Engineering Sketch Pad (ESP)

Exercise Solutions

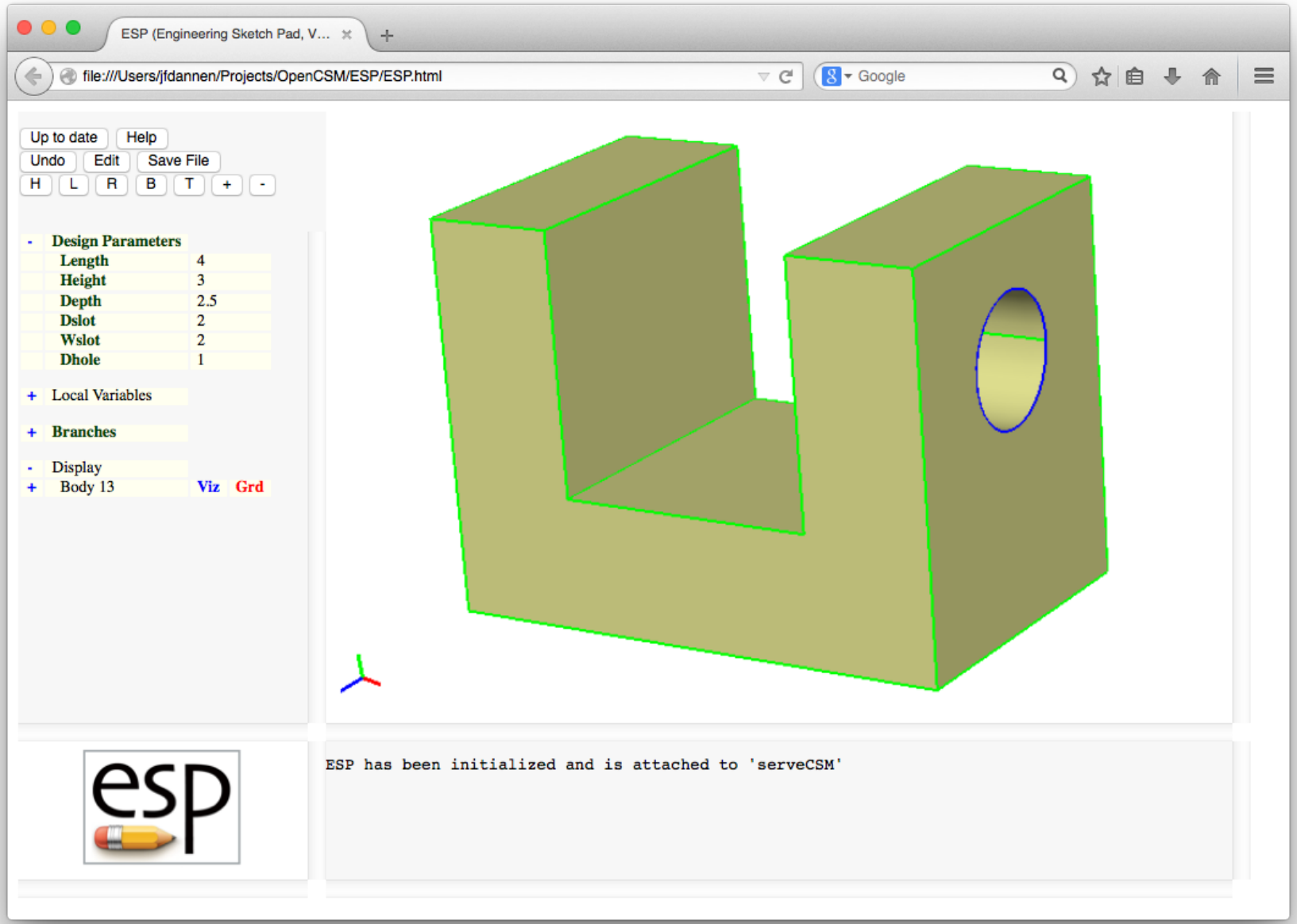
John F. Dannenhoffer, III
jfdannen@syr.edu
Syracuse University

Bob Haimes
haimes@mit.edu
Massachusetts Institute of Technology

updated for v1.19

Session 2 Solutions
Solids Fundamentals (1)

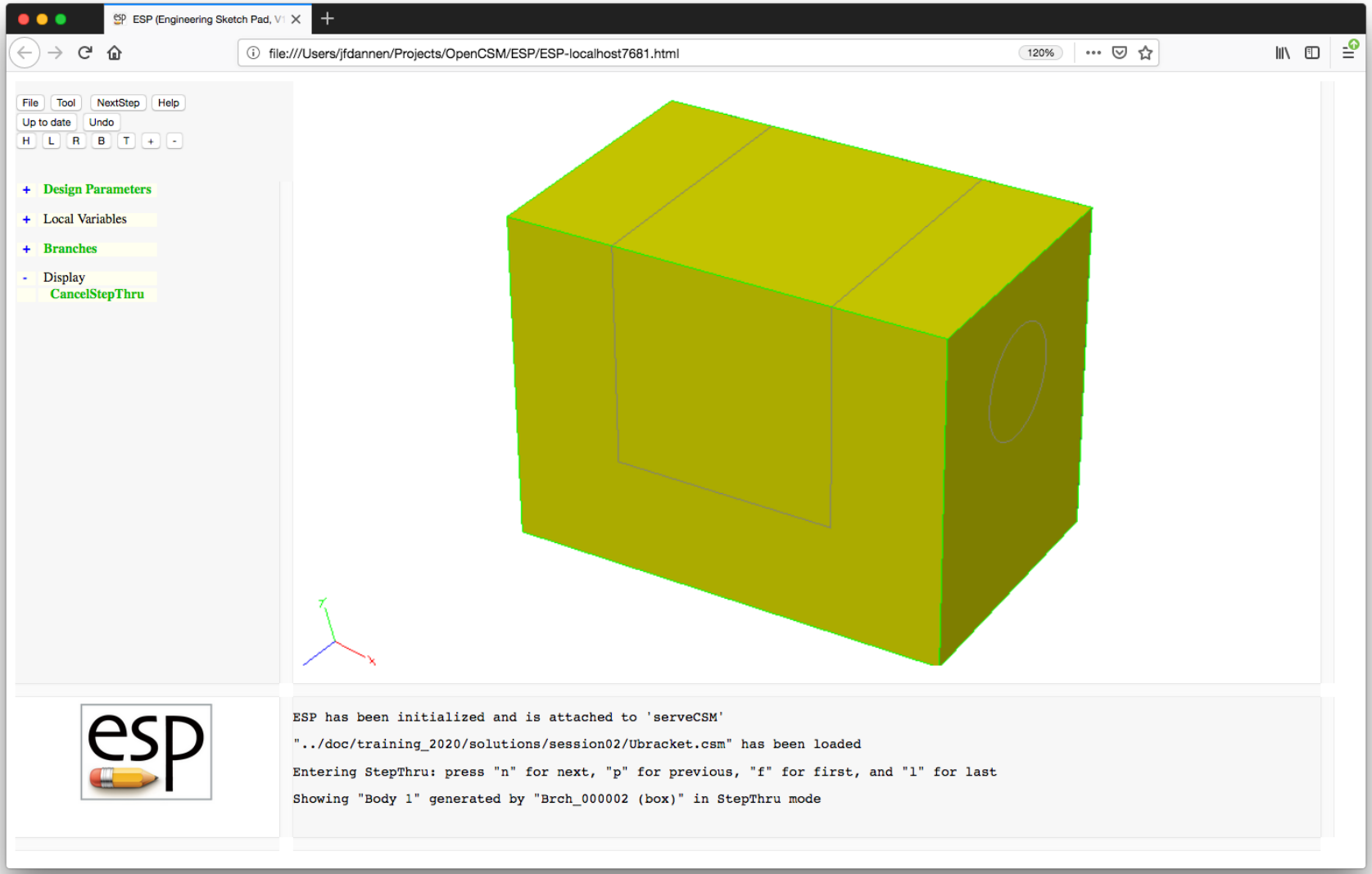
U-shaped Bracket with Hole (1)



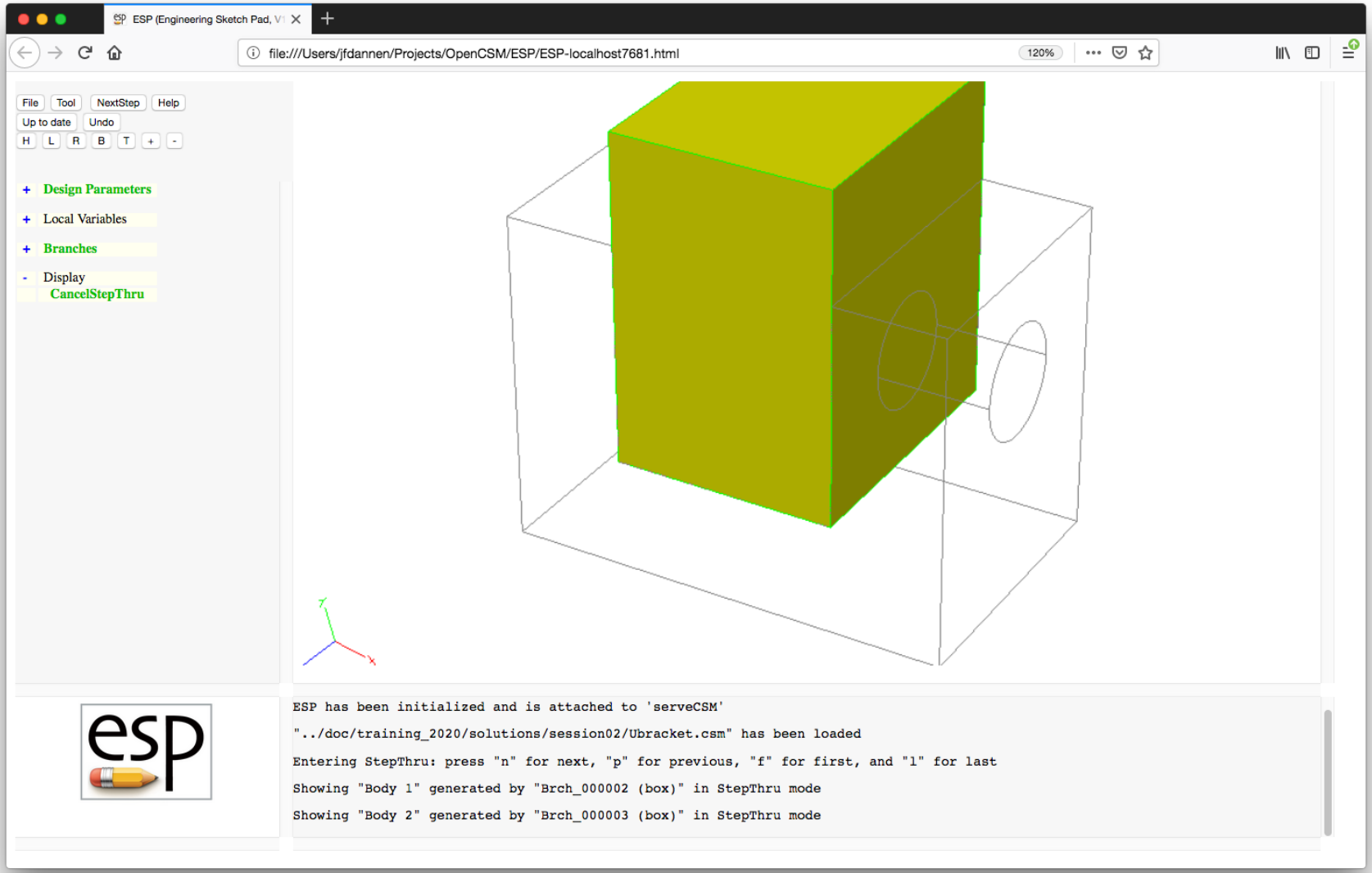
U-shaped Bracket with Hole (2)

Length	length in (X-direction)	4.00
Height	height of the two legs (Y-direction)	3.00
Depth	depth (in Z-direction)	2.50
Dslot	depth of slot (in Y-direction)	2.00
Wslot	width of slot (in X-direction)	2.00
	slot is centered in X-direction	
Dhole	diameter of hole	1.00
	hole is centered in Z-direction	
	center of hole is down Dhole from top	

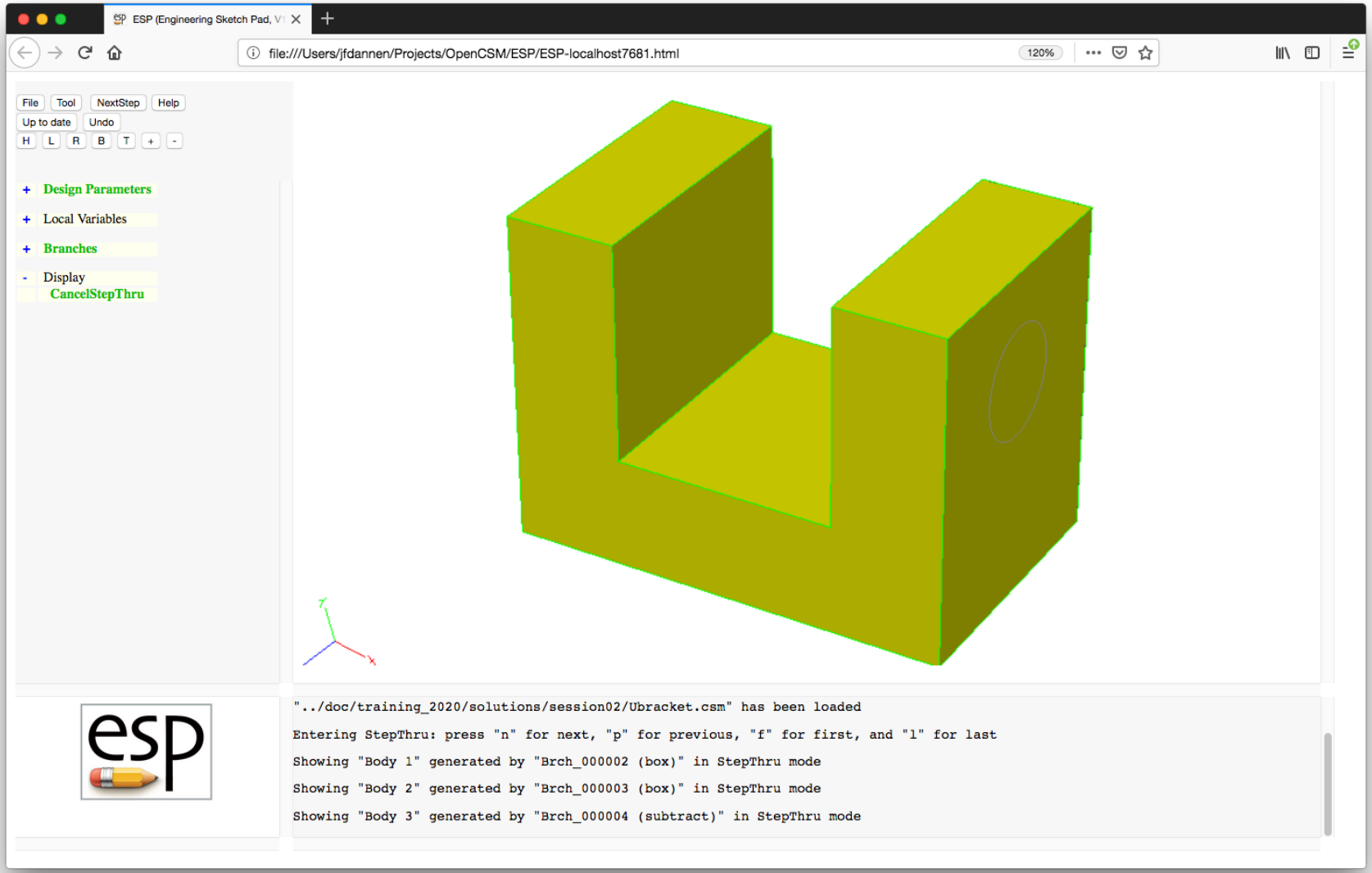
U-shaped Bracket — Step 1



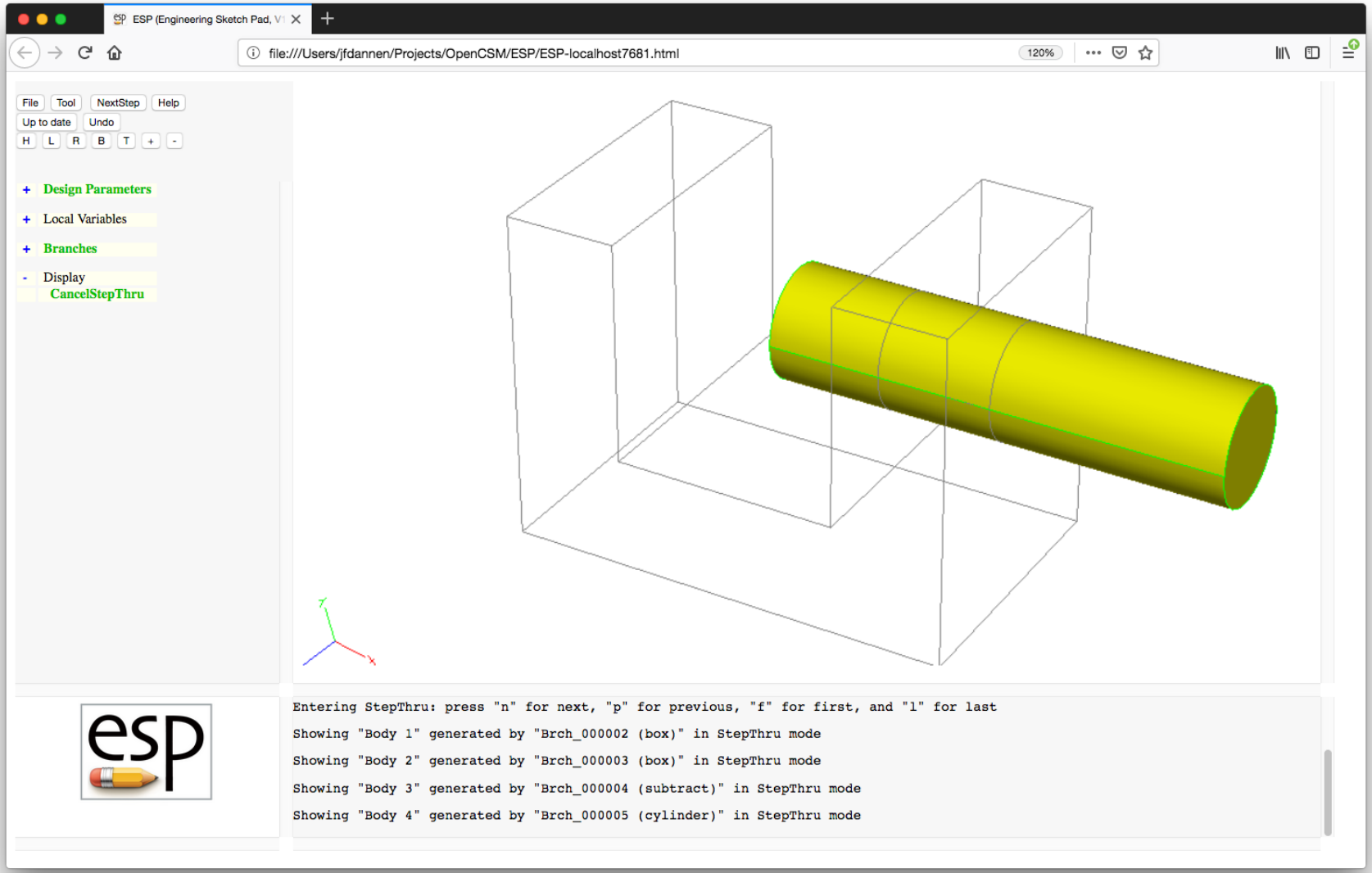
U-shaped Bracket — Step 2



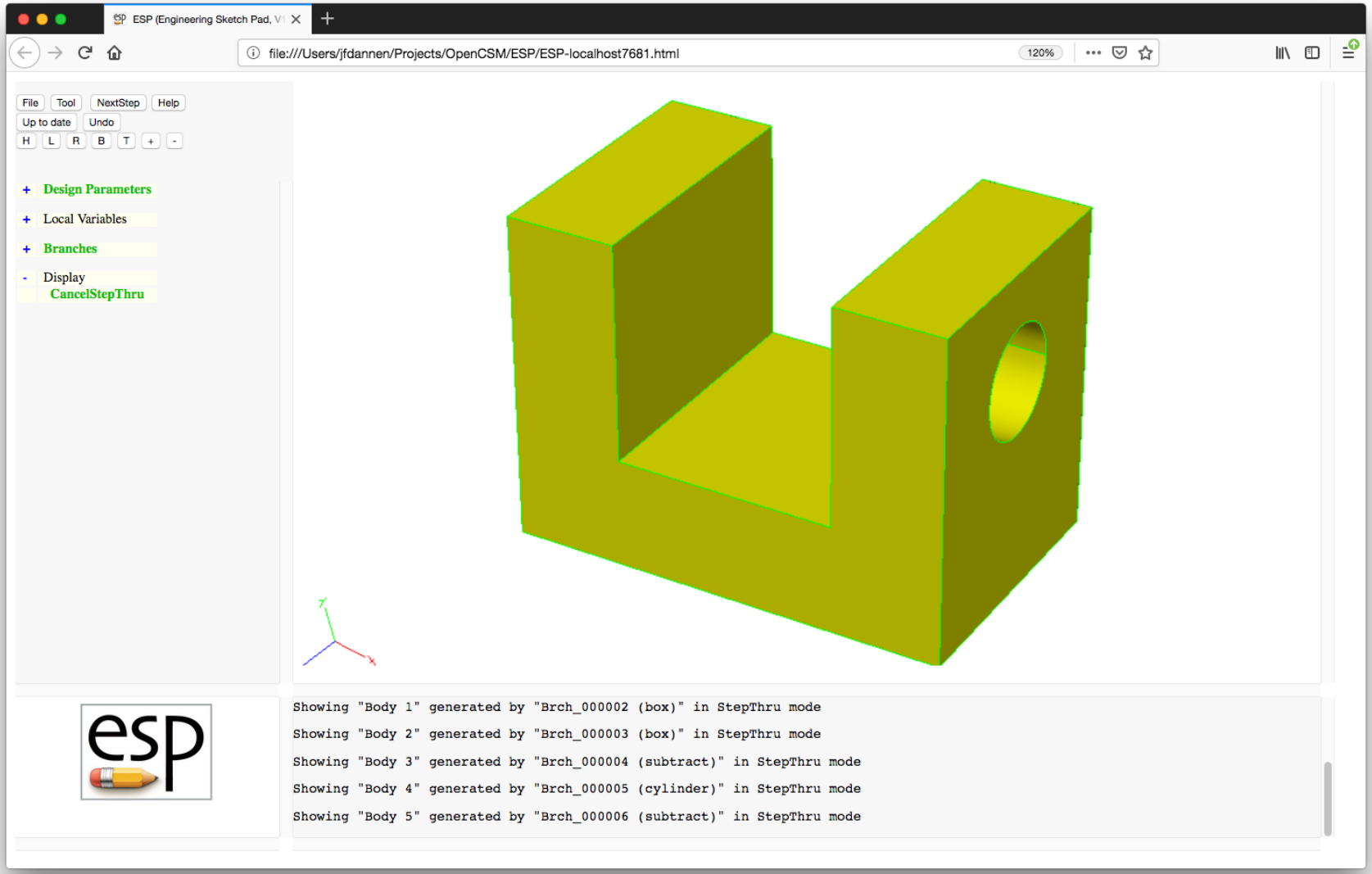
U-shaped Bracket — Step 3



U-shaped Bracket — Step 4



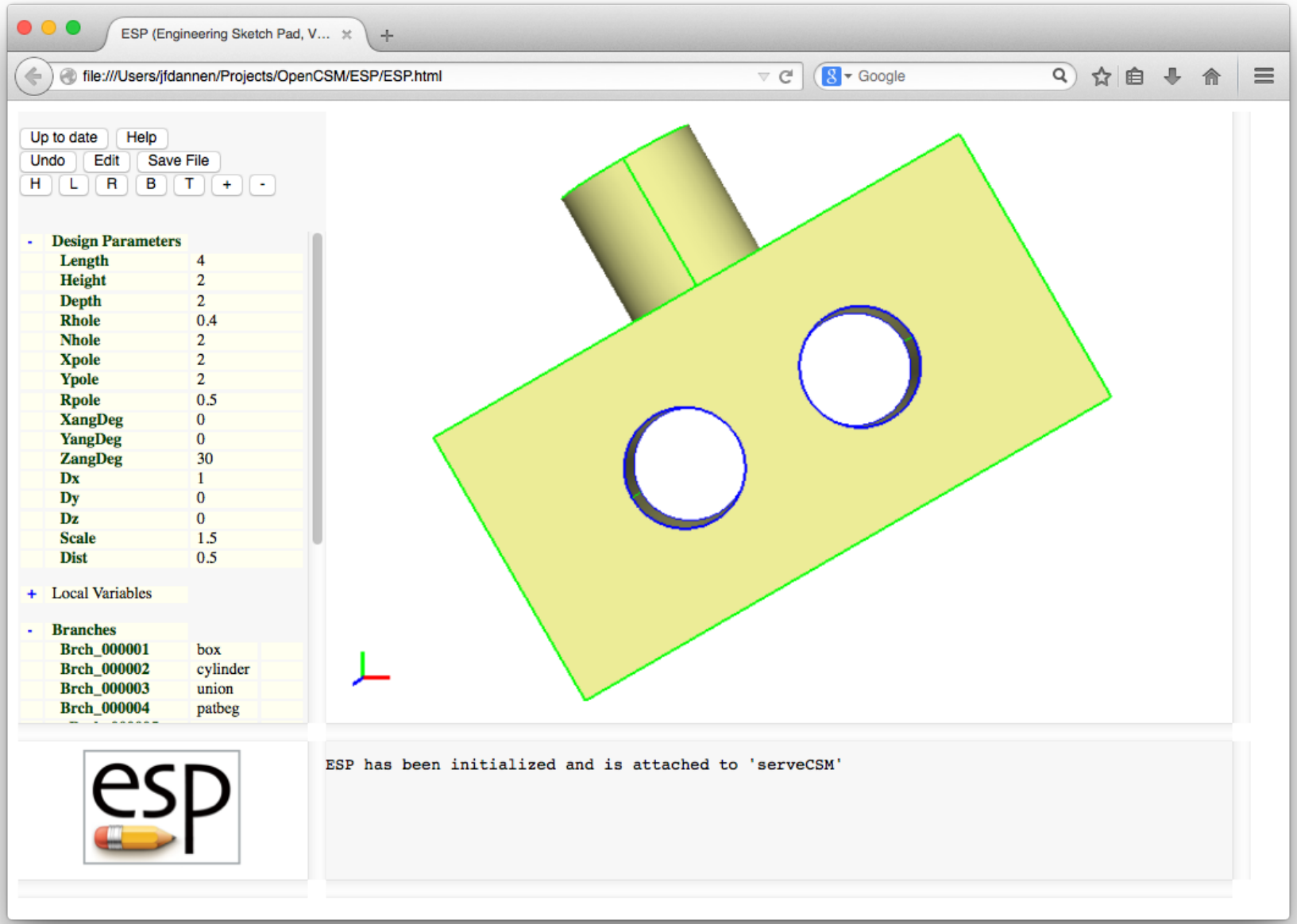
U-shaped Bracket — Step 5



U-shaped Bracket — `.csm` File

```
# Ubracket
# written by John Dannenhoffer

# design parameters
DESPMTR   Length    4.00      # length
DESPMTR   Height    3.00      # height
DESPMTR   Depth     2.50      # depth
DESPMTR   Dslot     2.00      # depth of slot
DESPMTR   Wslot     2.00      # width of slot
DESPMTR   Dhole     1.00      # diameter of hole

# bracket shape
SET       thick  (Length-Wslot)/2

BOX       0      0              0  Length          Height  Depth
BOX       thick  Height-Dslot   0  Length-2*thick  Height  Depth
SUBTRACT

# hole
CYLINDER  Length/2    Height-Dhole  Depth/2  \
          3*Length/2  Height-Dhole  Depth/2  Dhole/2
SUBTRACT

END
```

Simple Block (1)



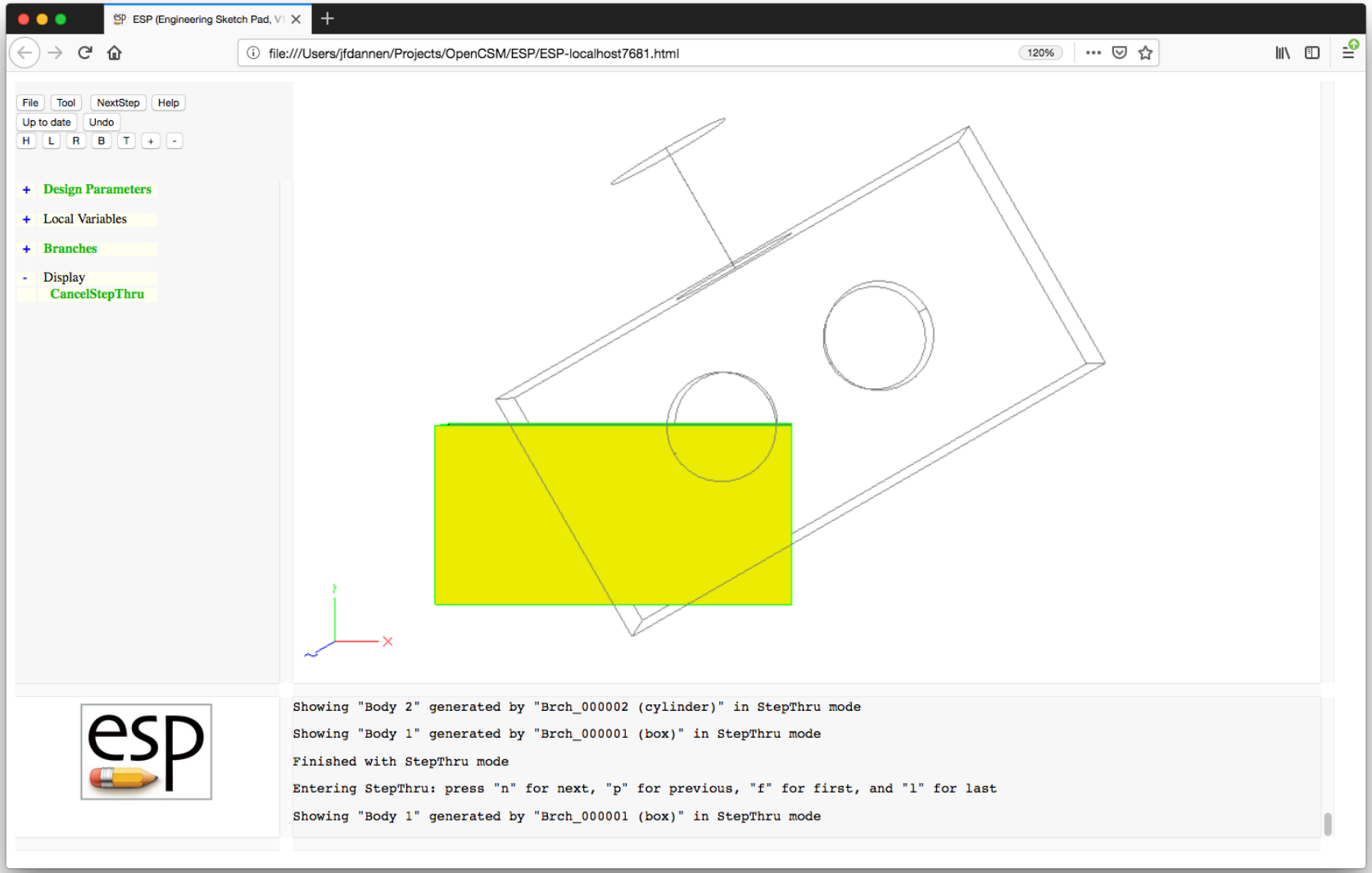
Simple Block (2)

Box		
Length	length of box	4.0
Height	height of box	2.0
Depth	depth of box	2.0
	anchored at $X = Z = 0$	
	centered at $Y = 0$	
Holes		
Rhole	radii of the holes	0.4
Nhole	number of holes	2
	holes are equally spaced	
Pole		
Xpole	X-location of top of pole	2.0
Ypole	Y-location of top of pole	2.0
Rpole	radius of pole	0.5

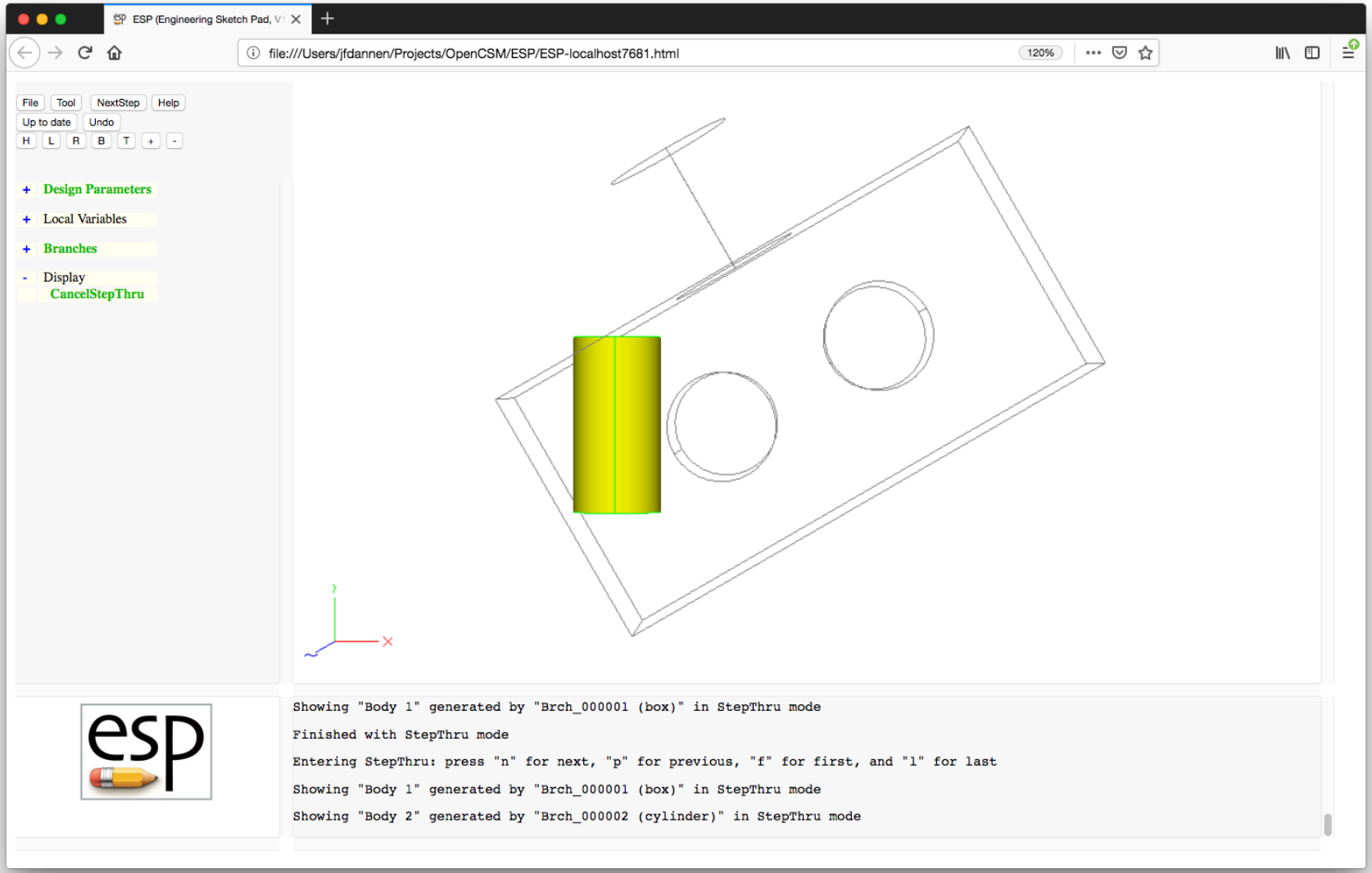
Simple Block (3)

Rotation about origin		
XangDeg	X rotation (deg)	0.
YangDeg	Y rotation (deg)	0.
ZangDeg	Z rotation (deg)	30.
Translation		
Dx		1.0
Dy		0.0
Dz		0.0
Scaling		
Scale	overall scaling factor	1.5

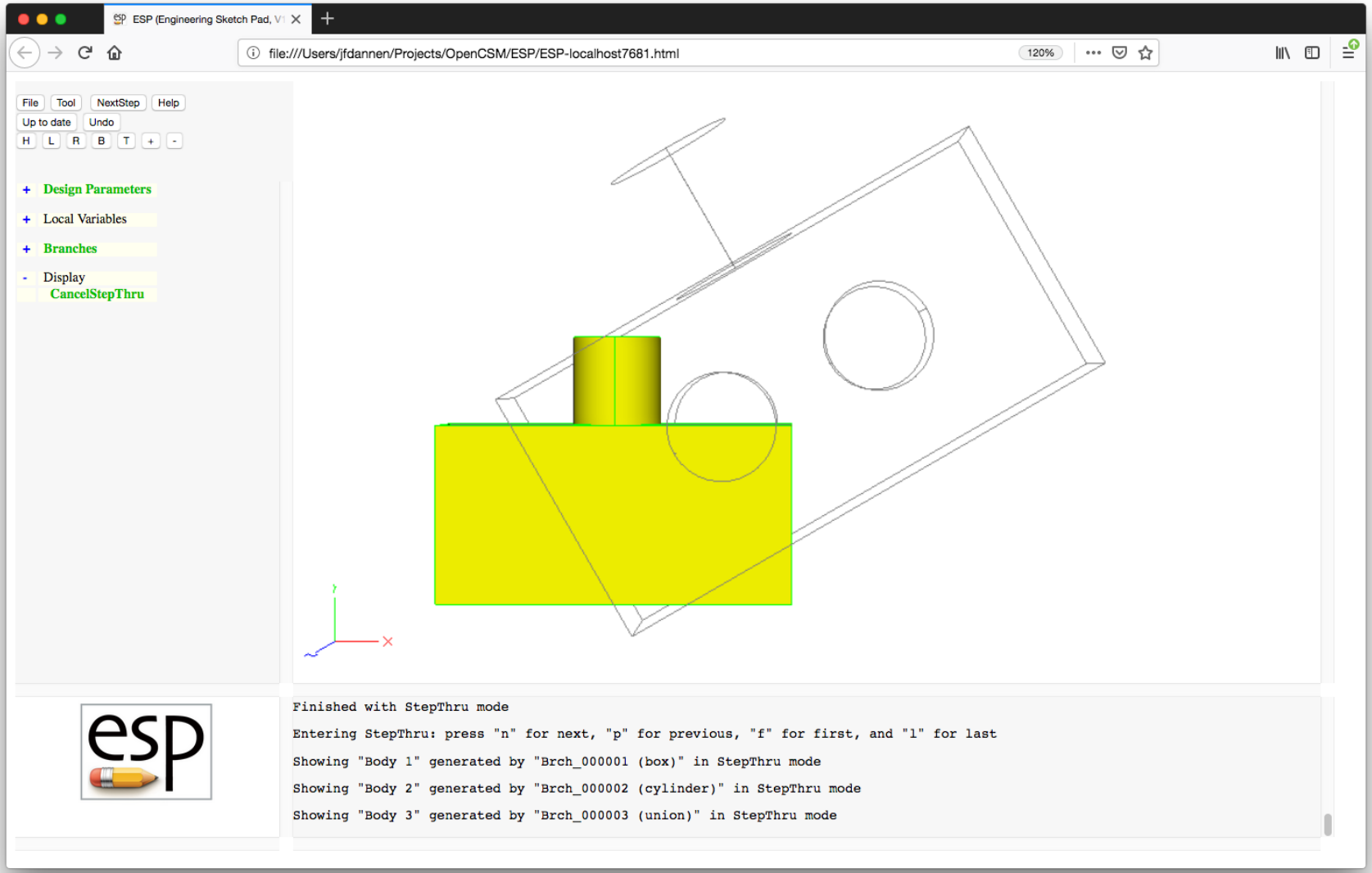
Simple Block — Step 1



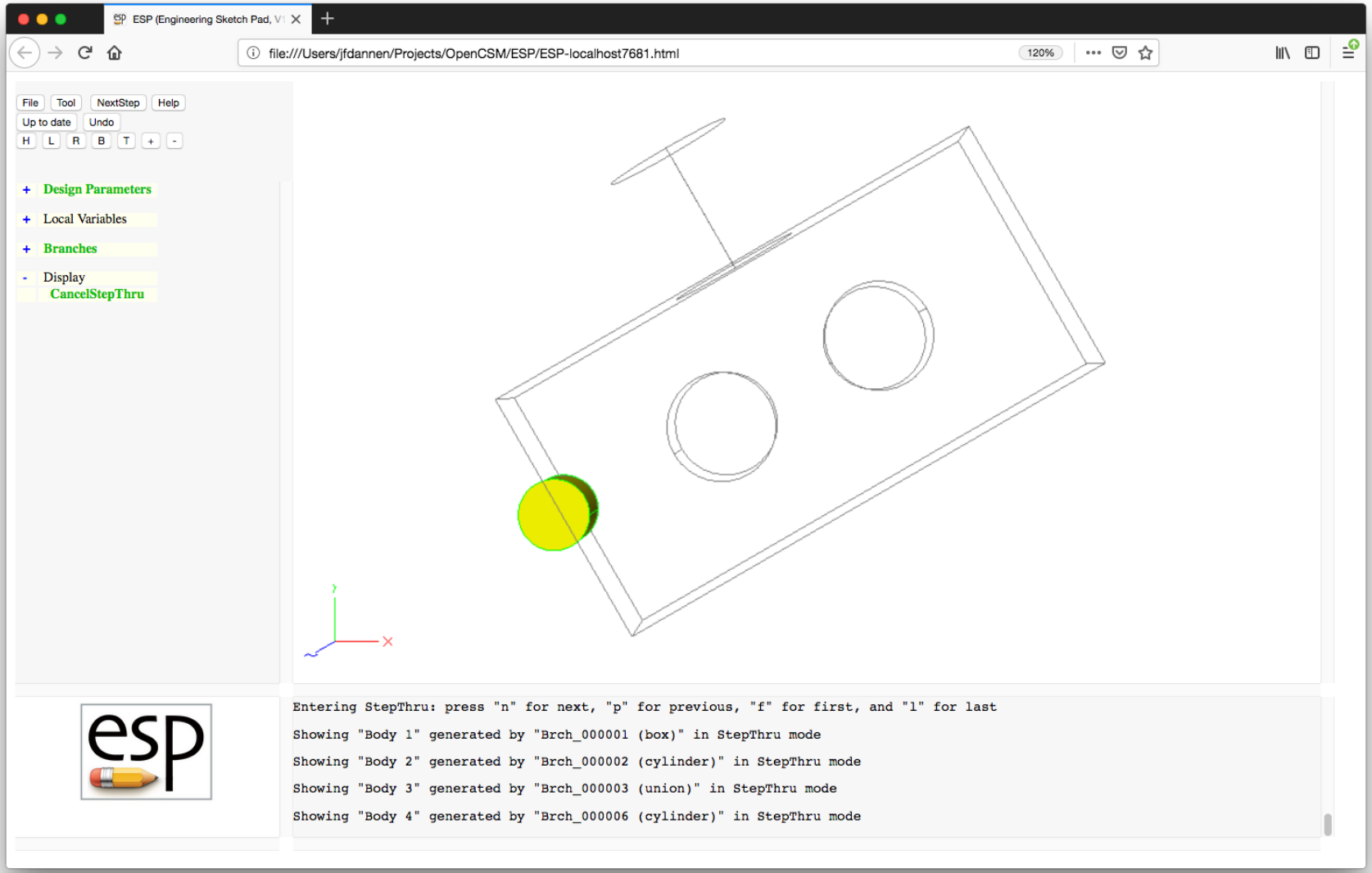
Simple Block — Step 2



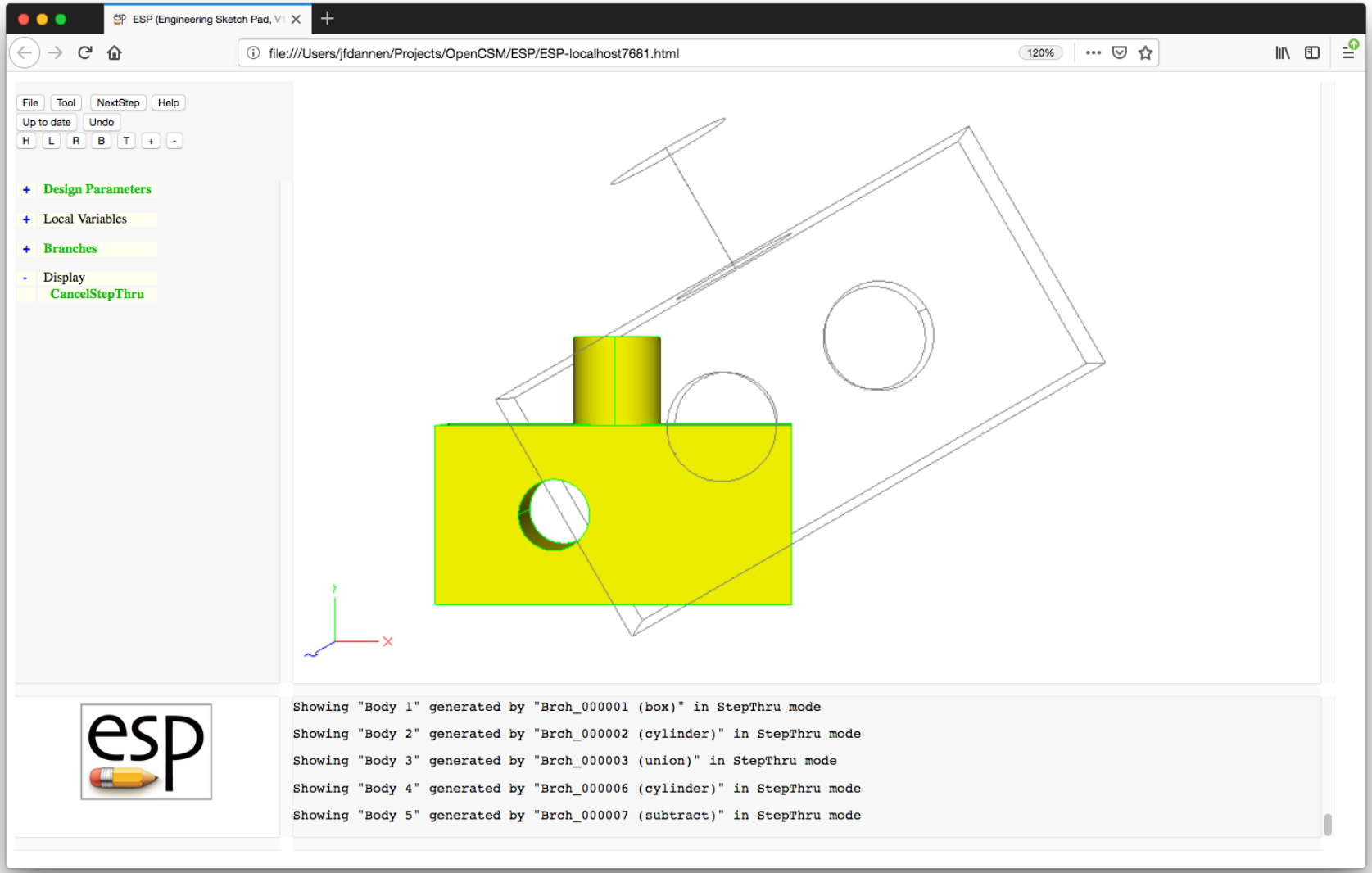
Simple Block — Step 3



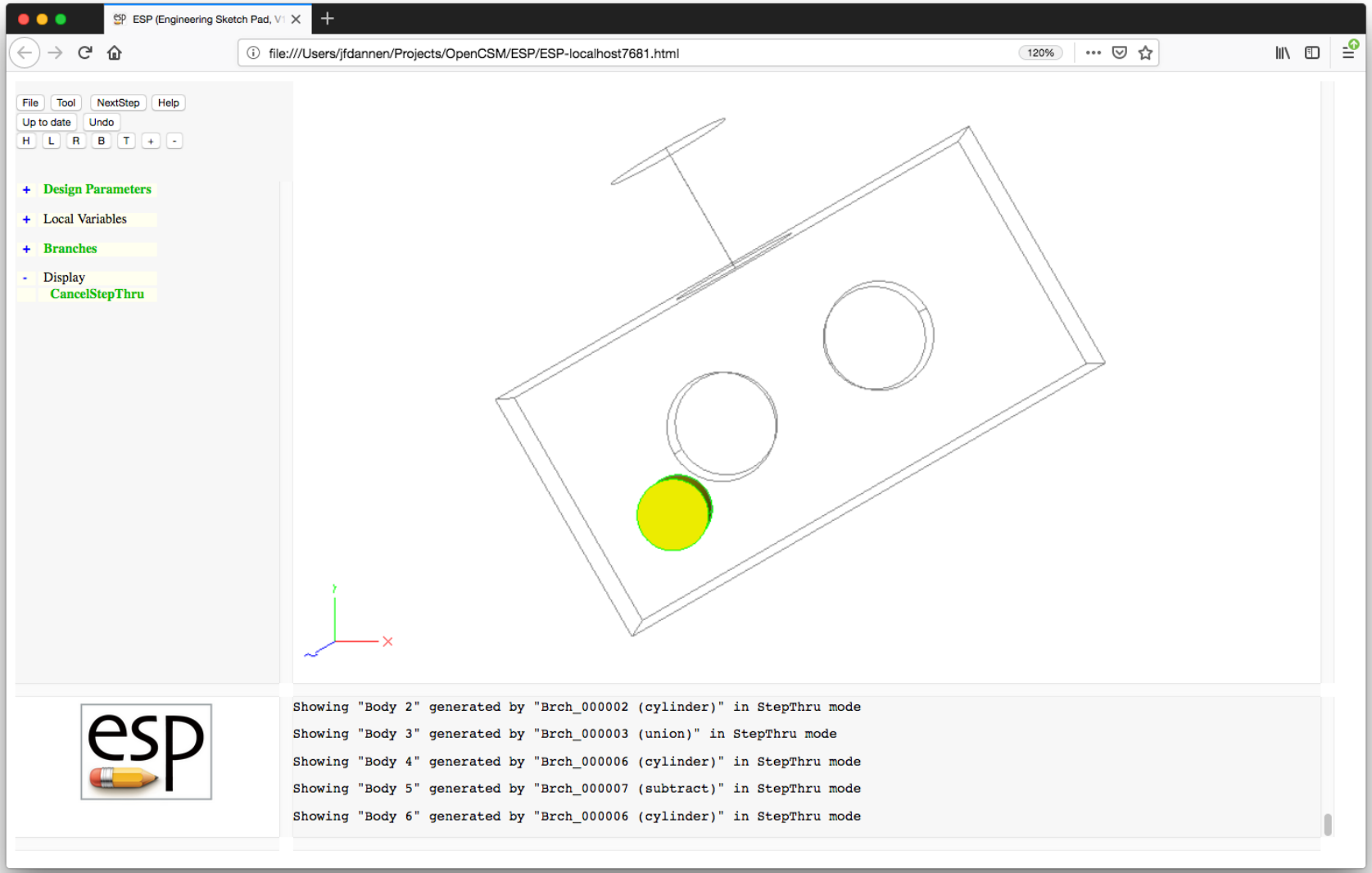
Simple Block — Step 4



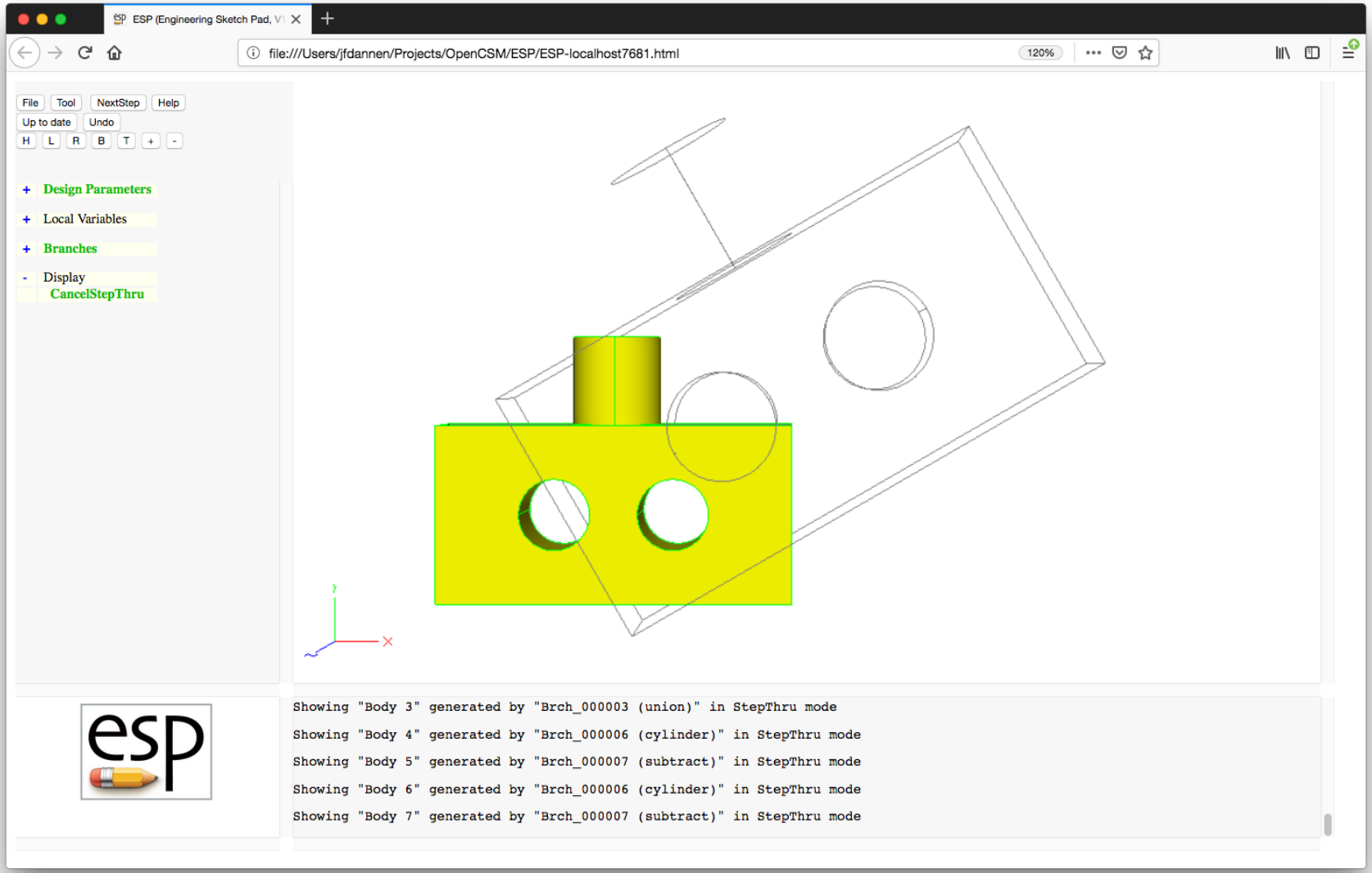
Simple Block — Step 5



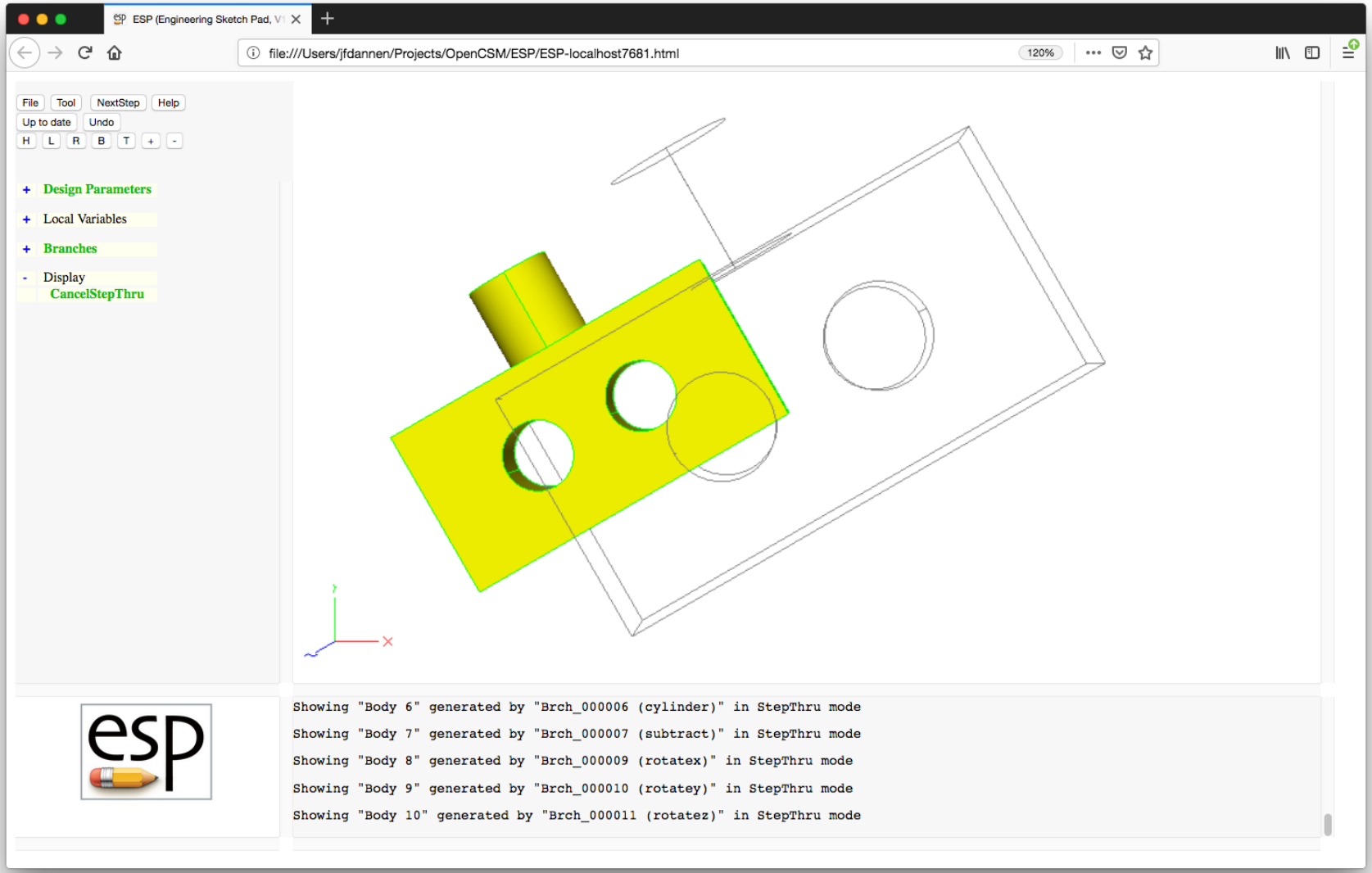
Simple Block — Step 6



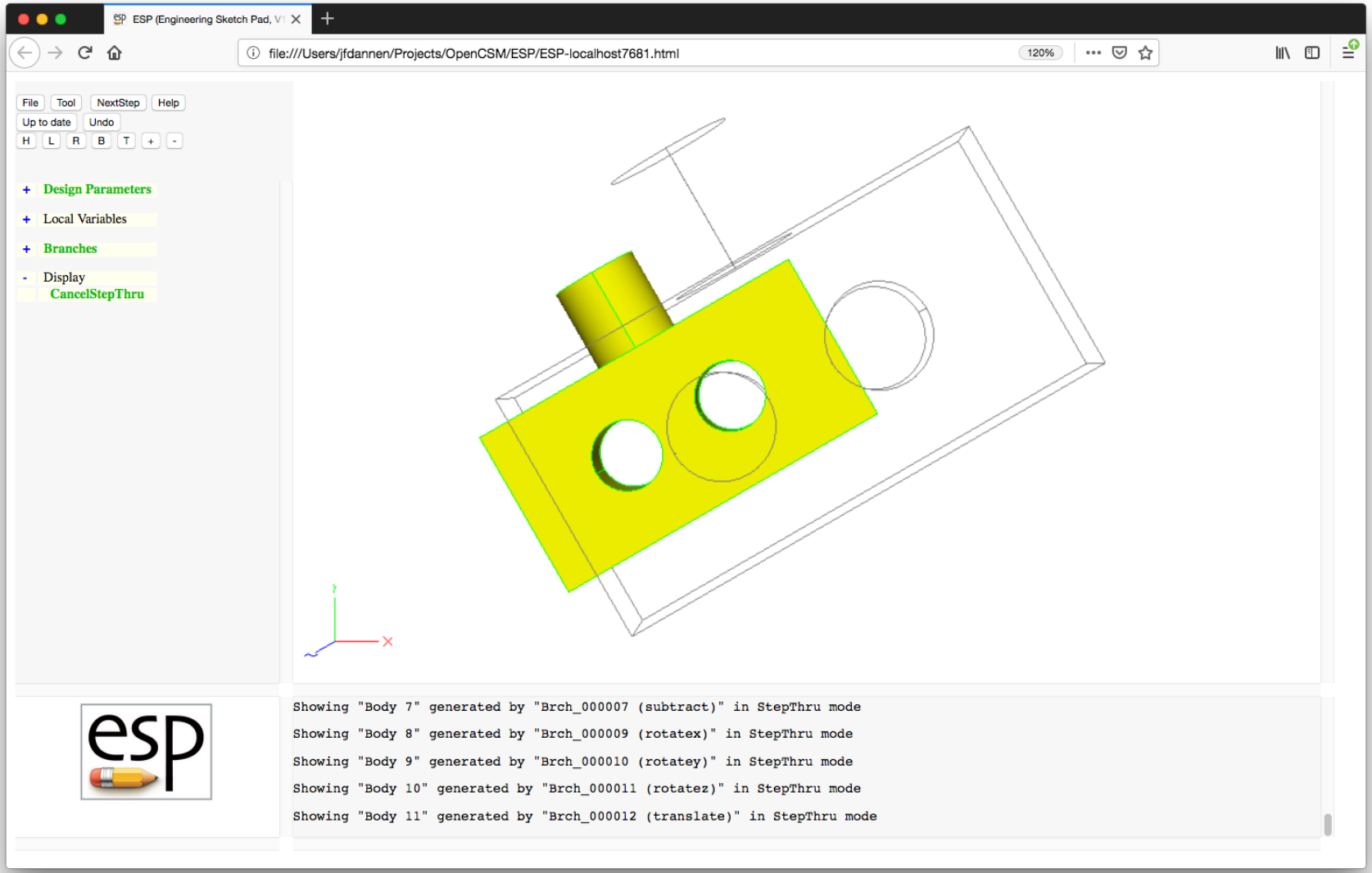
Simple Block — Step 7



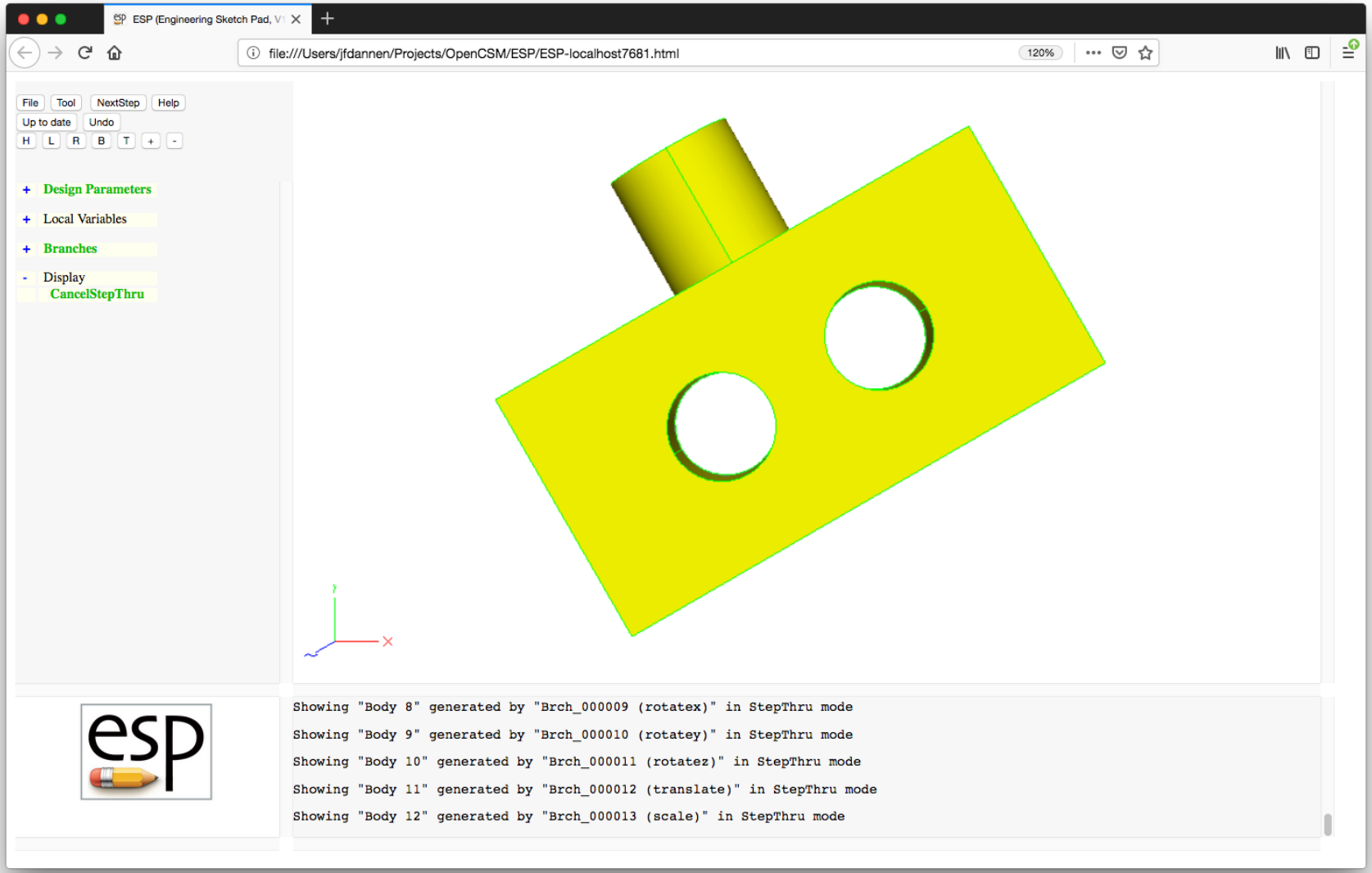
Simple Block — Step 8



Simple Block — Step 9



Simple Block — Step 10



Simple Block — `.csm` File (1)

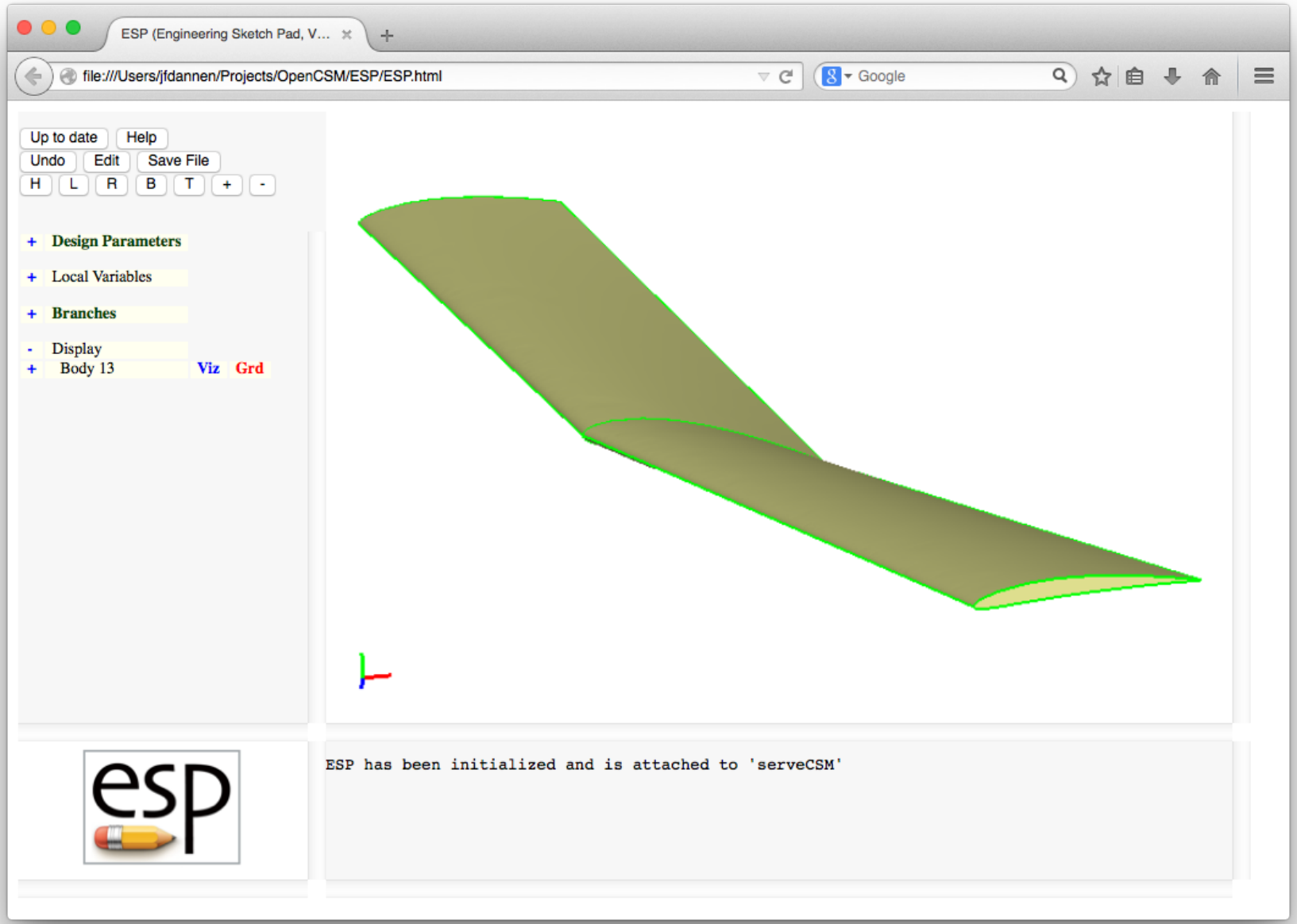
```
# block
# written by John Dannenhoffer

DESPMTR   Length    4.0
DESPMTR   Height    2.0
DESPMTR   Depth     2.0
DESPMTR   Rhole     0.4
DESPMTR   Nhole     2
DESPMTR   Xpole     2.0
DESPMTR   Ypole     2.0
DESPMTR   Rpole     0.5
DESPMTR   XangDeg    0.
DESPMTR   YangDeg    0.
DESPMTR   ZangDeg   30.
DESPMTR   Dx        1.0
DESPMTR   Dy        0.0
DESPMTR   Dz        0.0
DESPMTR   Scale     1.5
DESPMTR   Dist      0.5

# base block
BOX       0.0       -Height/2 0.0       Length    Height    Depth
```

Simple Block — `.csm` File (2)

```
# post
CYLINDER  Xpole     0.0       Depth/2   Xpole     Ypole     Depth/2   Rpole
UNION

# Nhole holes
PATBEG    ihole     Nhole
   SET       xhole  Length*ihole/(Nhole+1)
   CYLINDER  xhole  0.0       0.0       xhole     0.0       Depth     Rhole
   SUBTRACT
PATEND

# transformations
ROTATEX   XangDeg   0.0       0.0
ROTATEY   YangDeg   0.0       0.0
ROTATEZ   ZangDeg   0.0       0.0
TRANSLATE Dx        Dy        Dz
SCALE     Scale

END
```

Session 3 Solutions
Solids Fundamentals (2)

Simple Wing (1)



Simple Wing (2)

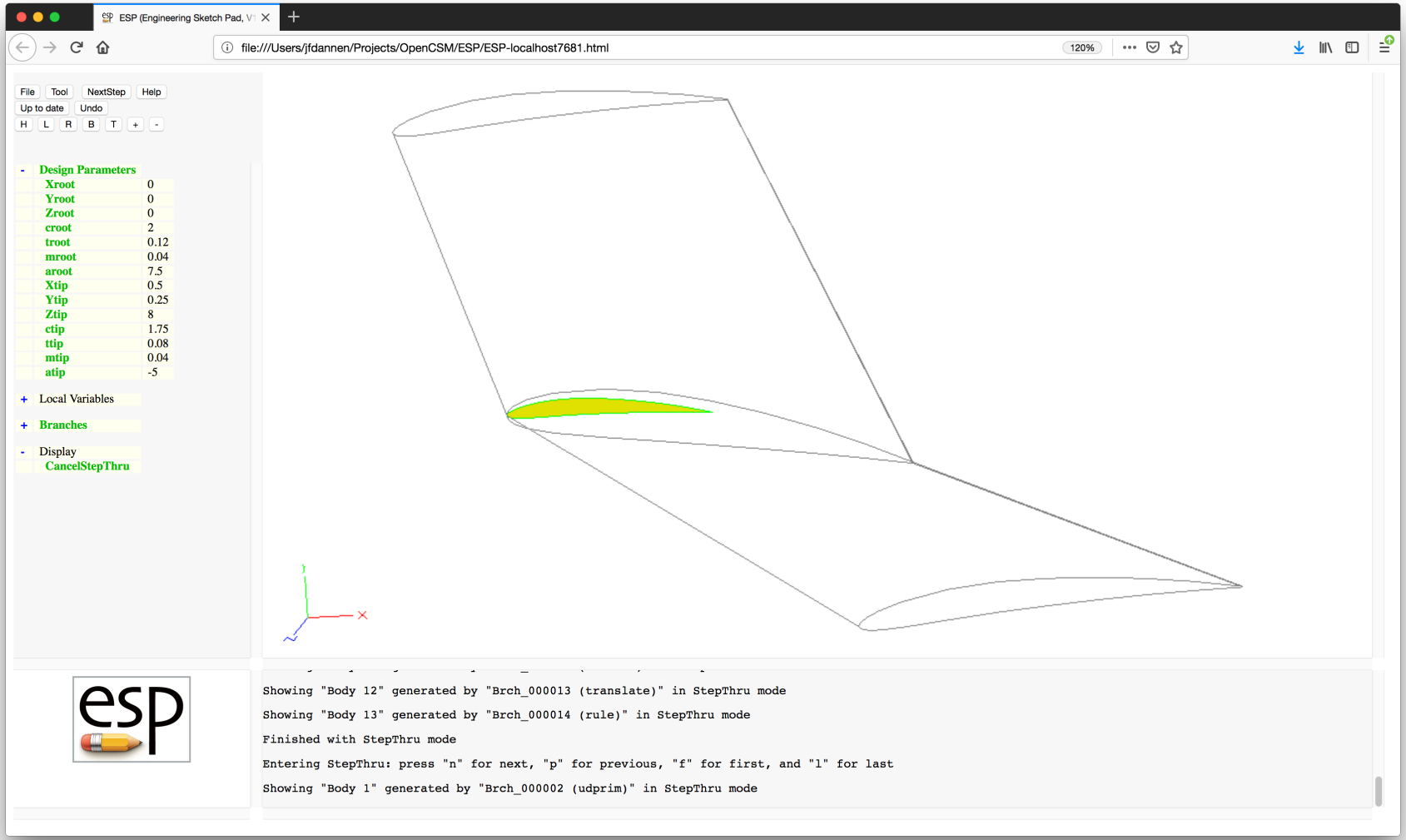
Xroot	X-coordinate of root leading edge	0.00
Yroot	Y-coordinate of root leading edge	0.00
Zroot	Z-coordinate of root leading edge	0.00
croot	chord of root	2.00
troot	thickness/chord of root	0.12
mroot	camber/chord of root	0.04
aroot	angle of attack of root (deg)	7.50
Xtip	X-coordinate of tip leading edge	0.50
Ytip	Y-coordinate of tip leading edge	0.25
Ztip	Z-coordinate of tip leading edge	8.00
ctip	chord of tip	1.75
ttip	thickness/chord of tip	0.08
mtip	camber/chord of tip	0.04
atip	angle of attack of tip (deg)	-5.00

Simple Wing (3)

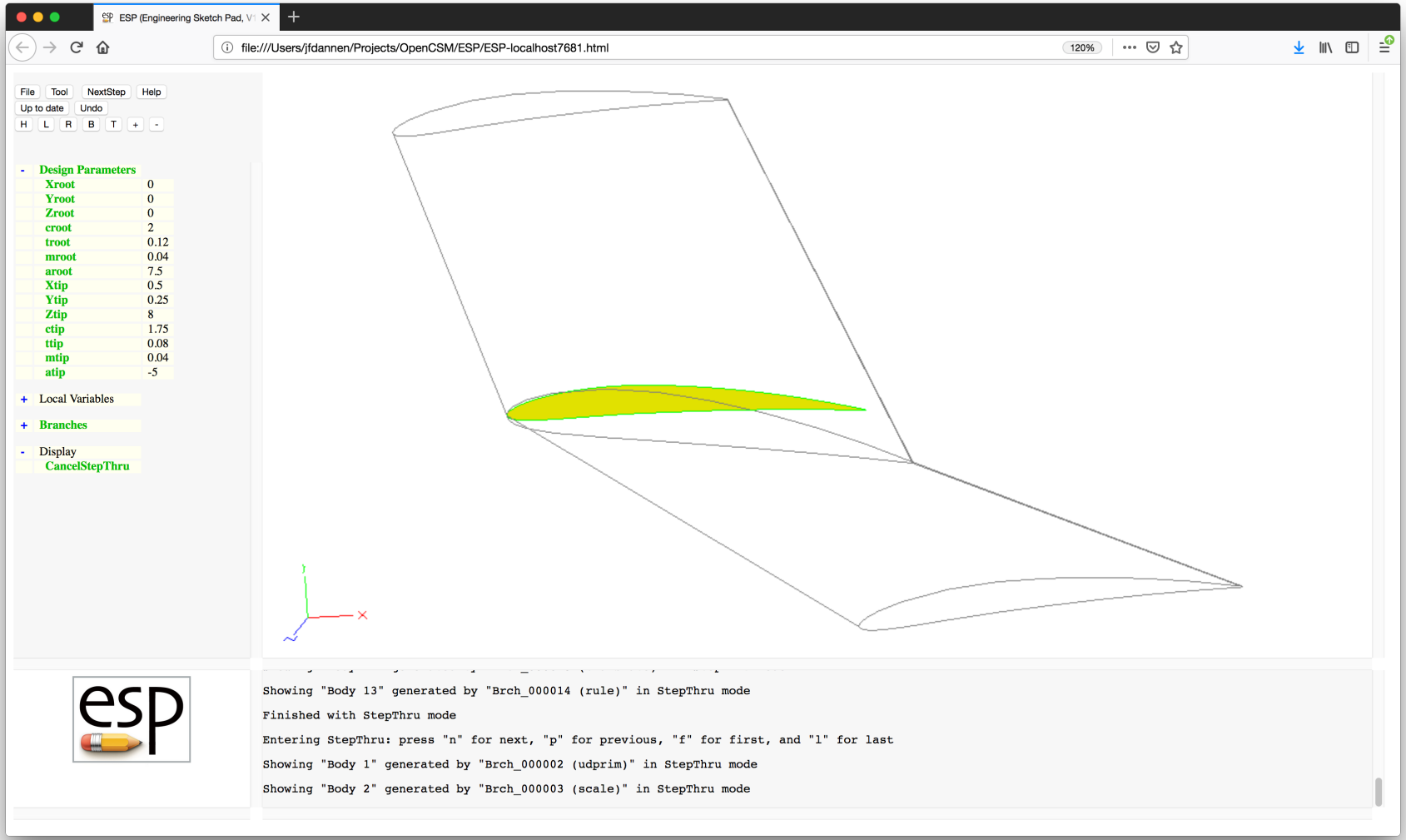
- What happens if you switch from `RULE` to `BLEND`?
- What happens if we change the sequence of transformations from `SCALE`, `ROTATEZ`, `TRANSLATE` to `ROTATEZ`, `SCALE`, `TRANSLATE`?
- What happens if we do the `TRANSLATE` first?
- Could you change the Design Parameters to `area`, `aspectRatio`, `taperRatio`, `sweep`, and `twist`?

$$AR = \frac{b^2}{S} \qquad S = b(c_{\text{tip}} + c_{\text{root}})/2 \qquad \tau = \frac{c_{\text{tip}}}{c_{\text{root}}}$$

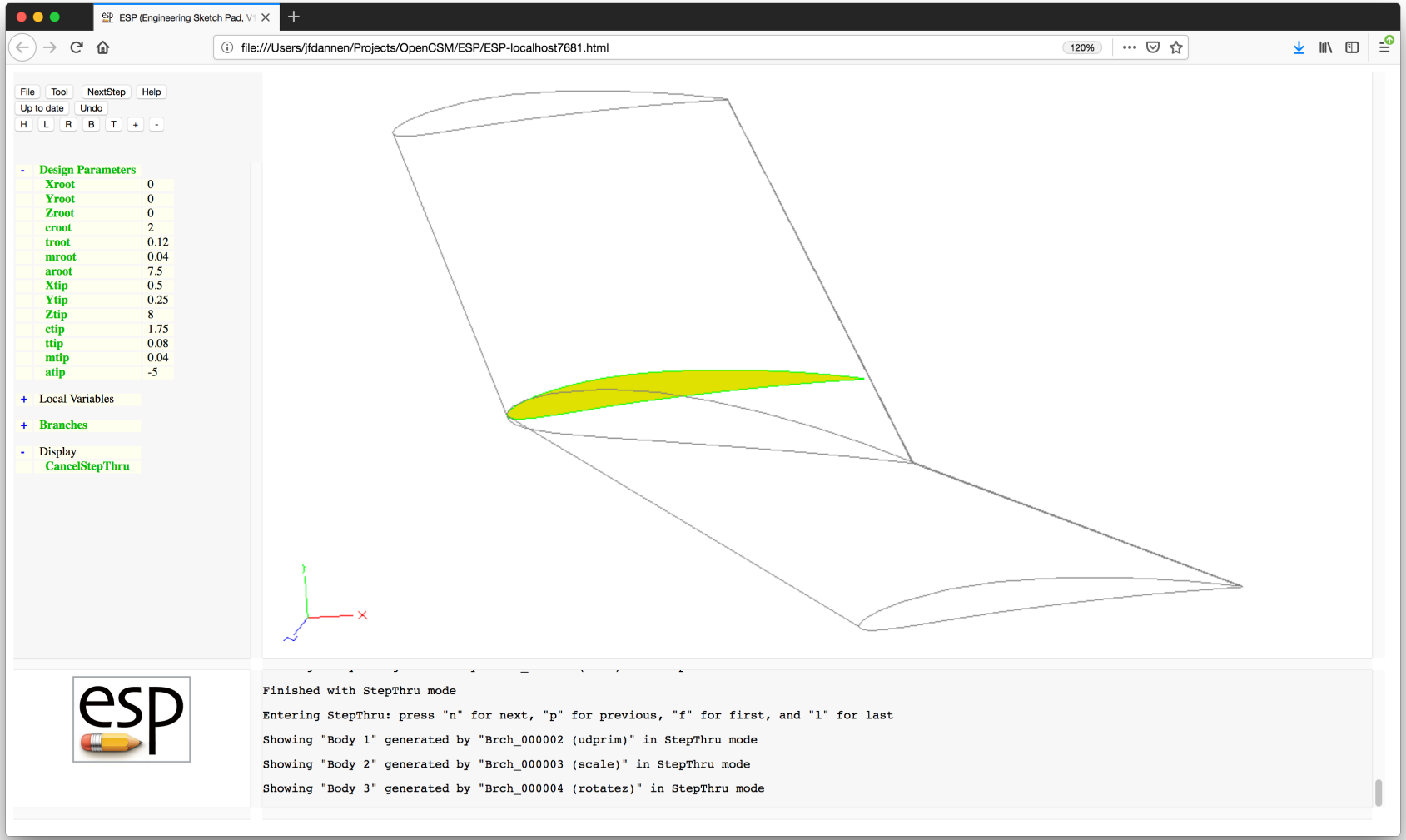
Simple Wing — Step 1



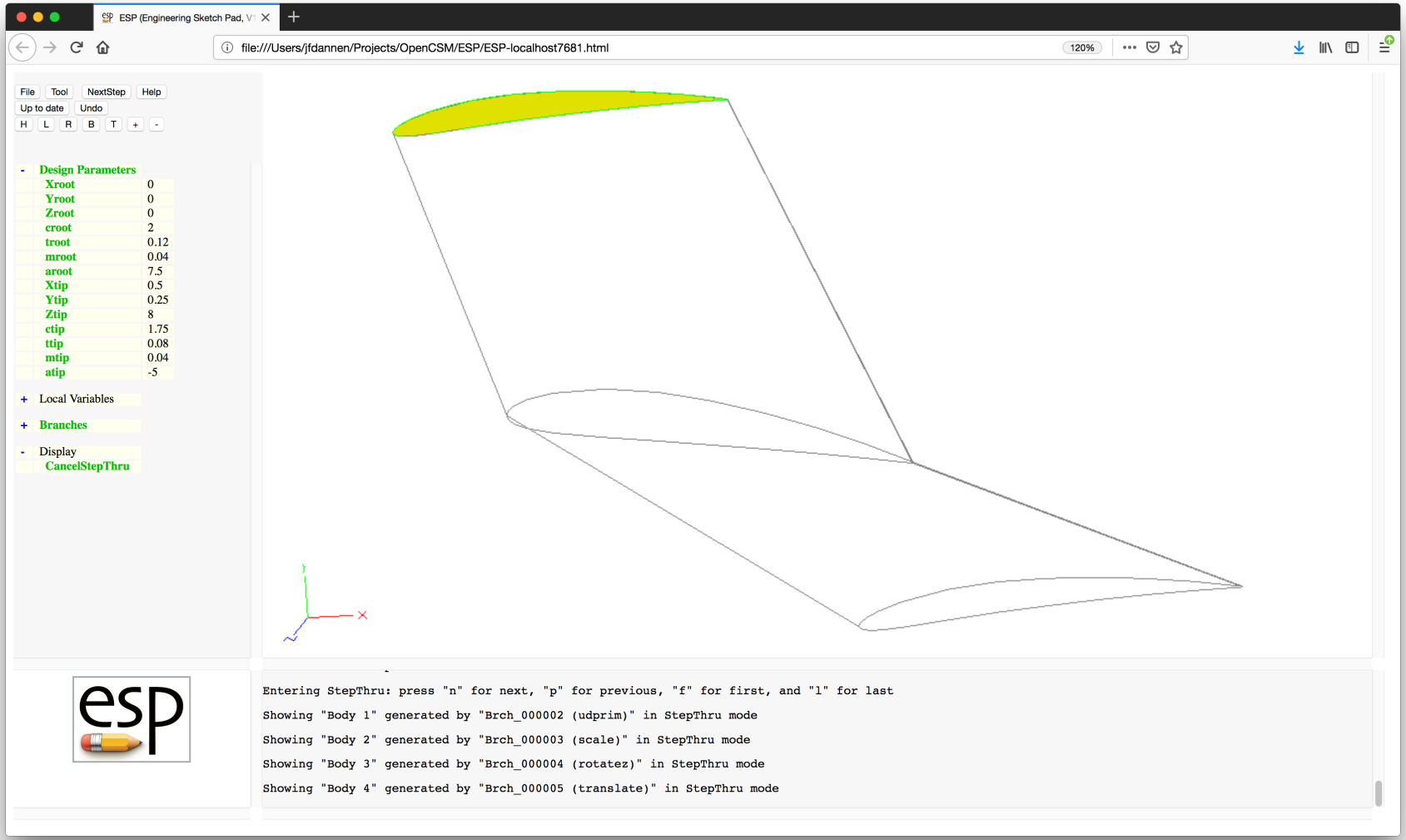
Simple Wing — Step 2



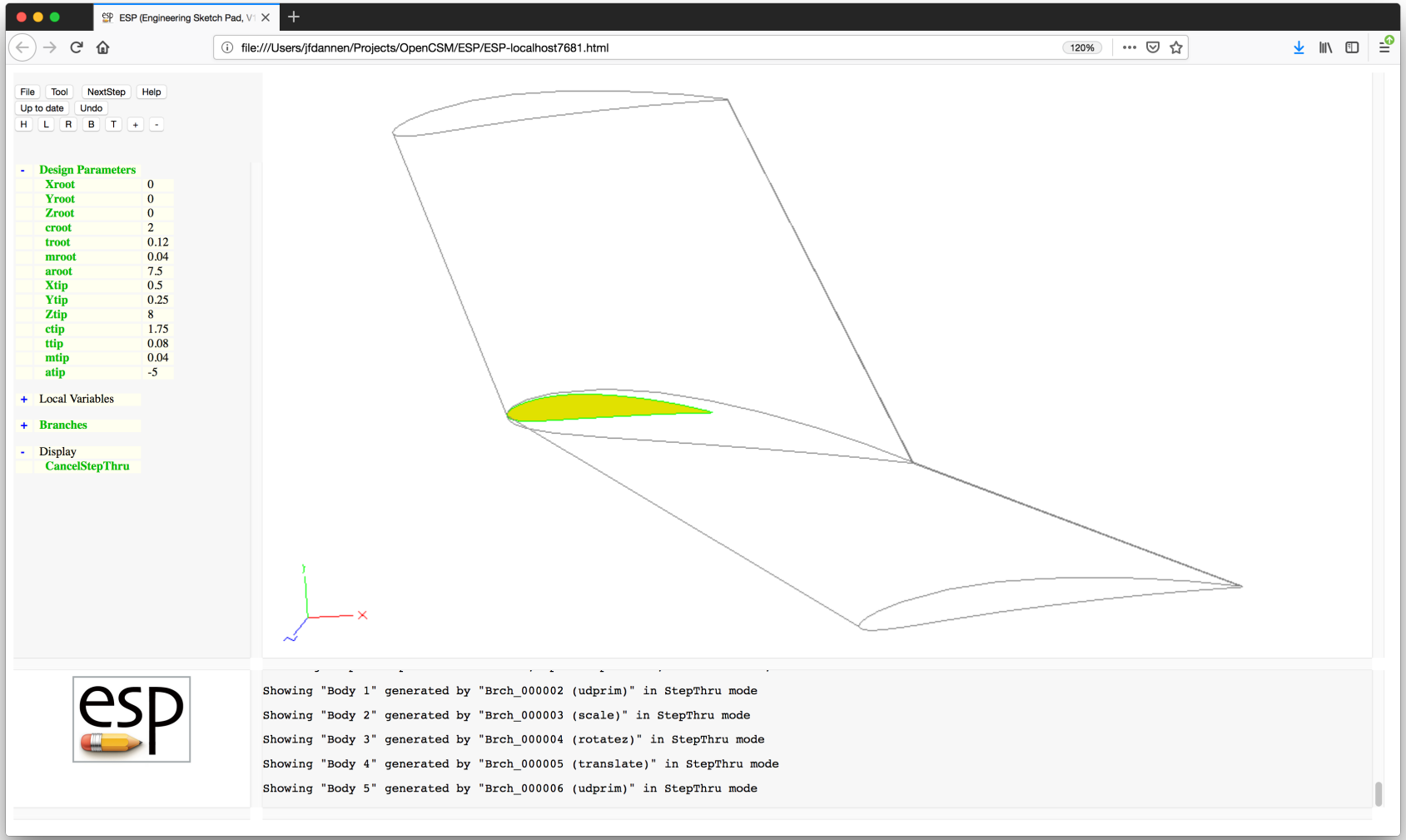
Simple Wing — Step 3



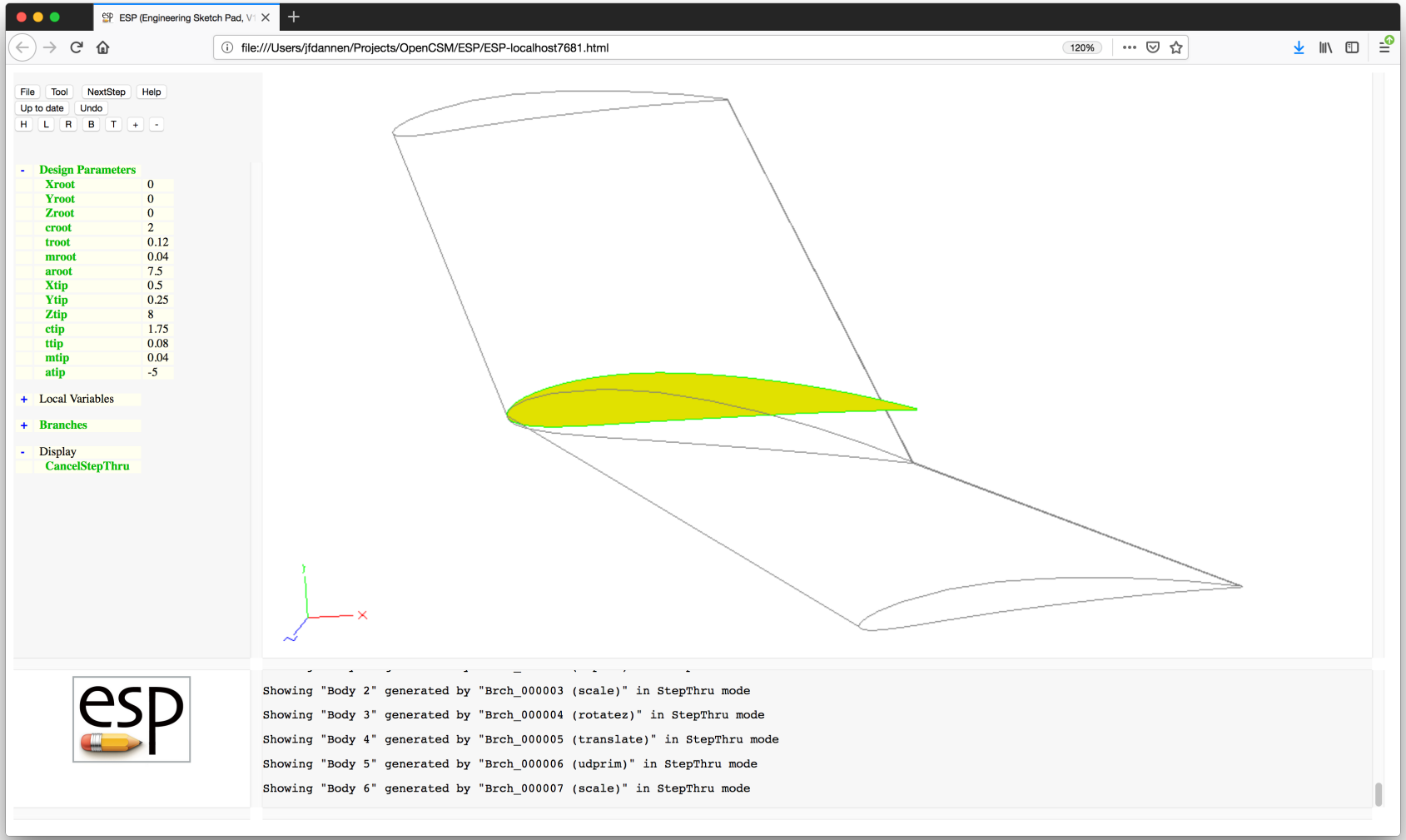
Simple Wing — Step 4



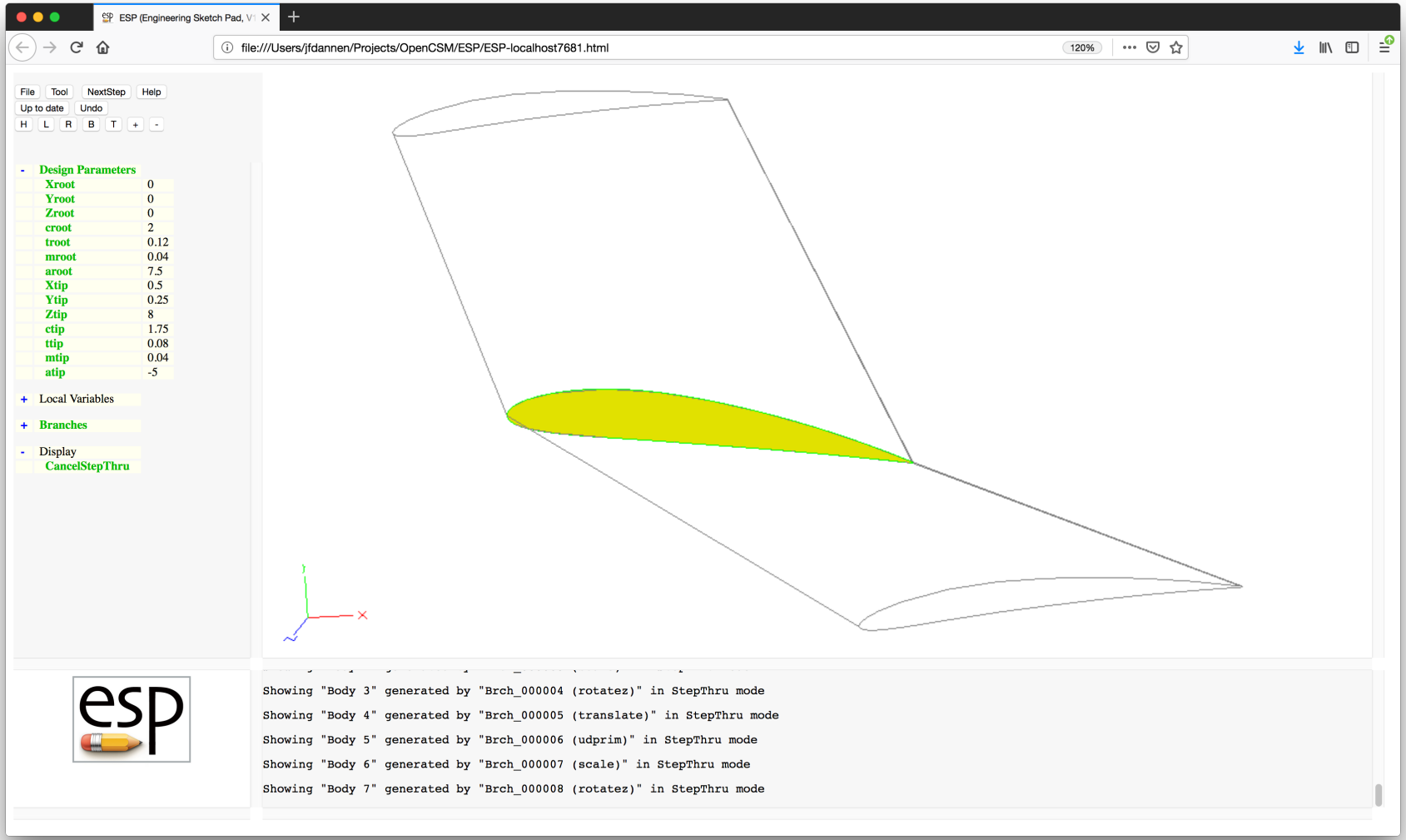
Simple Wing — Step 5



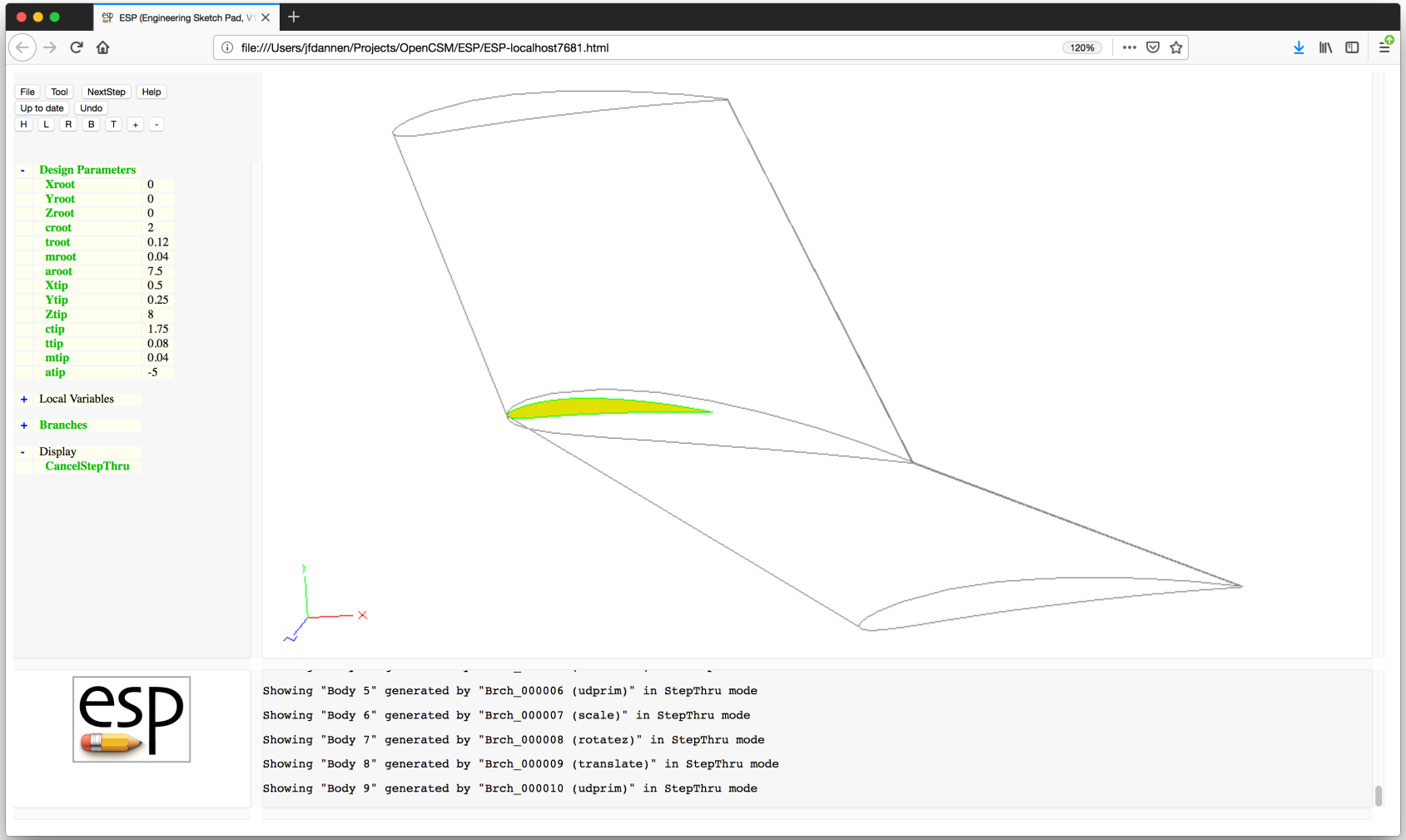
Simple Wing — Step 6



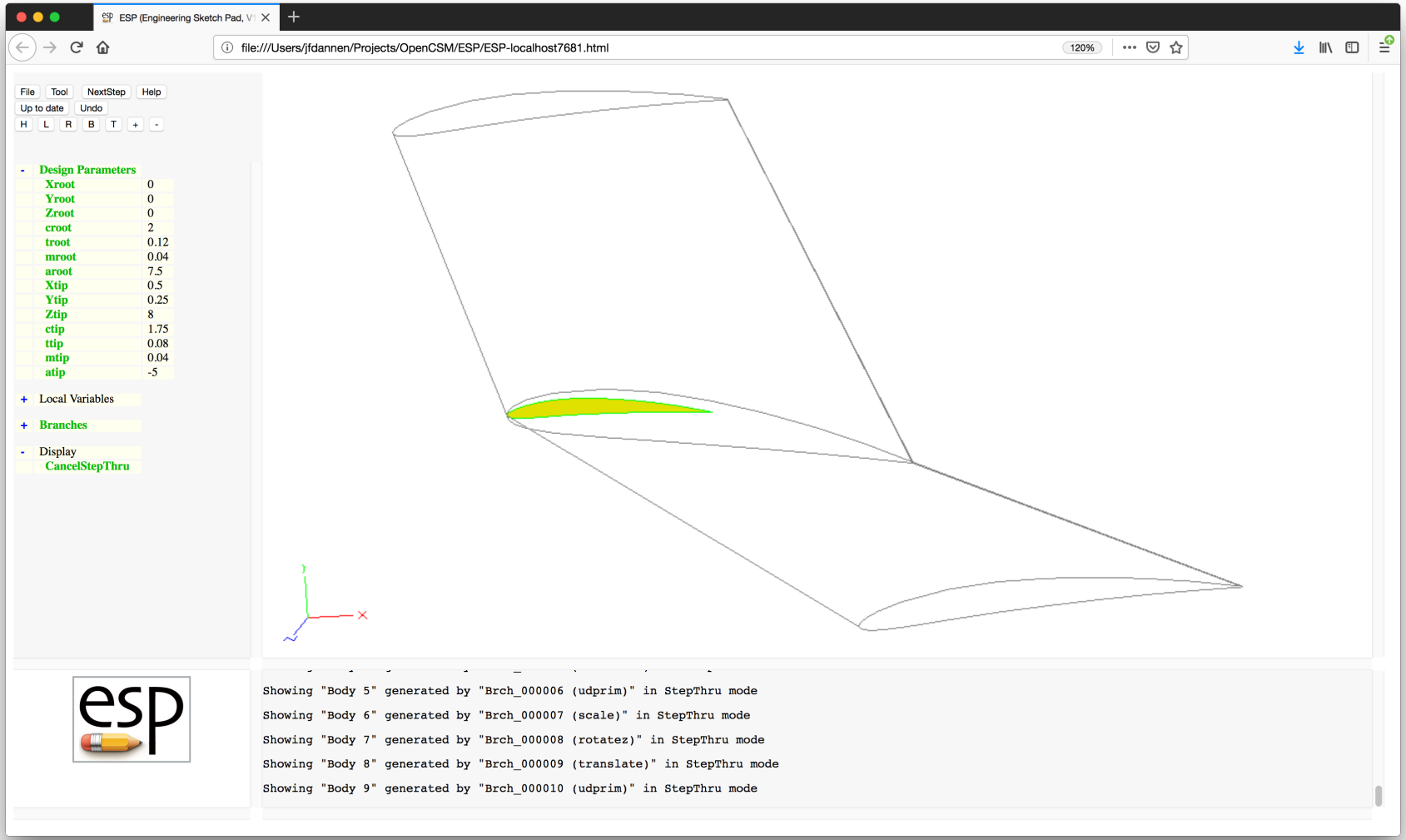
Simple Wing — Step 7



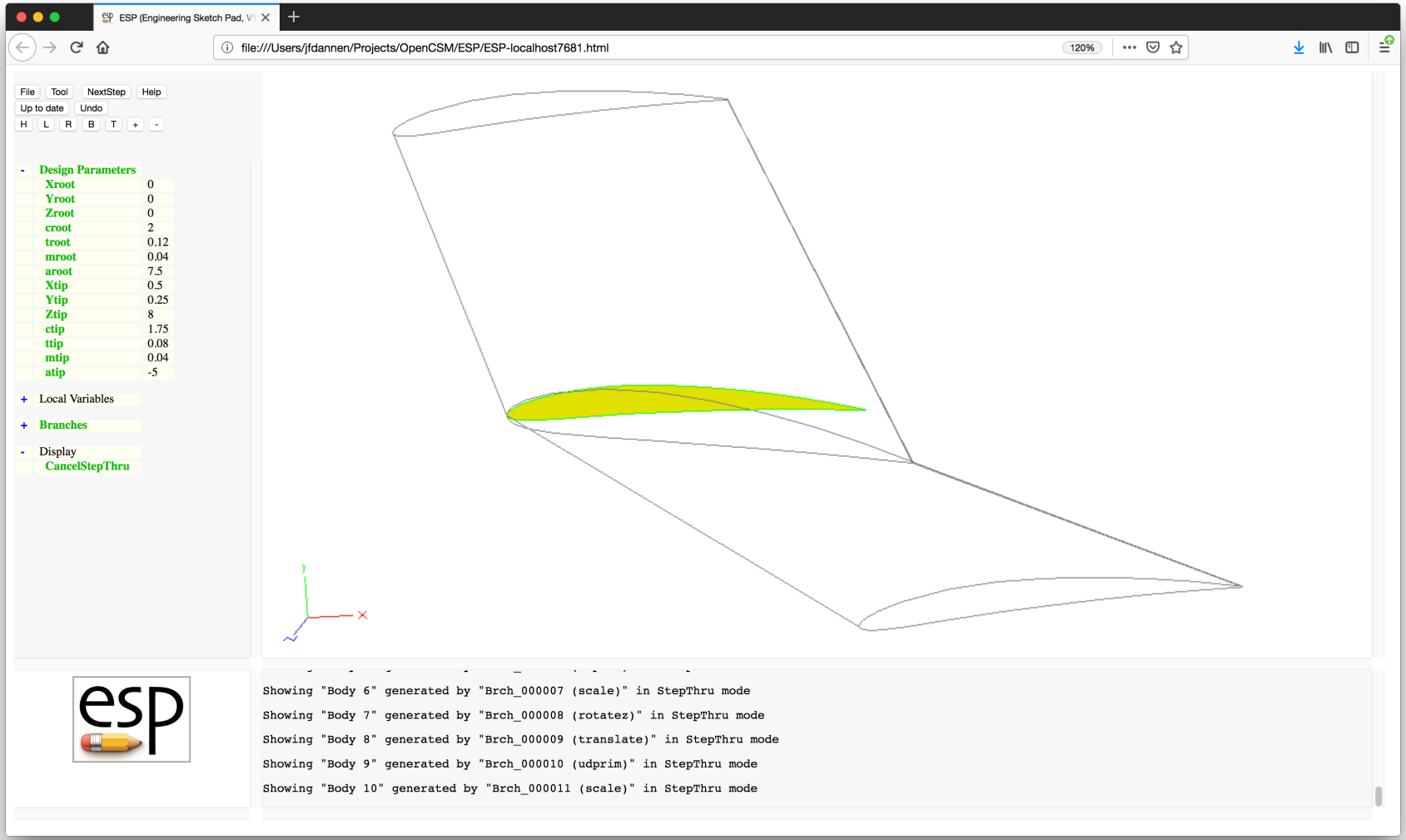
Simple Wing — Step 8



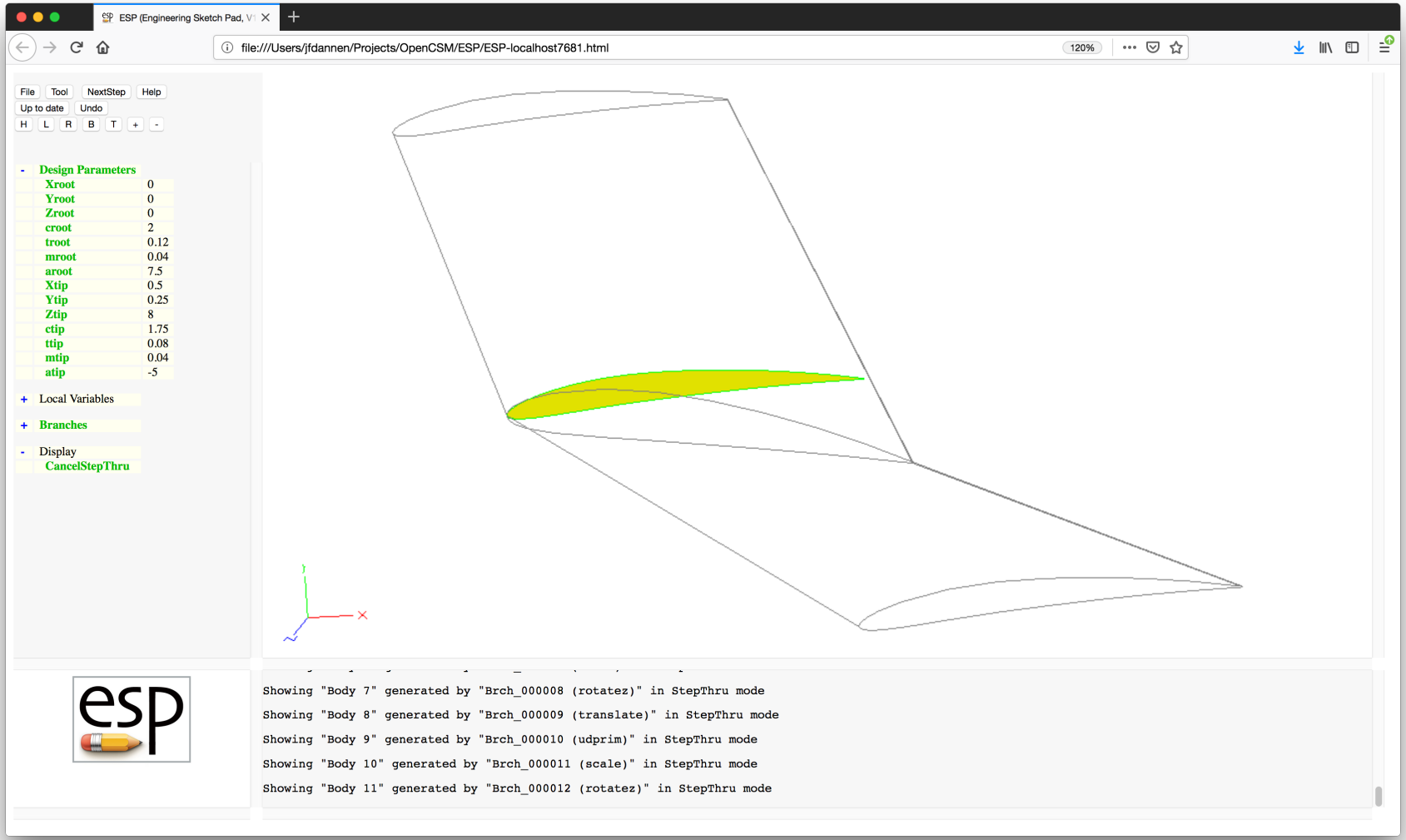
Simple Wing — Step 9



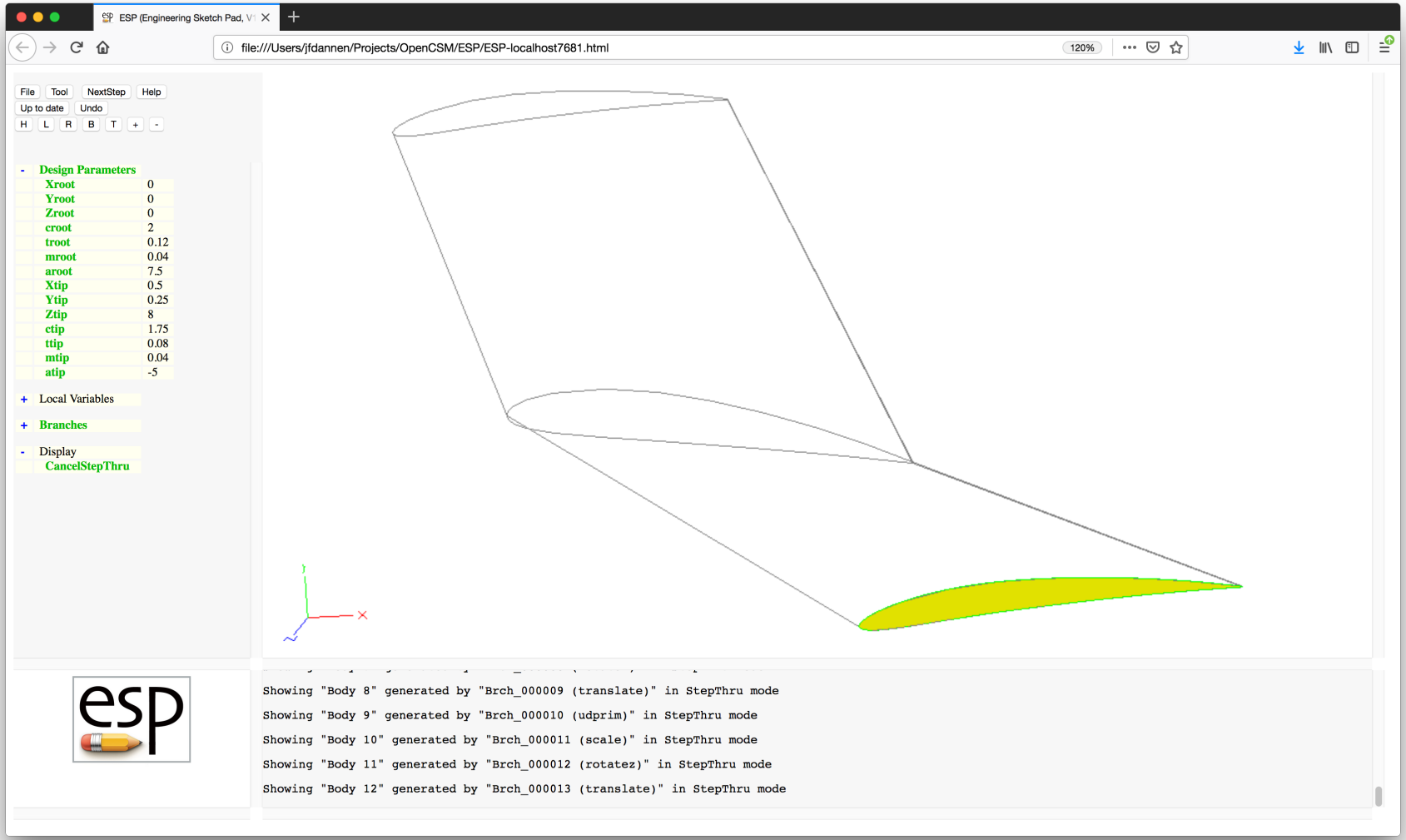
Simple Wing — Step 10



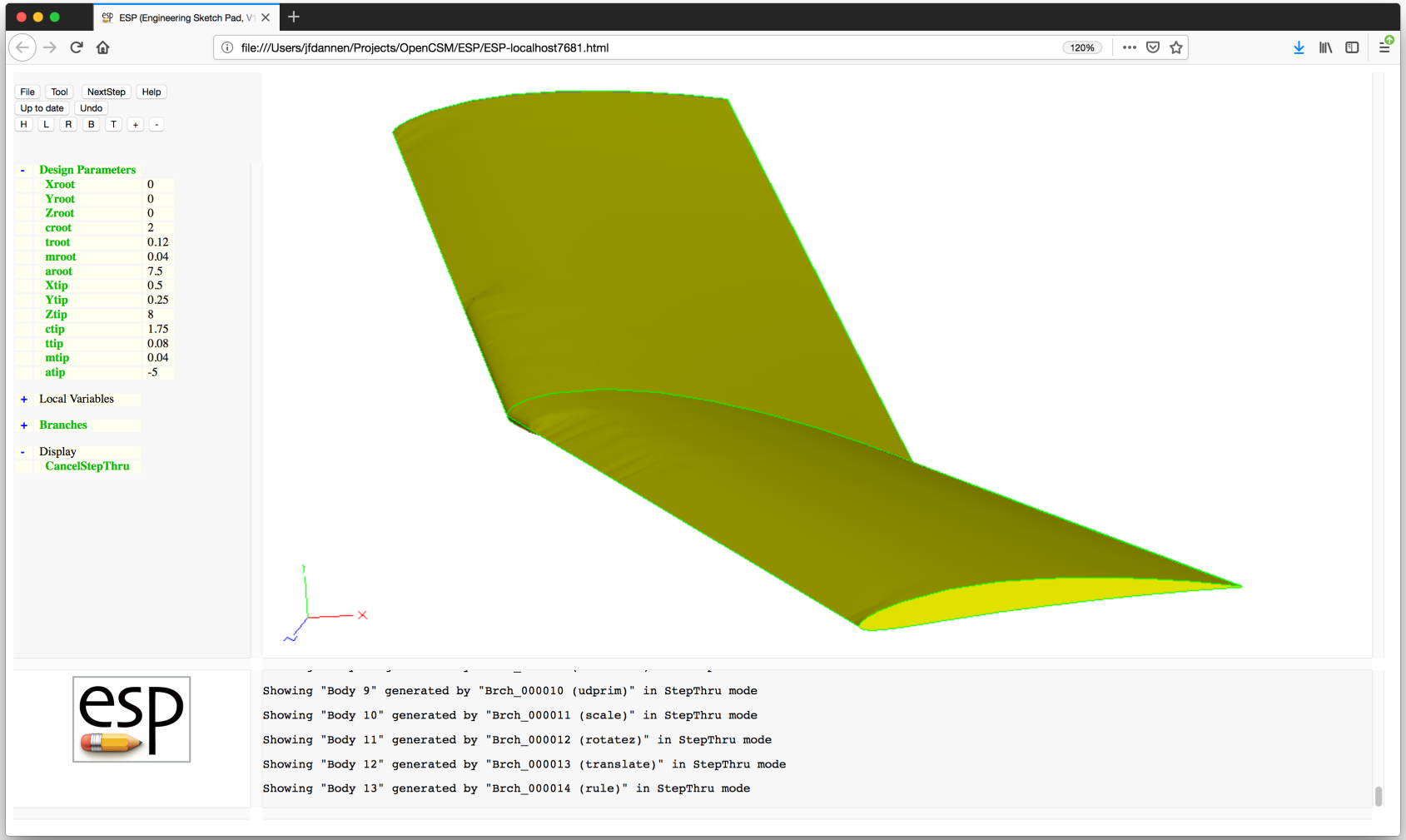
Simple Wing — Step 11



Simple Wing — Step 12



Simple Wing — Step 13



Simple Wing — `.csm` File (1)

```
# wing
# written by John Dannenhoffer

# design parameters
DESPMTR   Xroot     0.00      # X-coordinte of root leading edge
DESPMTR   Yroot     0.00      # Y-coordinte of root leading edge
DESPMTR   Zroot     0.00      # Z-coordinte of root leading edge
DESPMTR   croot     2.00      # chord of root
DESPMTR   troot     0.12      # thickness/chord of root
DESPMTR   mroot     0.04      # camber/chord of root
DESPMTR   aroot     7.50      # angle of attack of root (deg)
DESPMTR   Xtip      0.50      # X-coordinte of tip leading edge
DESPMTR   Ytip      0.25      # Y-coordinte of tip leading edge
DESPMTR   Ztip      8.00      # Z-coordinte of tip leading edge
DESPMTR   ctip      1.75      # chord of tip
DESPMTR   ttip      0.08      # thickness/chord of tip
DESPMTR   mtip      0.04      # camber/chord of tip
DESPMTR   atip     -5.00      # angle of attack of tip (deg)
```

Simple Wing — `.csm` File (2)

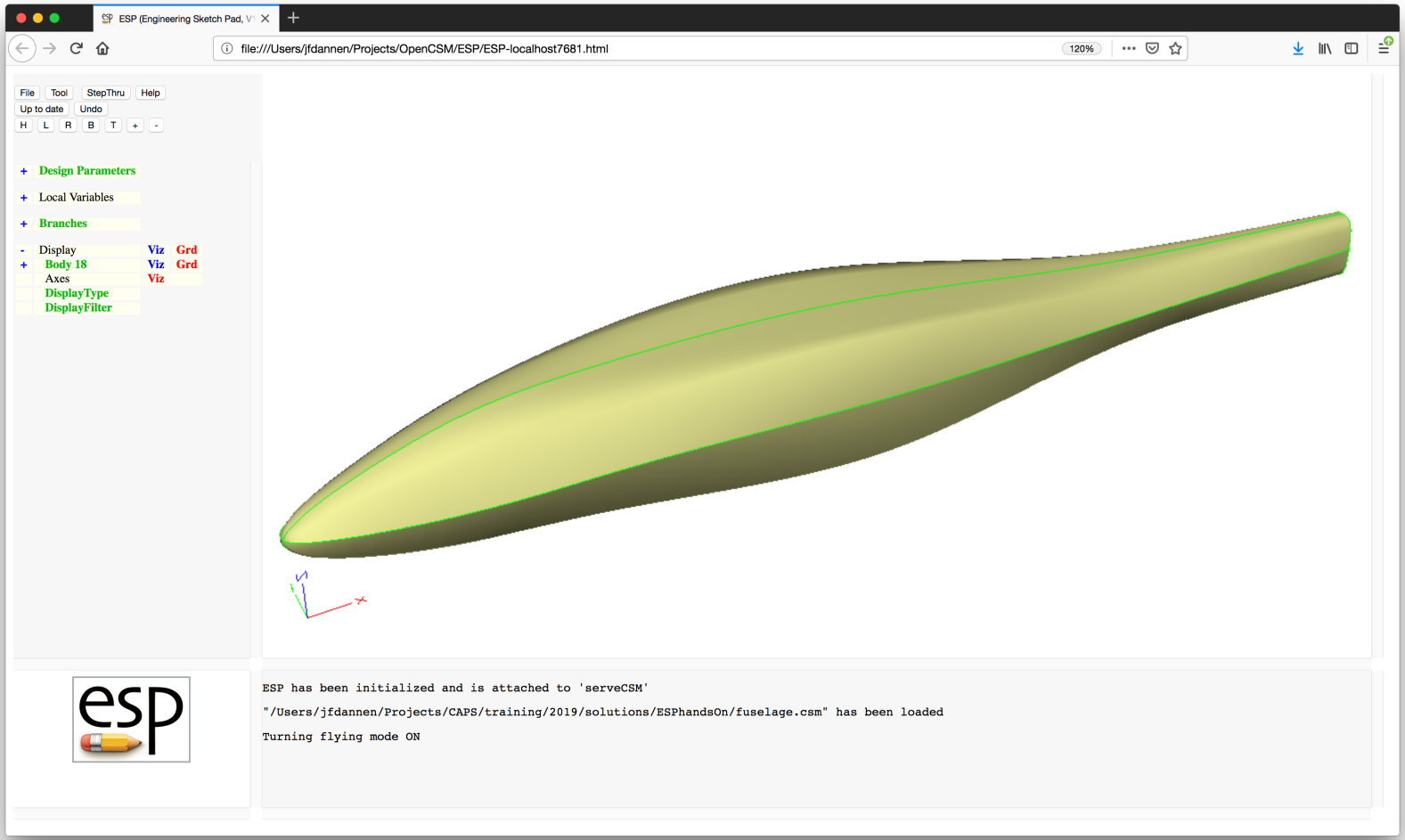
```
MARK

   # rite wing tip
   UDPRIM    naca   thickness  ttip   camber  mtip
   SCALE     ctip
   ROTATEZ  -atip   0      0
   TRANSLATE Xtip   Ytip  -Ztip

   # wing root
   UDPRIM    naca   thickness  troot  camber  mroot
   SCALE     croot
   ROTATEZ  -aroot  0      0
   TRANSLATE Xroot  Yroot  Zroot

   # left wing tip
   UDPRIM    naca   thickness  ttip   camber  mtip
   SCALE     ctip
   ROTATEZ  -atip   0      0
   TRANSLATE Xtip   Ytip   Ztip

# ruled surface
RULE

END
```

Simple Wing — Alternative DESPMTRs

```
DESPMTR   area      30.00     # wing area
DESPMTR   aspect    8.533     # aspect ratio
DESPMTR   taper     0.875     # taper  ratio
DESPMTR   sweep     3.583     # wing sweep (deg)
DESPMTR   dihedral  1.791     # dihedral (deg)

SET       span      sqrt(area*aspect)
SET       cmean     area/span
SET       croot     2*cmean/(1+taper)
SET       ctip      croot*taper
SET       Xtip      span/2*sind(sweep)
SET       Ytip      span/2*sind(dihedral)
SET       Ztip      span/2
```

Simple Fuselage (1)

- Fuselage by blending a series of super-ellipses (SUPELLs), where the dimensions of the X-sections are provided in arrays



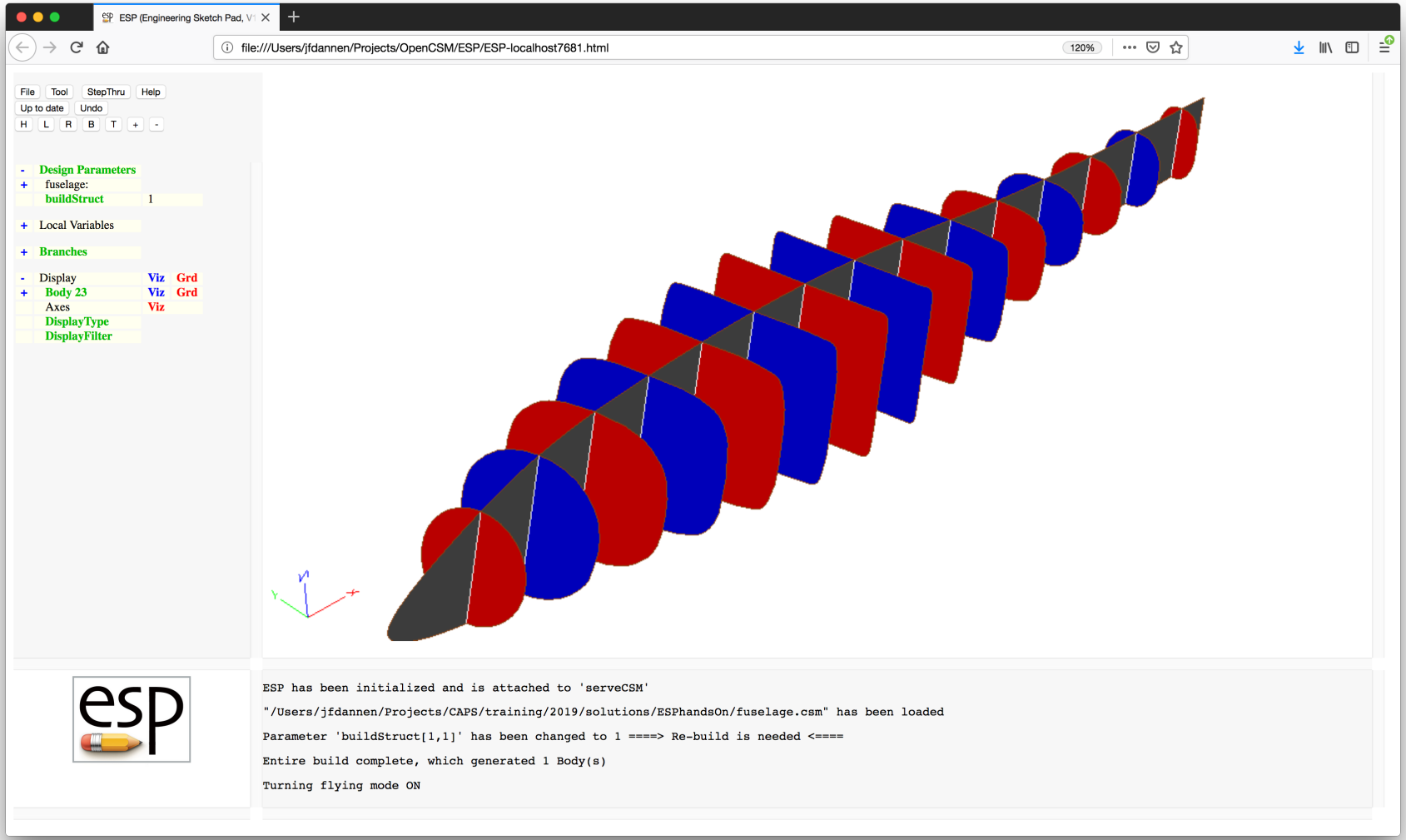
Simple Fuselage (2)

xloc	width	zcent	height	power
0.0	0.0	0.0	0.0	2
1.0	1.0	0.1	1.0	2
4.0	1.6	0.4	2.0	3
8.0	1.6	0.4	2.0	3
12.0	1.0	0.3	1.2	2
16.0	0.8	0.2	0.4	2

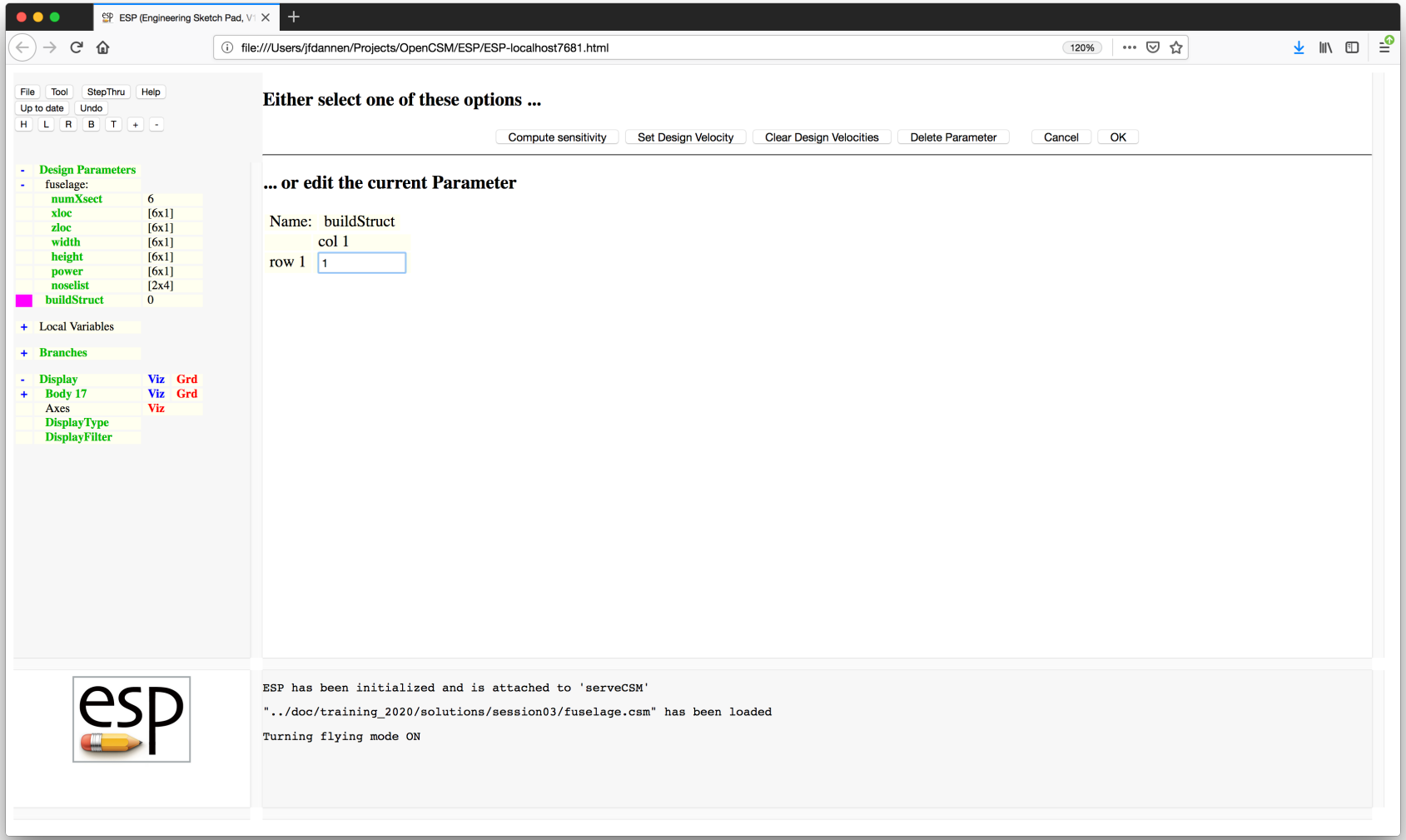
Simple Fuselage (3)

- Can you make the radius at the nose 0.2 in a top view and 0.1 in a side view?
- Can you make the fuselage between the two sections whose power is 3 have a constant cross-section?
- Can you create a SheetBody that has a plane of symmetry and cross-sections at every y, starting at $y = 1/2$ and spaced with $\Delta y = 1$?
- Can you color the odd-numbered bulkheads red and even-numbered bulkheads blue?
- Can you color the Edges at the intersections of the symmetry plane and bulkheads white?

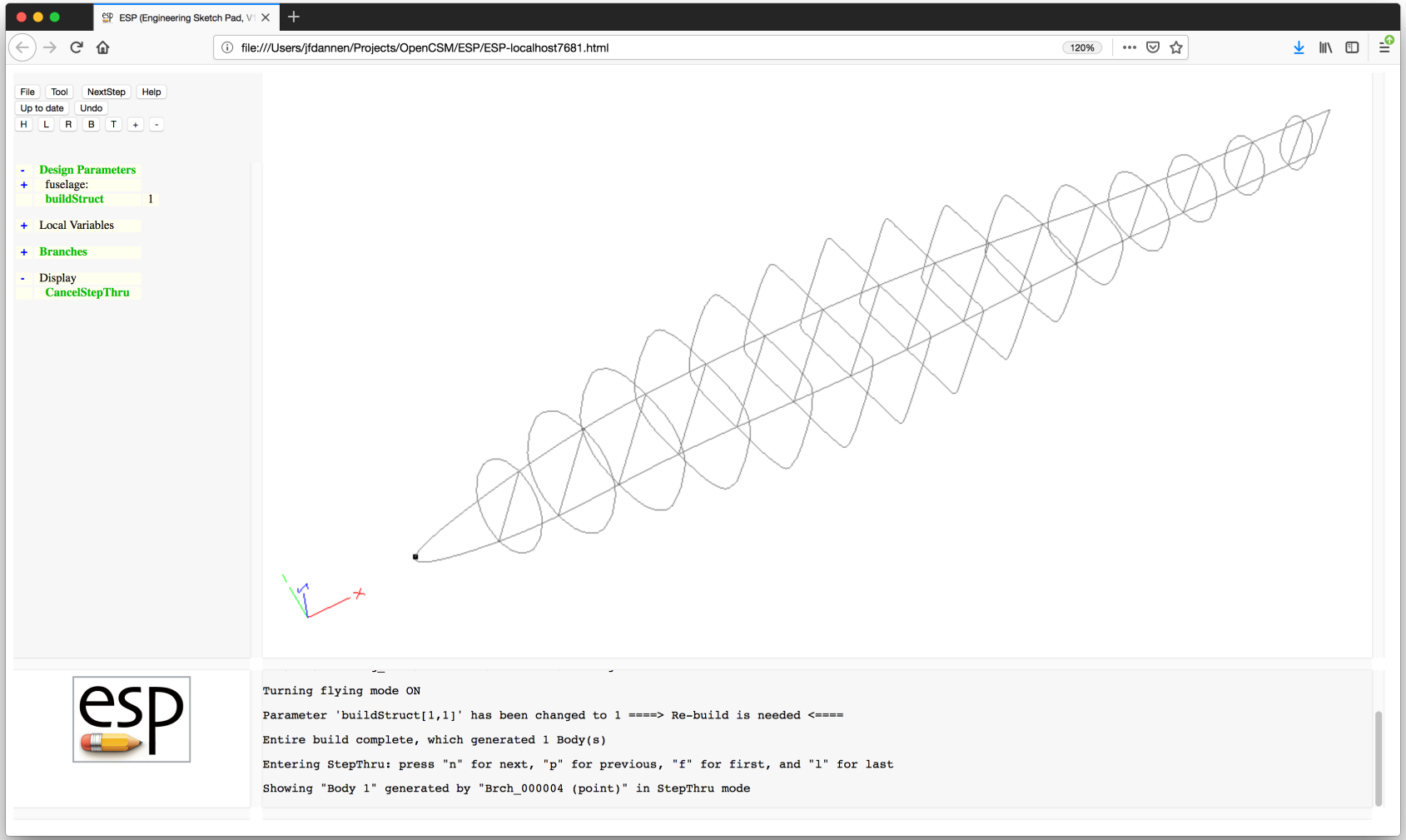
Simple Fuselage (4)



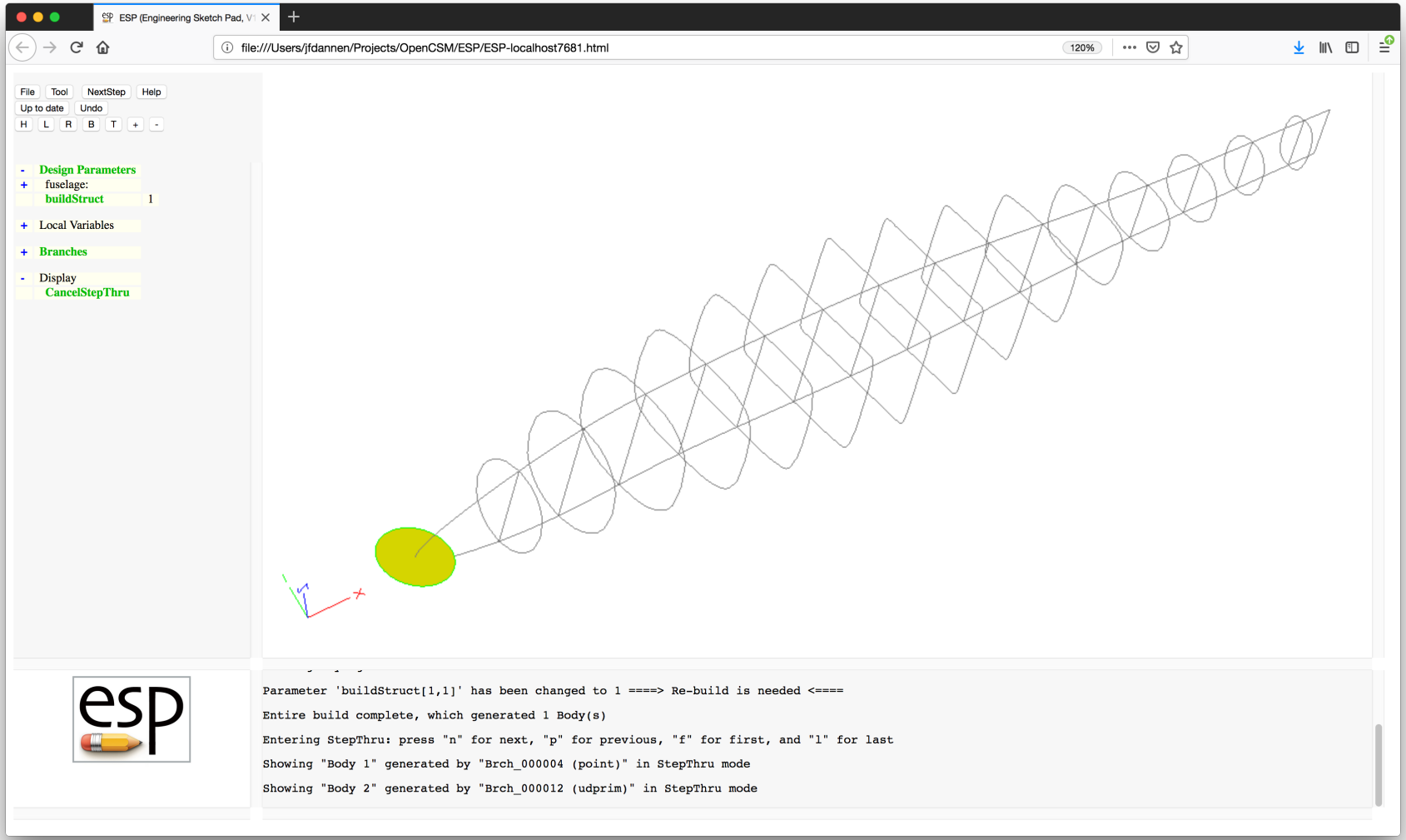
Simple Fuselage — Step 1



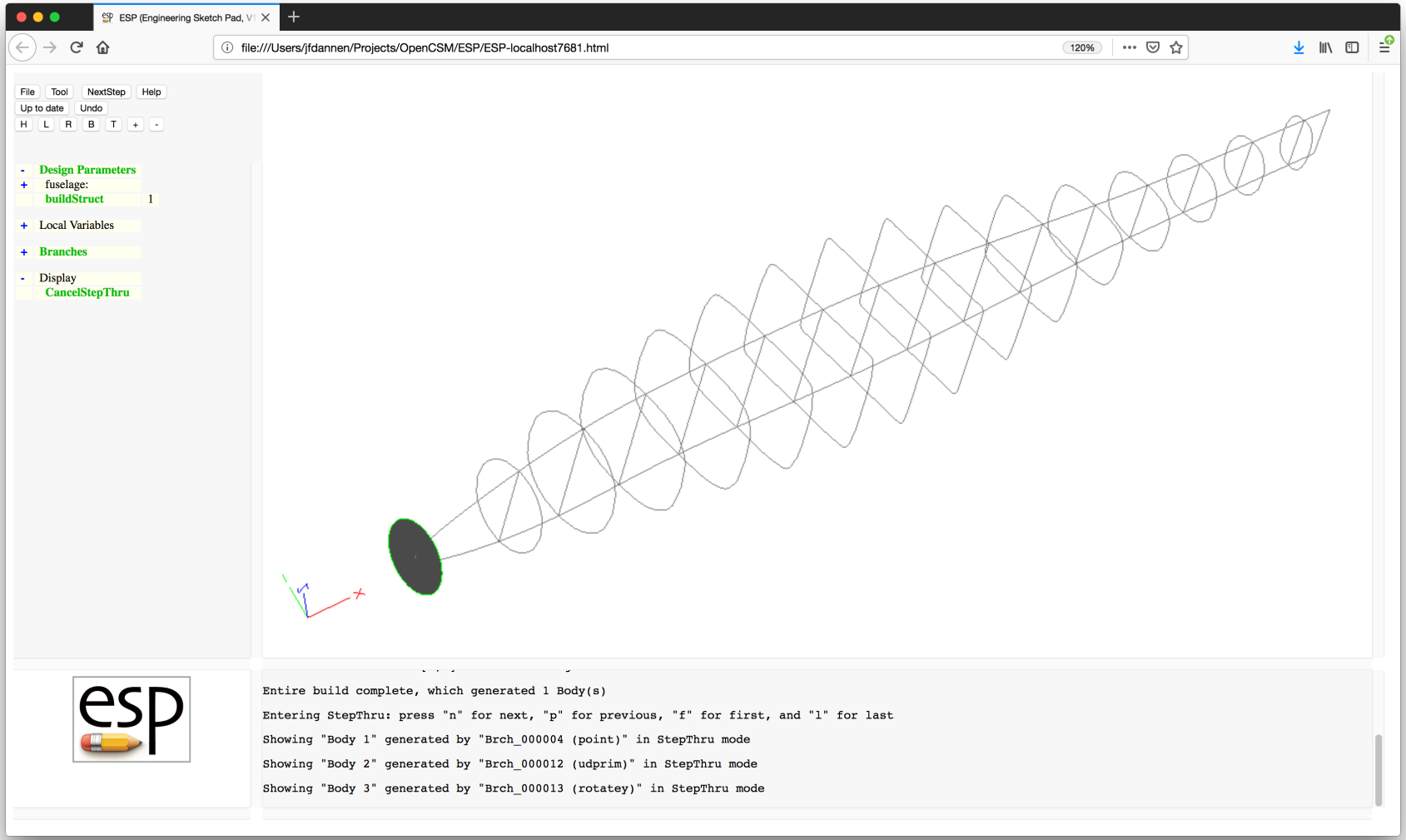
Simple Fuselage — Step 2



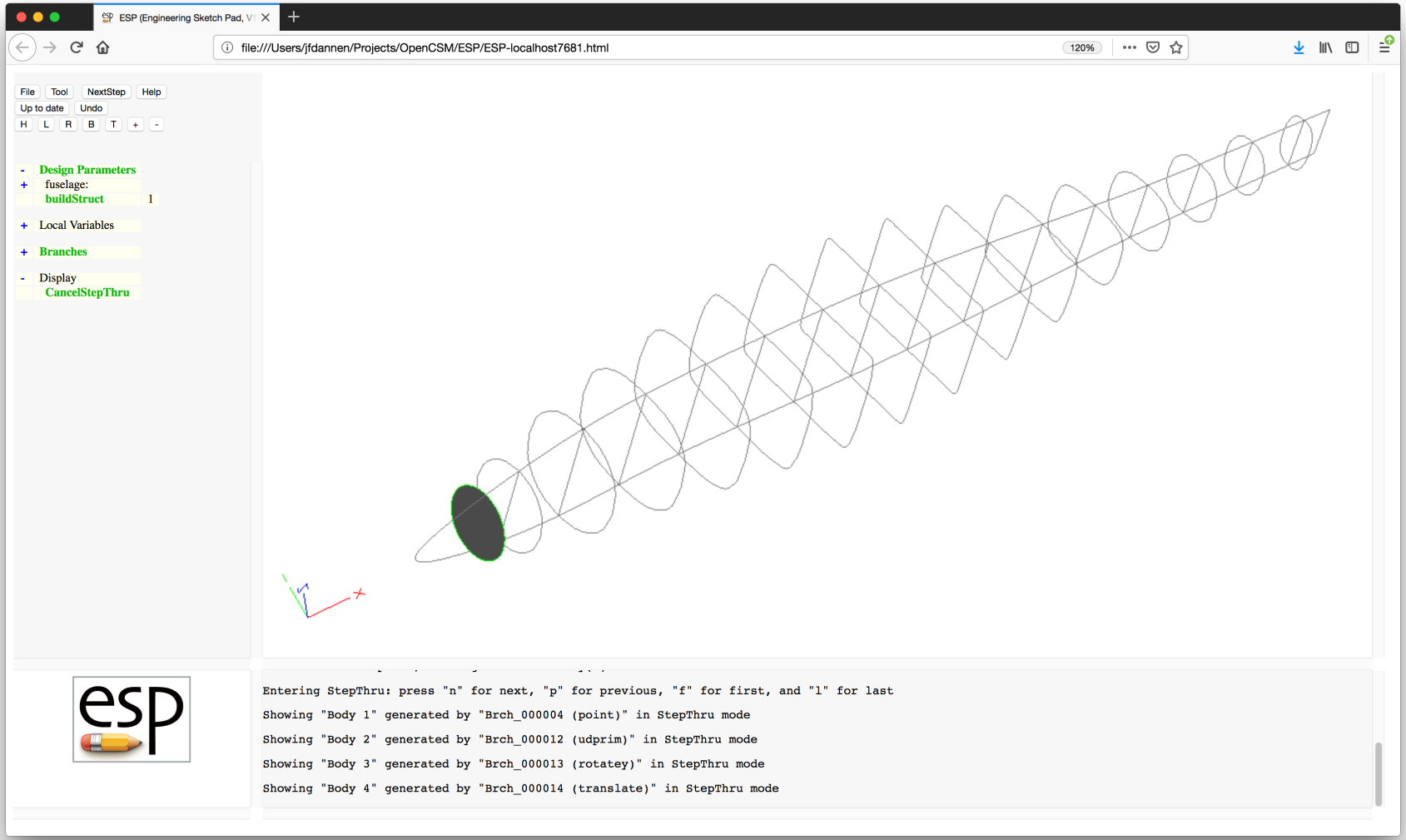
Simple Fuselage — Step 3



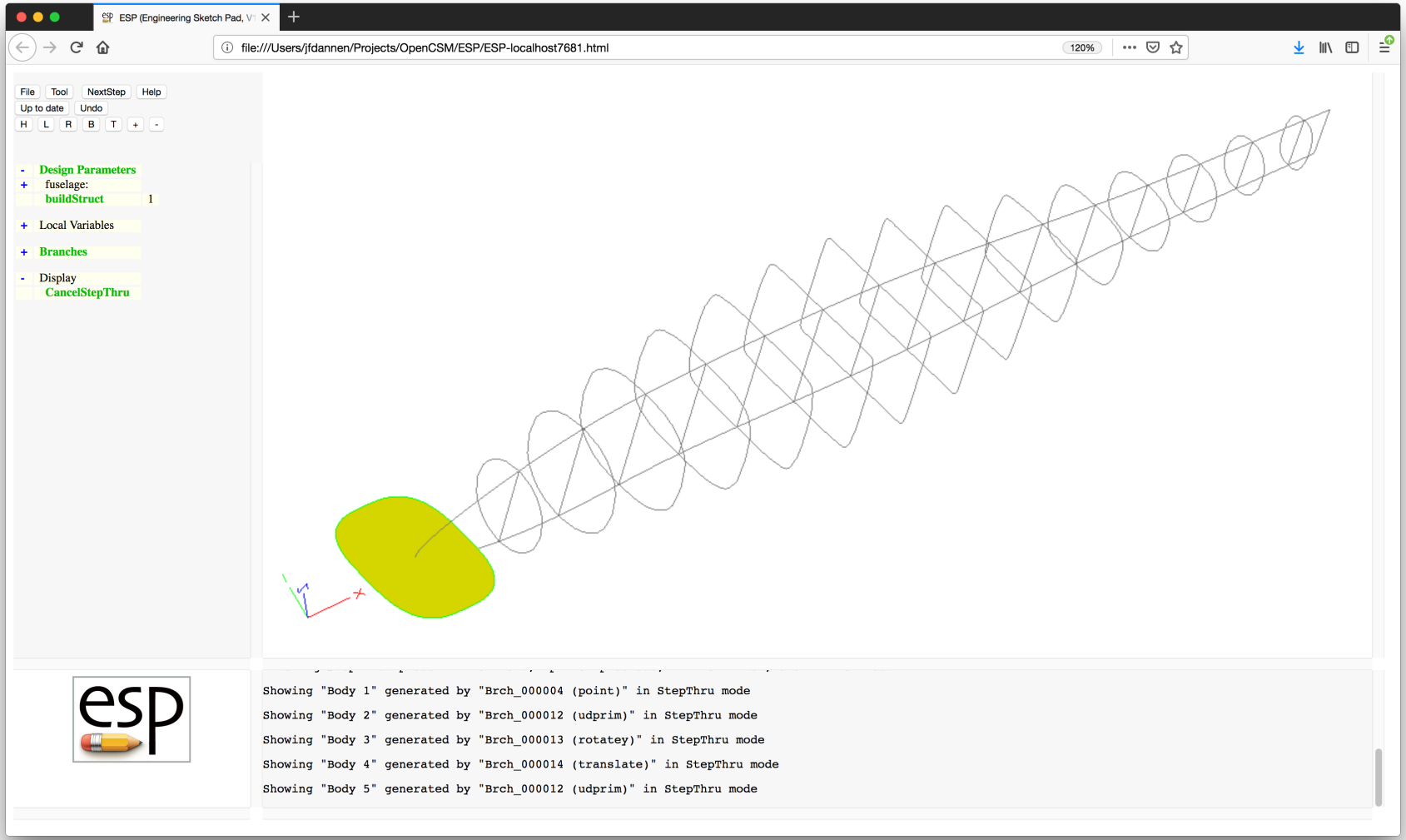
Simple Fuselage — Step 4



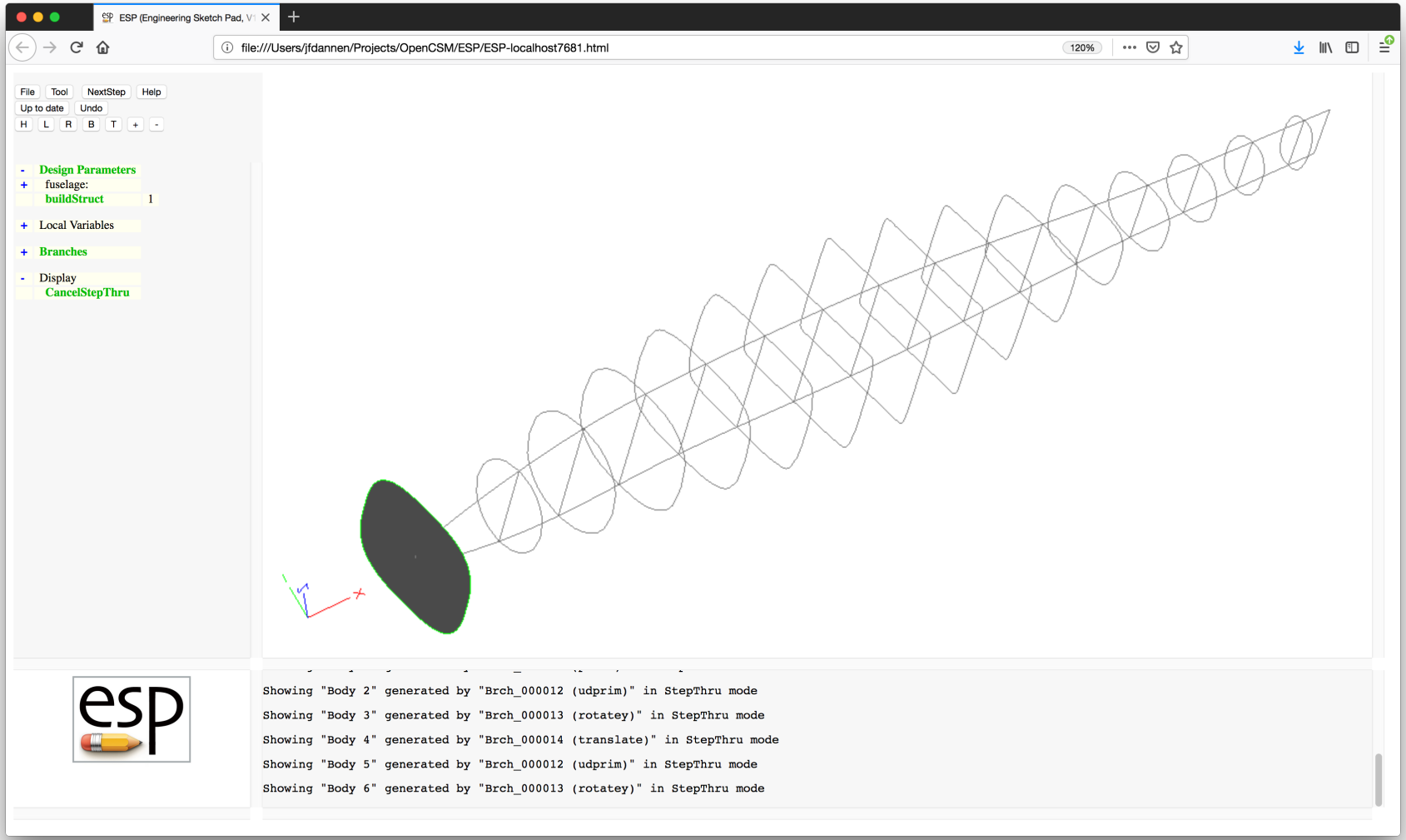
Simple Fuselage — Step 5



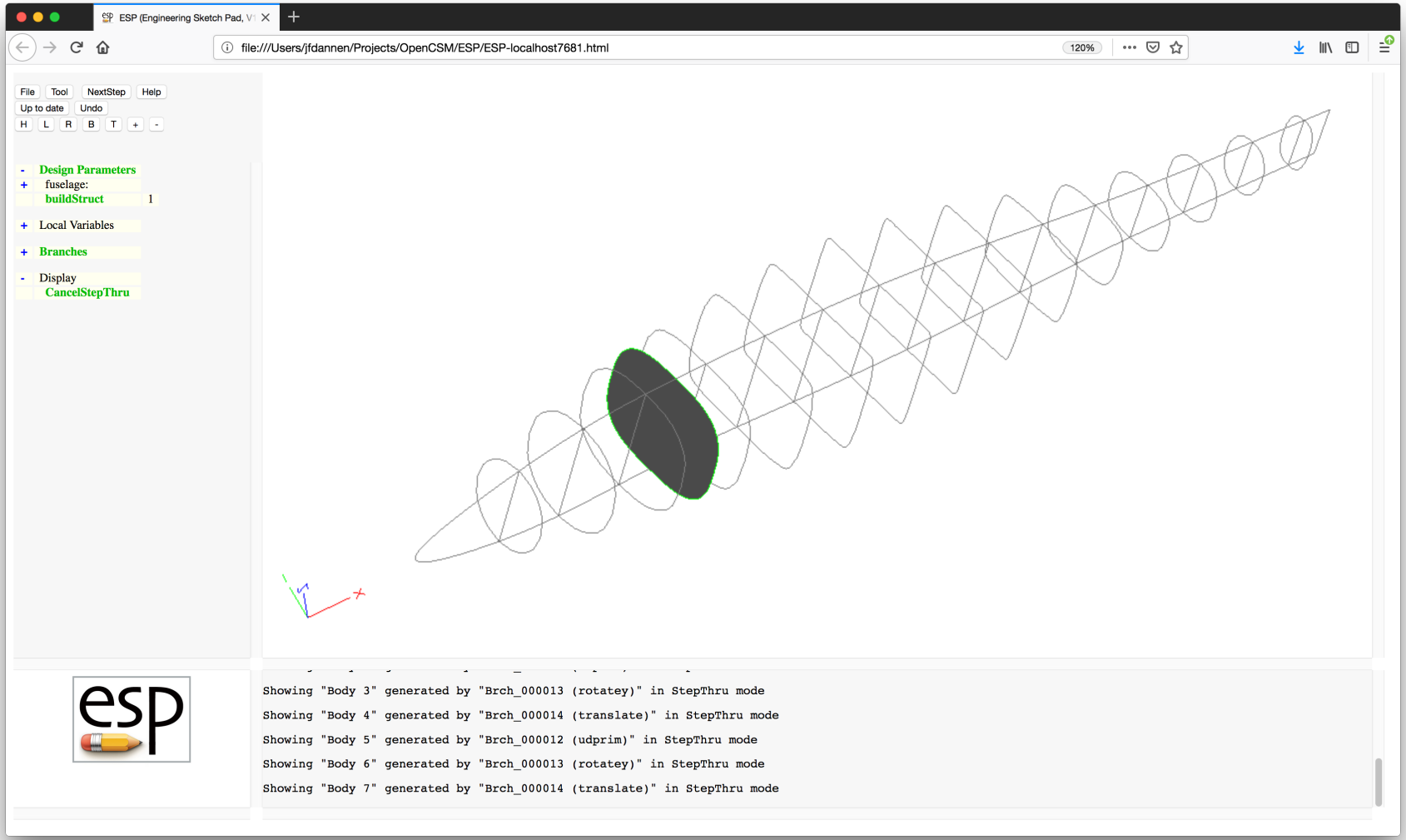
Simple Fuselage — Step 6



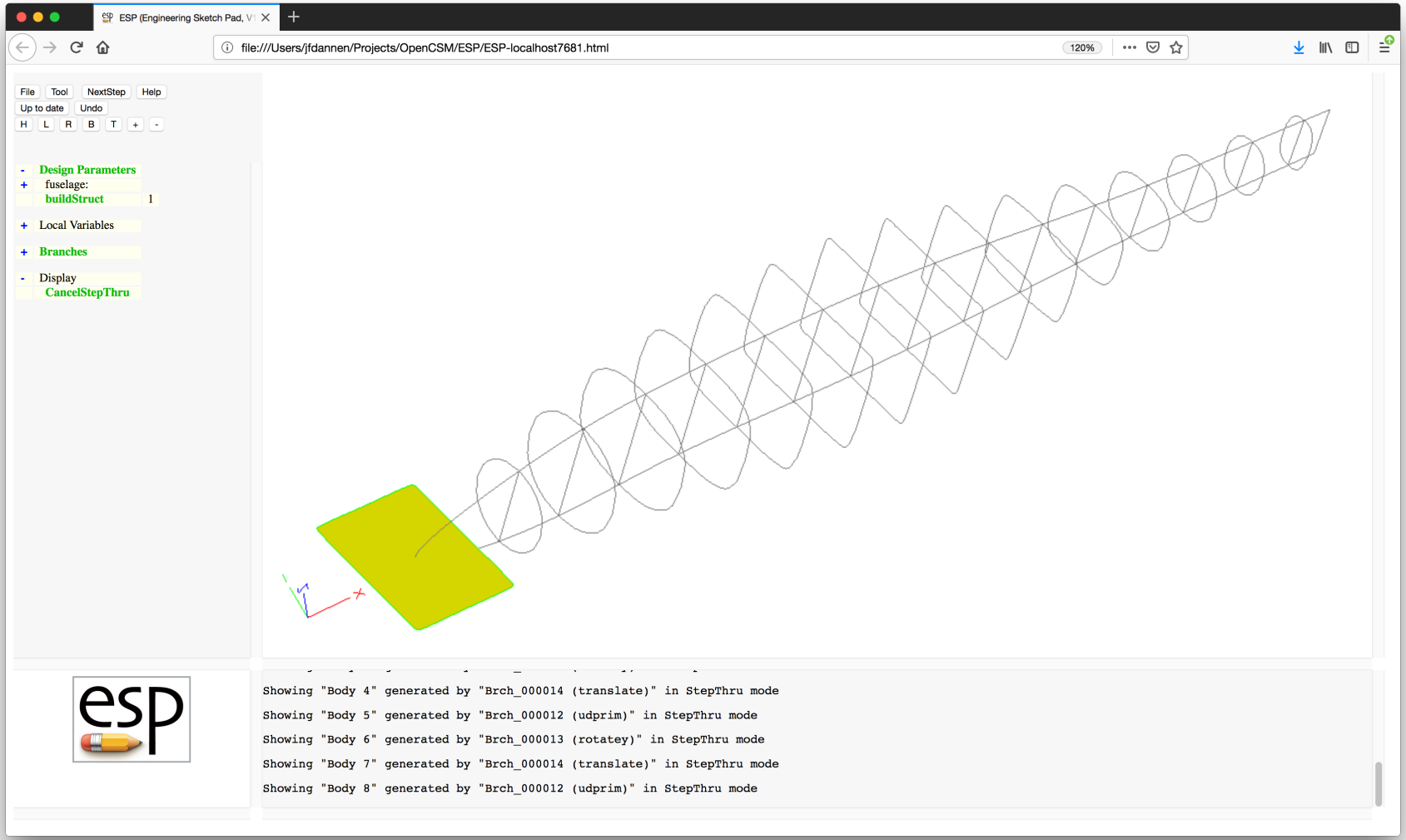
Simple Fuselage — Step 7



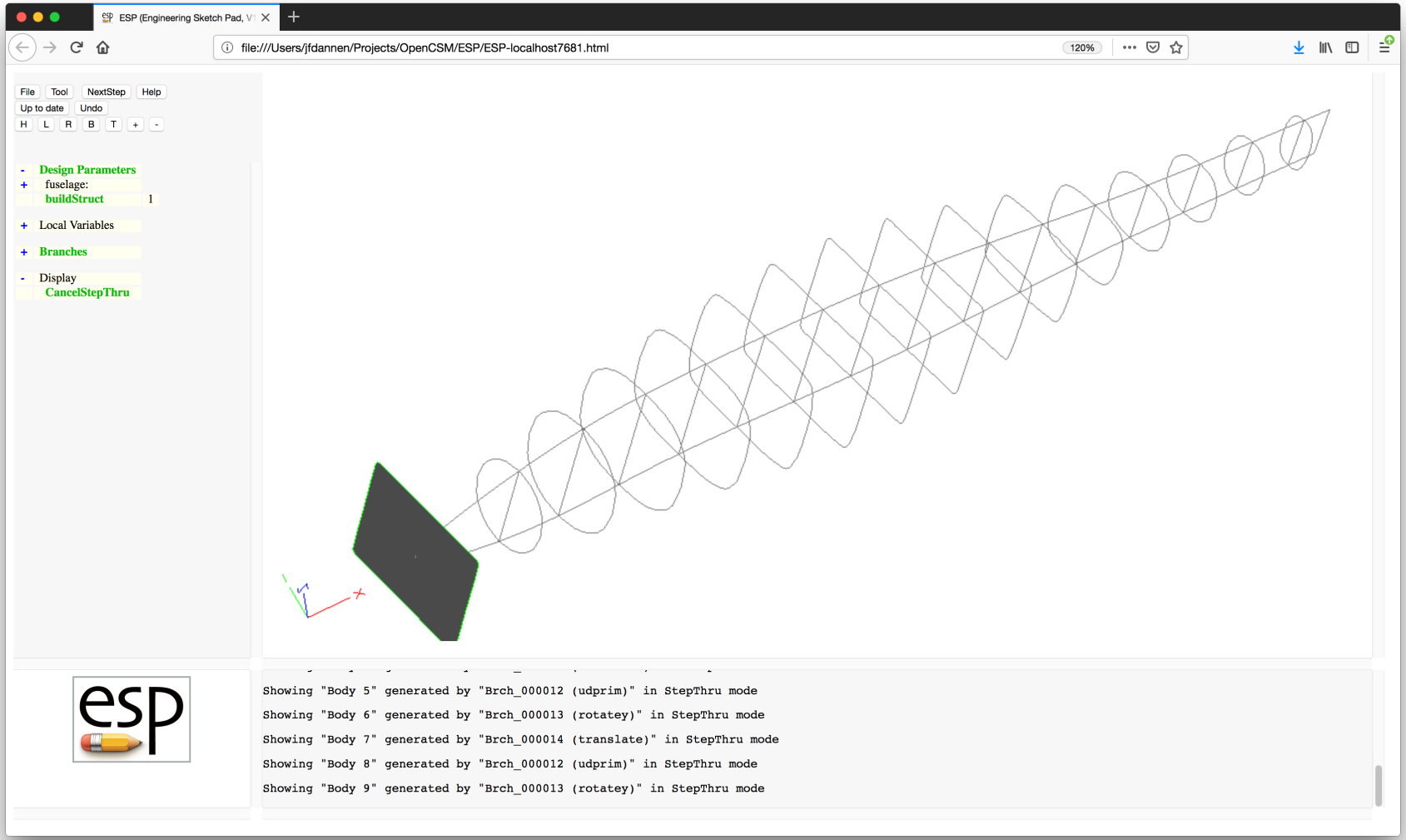
Simple Fuselage — Step 8



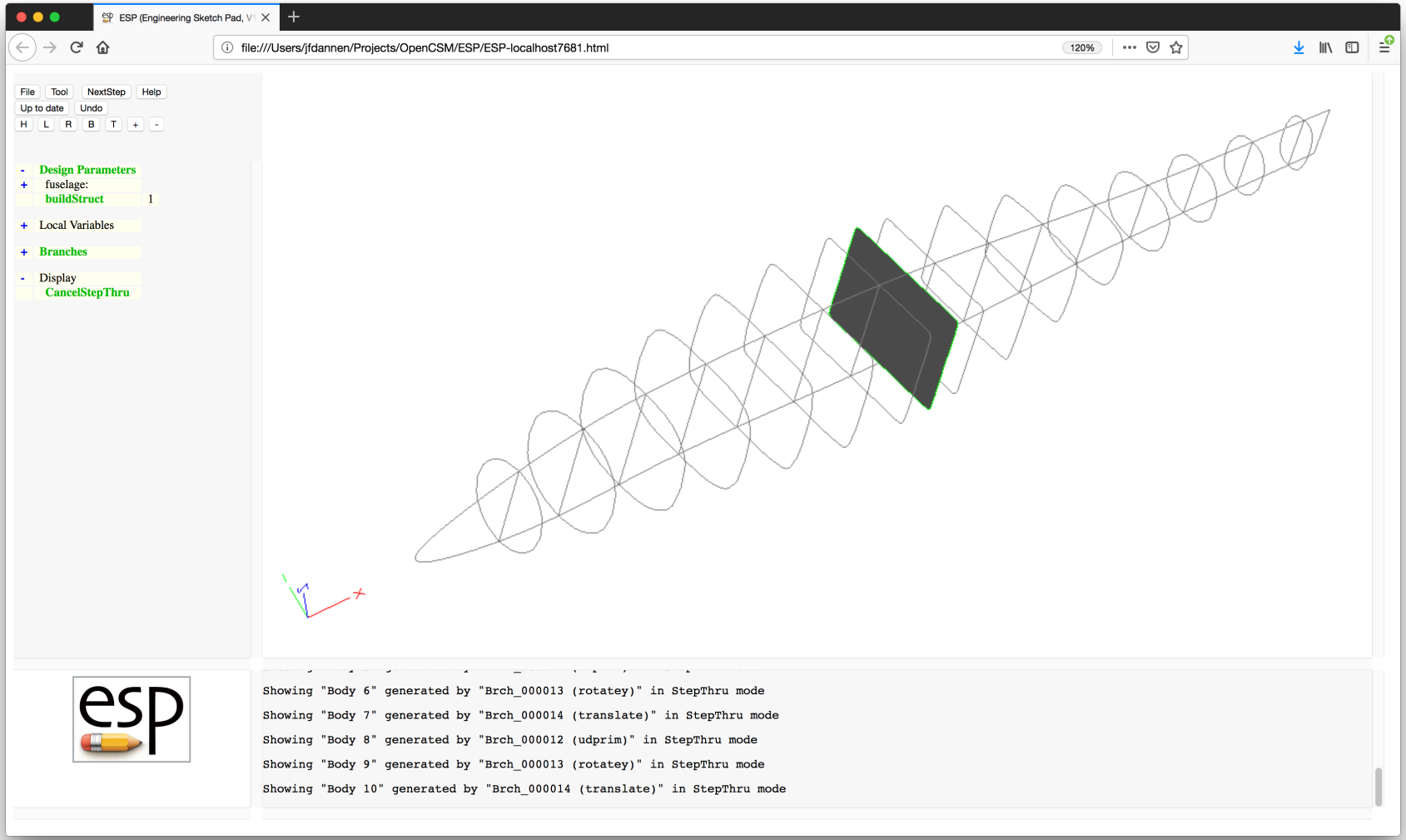
Simple Fuselage — Step 9



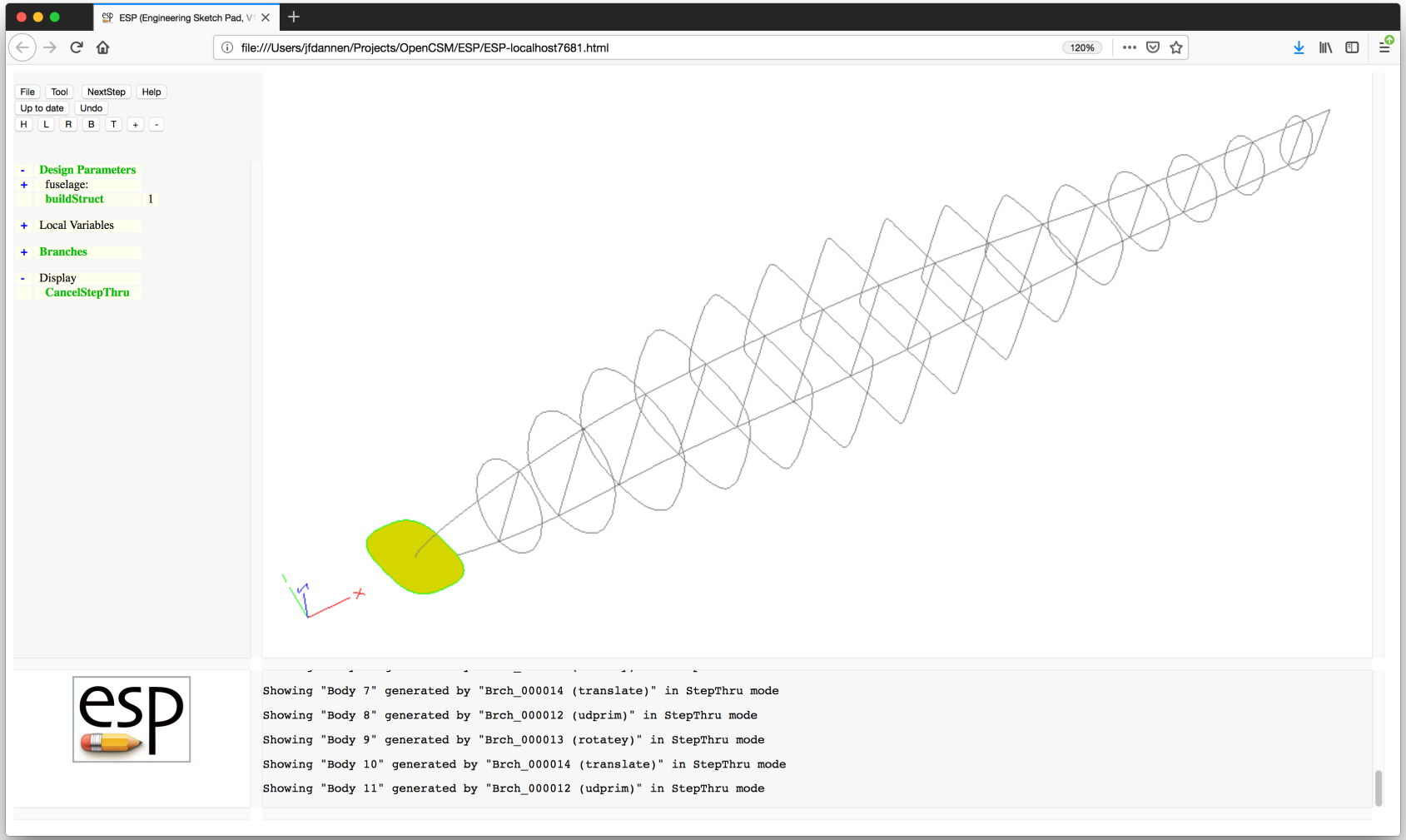
Simple Fuselage — Step 10



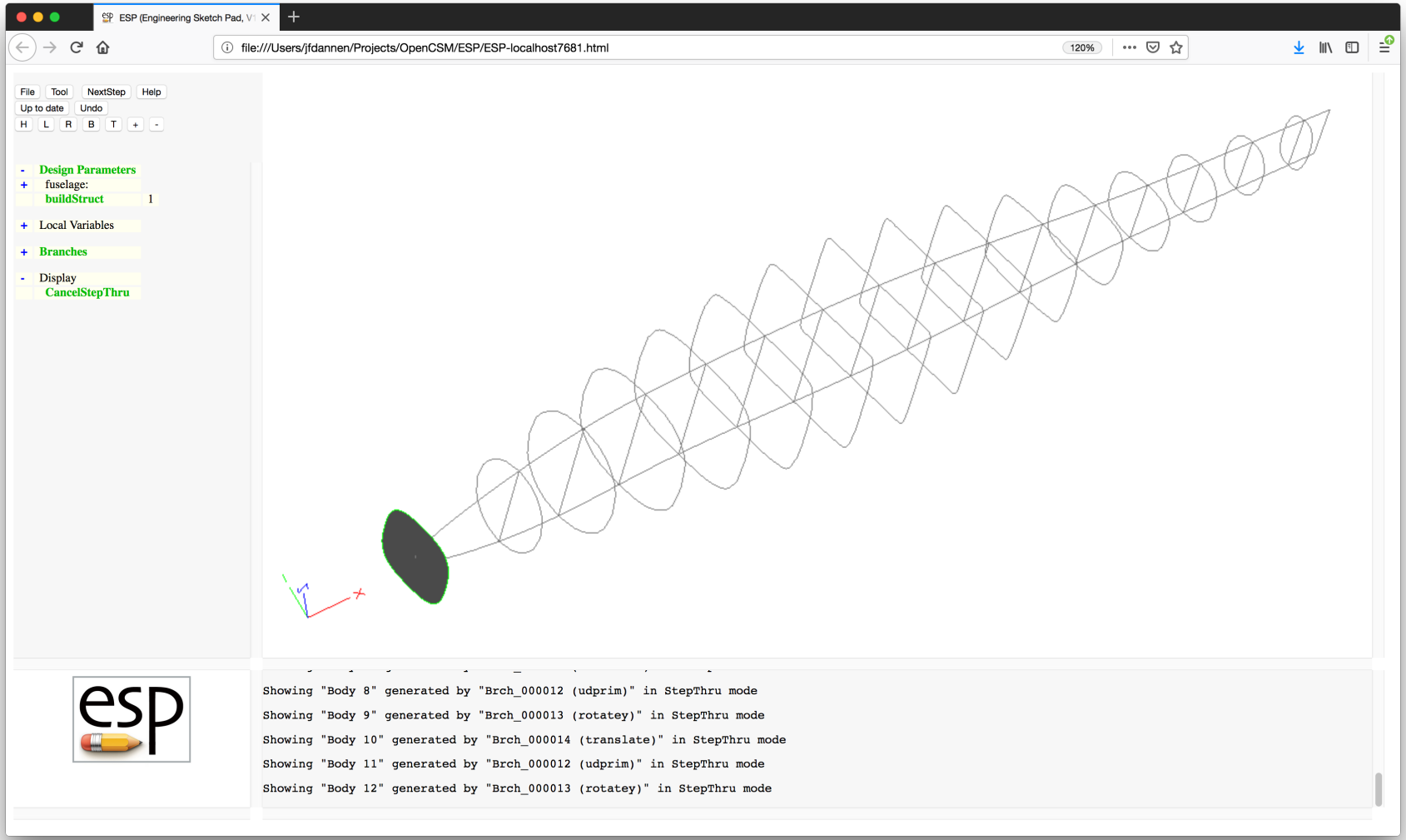
Simple Fuselage — Step 11



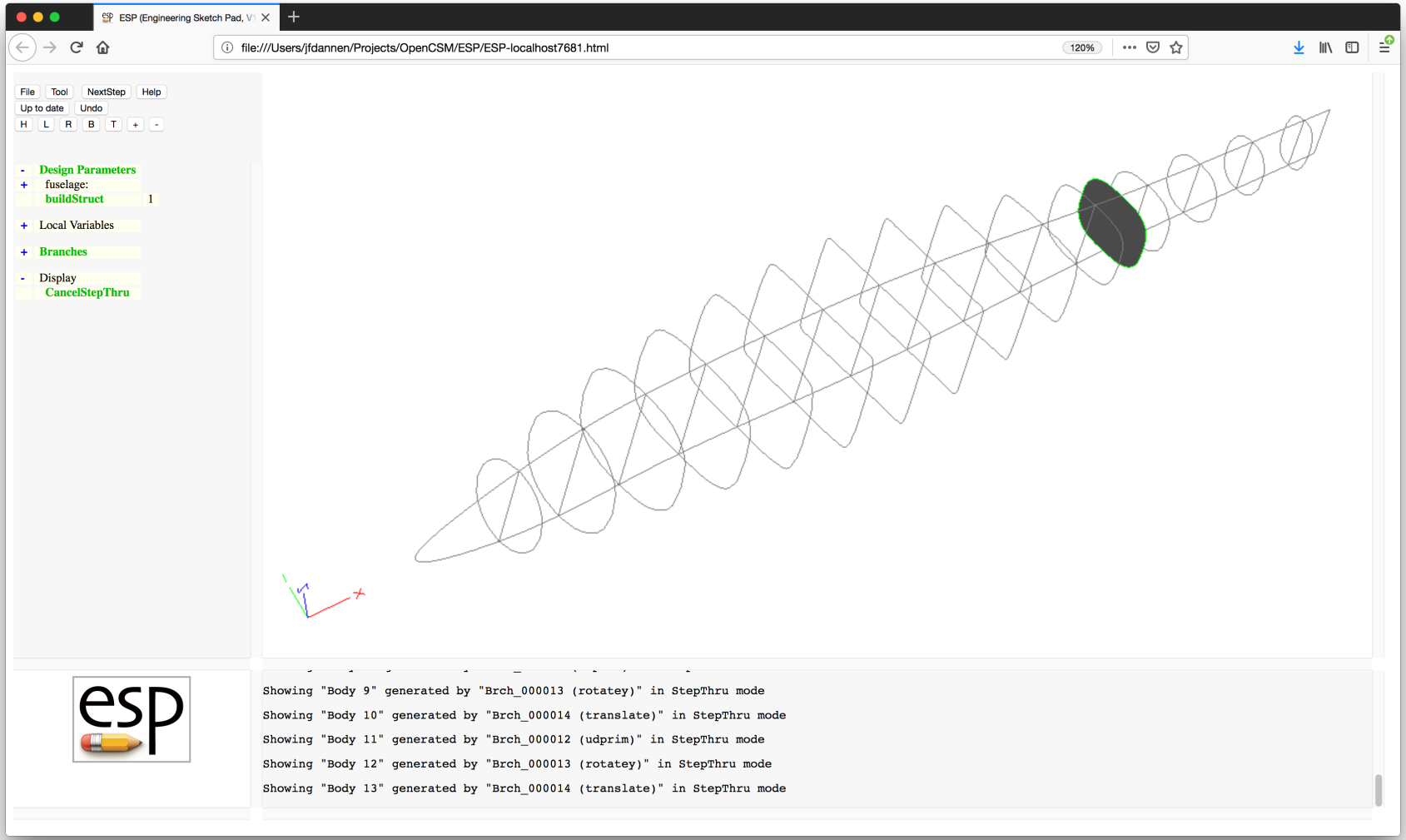
Simple Fuselage — Step 12



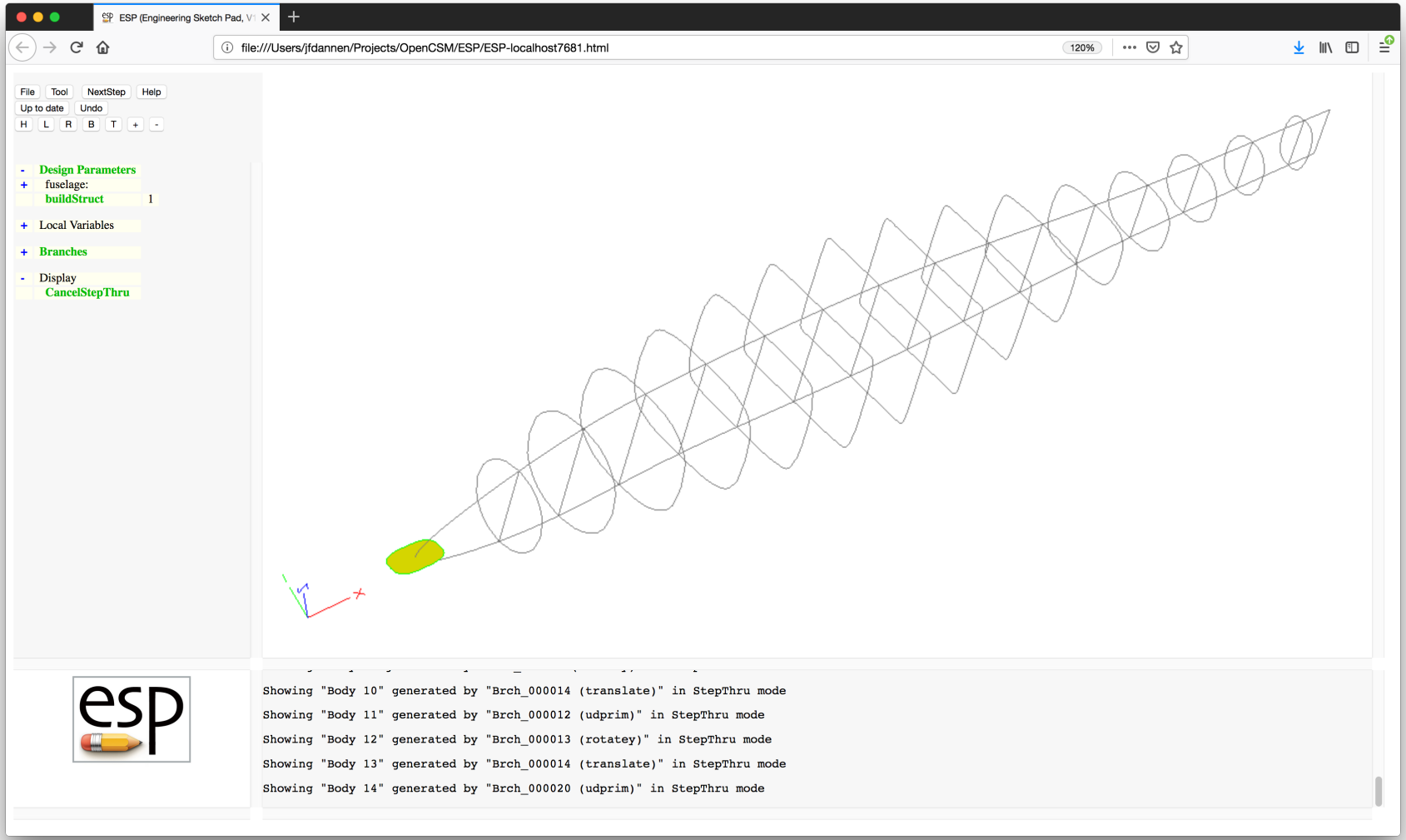
Simple Fuselage — Step 13



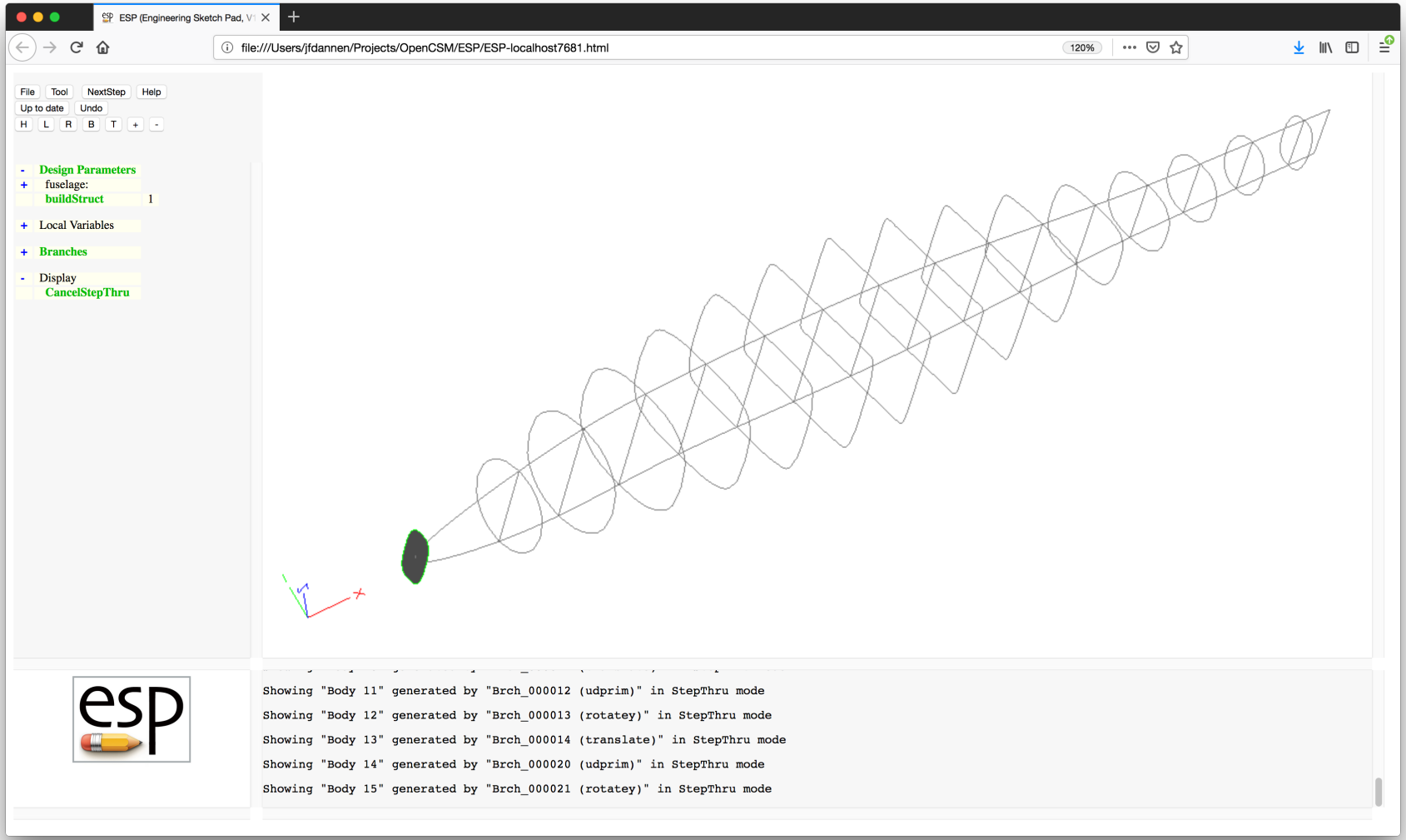
Simple Fuselage — Step 14



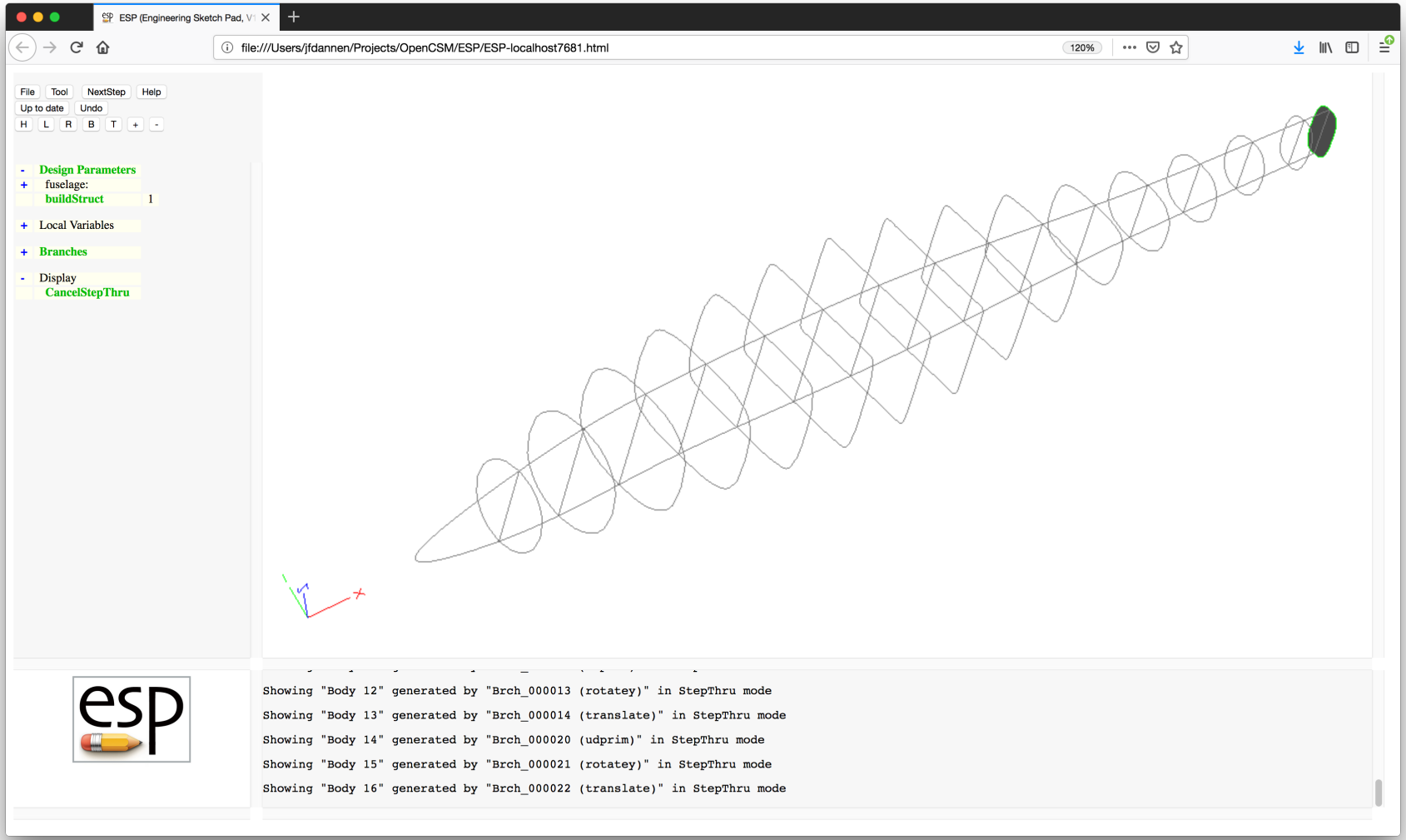
Simple Fuselage — Step 15



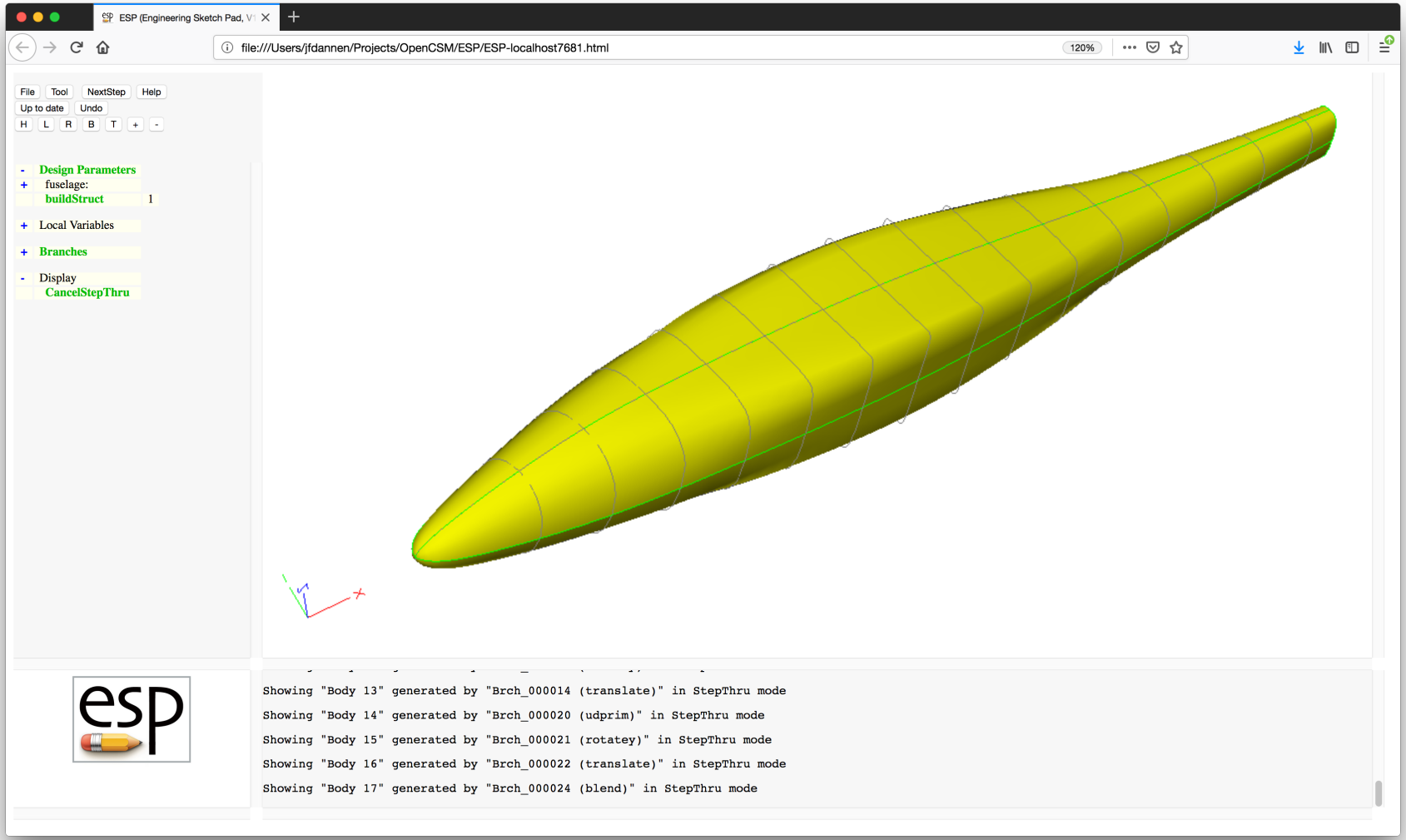
Simple Fuselage — Step 16



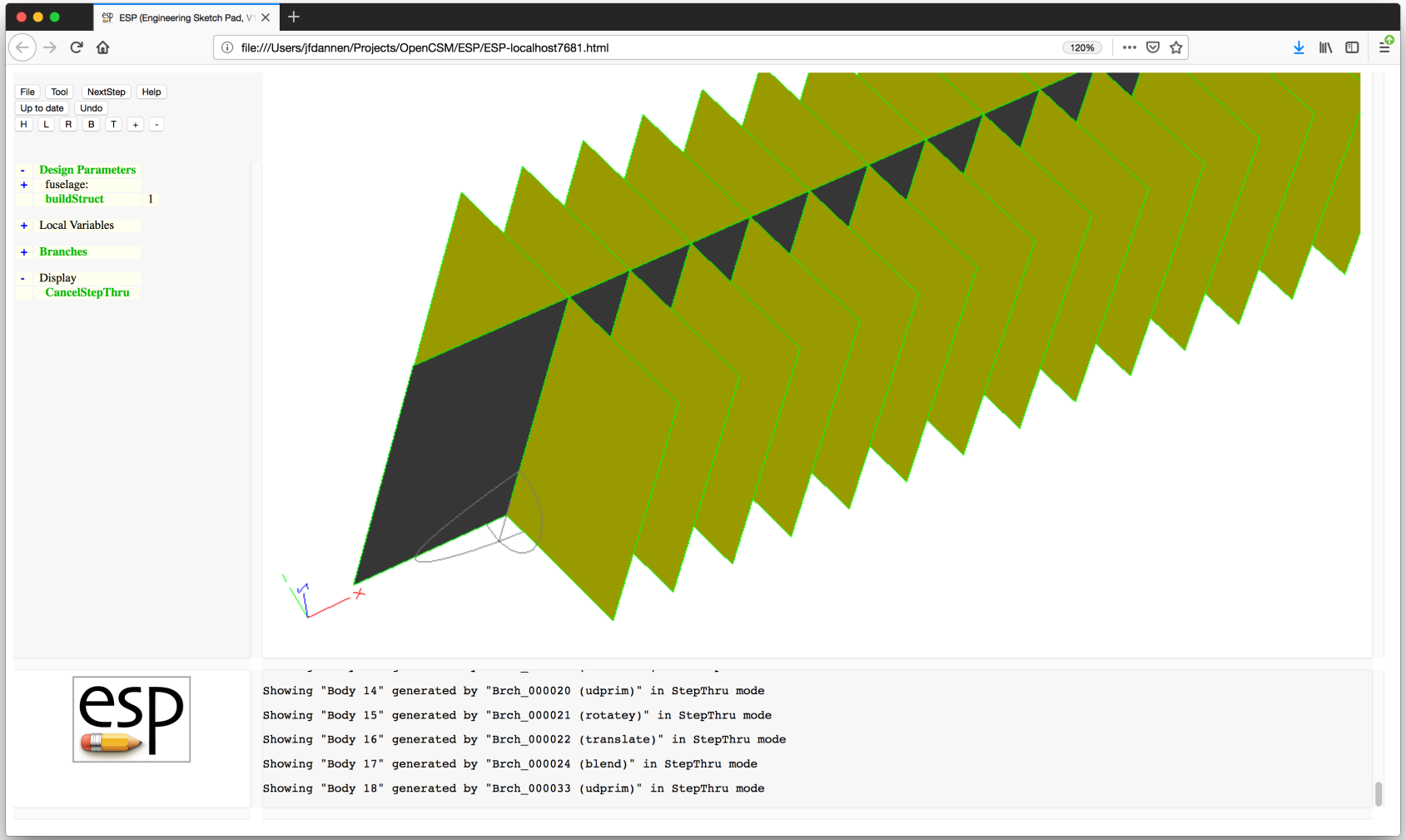
Simple Fuselage — Step 17



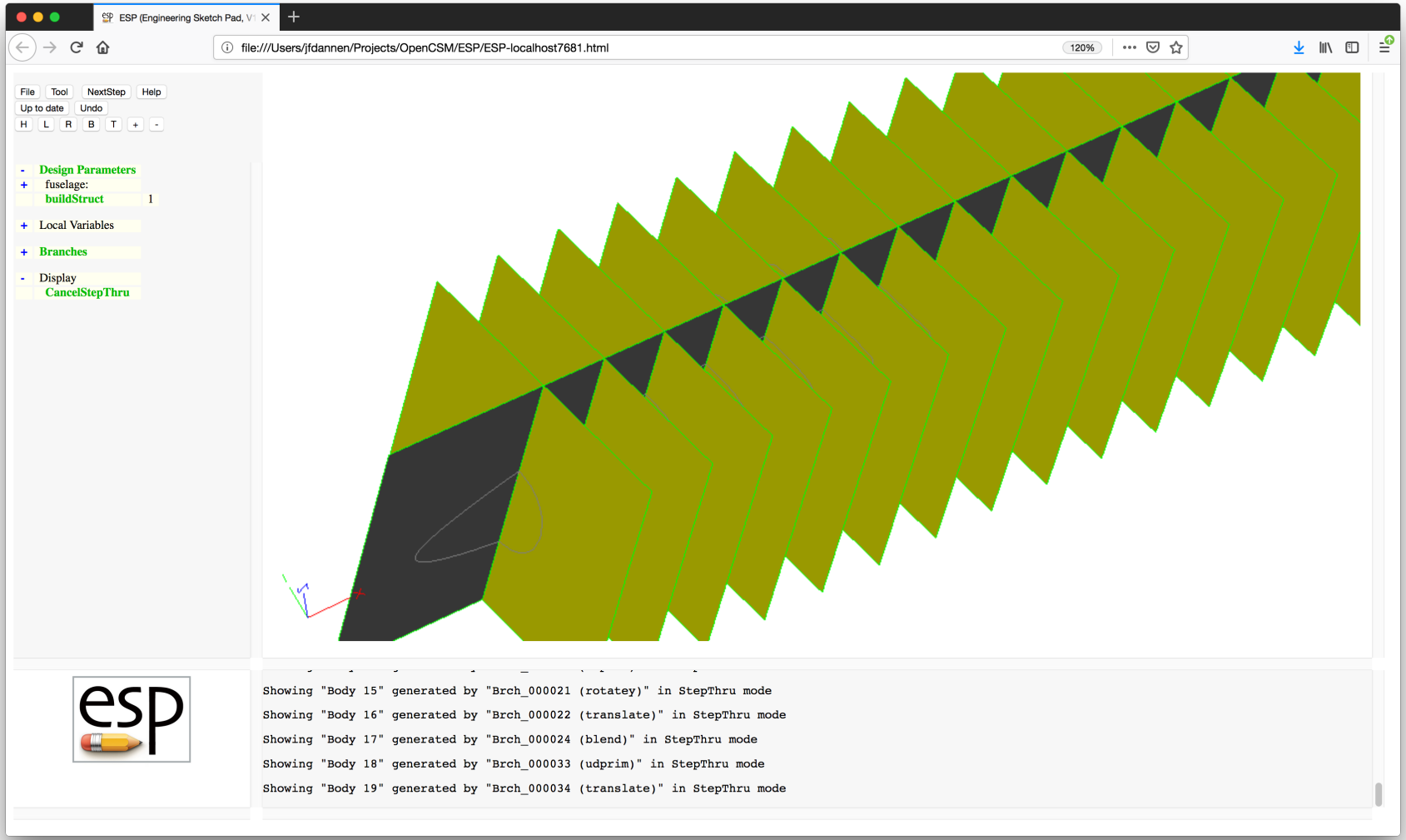
Simple Fuselage — Step 18



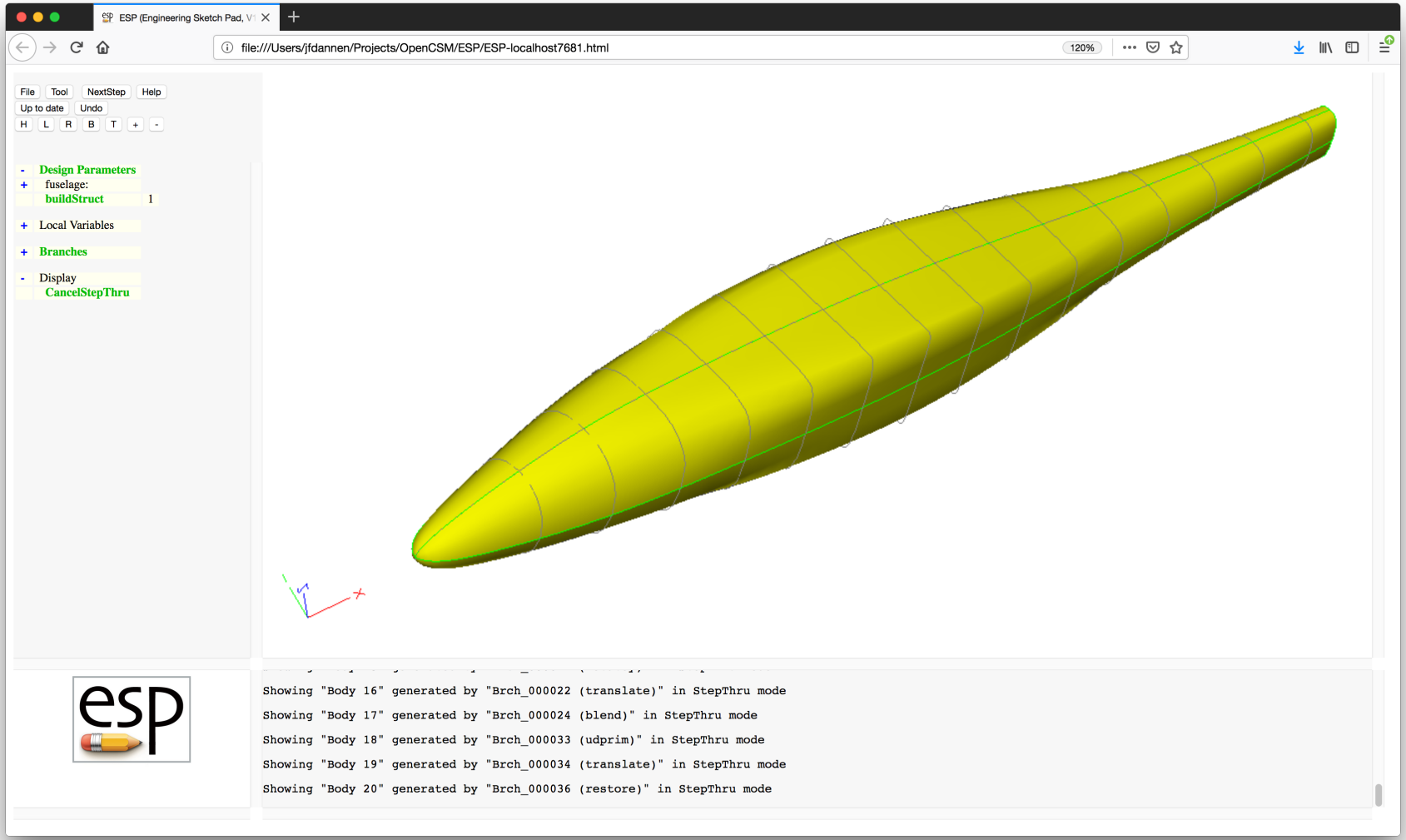
Simple Fuselage — Step 19



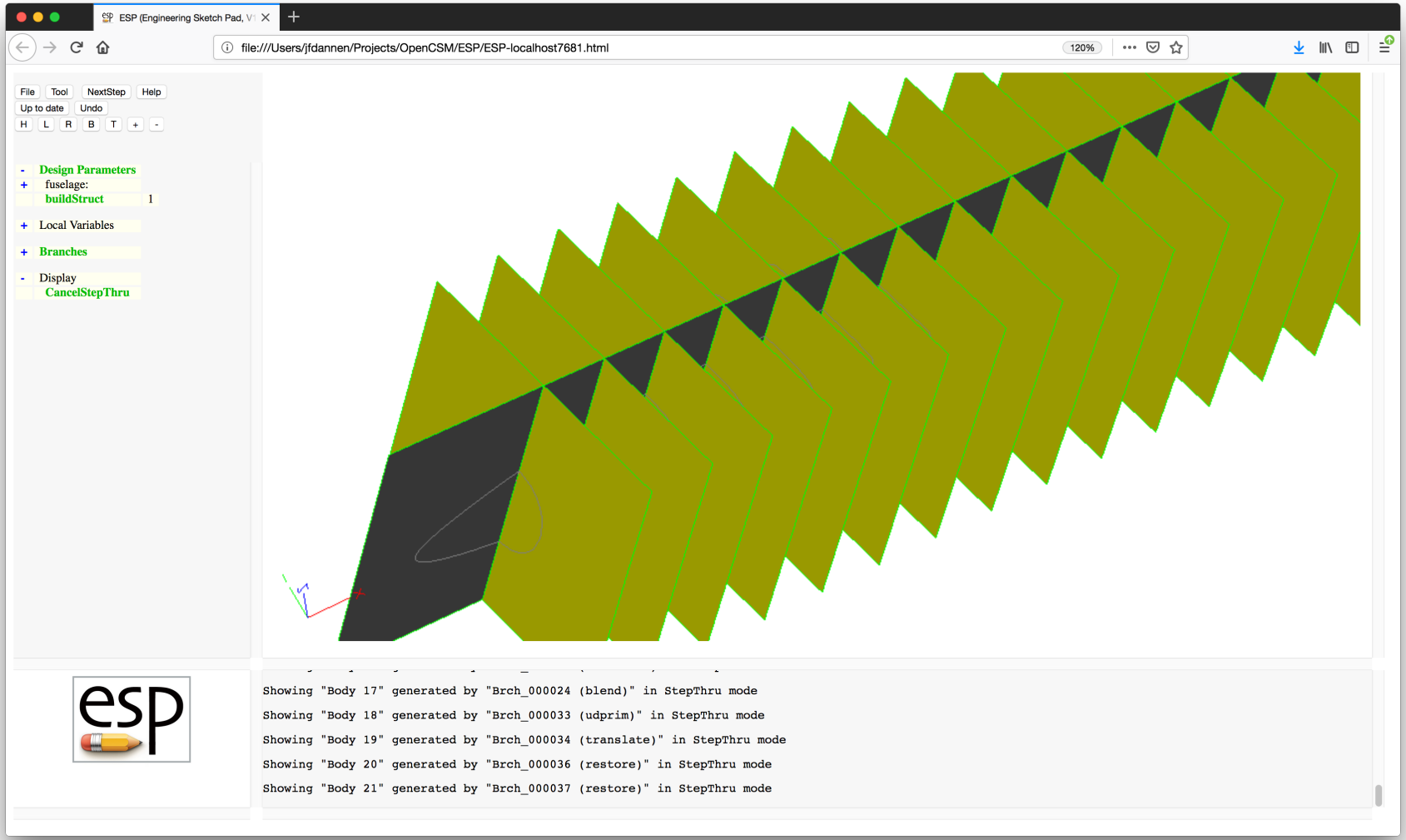
Simple Fuselage — Step 20



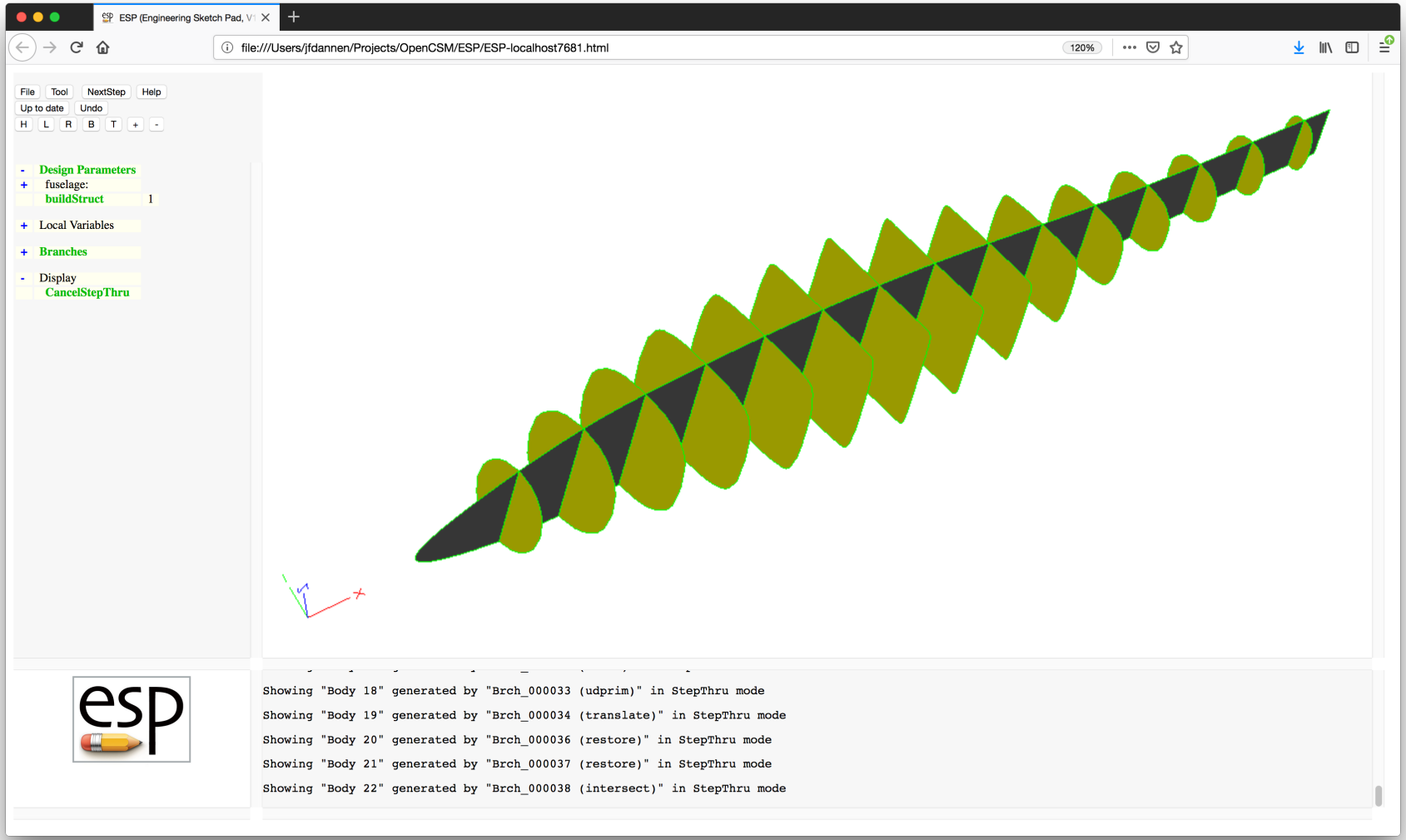
Simple Fuselage — Step 21



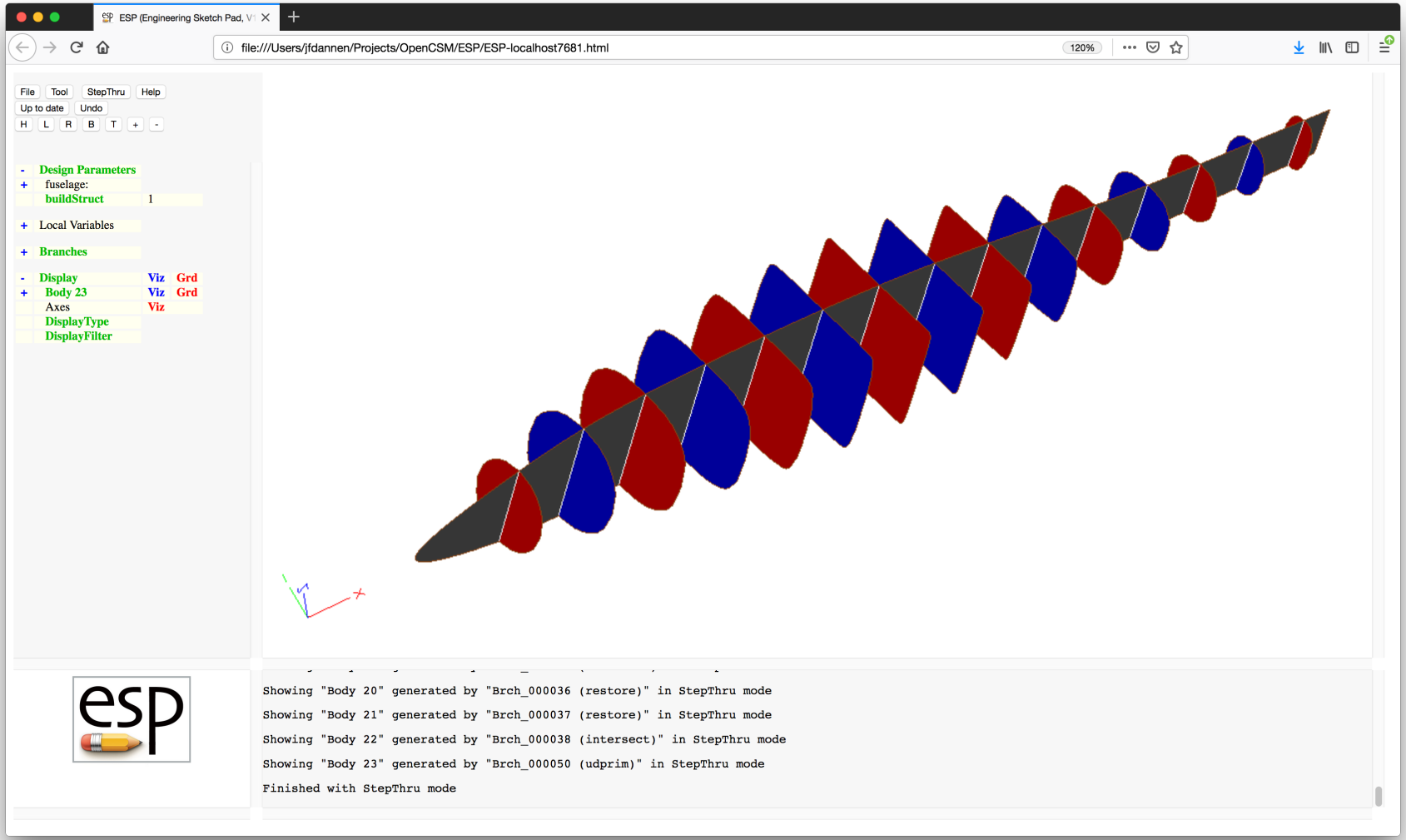
Simple Fuselage — Step 22



Simple Fuselage — Step 23



Simple Fuselage — Step 24



Simple Fuselage — `.csm` File (1)

```
# fuselageAlone
# written by John Dannenhoffer

# fuselage design Parameters
CFGPMTR                       fuselage:numXsect 6
DIMENSION fuselage:xloc       fuselage:numXsect 1 1
DIMENSION fuselage:zloc       fuselage:numXsect 1 1
DIMENSION fuselage:width      fuselage:numXsect 1 1
DIMENSION fuselage:height     fuselage:numXsect 1 1
DIMENSION fuselage:power      fuselage:numXsect 1 1
DIMENSION fuselage:noselist   2                 4 1

DESPMTR   fuselage:xloc       "0;  1.0;  4.0;  8.0;  12.0;  16.0;"
DESPMTR   fuselage:zloc       "0;  0.1;  0.4;  0.4;   0.3;   0.2;"
DESPMTR   fuselage:width      "0;  1.0;  1.6;  1.6;   1.0;   0.8;"
DESPMTR   fuselage:height     "0;  1.0;  2.0;  2.0;   1.2;   0.4;"
DESPMTR   fuselage:power      "2;    2;    3;    3      3;     3;"
DESPMTR   fuselage:noselist   "0.2; 0; 1; 0;\
                               0.1; 0; 0; 1;"

CFGPMTR   buildStruct         0   # set to 1 to build structure
```

Simple Fuselage — `.csm` File (2)

```
# build fuselage OML
MARK

   # sharp or rounded nose
   SET isect 1
   IFTHEN fuselage:width[isect] eq 0 and fuselage:height[isect] eq 0
      POINT  fuselage:xloc[isect] 0 fuselage:zloc[isect]

   # blunt nose
   ELSE
      UDPRIM supell rx fuselage:width[isect]/2 \
                    ry fuselage:height[isect]/2 \
                    n  fuselage:power[isect]
      ROTATEY 90 0 0
      TRANSLATE fuselage:xloc[isect] 0 fuselage:zloc[isect]
   ENDIF
```

Simple Fuselage — .csm File (3)

```
   # intermediate sections
   PATBEG jsect fuselage:numXsect-2
      SET isect jsect+1

      UDPRIM supell rx fuselage:width[isect]/2 ry fuselage:height[isect]/2 n fusela
      ROTATEY 90 0 0
      TRANSLATE fuselage:xloc[isect] 0 fuselage:zloc[isect]
   PATEND

   # sharp or rounded tail
   SET isect fuselage:numXsect
   IFTHEN fuselage:width[isect] eq 0 and fuselage:height[isect] eq 0
      POINT  fuselage:xloc[isect] 0 fuselage:zloc[isect]

   # blunt tail
   ELSE
      UDPRIM supell rx fuselage:width[isect]/2 ry fuselage:height[isect]/2 n fusela
      ROTATEY 90 0 0
      TRANSLATE fuselage:xloc[isect] 0 fuselage:zloc[isect]
   ENDIF

# blend the sections into the fuselage
BLEND fuselage:noselist
```

Simple Fuselage — `.csm` File (4)

```
# optionally build the structure
IFTHEN    buildStruct EQ 1

   # get the fuselage bounding box
   SET  xmin  @xmin
   SET  xmax  @xmax
   SET  ymin  @ymin
   SET  ymax  @ymax
   SET  zmin  @zmin
   SET  zmax  @zmax

   # store OML for later use
   STORE   fuseOML
```

Simple Fuselage — `.csm` File (5)

```
   # create a waffle that is "1" bigger than the OML
   UDPRIM  waffle  depth  zmax-zmin+2  filename <<

       # symmetry plane
       POINT A AT xmin-1 0
       POINT B AT xmax+1 0
       LINE  . A  B      tagType=symmetry

       # make the bulkheads
       PATBEG ibulk xmax-xmin-1
          POINT C AT ibulk+1/2 ymin-1
          POINT D AT x@C       ymax+1
          LINE  . C  D         tagType=bulkhead  tagIndex=!val2str(ibulk,0)
       PATEND
>>
```

Simple Fuselage — `.csm` File (6)

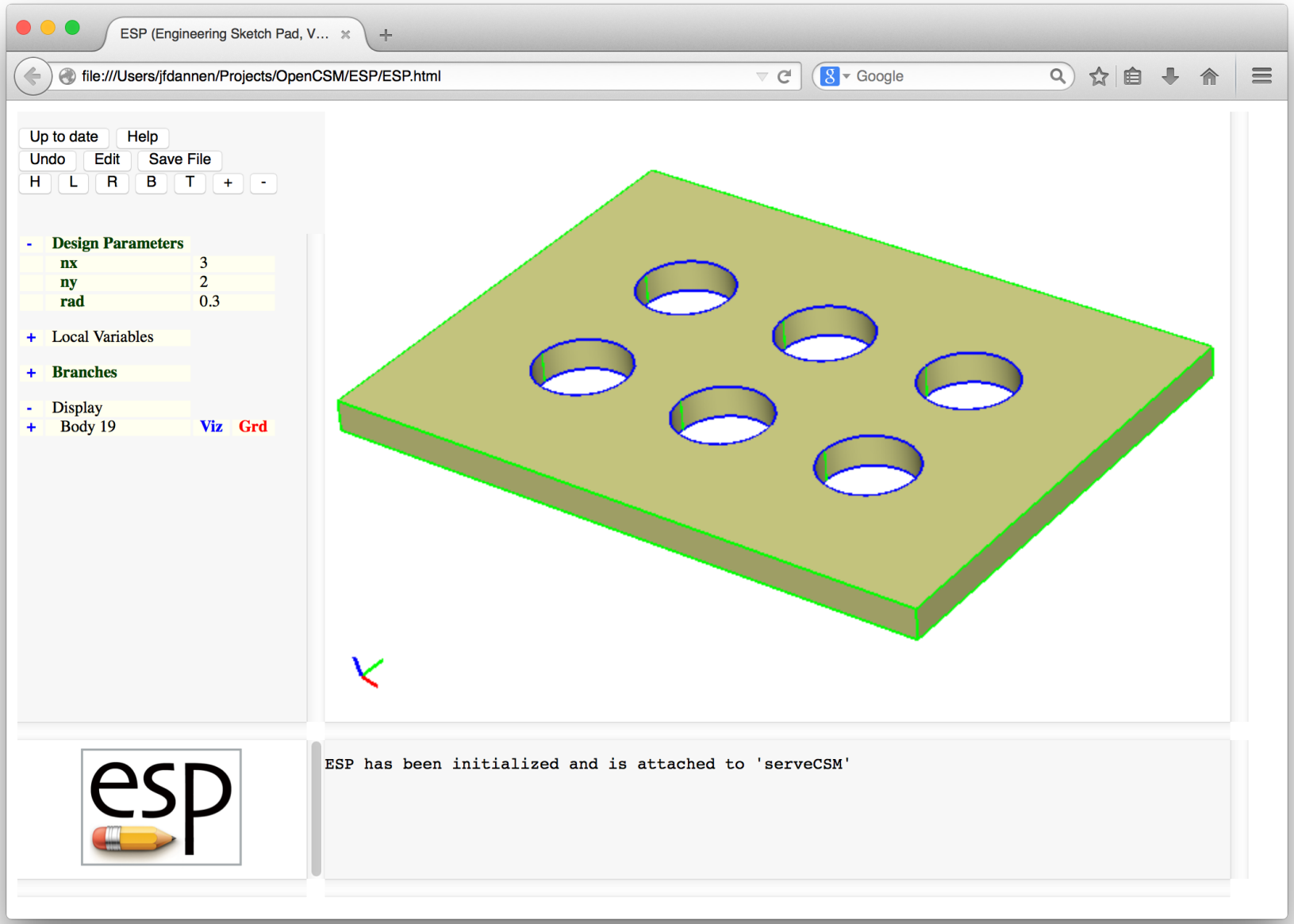
```
# translate the waffle down and store it
TRANSLATE 0  0  zmin-1
STORE   fuseWaffle

# trim the waffle to the fuselage
RESTORE  fuseOML
RESTORE  fuseWaffle
INTERSECT

# alternate the bulkhead colors red/blue/red/...
SET       color  $red
PATBEG    ibulk  99
   SELECT FACE   $tagType  $bulkhead  $tagIndex  val2str(ibulk,0)
      ATTRIBUTE  _color  color

   IFTHEN color EQ $red
      SET color    $blue
   ELSE
      SET color    $red
   ENDIF
PATEND
```

Simple Fuselage — `.csm` File (7)

```
   # this will get called when we run out of bulkheads
   CATBEG  $face_not_found
   CATEND

   # make the bulkhead/symmetry Edges white
   UDPRIM    editAttr  filename <<
      EDGE  ADJ2FACE  tagType=bulkhead
      AND   ADJ2FACE  tagType=symmetry
      SET             _color=white
>>

ENDIF

END
```

Session 5 Solutions
CSM Language (2)

Rectangular Plate with Holes (1)



Rectangular Plate with Holes (2)

nx	number of holes in X-direction	3.00
ny	number of holes in Y-direction	2.00
rad	radius of each hole	0.30
	distance between hole centers	1.00

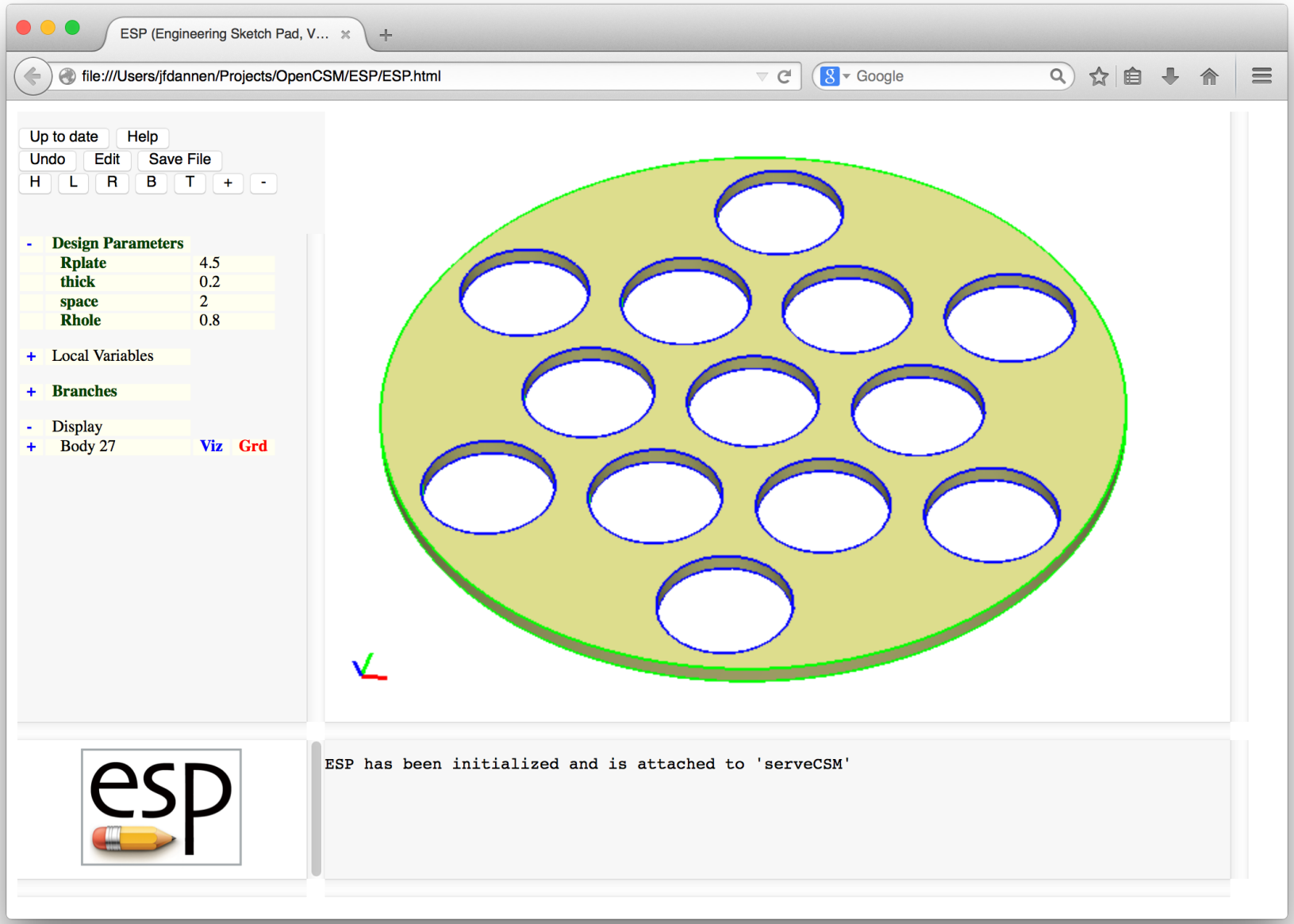
Rectangular Plate with Holes (3)

- Can you make a single hole in the center of the plate?
- Can you change your solution to have the holes spaced so that they fill the plate?
- What if you make the radius of the hole too big?

Rectangular Plate with Holes (4)

```
# rect_pat
# written by John Dannenhoffer

DESPMTR   nx        3
DESPMTR   ny        2
DESPMTR   rad       0.30
DESPMTR   space     1.00

# base plate (big enough to contain all holes)
BOX       0.00   0.00  -0.10   space*nx+1   space*ny+1   0.20

# 2D array of holes (with given spacing)
PATBEG ix nx
   PATBEG iy ny
      CYLINDER   ix*space  iy*space  -0.20 \
                 ix*space  iy*space  +0.20   rad
      SUBTRACT
   PATEND
PATEND

END
```

Round Plate with Holes (1)



Round Plate with Holes (2)

Rplate	radius or plate	4.50
thick	thickness of plate	0.20
space	distance between hole centers	2.00
Rhole	radius of holes	0.80
	number of holes selected automatically	

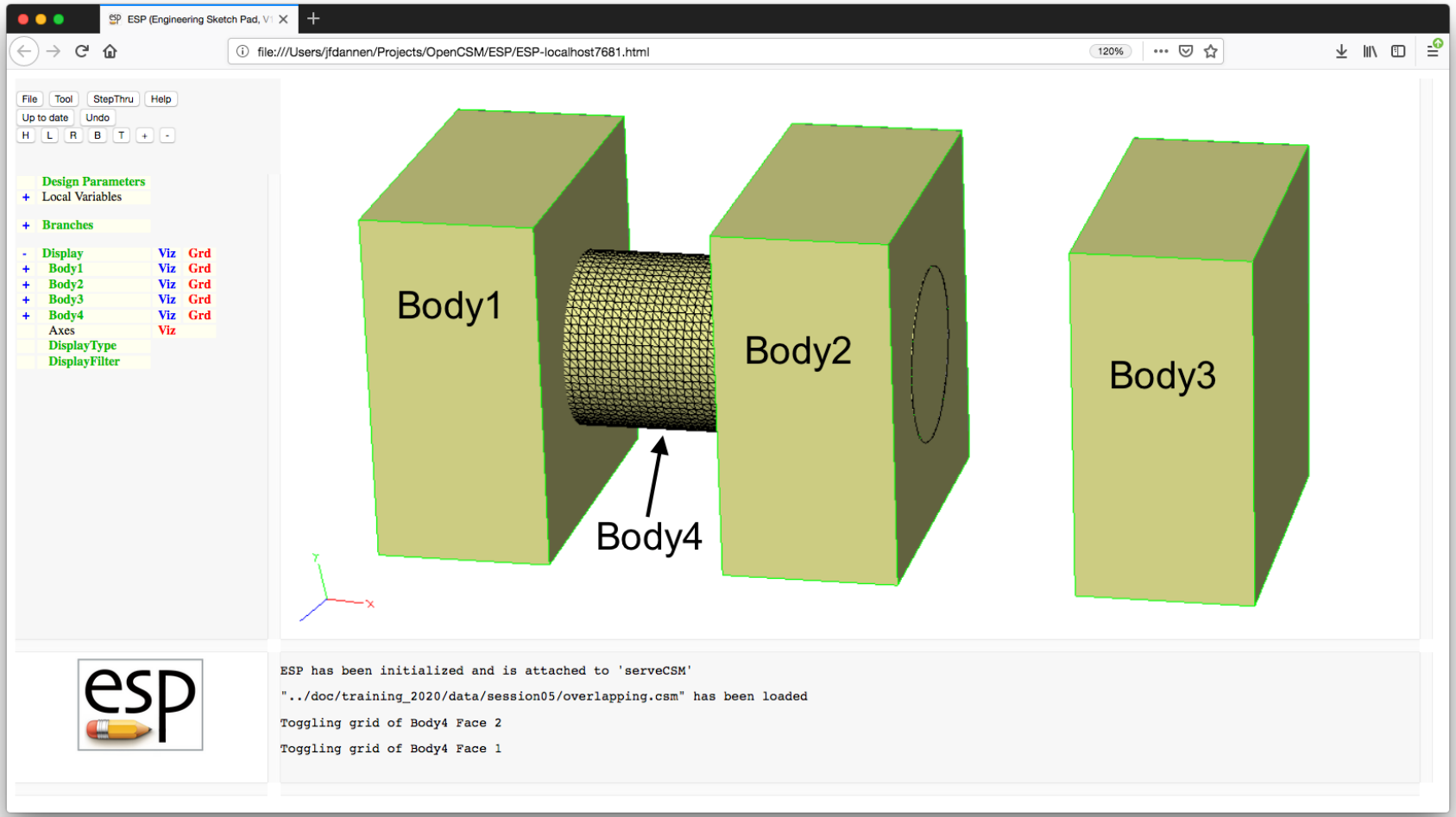
Round Plate with Holes (3)

```
# round_pat
# written by John Dannenhoffer

# default design parameters
DESPMTR   Rplate     4.5000   # radius    of plate
DESPMTR   thick      0.2000   # thickness of plate
DESPMTR   space      2.0000   # distance between hole centers
DESPMTR   Rhole      0.8000   # radius of holes

# make sure holes do not intersect with each other
IFTHEN    space LT 2*Rhole
   THROW  999
ENDIF

# overall plate
CYLINDER 0 0 -thick/2 0 0 +thick/2 Rplate
```

Round Plate with Holes (4)

```
# pattern for holes
SET nr int(Rplate/space)

PATBEG iy 1+2*nr
   PATBEG ix 1+2*nr

      SET xc "(ix-nr-1)*space + (iy-nr-1)*space*cosd(60)"
      SET yc                    "(iy-nr-1)*space*sind(60)"
      SET r  hypot(xc,yc)+Rhole

      # mask hole if not within circle
      IFTHEN r LT Rplate-0.001
         CYLINDER xc yc -thick xc yc +thick Rhole
         SUBTRACT
      ENDIF
   PATEND
PATEND

END
```

Overlapping Bodys (1)



Overlapping Bodys (2)

- Write `.csm` file to:
  - set `overlap1` to 1 if Bodys 1 and 4 overlap, otherwise set it to 0
  - set `overlap2` to 1 if Bodys 2 and 4 overlap, otherwise set it to 0
  - set `overlap3` to 1 if Bodys 3 and 4 overlap, otherwise set it to 0
- Try to use a pattern to do this compactly

Overlapping Bodys (3)

```
# overlapping
# written by John Dannenhoffer

# Body 1
BOX       0   0   0   1   2   2
STORE     body  1

# Body 2
BOX       2   0   0   1   2   2
STORE     body  2

# Body 3
BOX       4   0   0   1   2   2
STORE     body  3

# Body 4
CYLINDER  0   1   1   3   1   1   0.5
STORE     body  4
```

Overlapping Bodys (4)

```
# determine which or Bodys 1, 2, 3 intersect Body 4
PATBEG       ibody  3
   SET       !$overlap+ibody  1
   RESTORE   body  ibody
   RESTORE   body  4
   INTERSECT

   CATBEG    $did_not_create_body
      SET    !$overlap+ibody  0
   CATEND

   STORE     ...
PATEND
```

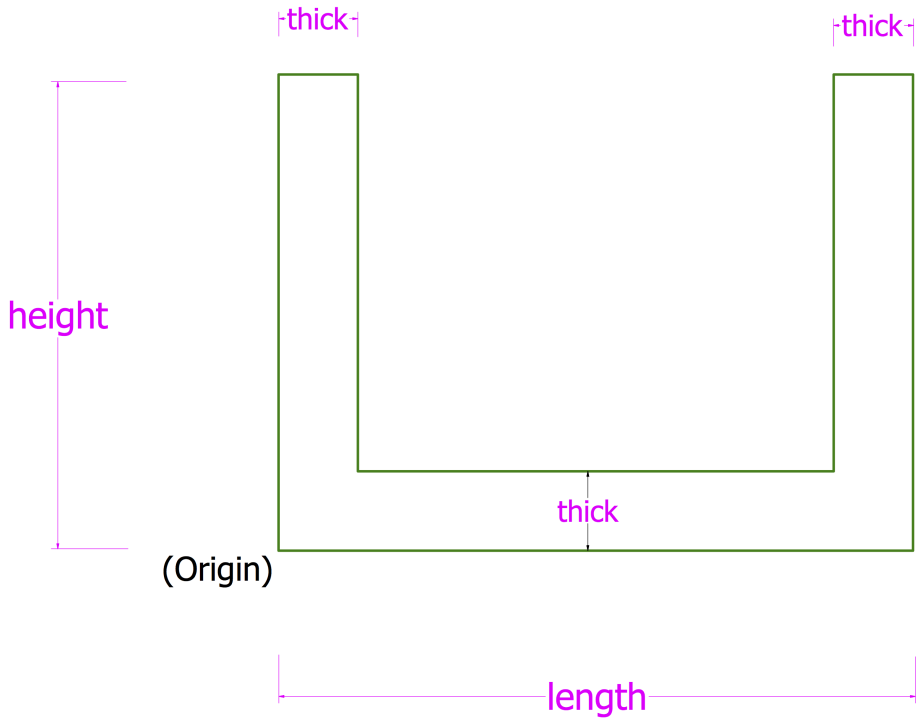
Overlapping Bodys (5)

```
# show Bodys
RESTORE    body  1
ATTRIBUTE  _name  $Body1

RESTORE    body  2
ATTRIBUTE  _name  $Body2

RESTORE    body  3
ATTRIBUTE  _name  $Body3

RESTORE    body  4
ATTRIBUTE  _name  $Body4

END
```

Session 7 Solutions
Sketcher Fundamentals

U-bracket (version 1)

Problem



Measurements
length = 4.00
height = 3.00
thick = 0.5

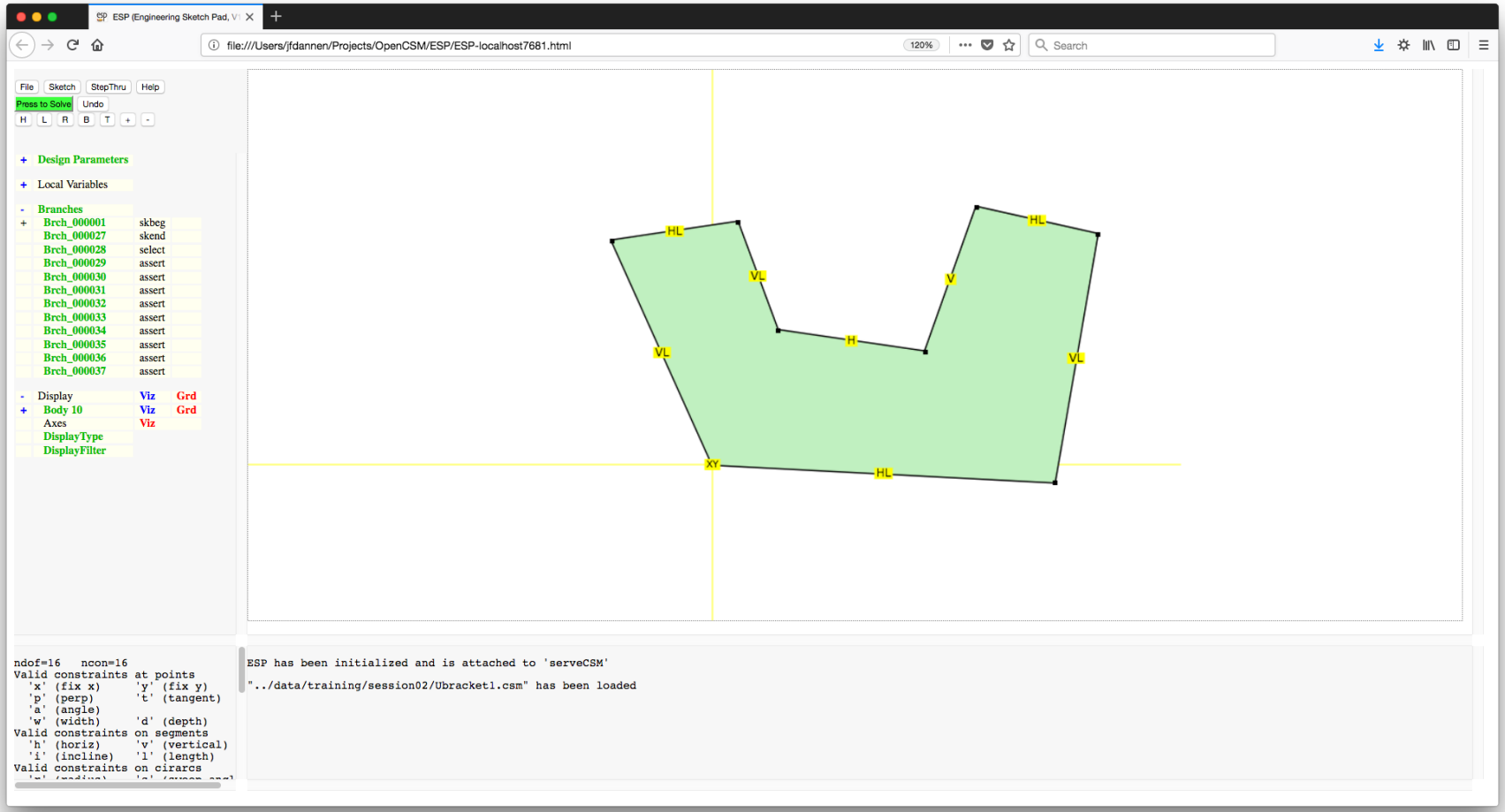
U-bracket (version 1)

Programmatic Solution

```
DESPMTR   length    4.00000
DESPMTR   height    3.00000
DESPMTR   thick     0.50000

SKBEG     0.0           0.0       0.0
   LINSEG length        0.0       0.0
   LINSEG length        height    0.0
   LINSEG length-thick  height    0.0
   LINSEG length-thick  thick     0.0
   LINSEG thick         thick     0.0
   LINSEG thick         height    0.0
   LINSEG 0.0           height    0.0
   LINSEG 0.0           0.0       0.0
SKEND
```

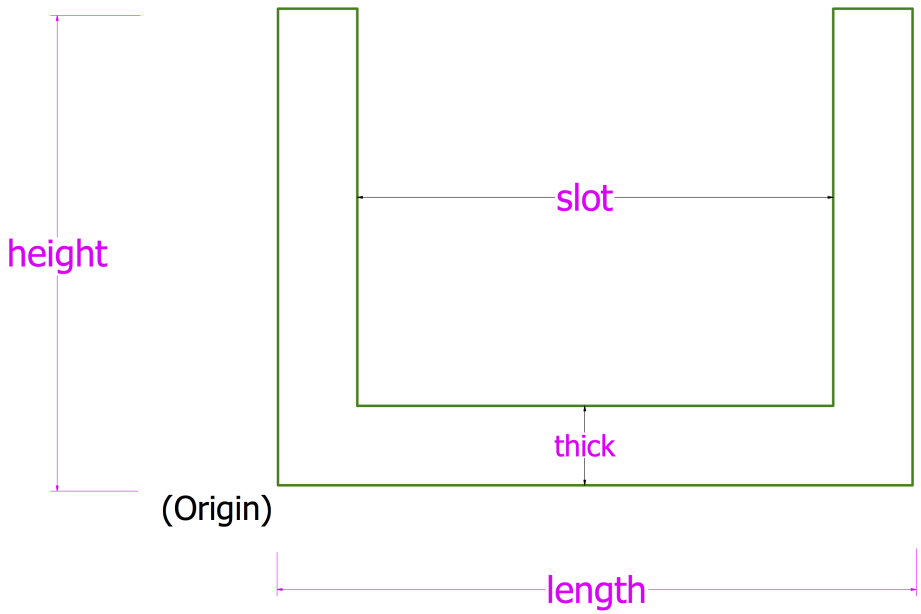
U-bracket (version 1)

Sketcher Solution



U-bracket (version 2)

Problem



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

Note: slot
is centered

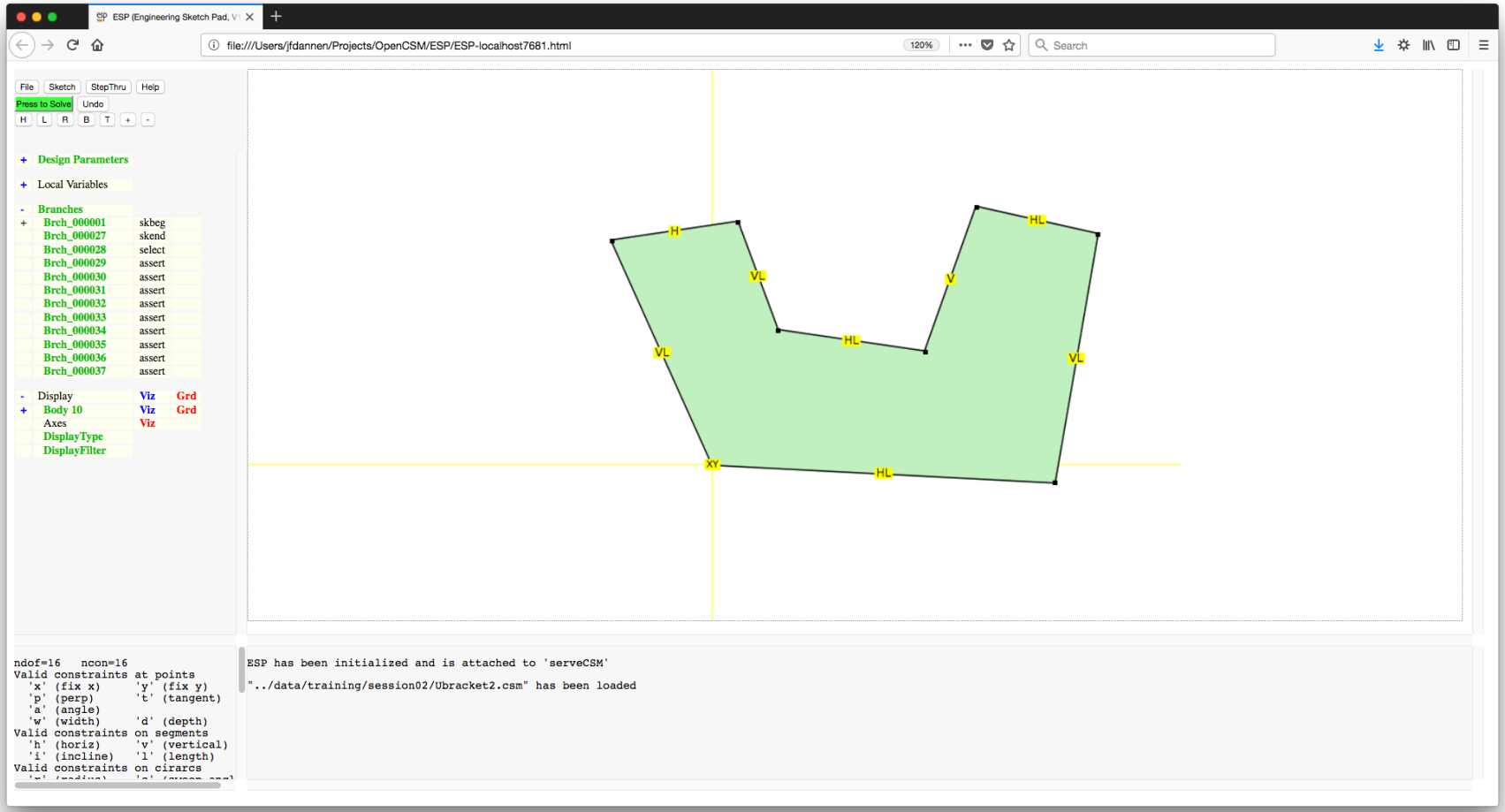
U-bracket (version 2)

Programmatic Solution

```
DESPMTR   height    3.00000
DESPMTR   thick     0.50000
DESPMTR   slot      2.00000

SET       length    slot+2*thick

SKBEG     0.0             0.0     0.0
   LINSEG length          0.0     0.0
   LINSEG length          height  0.0
   LINSEG length-thick    height  0.0
   LINSEG length-thick    thick   0.0
   LINSEG thick           thick   0.0
   LINSEG thick           height  0.0
   LINSEG 0.0             height  0.0
   LINSEG 0.0             0.0     0.0
SKEND
```

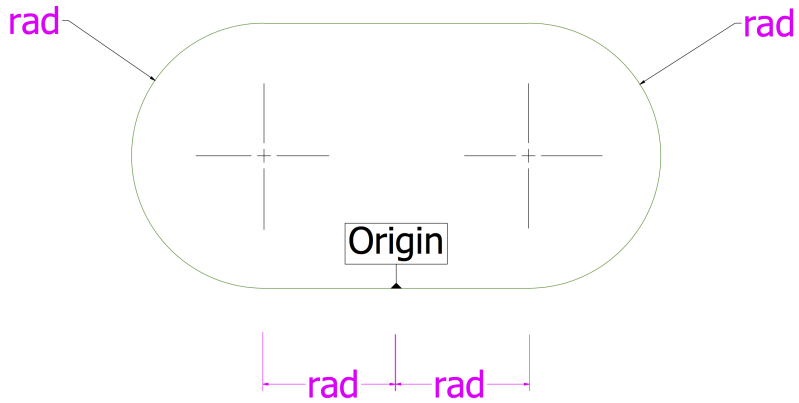
U-bracket (version 2)

Sketcher Solution



Oval

Problem



Measurements:
rad = 0.50

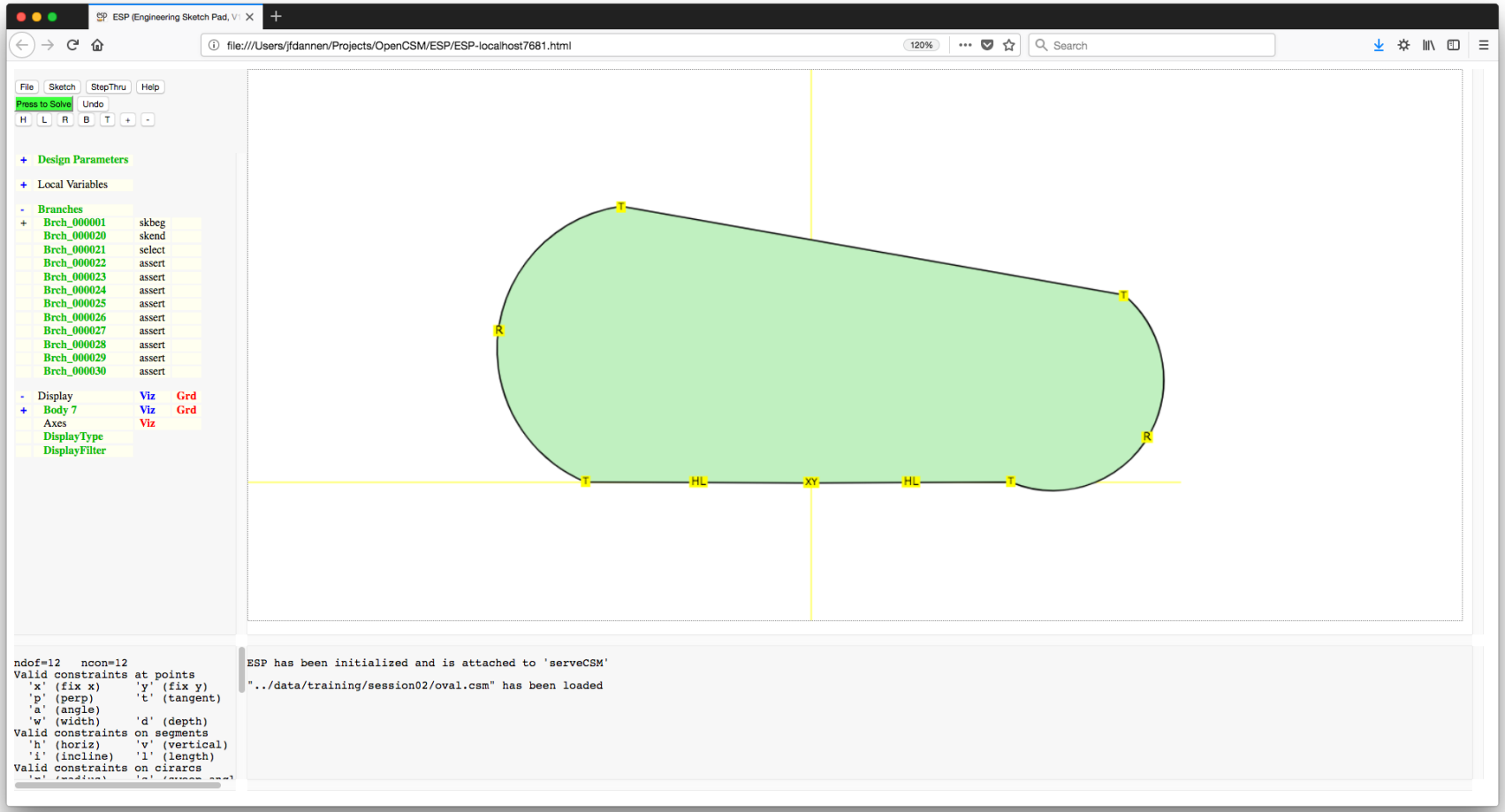
Oval

Programmatic Solution

```
DESPMTR   rad        0.50000

SKBEG        0.0     0.0  0.0
   LINSEG    rad     0.0  0.0
   CIRARC  2*rad     rad  0.0   rad  2*rad  0.0
   LINSEG   -rad   2*rad  0.0
   CIRARC -2*rad     rad  0.0  -rad    0.0  0.0
   LINSEG    0.0     0.0  0.0
SKEND
```

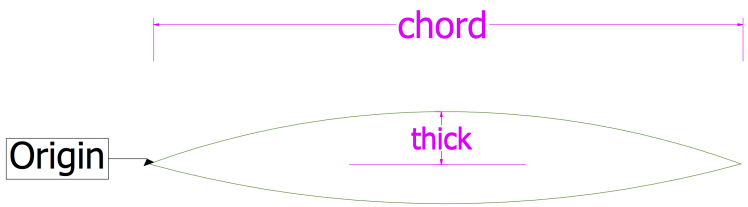
Oval

Sketcher Solution



Biconvex airfoil (with arcs)

Problem



Measurements:
chord = 2.00
thick = 0.10

Note:
Circular Arcs

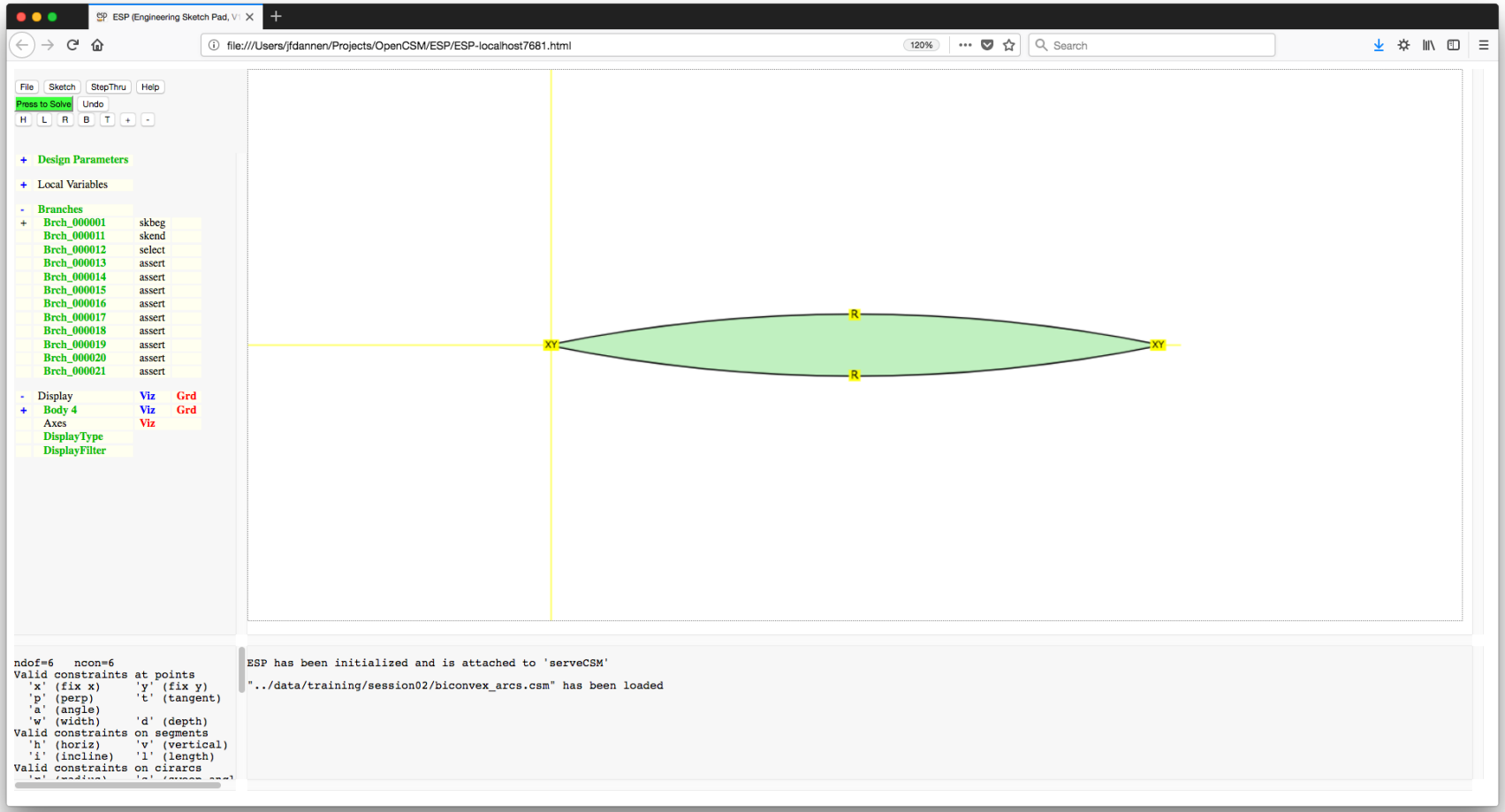
Biconvex airfoil (with arcs)

Programmatic Solution

```
DESPMTR   chord     2.00000
DESPMTR   thick     0.10000

SET       rad       radius(0,0,thick,chord,0)

SKBEG     0.0       0.0    0.0
   CIRARC chord/2  -thick  0.0  chord  0.0  0.0
   CIRARC chord/2   thick  0.0  0.0    0.0  0.0
SKEND
```

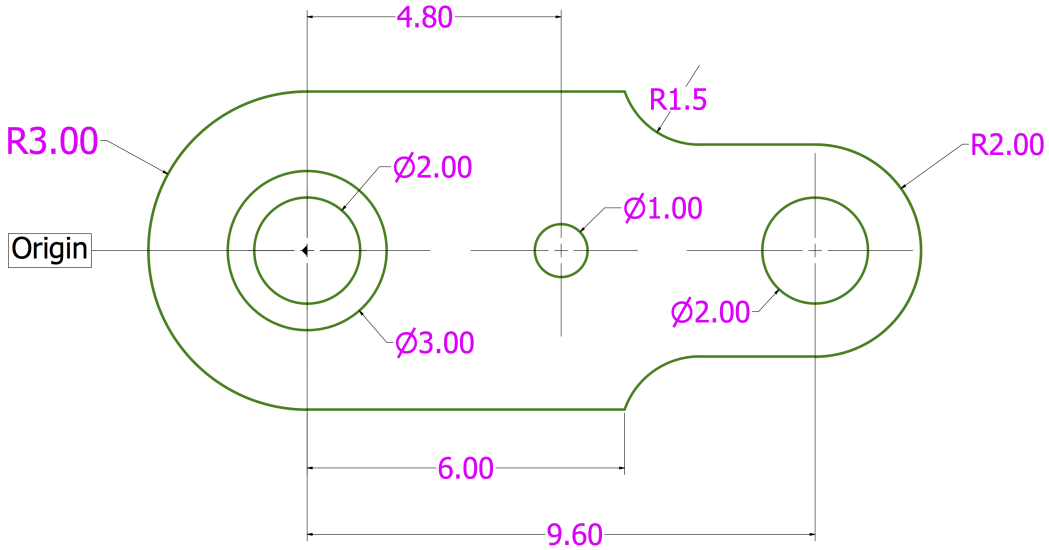
Biconvex airfoil (with arcs)

Sketcher Solution



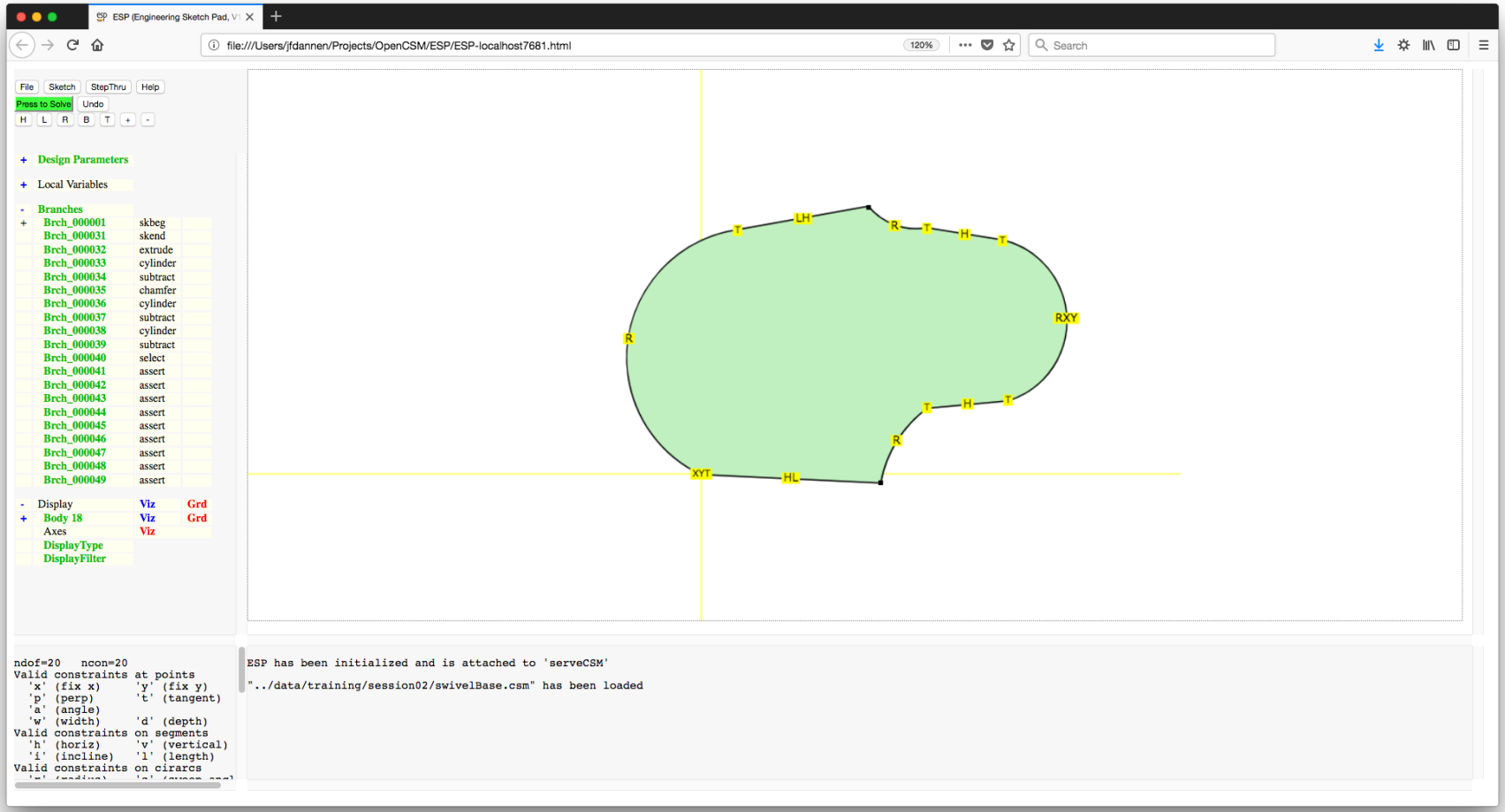
Swivel Base

Problem



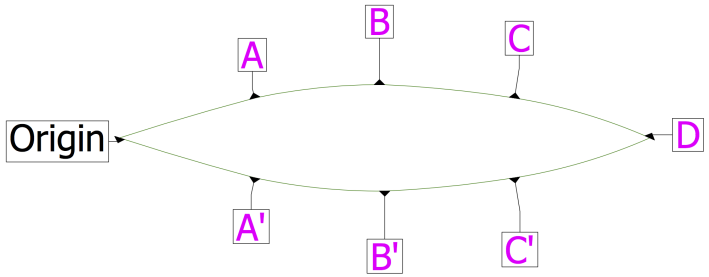
Swivel Base

Sketcher Solution



Biconvex Airfoil (with splines)

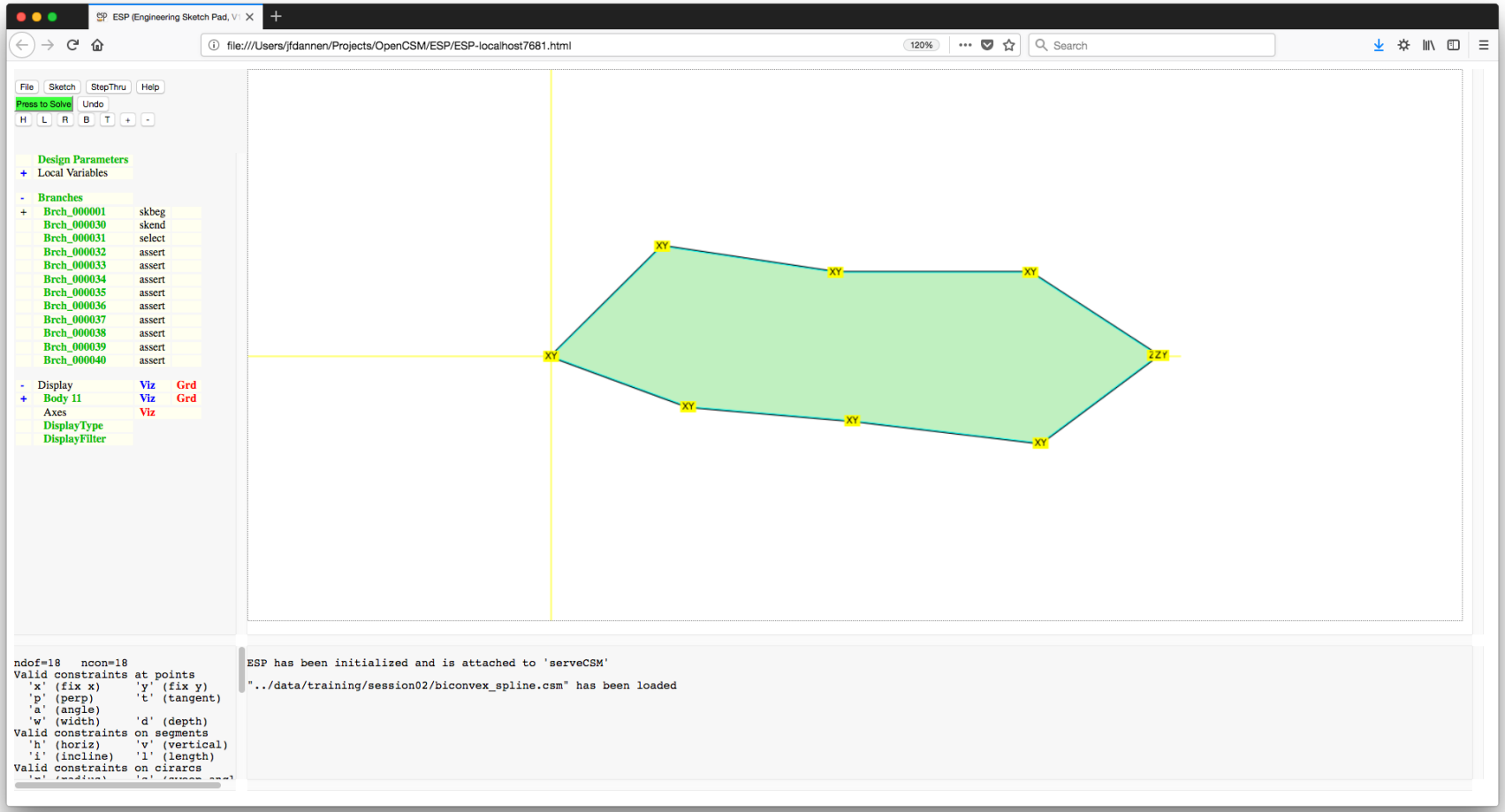
Problem



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

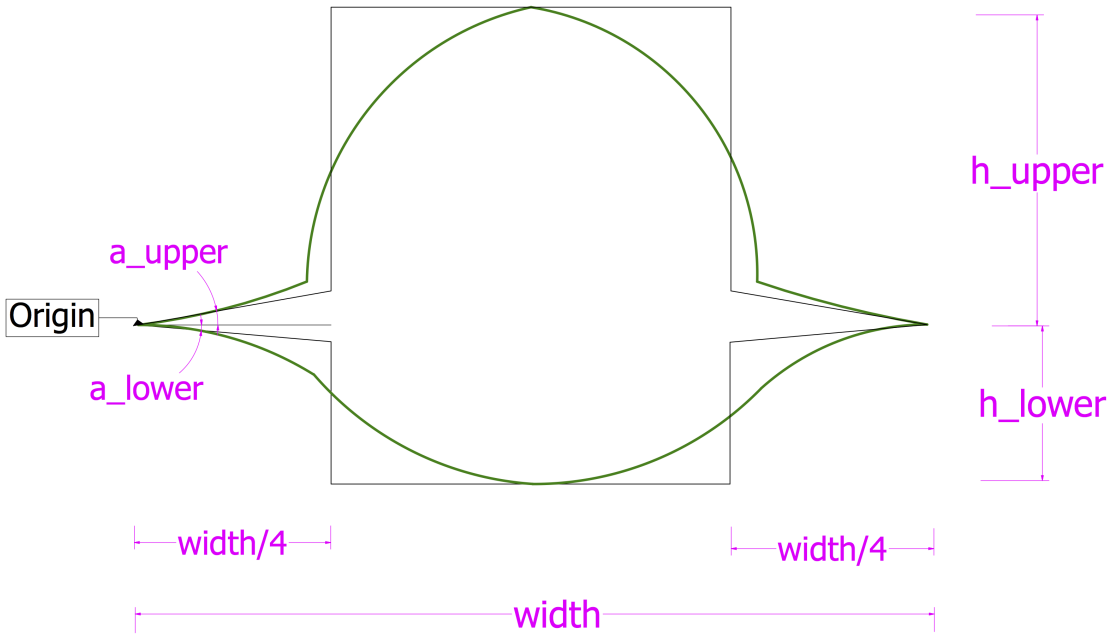
Biconvex Airfoil (with splines)

Sketcher Solution



Fuselage X-section (with Beziers)

Problem

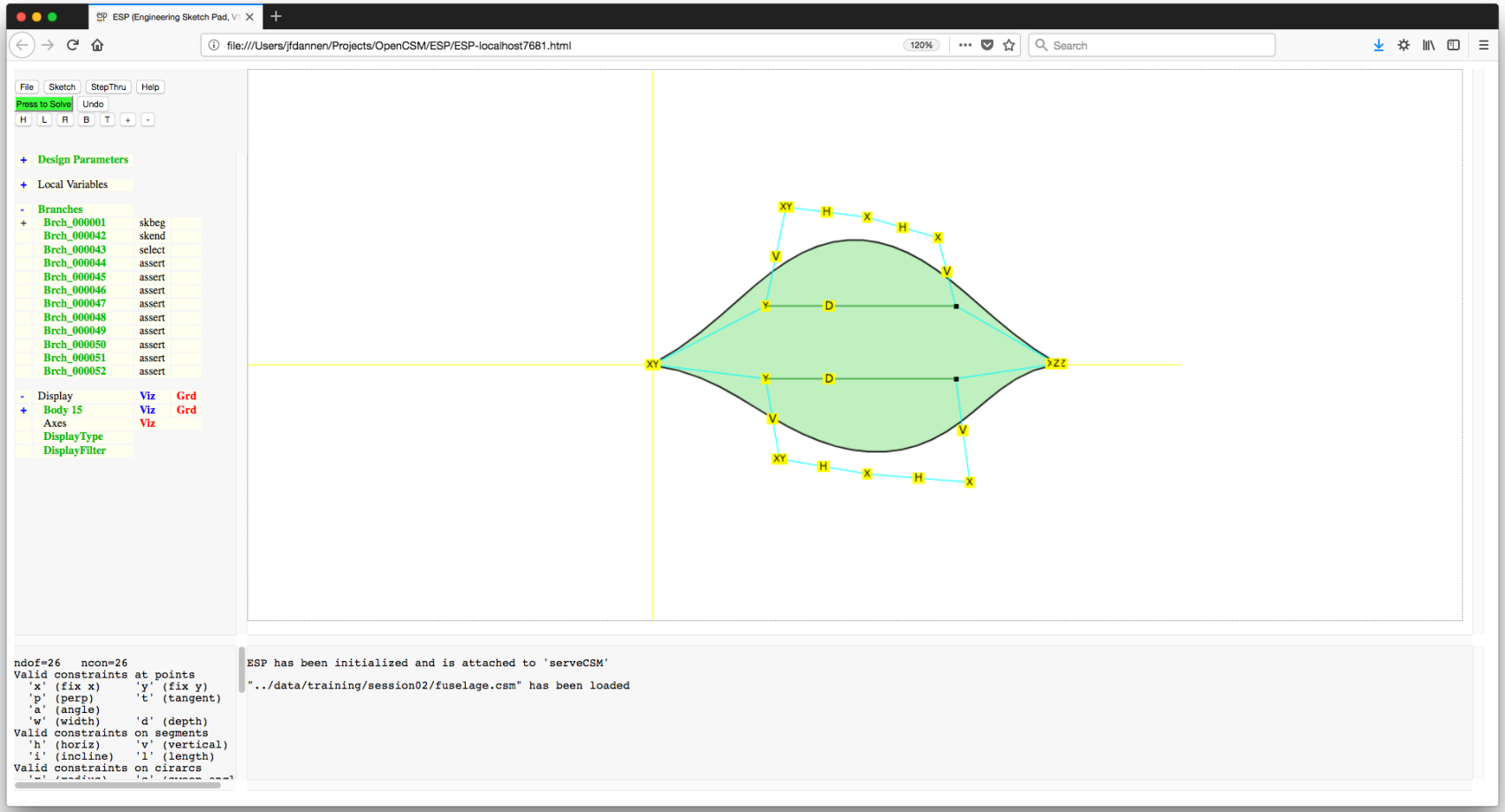


Measurements:
width = 5.00
h_upper = 2.00
h_lower = 1.00
a_upper = 10°
a_lower = 5°

Note:
4 Bezier Cubics

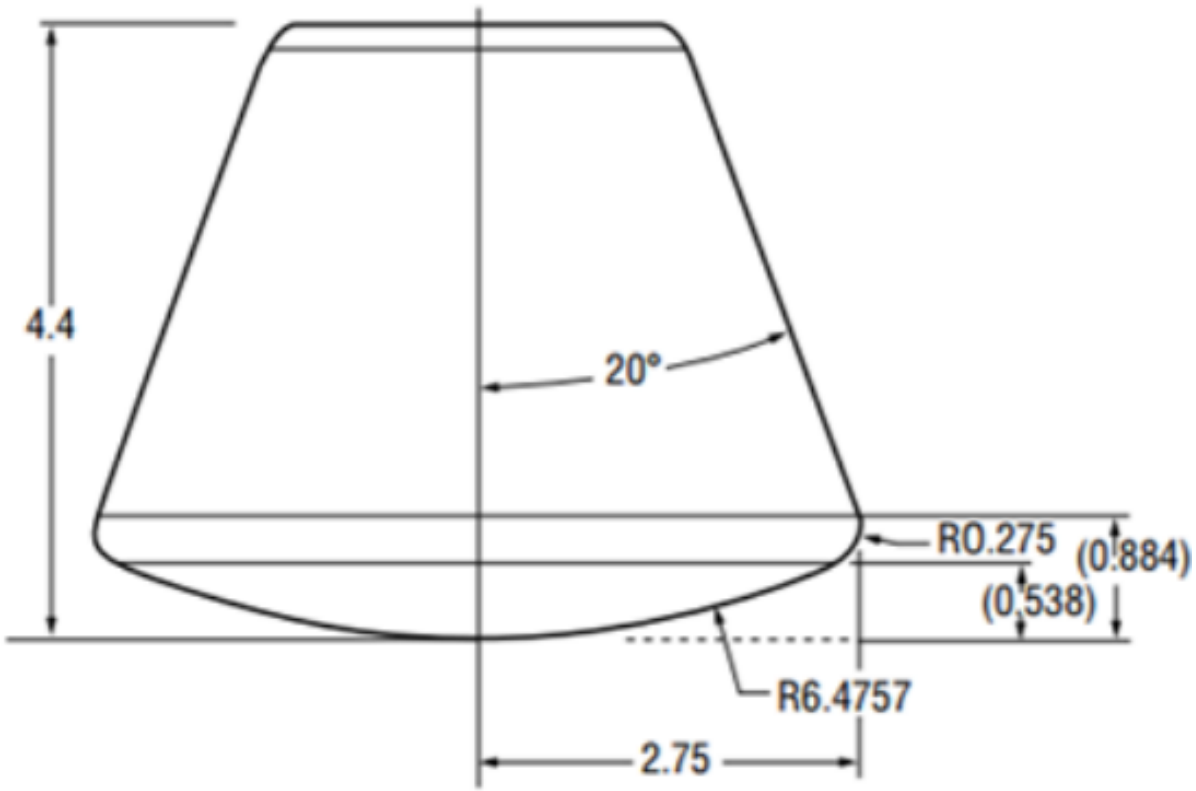
Fuselage X-section (with Beziers)

Sketcher Solution



Capsule

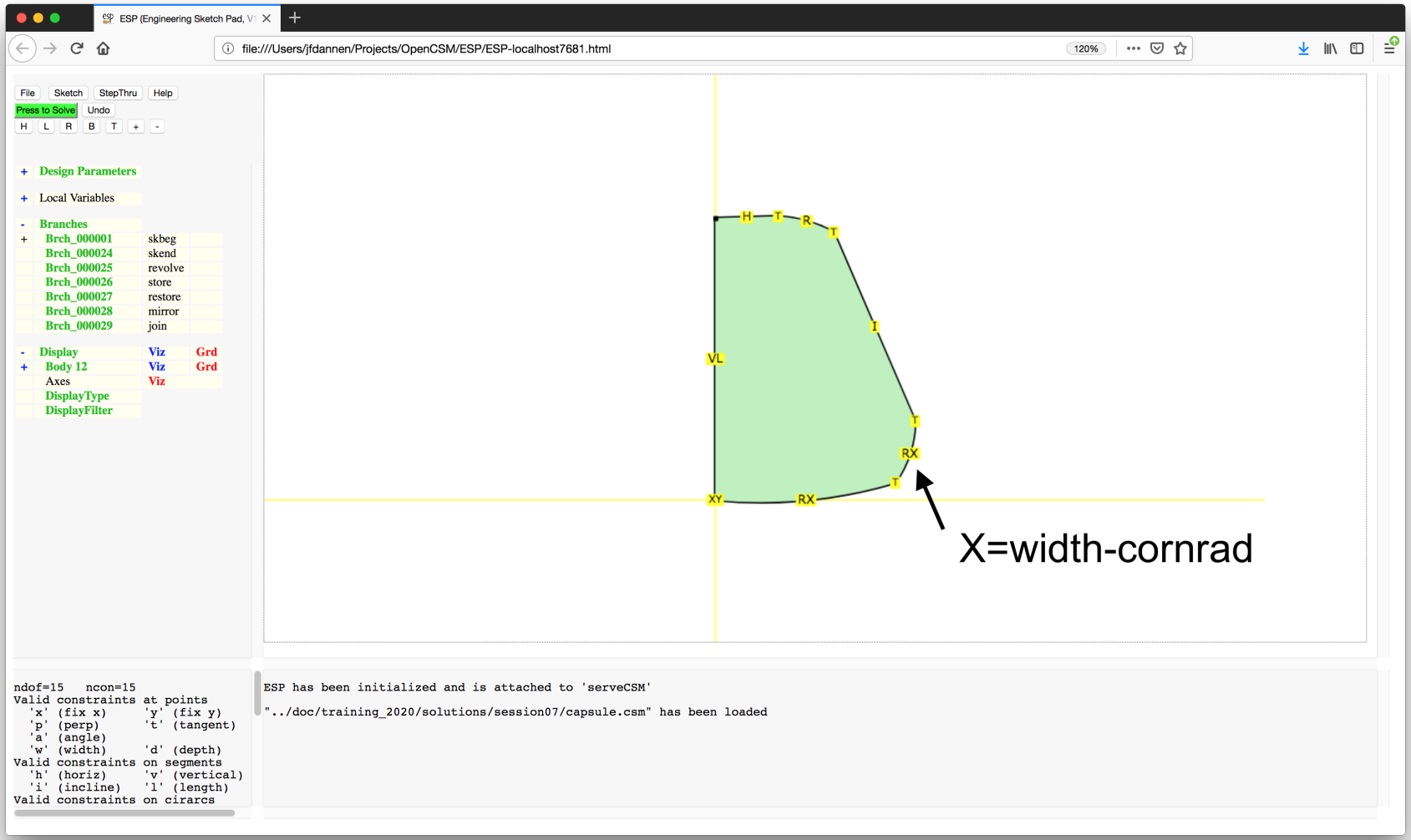
Problem




```
width     = 2.75000
baserad   = 6.47570
cornrad   = 0.27500
coneangle = 20.00000
height    = 4.40000
```

Capsule

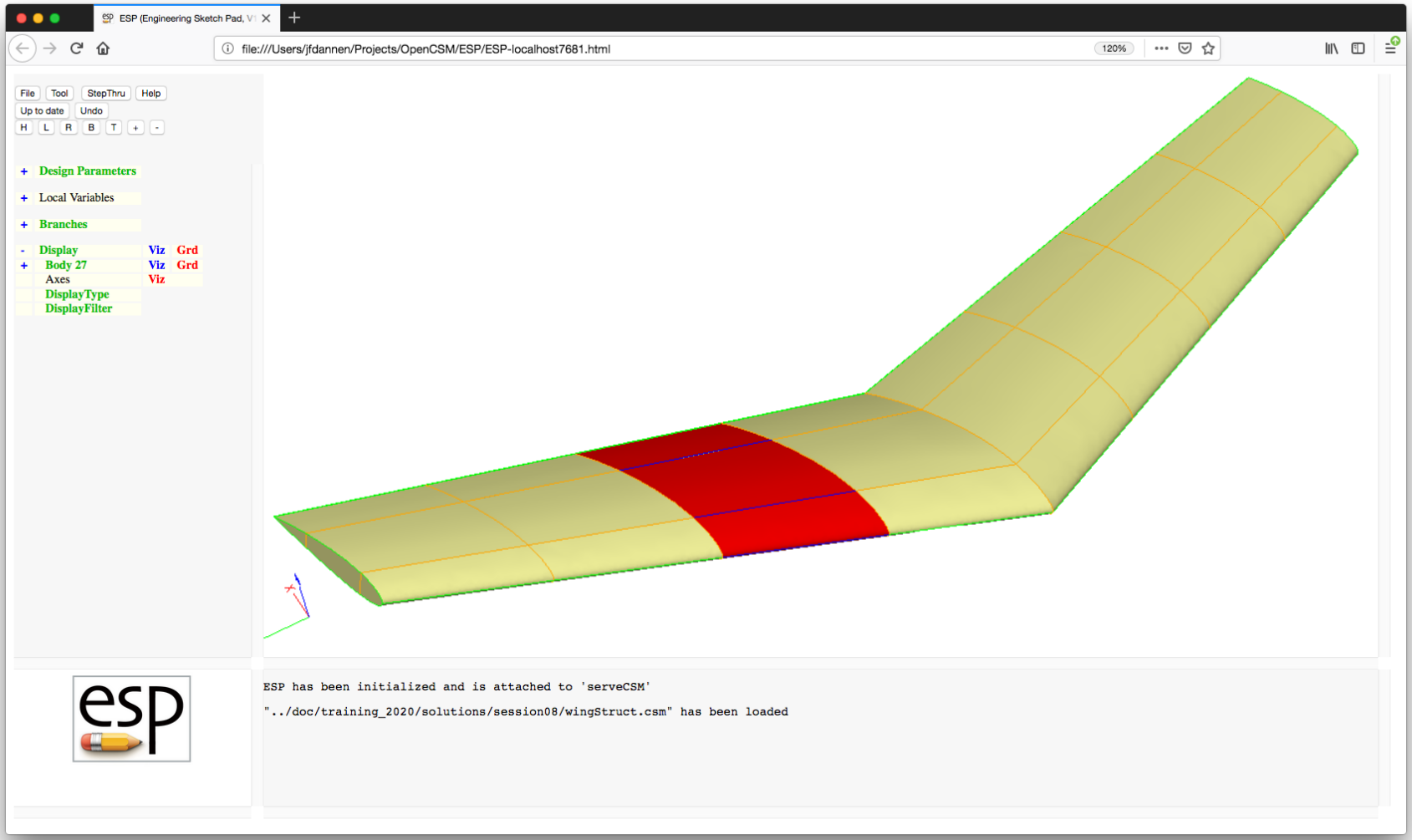
Sketcher Solution



Session 8 Solutions
Selection & Attribution

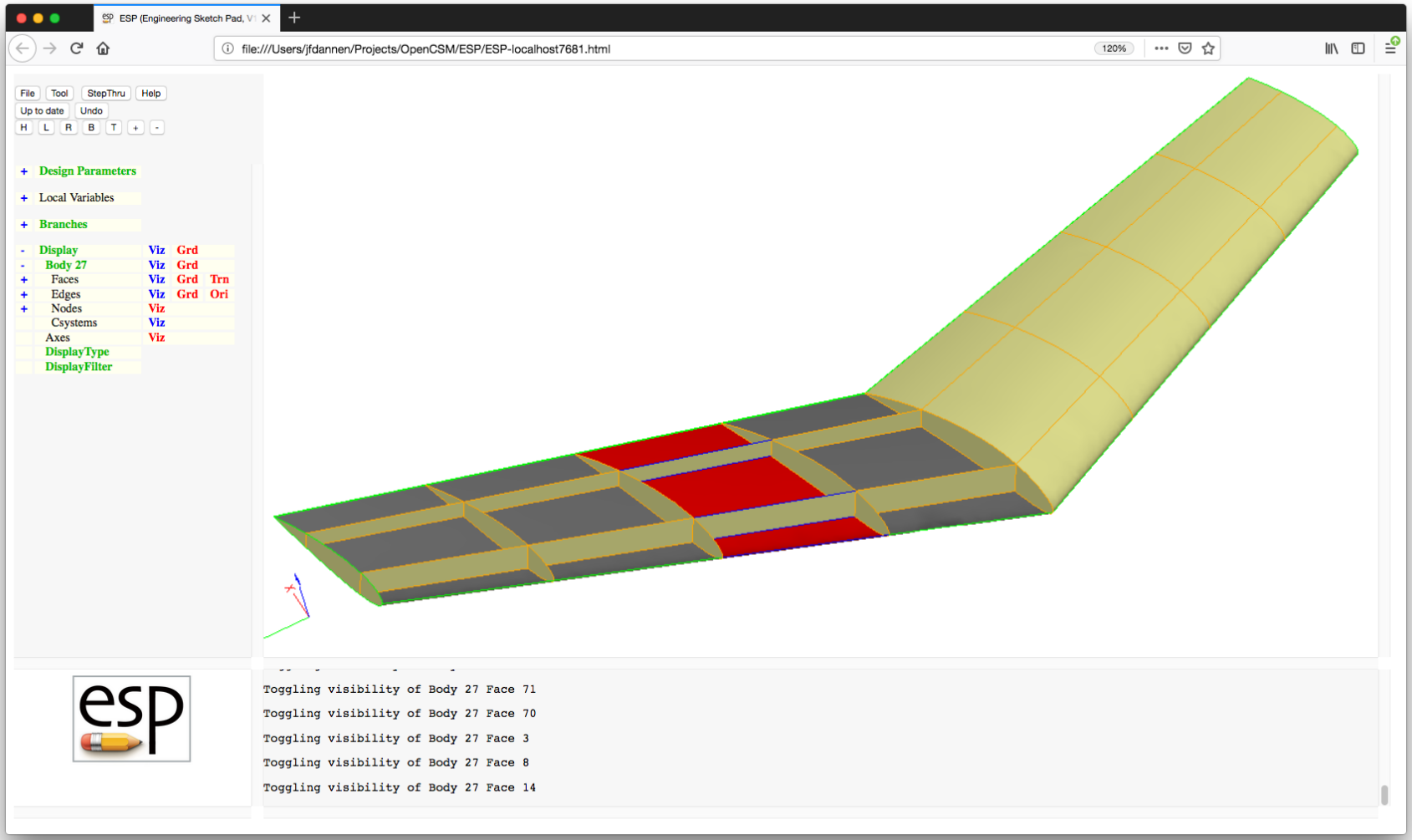
Wing with structure

Structure is not shown



Wing with structure

Structure is shown for right wing



Attribution Scheme (1)

- Right wing upper skin panels (Faces)
  - `tagComp=riteWing`
  - `tagType=upper`
- Right wing lower skin panels (Faces)
  - `tagComp=riteWing`
  - `tagType=lower`
- Right wing leading edge (Edge)
  - `tagComp=riteWing`
  - `tagType=leadingEdge`
- Right wing trailing edge panels (Faces)
  - `tagComp=riteWing`
  - `tagType=trailingEdge`
- Right wing tip panels (Faces)
  - `tagComp=riteWing`
  - `tagType=tip`

Attribution Scheme (2)

- Right wing spars (Faces)
  - `tagComp=riteWing`
  - `tagType=spar`
  - `tagIndex=1` for forward spar or `tagIndex=2` for rearward spar
- Right wing ribs (Faces)
  - `tagComp=riteWing`
  - `tagType=rib`
  - `tagIndex=1` for inboard rib, . . . , `tagIndex=3` for outboard rib
- Left wing is attributed similarly to right wing (Faces & Edges)
- Ribs at the wing root (Faces)
  - `tagComp=rootWing`
  - `tagType=rib`
  - `tagIndex=0`

wingStruct.csm (1)

```
# Design Parameters for OML
DESPMTR   wing:area       10.0      # wing area
DESPMTR   wing:aspect     6.00      # aspect ratio
DESPMTR   wing:taper      0.60      # taper ratio
DESPMTR   wing:sweep      20.0      # deg (of leading edge)
DESPMTR   wing:thickr     0.12      # thickness ratio at root
DESPMTR   wing:camberr    0.06      # camber    ratio at root
DESPMTR   wing:thickt     0.16      # thickness ratio at tip
DESPMTR   wing:cambert    0.02      # camber    ratio at tip
DESPMTR   wing:alphat    -5.00      # setting angle   at tip
DESPMTR   wing:dihedral   4.00      # deg
DESPMTR   wing:xroot      0.00      # xloc at root LE
DESPMTR   wing:yroot      0.00      # yloc at root LE
DESPMTR   wing:zroot      0.00      # zloc at root LE

CFGPMTR   SHARP_TE        0         # make the trailing edge blunt
```

wingStruct.csm (2)

```
# Design Parameters for structure
DESPMTR   wing:spar1     0.20      # location of fwrd spar
DESPMTR   wing:spar2     0.70      # location of rwrd spar
CFGPMTR   wing:nrib      3.00      # number of ribs per wing

# wing local variables
SET       wing:span      sqrt(wing:aspect*wing:area)
SET       wing:chordr    2*wing:area/wing:span/(1+wing:taper)
SET       wing:chordt    wing:chordr*wing:taper
SET       wing:ytip     -wing:span/2
SET       wing:xtip     -wing:ytip*tand(wing:sweep)
SET       wing:ztip     -wing:ytip*tand(wing:dihedral)
SET       wing:mac       sqrt(wing:area/wing:aspect)
```

wingStruct.csm (3)

```
# make wing OML
# lay out left wing
MARK
    # root
    UDPRIM    naca      thickness  wing:thickr   camber  wing:camberr\
                        sharpte  SHARP_TE
    SCALE     wing:chordr
    ROTATEX   90  0  0

    # left tip
    UDPRIM    naca      thickness  wing:thickt   camber  wing:cambert\
                        sharpte  SHARP_TE
    SCALE     wing:chordt
    ROTATEX   90  0  0
    ROTATEY   wing:alphat  0           0
    TRANSLATE wing:xtip    wing:ytip   wing:ztip
RULE
    ATTRIBUTE tagComp  $leftWing
SET       ruledBody @nbody
```

wingStruct.csm (4)

```
SELECT    FACE ruledBody  1
    ATTRIBUTE tagType  $root
SELECT    FACE ruledBody  2
    ATTRIBUTE tagType  $tip
SELECT    FACE ruledBody  3
    ATTRIBUTE tagType  $upper
SELECT    FACE ruledBody  4
    ATTRIBUTE tagType  $lower
SELECT    EDGE  ruledBody 3 ruledBody 4 1
    ATTRIBUTE tagComp  $leftWing
    ATTRIBUTE tagType  $leadingEdge
IFTHEN    SHARP_TE EQ 0
    SELECT    FACE  ruledBody 5
        ATTRIBUTE tagType  $trailingEdge
ELSE
    SELECT    EDGE  ruledBody 3 ruledBody 4 2
        ATTRIBUTE tagComp  $leftWing
        ATTRIBUTE tagType  $trailingEdge
ENDIF
```

wingStruct.csm (5)

```
# right wing too
STORE     LeftWing 0 1
RESTORE   LeftWing
    ATTRIBUTE tagComp $riteWing
    SELECT    EDGE  $tagType $leadingEdge
    IFTHEN    @iedge GT 0
        SELECT EDGE  $tagType $leadingEdge
            ATTRIBUTE tagComp $riteWing
    ENDIF
    SELECT    EDGE  $tagType $trailingEdge
    IFTHEN    @iedge GT 0
        SELECT EDGE  $tagType $trailingEdge
            ATTRIBUTE tagComp $riteWing
    ENDIF
    CATBEG   $edge_not_found
    CATEND
MIRROR    0   1  0
JOIN

SELECT    EDGE  ruledBody 3 ruledBody 3 1
    ATTRIBUTE tagType  $root
SELECT    EDGE  ruledBody 4 ruledBody 4 1
    ATTRIBUTE tagType  $root
STORE     WingOml
```

wingStruct.csm (6)

```
# make wing waffle
RESTORE   WingOml
SET       xmin       @xmin-0.1
SET       xmax       @xmax+0.1
SET       ymin       0
SET       ymax       @ymax+0.1
SET       zmin       @zmin-0.1
SET       zmax       @zmax+0.1
STORE     .

UDPARG    waffle     depth wing:nrib     # ensures rebuild
UDPARG    waffle     depth wing:spar1
UDPARG    waffle     depth wing:spar2
UDPRIM    waffle     depth zmax-zmin  filename <<
```

wingStruct.csm (7)

```
# construction lines for spars
CPOINT A   AT          0+wing:spar1*wing:chordr 0
CPOINT B   AT  wing:xtip+wing:spar1*wing:chordt -wing:ytip
CPOINT C   AT          0+wing:spar2*wing:chordr 0
CPOINT D   AT  wing:xtip+wing:spar2*wing:chordt -wing:ytip

CLINE  AB      A  B
CLINE  CD      C  D

# rite spars
POINT  E   ON  AB   YLOC  ymin
POINT  F   ON  AB   YLOC  ymax
LINE   EF  E   F  tagComp=riteWing  tagType=spar  tagIndex=1

POINT  G   ON  CD   YLOC  ymin
POINT  H   ON  CD   YLOC  ymax
LINE   GH  G   H  tagComp=riteWing  tagType=spar tagIndex=2
```

wingStruct.csm (8)

```
# rite ribs
PATBEG  irib  wing:nrib
    CPOINT  I  AT  xmin  -wing:ytip*irib/(wing:nrib+1)
    CPOINT  J  AT  xmax  y@I
    LINE    .   I  J  tagComp=riteWing  tagType=rib   tagIndex=!val2str(irib,0)
PATEND

# root rib
CPOINT  I  AT  xmin  0
CPOINT  J  AT  xmax  y@I
LINE    .   I  J  tagComp=rootWing  tagType=rib   tagIndex=0

# left spars
POINT  E   AT  x@E  -y@E
POINT  F   AT  x@F  -y@F
LINE   EF  E   F  tagComp=leftWing  tagType=spar  tagIndex=1

POINT  G   AT  x@G  -y@G
POINT  H   AT  x@H  -y@H
LINE   GH  G   H  tagComp=leftWing  tagType=spar tagIndex=2
```

wingStruct.csm (9)

```
    # left ribs
    PATBEG  irib  wing:nrib
        CPOINT  I  AT  xmin  wing:ytip*irib/(wing:nrib+1)
        CPOINT  J  AT  xmax  y@I
        LINE    .   I  J  tagComp=leftWing  tagType=rib   tagIndex=!val2str(irib,0)
    PATEND
>>
TRANSLATE  0  0  zmin
STORE     WingWaffle
```

wingStruct.csm (10)

```
# trim the waffle to be the ribs and spars
RESTORE   WingOml
RESTORE   WingWaffle
INTERSECT

# score the wing skin with the waffle
RESTORE   WingOml
RESTORE   WingWaffle
SUBTRACT
EXTRACT   0

# combine the two
UNION
```

Attributing: LoadPoint=leftTip

- Put the Attribute `LoadPoint=leftTip` on the Node that is at the intersection of the forward spar, wing tip, and upper skin on the left wing

```
UDPRIM    editAttr  filename <<
    NODE  ADJ2FACE  tagComp=leftWing  tagType=spar  tagIndex=1
    AND   ADJ2FACE  tagComp=leftWing  tagType=upper
    AND   ADJ2FACE  tagComp=leftWing  tagType=tip
    SET             LoadPoint=leftTip
>>
```

Attributing: Right skin panels red

- For the upper and lower skin panels on the rite wing that are between the first and second rib, make their color red and their grid white

```
UDPRIM    editAttr  filename <<
    FACE  HAS       tagComp=riteWing  tagType=upper
    AND   ADJ2FACE  tagType=rib  tagIndex=1
    AND   ADJ2FACE  tagType=rib  tagIndex=2
    SET             _color=red
    SET             _bcolor=red
    SET             _gcolor=white

    FACE  HAS       tagComp=riteWing  tagType=lower
    AND   ADJ2FACE  tagType=rib  tagIndex=1
    AND   ADJ2FACE  tagType=rib  tagIndex=2
    SET             _color=red
    SET             _bcolor=red
    SET             _gcolor=white
>>
```

Attributing: Blue Edges around red panels

- Make the Edges blue that are between two red panels

```
UDPRIM    editAttr  filename <<
    EDGE  ADJ2FACE  _color=red
    AND   ADJ2FACE  tagType=spar
    SET             _color=blue

    EDGE  HAS       tagType=leadingEdge
    AND   ADJ2FACE  _color=red
    SET             _color=blue
>>
```