

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

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

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

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

- 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

- 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)



```
# 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
```

- 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

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

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

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**
 - ...

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

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

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**

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**

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**

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** or **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

- 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

- 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

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

- 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

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

- 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)

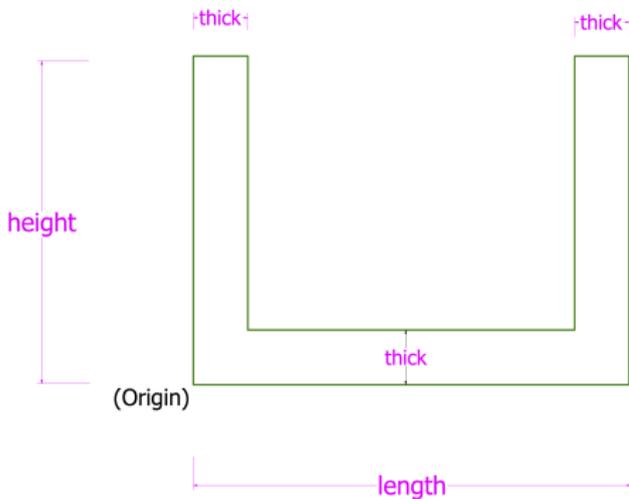
- Recenter Sketch
 - **Ctrl-h** key or **H** button
- Move the Sketch to the left
 - **Ctrl-l** key or **L** button or ← key
- Move the Sketch to the right
 - **Ctrl-r** key or **R** button or → key
- Move the Sketch to the bottom
 - **Ctrl-b** key or **B** button or ↓ key
- Move the Sketch to the top
 - **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

- 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

- 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



Measurements

length = 4.00

height = 3.00

thick = 0.5

The screenshot displays the ESP (Engineering Simulation Platform) software interface. The main workspace shows a 2D model of a U-bracket, colored green, with vertices labeled H1 through H10. The interface includes a file explorer on the left, a design parameters panel, and a console window at the bottom.

Design Parameters:

- Local Variables:
 - Branches:
 - Branch_000001: skbg
 - Branch_000027: skrd
 - Branch_000028: sldi
 - Branch_000029: asct
 - Branch_000030: asct
 - Branch_000031: asct
 - Branch_000032: asct
 - Branch_000033: asct
 - Branch_000034: asct
 - Branch_000035: asct
 - Branch_000036: asct
 - Branch_000037: asct
 - Display:
 - Via: Crd
 - Body 10: Via
 - Axis: Via
 - Display Type: Via
 - Display Filter: Via

Console Output:

```

ndof=16  ndof=16
Valid constraints at points
' (fx x) ' (fy y)
' (zprz) ' (tanp65)
' (ang1)
' (width) ' (depth)
Valid constraints on segments
' (horiz) ' (vertical)
' (isoline) ' (length)
Valid constraints on circles

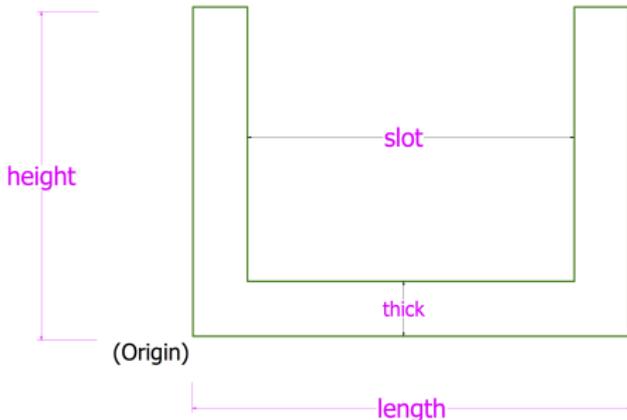
```

ESP has been initialized and is attached to 'servcom'.
 ../data/training/session2/Ubracket1.csm has been loaded



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).



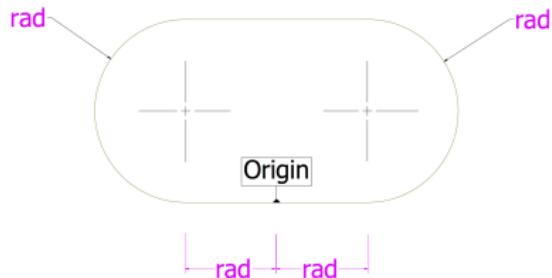
Measurements

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

Note: slot
is centered

Example: Oval

Hint: tangency constraints may be useful for this case



Measurements:

rad = 0.50

The screenshot shows the ESP (Engineering Sketch Pad) interface. The main workspace displays a green oval shape with several yellow constraint points (A through S) placed around its perimeter. A coordinate system with X and Y axes is visible. The left sidebar contains a tree view with sections for Design Parameters, Local Variables, Branches, and Display. The bottom panel shows a command window with the following text:

```

ndof=12  nump=12
Valid constraints at points
  ' ( (x x) ' ( (x y)
  ' ( (y x) ' ( (y y)
  ' ( (x) ' ( (tanget))
  ' ( (width) ' ( (depth)
Valid constraints on segments
  ' (horiz) ' ( (vertical)
  ' ( (x) ' ( (y)
Valid constraints on circles
  ' ( (radius)
  
```

Below the command window, a status message reads: "ESP has been initialized and is attached to 'servicem'". A log entry below that states: ".../data/training/session2/oval.cem" has been loaded".

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

File Search Snap/Pro Help

Search/Find Undo

M L R B T

- Design Parameters
- Local Variables
- Branches
 - Branch_000001 skbcg
 - Branch_000001 skrcd
 - Branch_000002 skcsi
 - Branch_000003 skcst
 - Branch_000004 skcst
 - Branch_000005 skcst
 - Branch_000006 skcst
 - Branch_000007 skcst
 - Branch_000008 skcst
 - Branch_000009 skcst
 - Branch_000010 skcst
 - Branch_000011 skcst
- Display
 - Body 4 Via Grid
 - Axis Via Grid
 - DisplayType
 - DisplayFilter

ndof=4 drcs=6

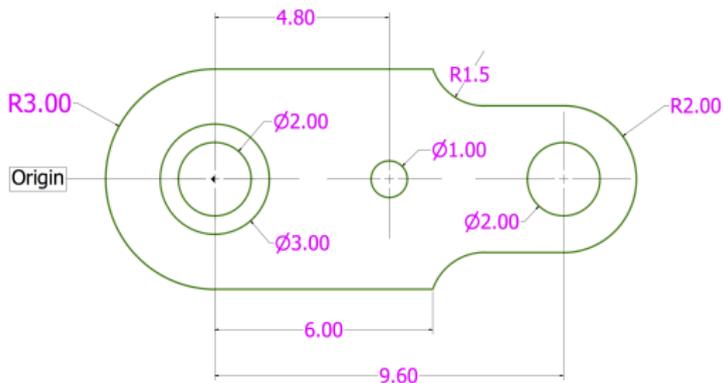
```
Valid constraints at points
'x' (fix x) 'y' (fix y)
'p' (geom) 't' (tangeth)
'g' (angl)
'w' (width) 'd' (depth)
Valid constraints on segments
'b' (horiz) 'v' (vertical)
'c' (cylind) 'l' (length)
Valid constraints on circles
'c' (radius) 't' (tangent)
```

ESP has been initialized and is attached to 'servoCOM'

../data/training/session2/biconvex_arsm.csm has been loaded

Example: Swivel Base

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



The screenshot shows the ESP software interface. The main window displays a 2D CAD model of a swivel base, which is a green, irregularly shaped object with several points labeled with letters (A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z). The model is centered on a coordinate system with yellow axes.

The left sidebar shows the design tree with the following structure:

- Design Parameters
- Local Variables
- Branches
 - Branch_000001 skleg
 - Branch_000031 skend
 - Branch_000032 outside
 - Branch_000033 cylinder
 - Branch_000034 subtract
 - Branch_000035 chamfer
 - Branch_000036 cylinder
 - Branch_000037 subtract
 - Branch_000038 cylinder
 - Branch_000039 subtract
 - Branch_000040 select
 - Branch_000041 insert
 - Branch_000042 insert
 - Branch_000043 insert
 - Branch_000044 insert
 - Branch_000045 insert
 - Branch_000046 insert
 - Branch_000047 insert
 - Branch_000048 insert
 - Branch_000049 insert
- Display
 - Axis Via Grid
 - Ready 18 Via Grid
 - Axis Via
 - DisplayType
 - DisplayFilter

The bottom console window shows the following output:

```

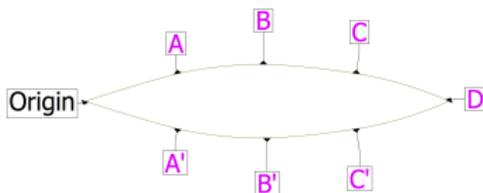
ndof=20  sloop=20
Valid constraints at points
' (fx x) ' (fy y)
' (px px) ' (tangent)
' (tangent)
' (width) ' (depth)
Valid constraints on segments
' (horiz) ' (vertical)
' (isoline) ' (length)
Valid constraints on circles

```

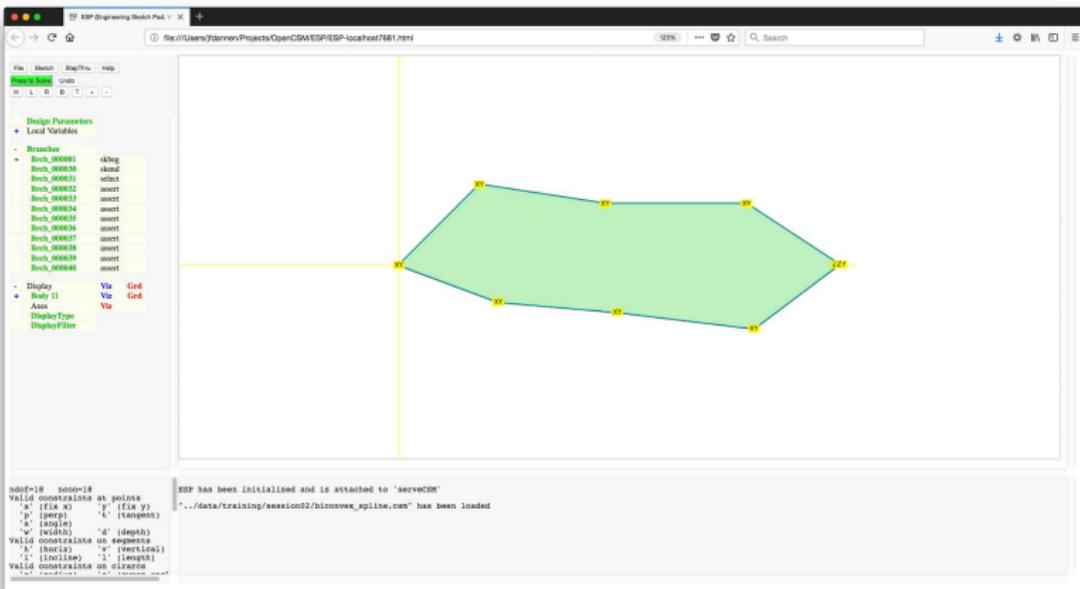
Below the console, a message states: "ESP has been initialized and is attached to 'servoCM'. './data/training/session2/swivelbase.cm' has been loaded."

Example: Biconvex Airfoil (with splines)

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

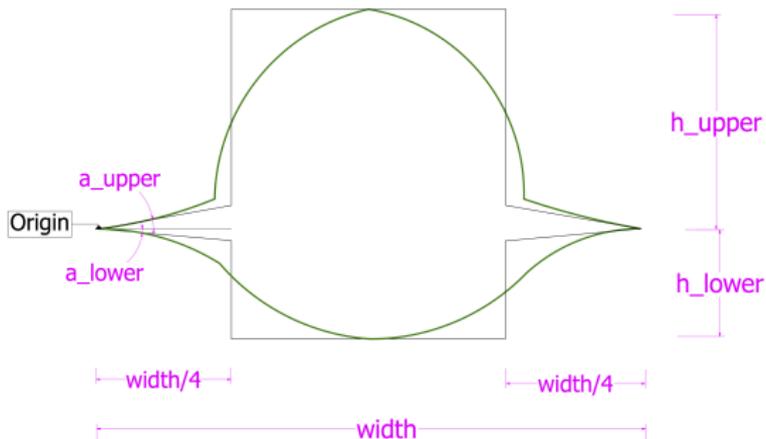


	x	y
A:	.255	.075
B:	.500	.100
C:	.745	.075
D:	1.00	0.00



Example: Fuselage X-section (with Beziers)

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

**Measurements:**

$width = 5.00$
 $h_{upper} = 2.00$
 $h_{lower} = 1.00$
 $a_{upper} = 10^\circ$
 $a_{lower} = 5^\circ$

Note:

4 Bezier Cubics

The screenshot displays the ESP (Engineering Sketch Pad) interface. The main workspace shows a green-shaded fuselage X-section defined by a series of Bezier curves. The curves are controlled by yellow control points labeled with letters (M, N, H, K, X, Y, D, Z). A coordinate system is shown with a vertical yellow line and a horizontal yellow line. The left sidebar contains a 'Design Parameters' panel with a tree view showing 'Local Variables' and 'Beziers'. The bottom panel shows a console window with the following text:

```

ndof=26  nump=24
Valid constraints at points
' (fx x) ' (fy y)
' (px) ' (tangent)
' (angle)
' (width) ' (depth)
Valid constraints on segments
' (horiz) ' (vertical)
' (isoline) ' (length)
Valid constraints on circles

```

Below the console window, a message states: "ESP has been initialized and is attached to 'servicem'". A file path is also visible: ".../data/training/session2/fuselage.cm".