketch to the right
  - **Ctrl-r** key or **R** button or → key
- Move the Sketch to the bottom
  - **Ctrl-b** key or **B** button or ↓ key
- Move the Sketch to the top
  - **Ctrl-t** key or **T** button or ↑ key
- Zoom in
  - **Ctrl-i** key or **PgUp** key or **+** button
- Zoom out
  - **Ctrl-o** key or **PgDn** key or **-** button

Sketching Best Practices

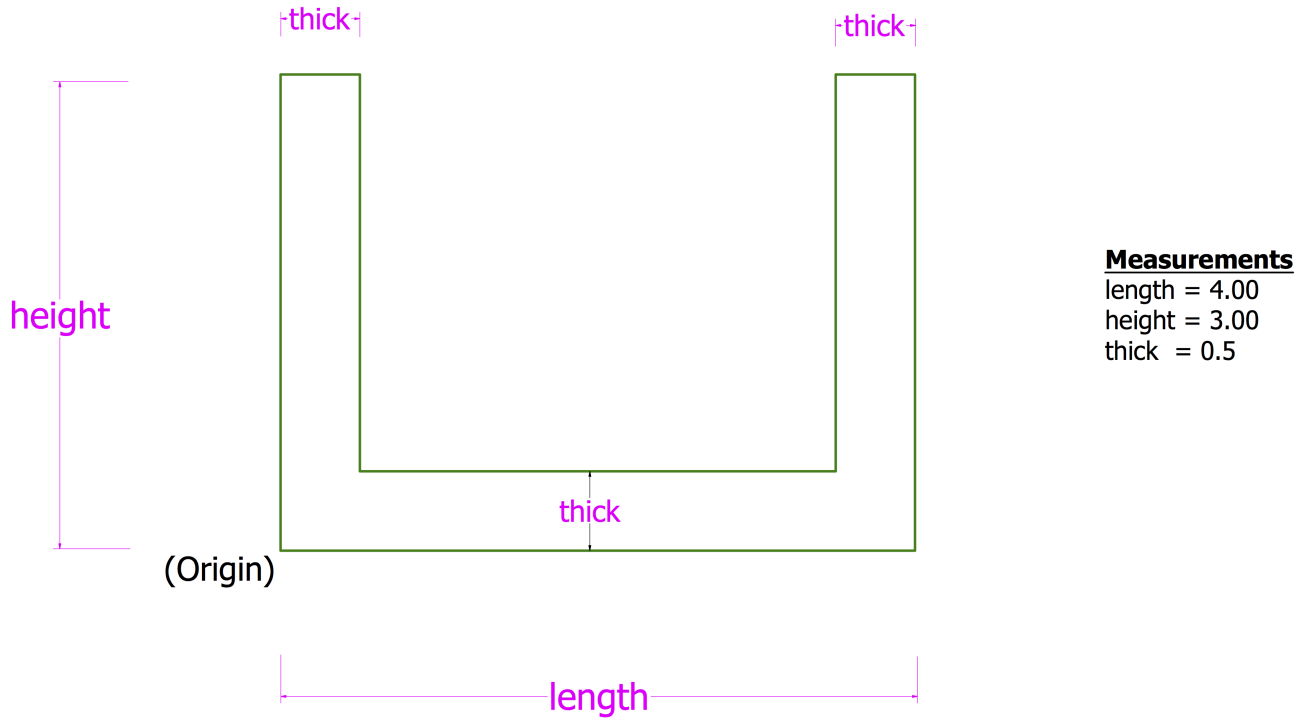
- Try to start the Sketch at a point with known coordinates
- Proceed around the sketch in a counter-clockwise direction
- Constrain the X-coordinate at one or more points (or arc centers)
- Constrain the Y-coordinate at one of more points (or arc centers)
- Specify the orientation of one or more Segments
  - this is sometimes done by specifying the coordinates of both ends
- Avoid redundancies, such as:
  - points at which angles are constrained and which are adjacent to Segments in which the inclination is constrained
  - dimensions specified for both a series of Segments as well as their combination

Homework Exercises

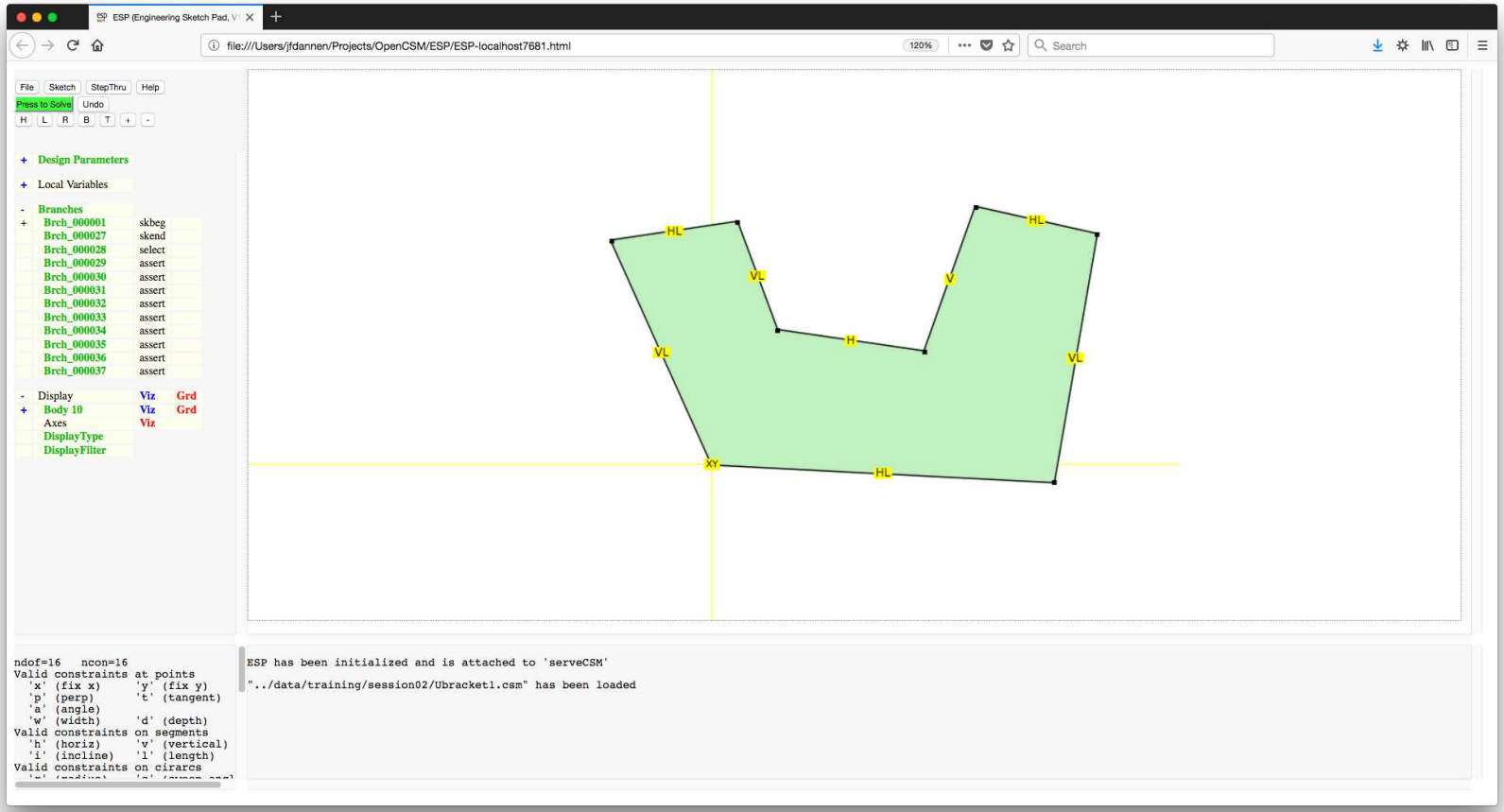
- U-shaped bracket (version 1)
- U-shaped bracket (version 2)
- oval
- bi-convex airfoil (with arcs)
- swivel base
- V-slide plate
- bi-convex airfoil (with splines)
- fuselage cross-section (with Beziers)

Example: U-bracket (version 1)

Hint: move mouse until blue line turns orange to automatically generate horizontal and vertical constraints

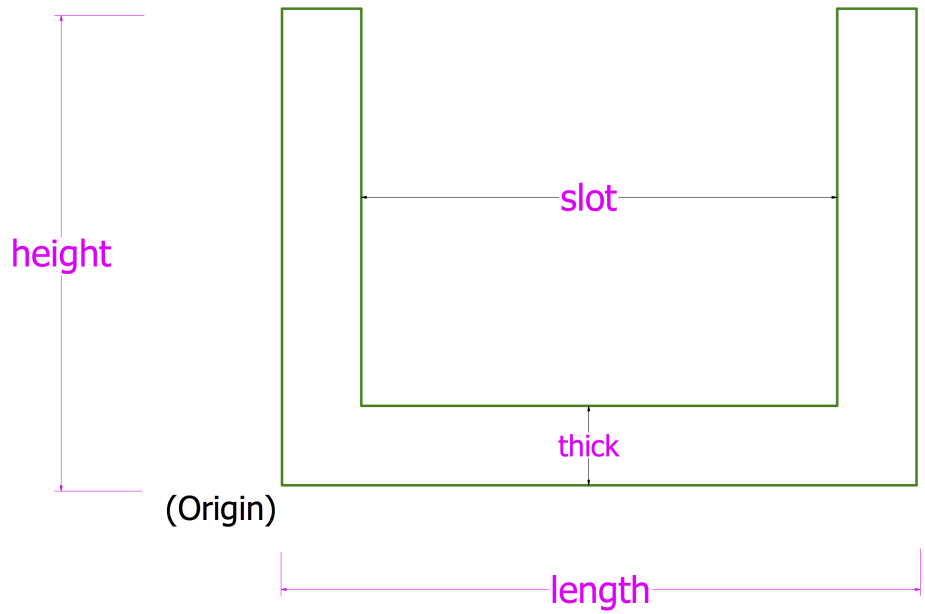


Solution: U-bracket (version 1)



Example: U-bracket (version 2)

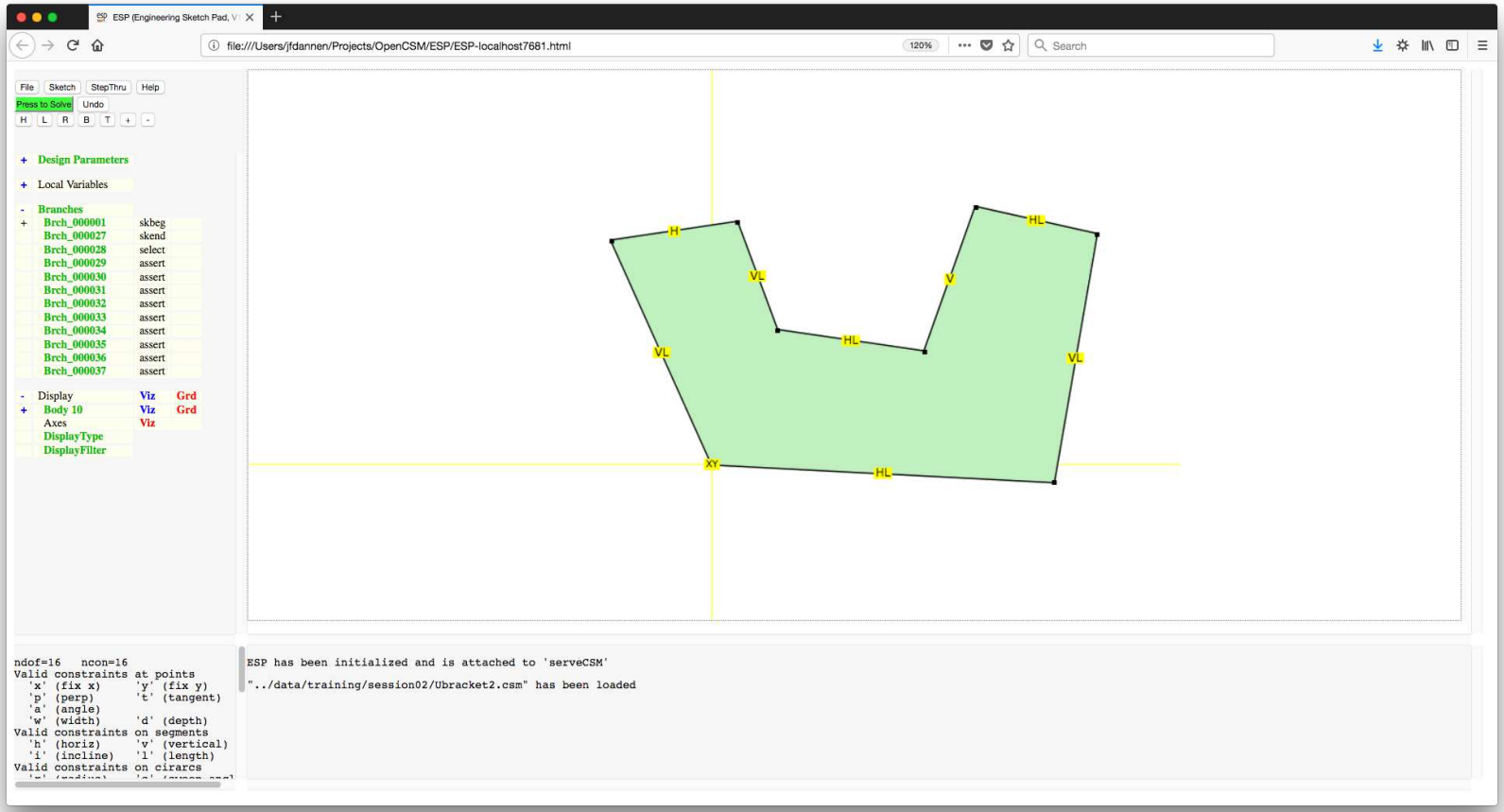
Hint: You can specify the length of a Segment to be equal to Segment 5's length with `::L[5]` (where the Segment number can be obtained with the "**?**" command).



Measurements
length = 4.00
height = 3.00
thick = 0.5
slot = 2.00

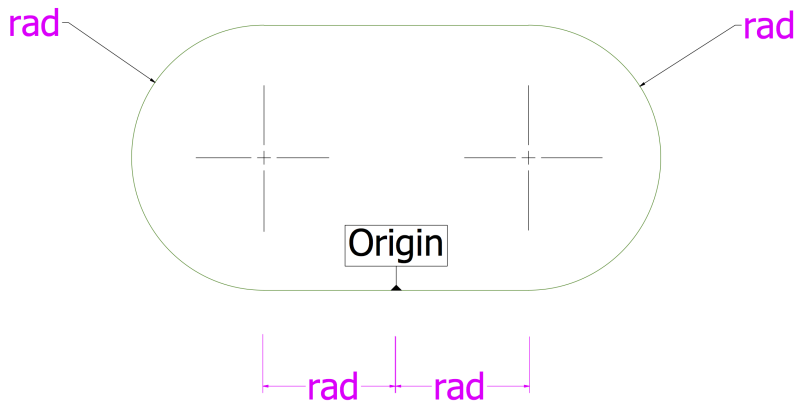
Note: slot
is centered

Solution: U-bracket (version 2)



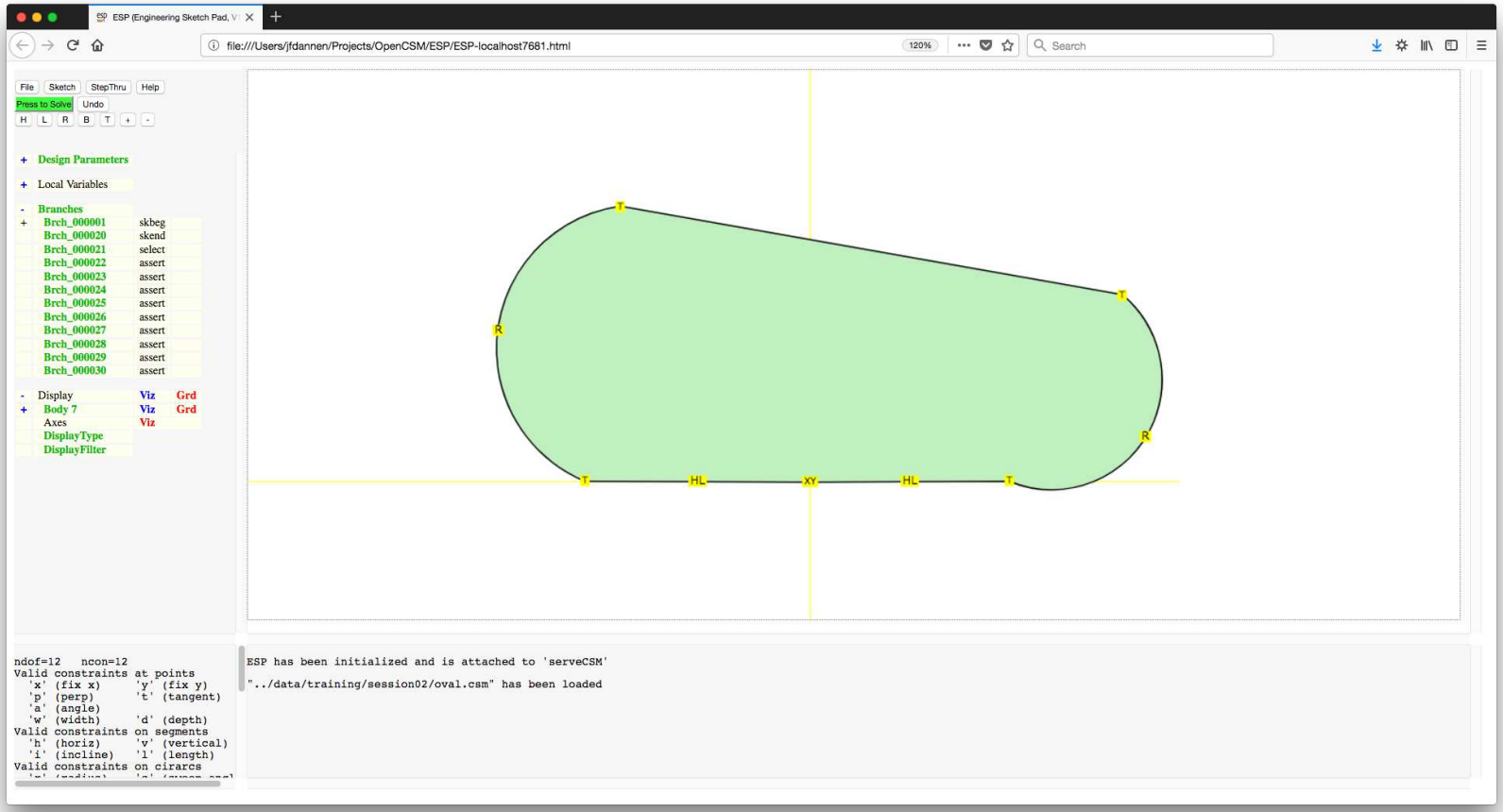
Example: Oval

Hint: tangency constraints may be useful for this case



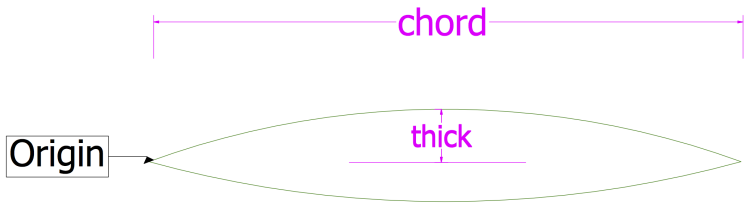
Measurements:
rad = 0.50

Solution: Oval



Example: Biconvex airfoil (with arcs)

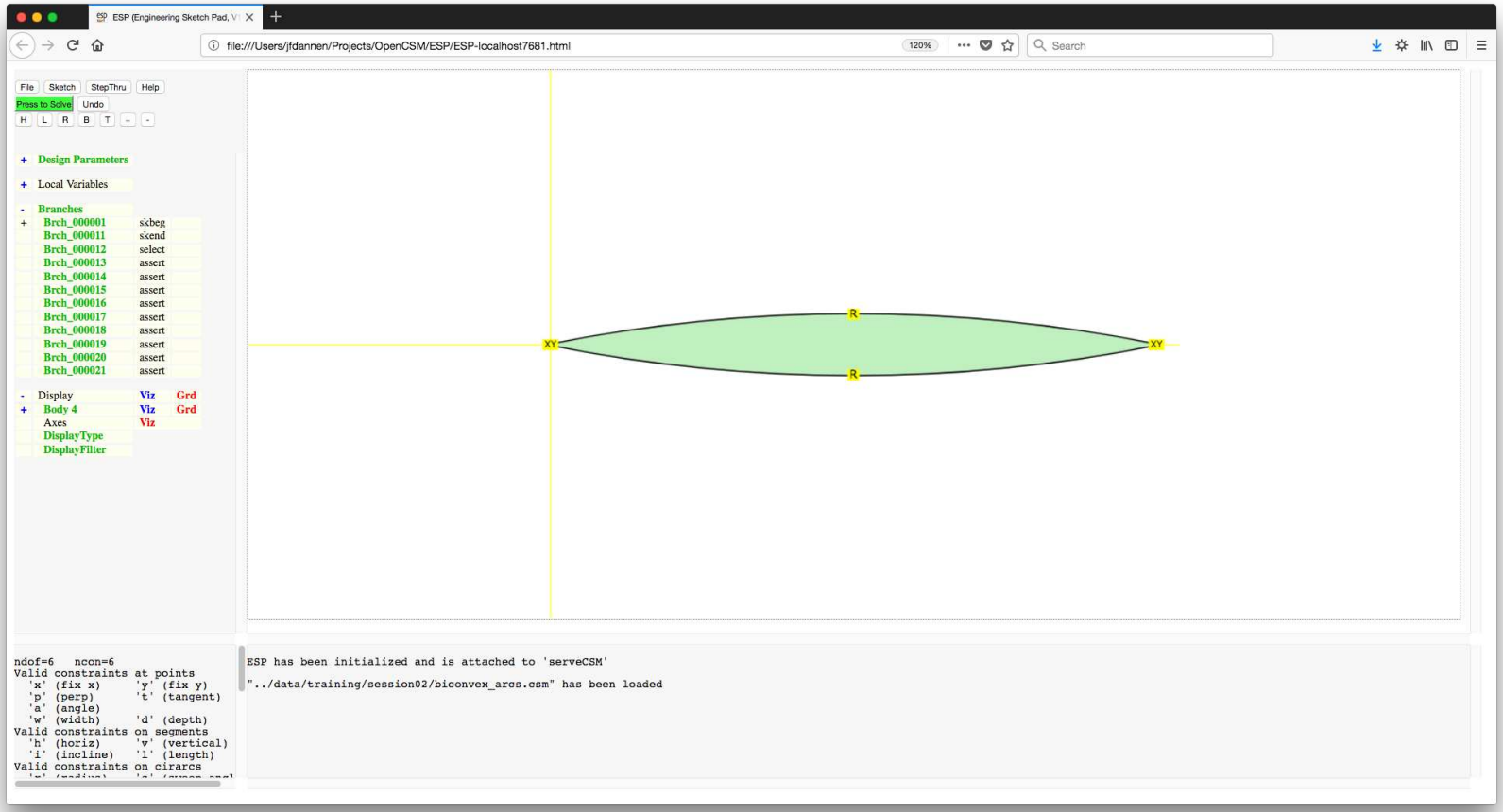
Hint: the `radius()` function can be used if one knows the bounding coordinates and the "dip" (see "Help" for details)



Measurements:
chord = 2.00
thick = 0.10

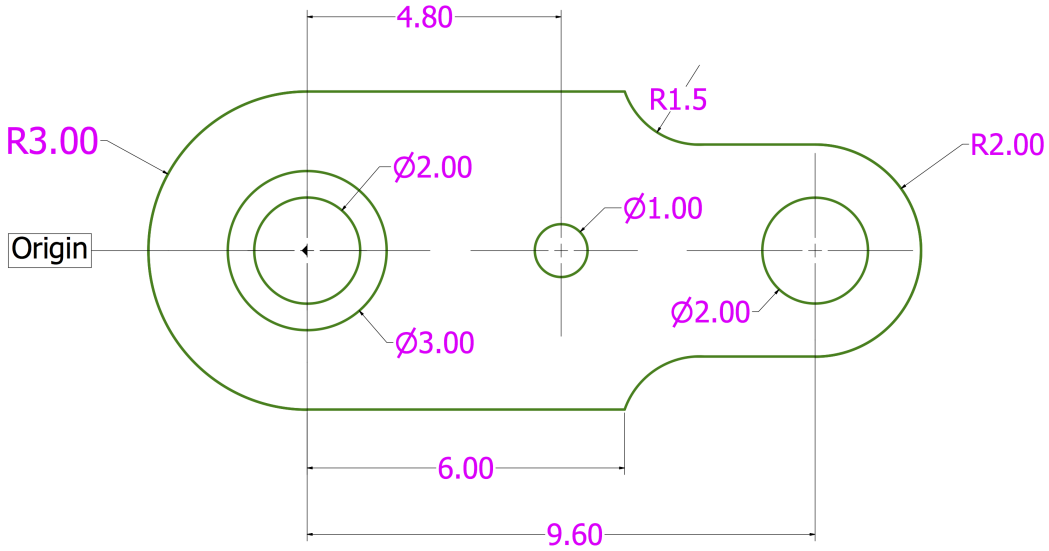
Note:
Circular Arcs

Solution: Biconvex airfoil (with arcs)

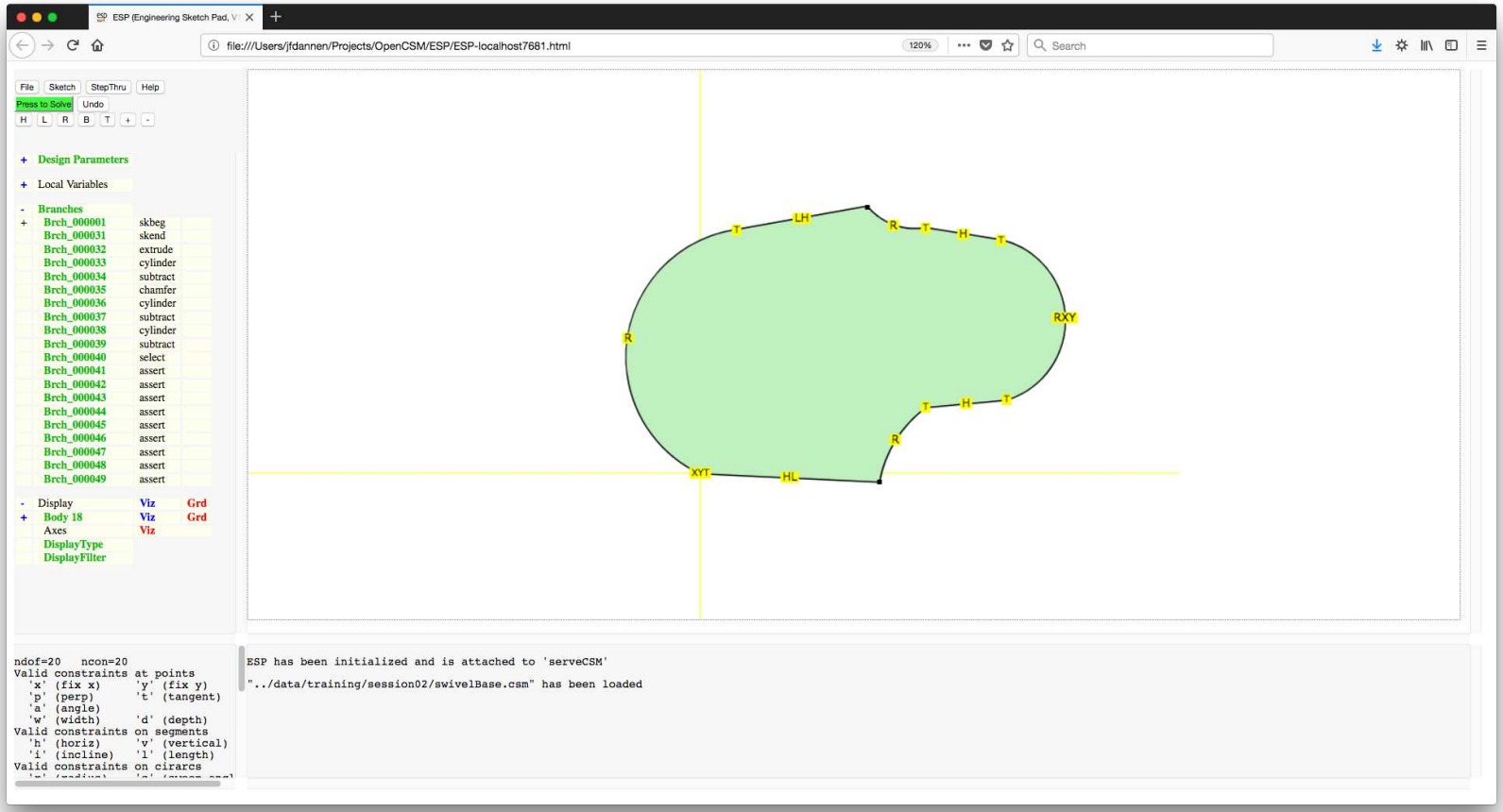


Example: Swivel Base

Hint: nested Sketches can be generated with a series of Sketches

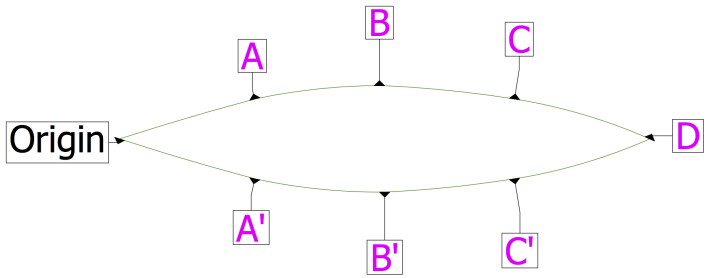


Solution: Swivel Base



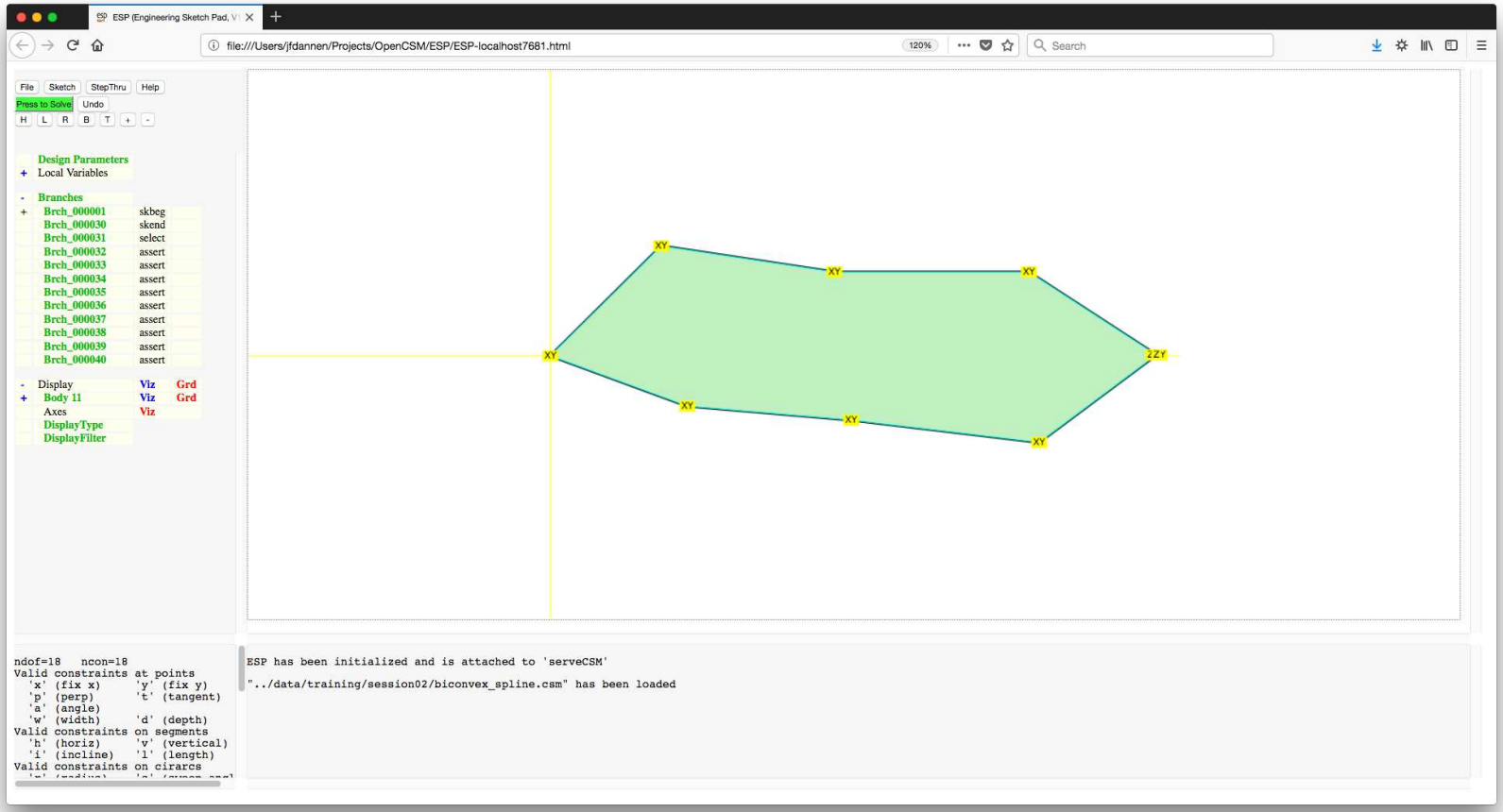
Example: Biconvex Airfoil (with splines)

Hint: adjacent splines (with slope discontinuities) can be obtained by putting a zero-length line Segments between them



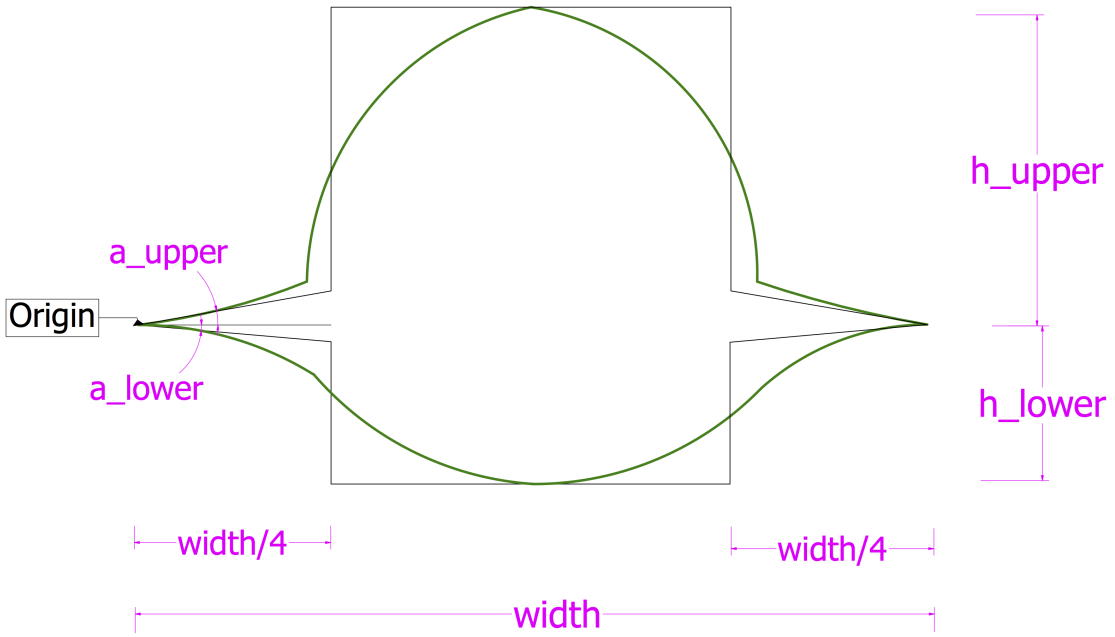
	x	y
A:	.255	.075
B:	.500	.100
C:	.745	.075
D:	1.00	0.00

Solution: Biconvex Airfoil (with splines)



Example: Fuselage X-section (with Beziers)

Hint: the Bezier control points are constrained in the same way as any other point



Measurements:
width = 5.00
h_upper = 2.00
h_lower = 1.00
a_upper = 10°
a_lower = 5°

Note:
4 Bezier Cubics

Solution: Fuselage X-section (with Beziers)