Engineering Sketch Pad (ESP)

User Training Session 13
Associated Tools

John F. Dannenhoffer, III
john@geocentrictech.com
Geocentric Technologies LLC

updated for v1.28

Overview

- Other `.csm` statements
- CAPS (described earlier)
- ErepEd
- Plugs
- Pyscript (described earlier)
- VspSetup
- Homework exercise

Other `.csm` Statements (1)

- `APPLYCSYS $csysName ibody=0`
  - transforms Group on top of stack so that their origins/orientations coincide with given csys
- `ASSERT arg1 arg2 toler=0 verify=0`
  - return error if arg1 and arg2 differ
- `CHAMFER radius edgeList=0`
  - apply a chamfer to a Body
- `FILLET radius edgeList=0 listStyle=0`
  - apply a fillet to a Body

Other `.csm` Statements (2)

- `HOLLOW thick=0 entList=0 listStyle=0`
  - hollow out a SolidBody or SheetBody
- `MESSAGE $text $schar=_ $fileName=.  $openType=a`
  - generate a message to be displayed
- `PROJECT x y z dx dy dz useEdges=0`
  - find the first projection from given point in given direction
- `SOLBEG / SOLCON / SOLEND`
  - solve simultaneous non-linear equations

`EVALUATE` Statement (1)

- In general, the `EVALUATE` command evaluates a Node, Edge, Face, or distance and puts its results into `@edata`
- See Help for a complete dsscription

- `EVALUATE NODE ibody inode`
  - return coordinates

EVALUATE Statement (2)

- `EVALUATE EDGE ...`
  - return (clipped) t, coordinates, and derivatives
- `EVALUATE EDGEBBOX ibody iedge`
  - return bounding box
- `EVALUATE EDGERNG ibody iedge`
  - return $t_{\texttt{min}}$ and $t_{\texttt{max}}$
- `EVALUATE EDGEINV ibody iedge x y z`
  - find closest point to (x, y, z)
- `EVALUATE EDGEKT ibody iedge`
  - return B-spline knot vector
- `EVALUATE EDGECP ibody iedge`
  - return B-spline control points
- `EVALUATE EDGETESS ibody iedge`
  - return number of tessellation points

EVALUATE Statement (3)

- `EVALUATE FACE iface u v`
  - return (clipped) (u, v), coordinates, and derivatives
- `EVALUATE FACEBBOX ibody iface`
  - return bounding box
- `EVALUATE FACERNG ibody iface`
  - return $u_{\texttt{min}}$, $u_{\texttt{max}}$, $v_{\texttt{min}}$, and $v_{\texttt{max}}$
- `EVALUATE FACEINV ibody iface x y z`
  - find closest point to (x, y, z)
- `EVALUATE FACEUKT ibody iface`
  - return B-spline U-knot vector
- `EVALUATE FACEVKT ibody iface`
  - return B-spline V-knot vector
- `EVALUATE FACECP ibody iface`
  - return B-spline control points
- `EVALUATE FACETESS ibody iface`
  - return number of tessellation points and triangles

EVALUATE Statement (4)

- `EVALUATE DIST +ibody1 +ibody2`
  - find the minimum distance between `ibody1` and `ibody2` and return the distance and closest points
  - a negative distance indicates that the Bodys intersect
- `EVALUATE DIST -ibody1 +ibody2`
  - find the minimum clearance between enclosing `ibody1` and enclosed `ibody2` and return the clearance and closest points
  - a negative distance indicates that `ibody2` is not fully enclosed in `ibody1`

GETATTR Statement

- `GETATTR $pmtrName attrID global=0`
  - If `global` is zero, this command applies to the selected Body, otherwise it refers to global Attributes
  - If `attrID` is the string "`_nattr`", set `$pmtrName` to the number Attributes
  - If `attrID` is an integer, the `attrID`th Attribute value (bias-1) is stored in `$pmtrName`
  - Otherwise the value of the Attribute whose name is `attrID` is stored in `$pmtrName`

Erep Editor (1)

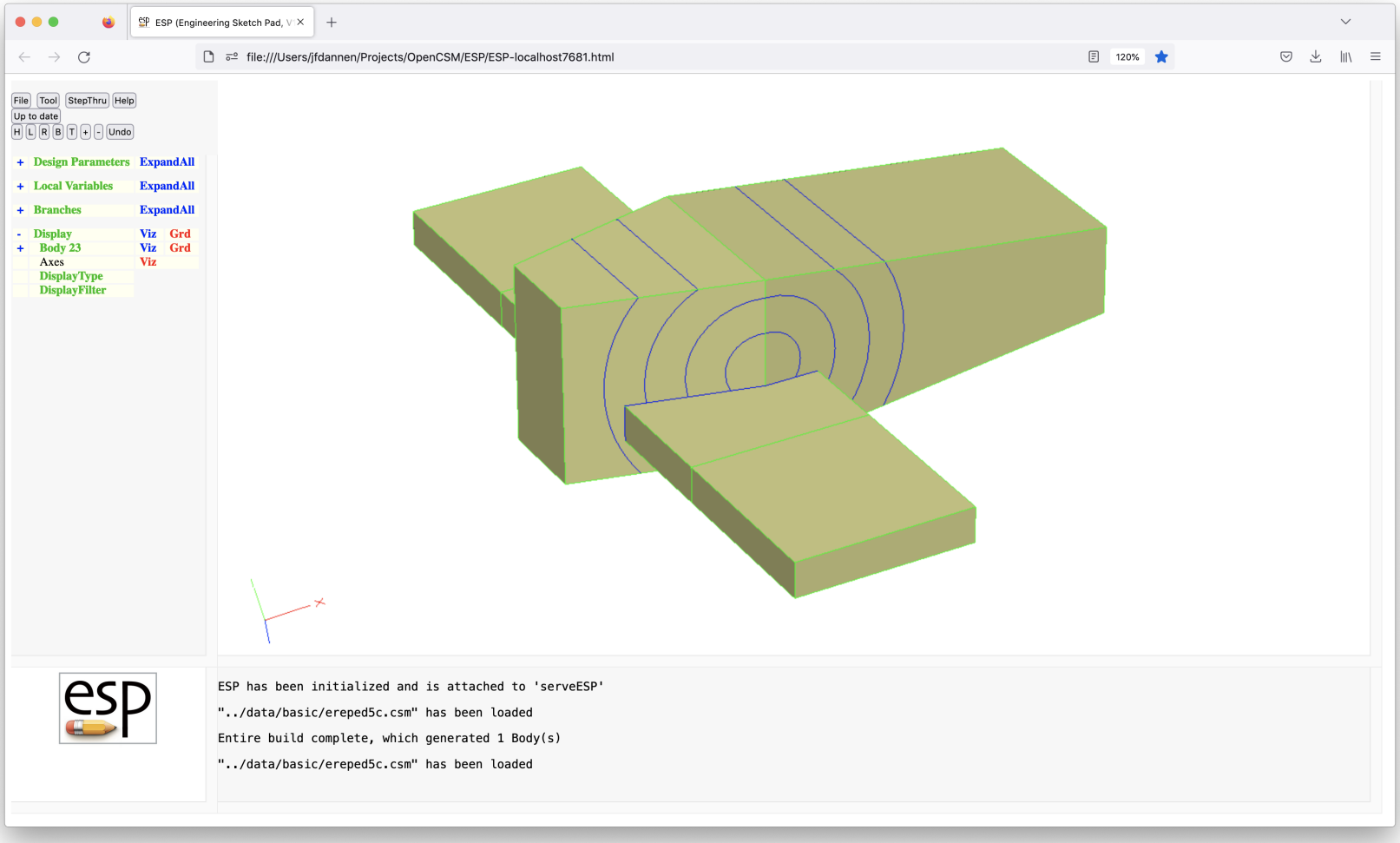
- An Erep is an effective topology the may be used in a down-stream analysis (such as `CAPS`)
- Ereps share geometry with a supporting Brep (which is comprised of Nodes, Edges, and Faces)
- ENodes in an Erep are a subset of the Brep's Nodes
- EEdges in an Erep are combinations of the Brep's Edges
  - the EEdges are the concatenation of one or more contiguous Edges
  - note: not all Edges are associated with an EEdge (that is, some Edges may be interior to an EFace)
- EFaces are combinations of one or more of the Brep's Faces
  - note: every Face is associated with an EFace

Erep Editor (2)

- Ereps are specified via attributes on the Brep
- Nodes that have a `.Keep` Attribute are forced to be ENodes
- Edges that have a `.Keep` Attribute are forced to be part of an EEdge
- Attributes on the Faces define which ones are to be combined into an EFace
  - The Attribute name is user-selected (for example `_erep`)
  - All Brep Faces that have a `_erep` Attribute with the same integer value are candidates to be combined into an EFace
    - the Faces must be contiguous
    - the dihedral angle between adjacent Faces must not exceed a user-specified tolerance (typically 5 degrees)
  - The Erep Editor uses "colors" to specify the Attribute values

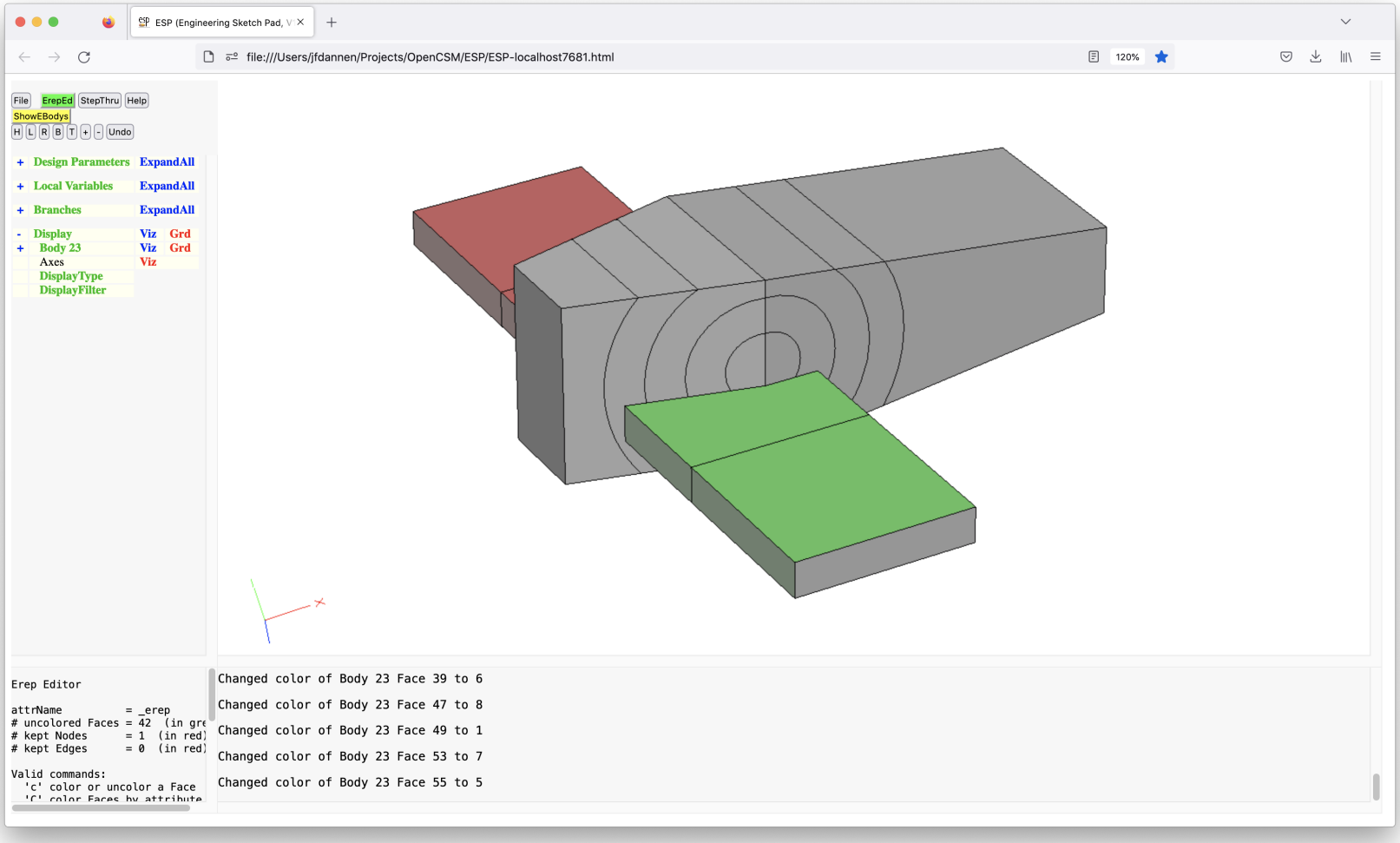
Erep Editor (3)

Original Brep configuration



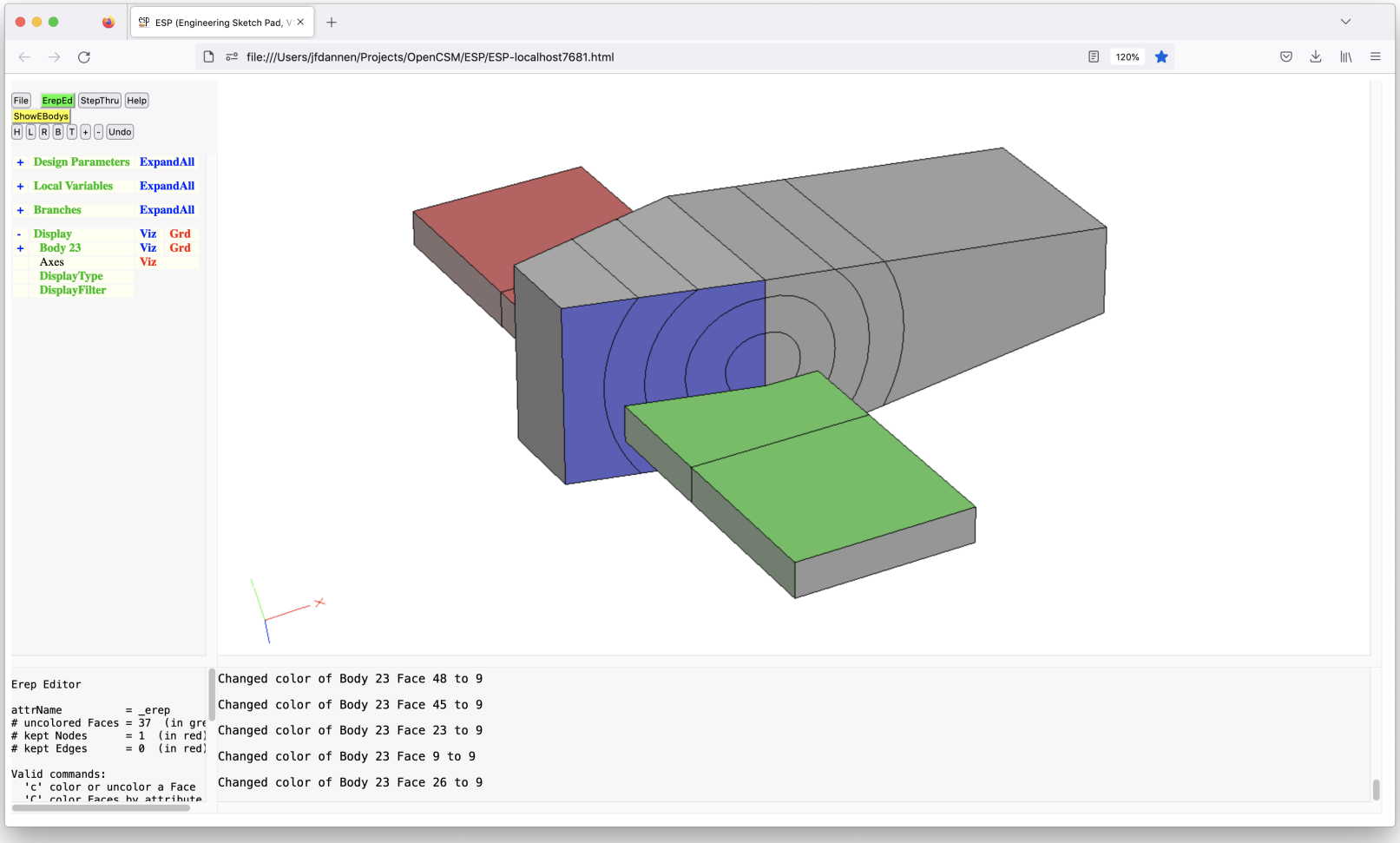
Erep Editor (4)

Wing Faces colored into 8 groups (using the "C" option)



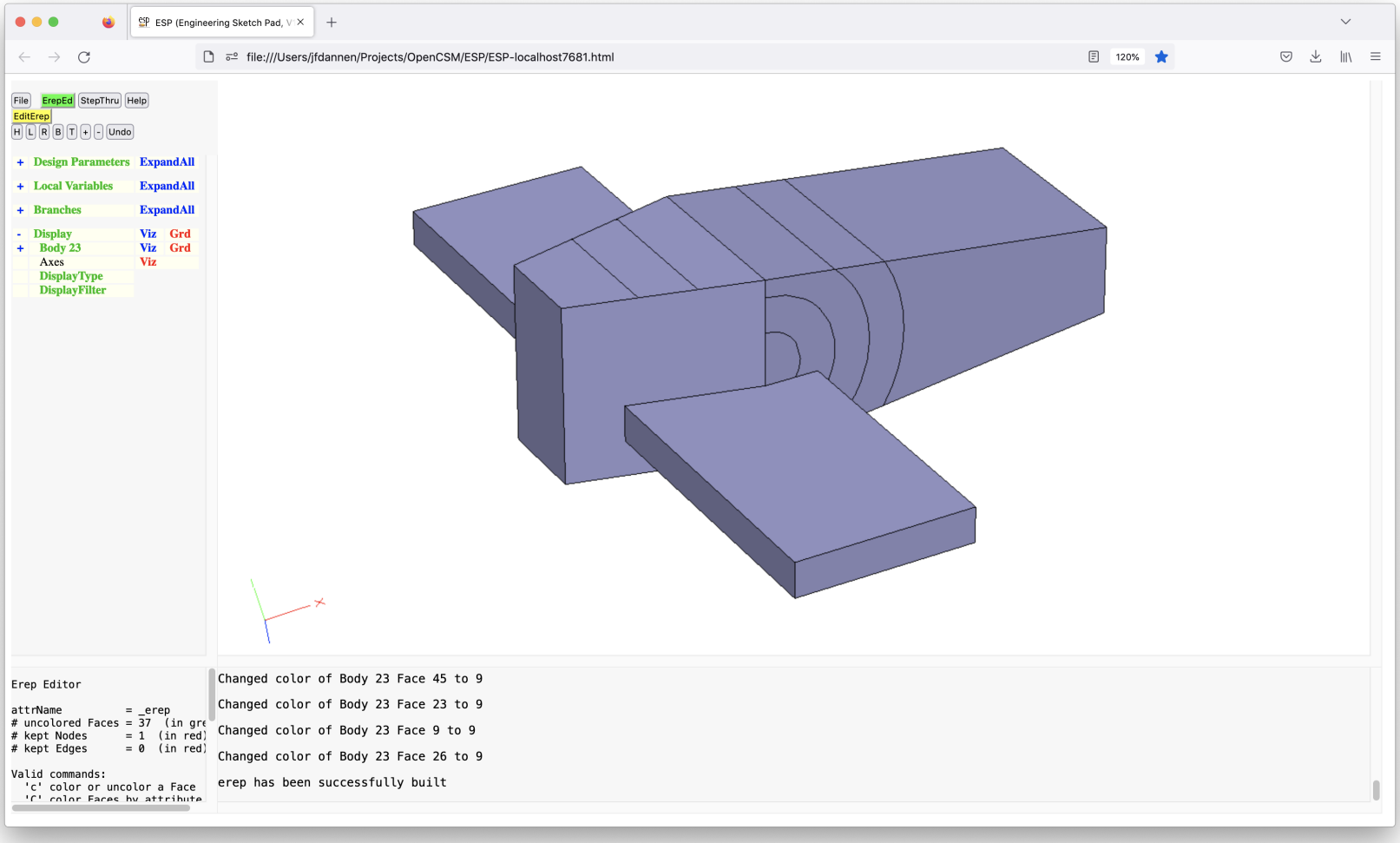
Erep Editor (5)

Fuselage side colored into one group (using the “c” option five times)



Erep Editor (6)

Associated Erep



Plugs (1)

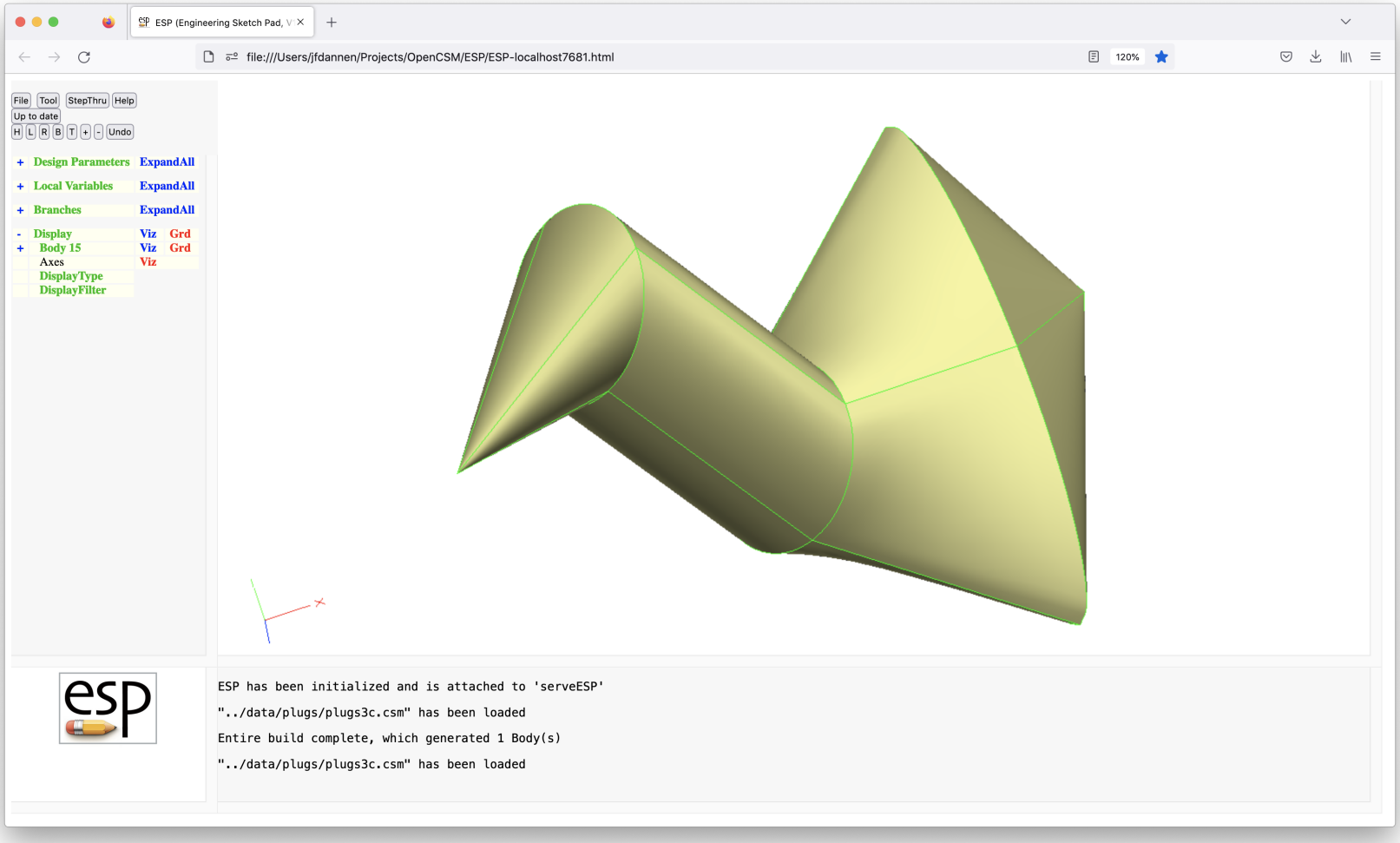
- `Plugs` is a tool for finding the Design Parameter values such that a Body matches a cloud of points
- Inputs:
  - a single parameterized Body
  - a cloud of (unclassified) points
- Outputs:
  - Design Parameter values
- Executed via **Tool** $\rightarrow$ **Plugs**
- Algorithm is described in Jia, P, and Dannenhoffer, J.F., "Generation of Parametric Aircraft Models from a Cloud of Points", AIAA-2016-1926, presented at AIAA SciTech 2016, January 2016.

Plugs (2)

- Phase 1
  - Take up to 50 Levenberg-Marquardt optimization steps to vary the Design Parameters to match the bounding box of the point cloud
  - Unclassify all points in the cloud
- Phase 2
  - In a sequence of passes
    - classify each point in the cloud by determining the most likely Face to associated it with; if there is no obvious Face, leave the point unclassified
    - if all points are classified and the point classifications are the same as in the previous pass, exit
    - perform up to 50 steps of the Levenberg-Marquardt optimizer

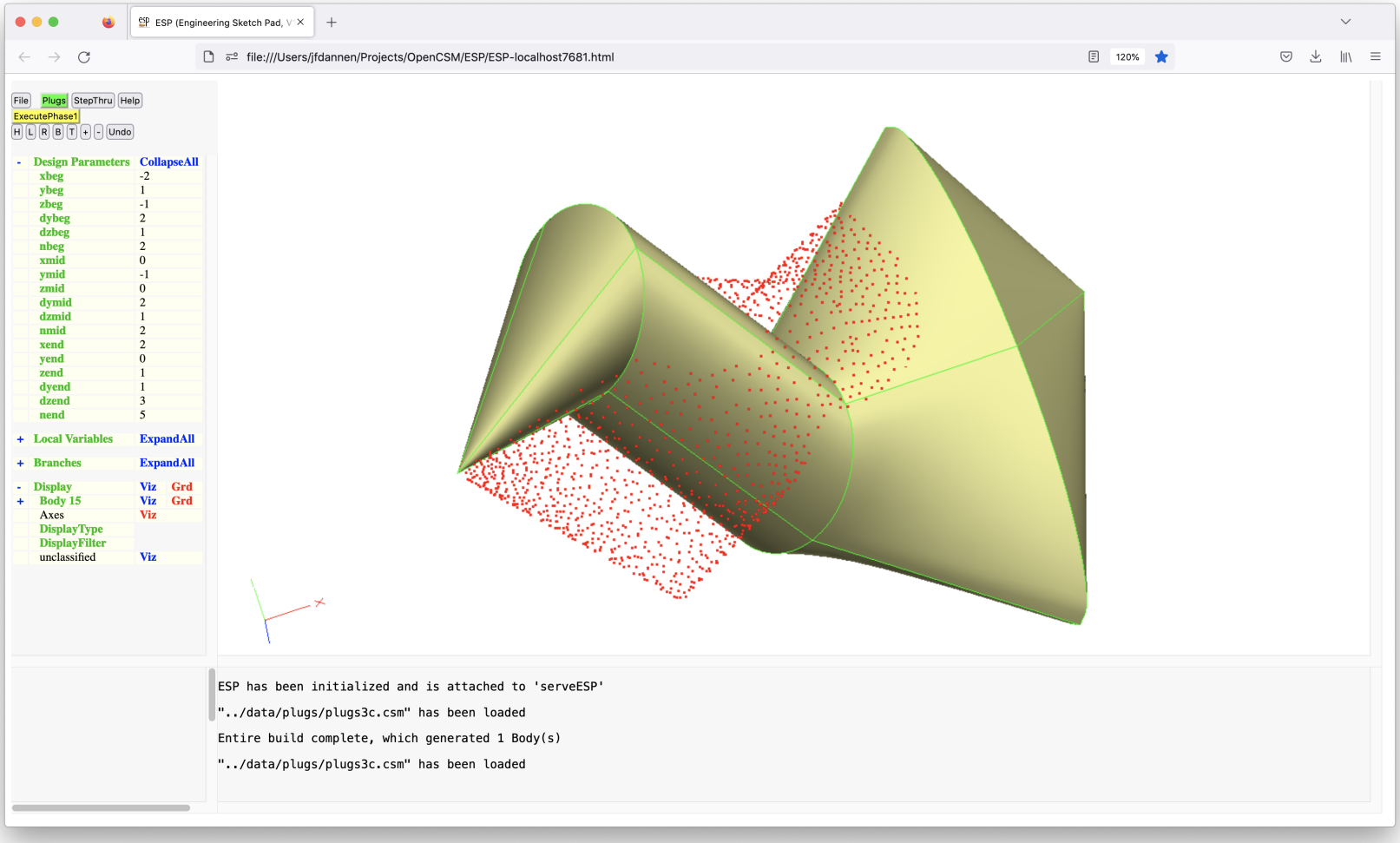
Plugs (3)

Initial Body with user-guessed Design Parameters



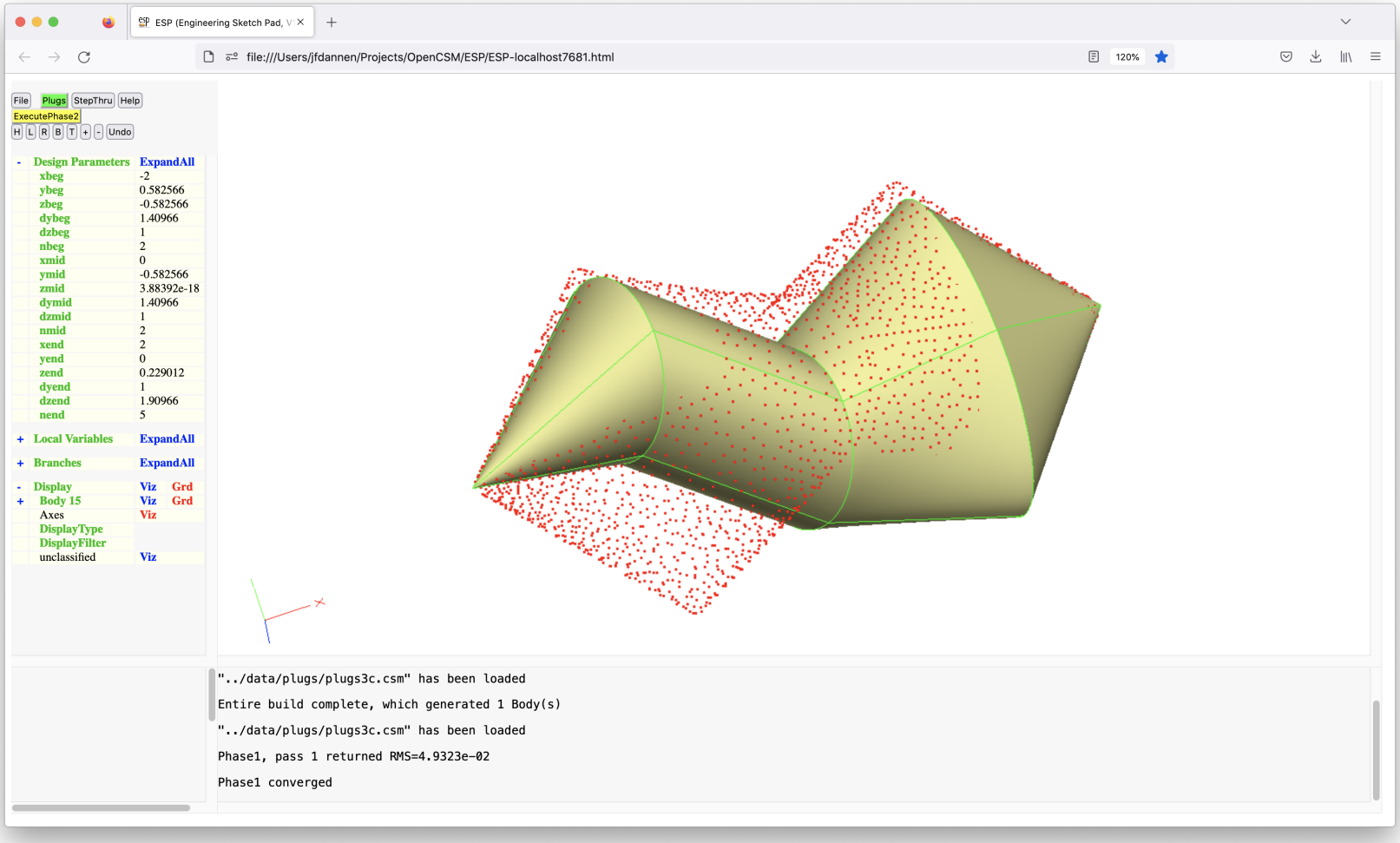
Plugs (4)

Initial Body with cloud of points (red points are unclassified)



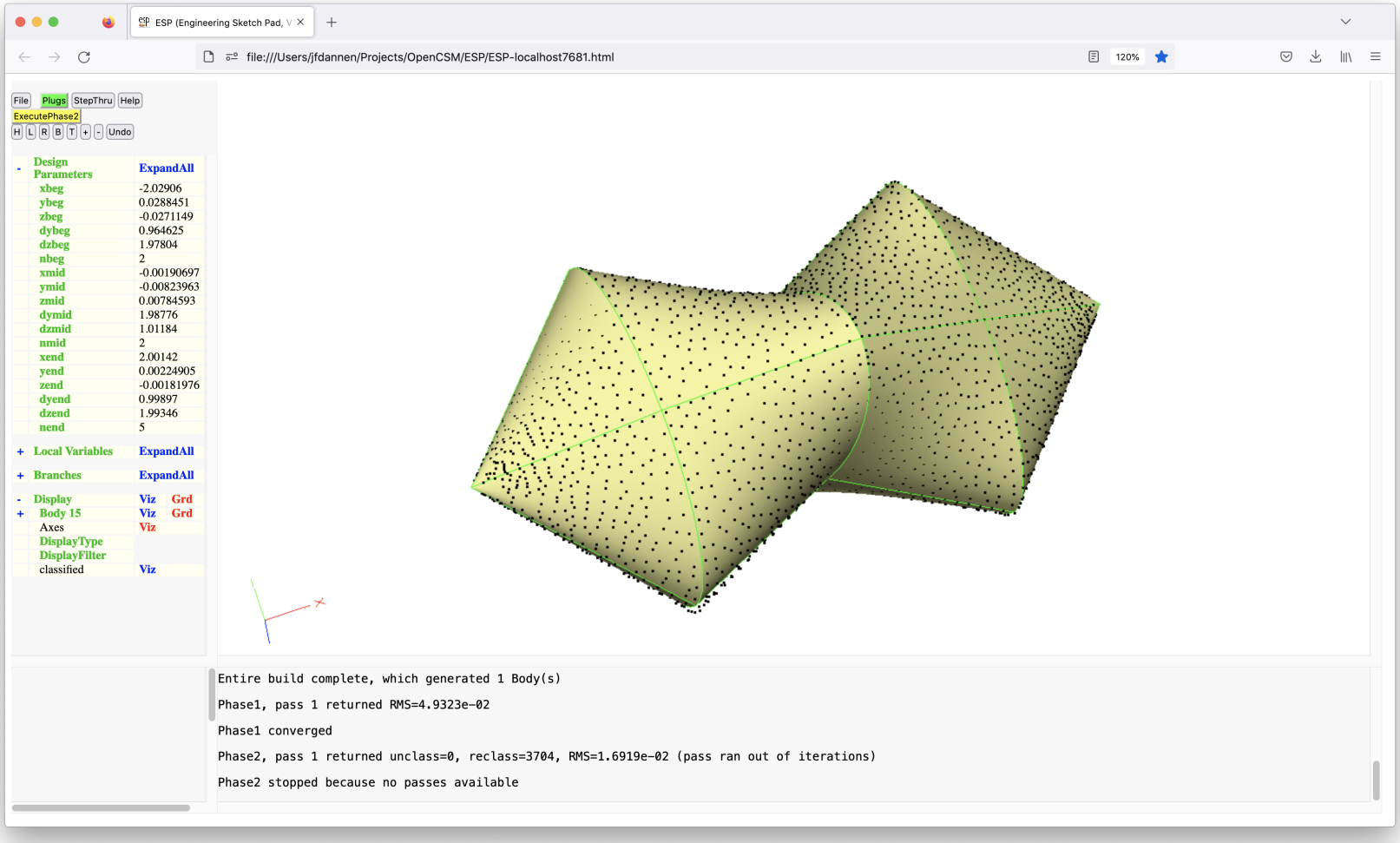
Plugs (5)

After Phase 1 — bounding boxes of Body and point cloud match



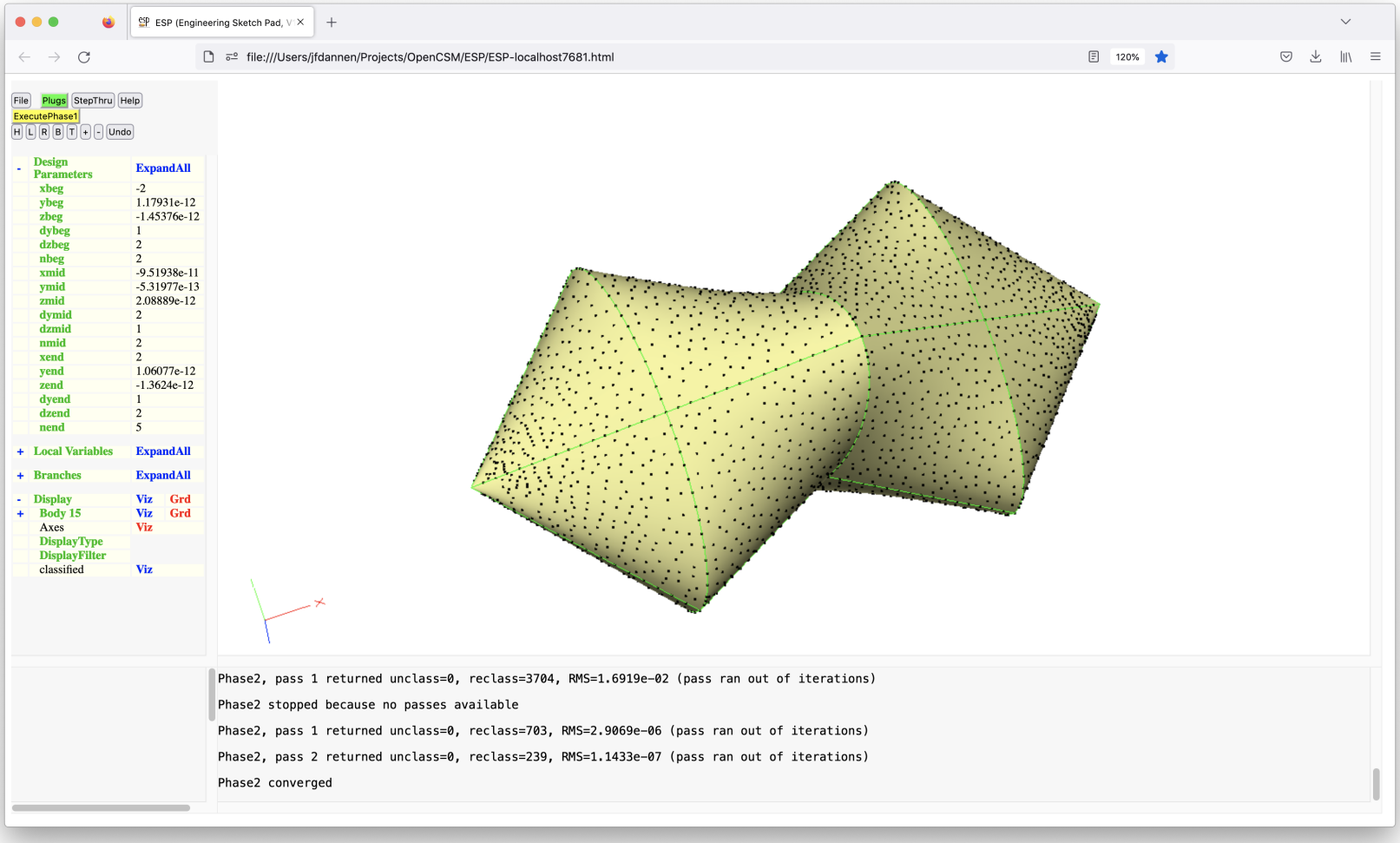
Plugs (6)

After Pass 1 of Phase 2 (black points are classified)



Plugs (7)

After Pass 4 of Phase 2 (converged)



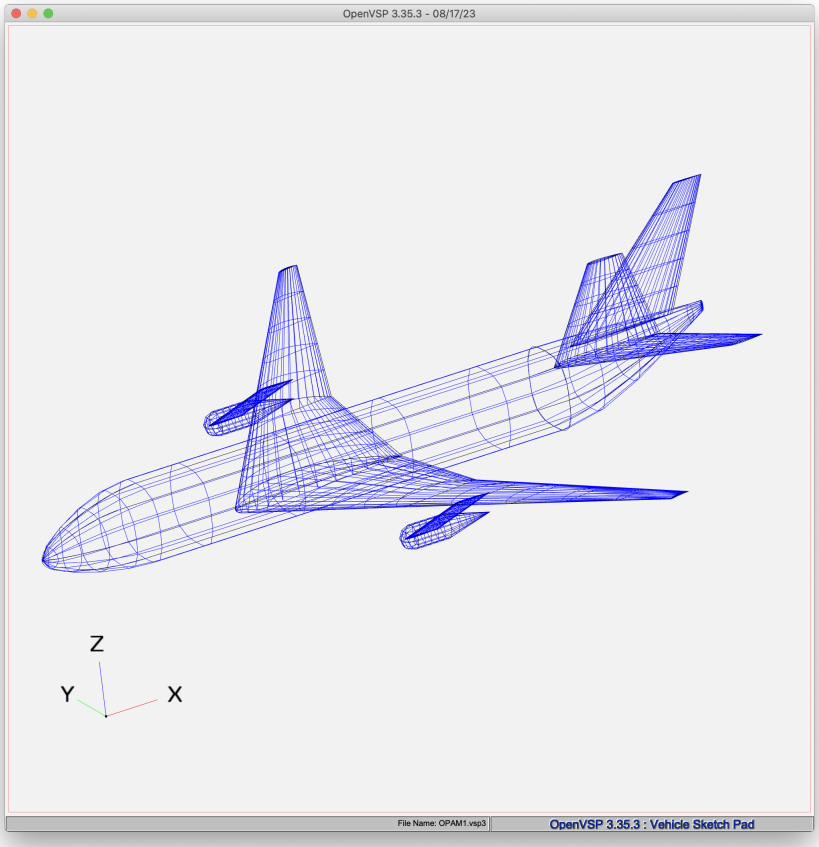
VspSetup (1)

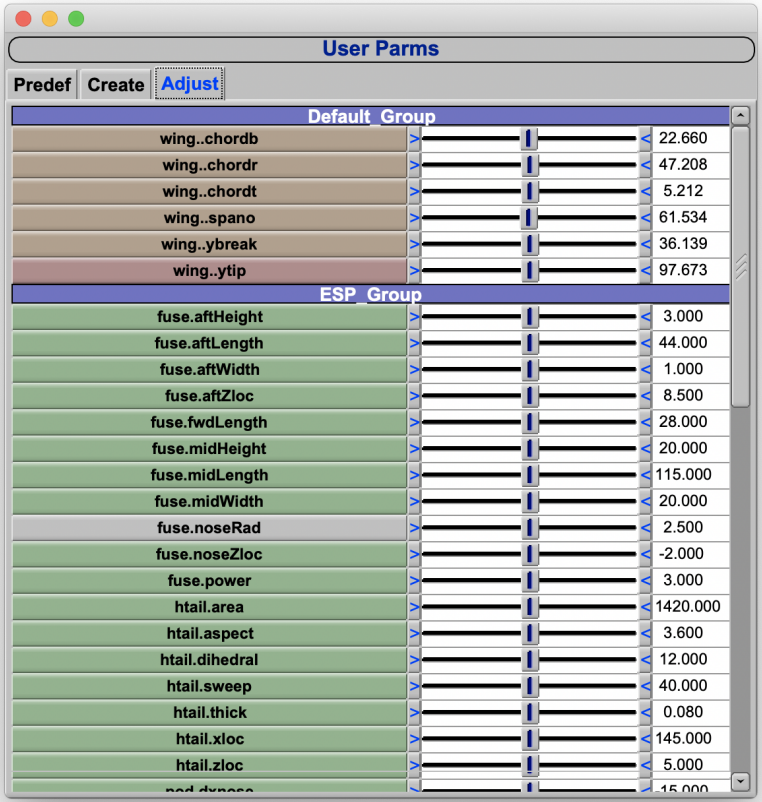
- The `vsp3` UDP can be used to import an `OpenVSP` model into `ESP`
- Each component in the `OpenVSP` model comes into `ESP` as a separate SolidBody
- The SolidBodys are static (i.e., they have fixed geometry)
- In order to drive an `OpenVSP` model parametrically, one needs to use the `VspSetup` tool
  - creates a `.udc` that defines the **UserParms** in `OpenVSP`
  - calls the `vsp3` UDP
- In order to use the `vsp3` UDC, either `vspscript` must be in the user's path or the user must set the `VSP3_ROOT` environment variable to the directory containing `vspscript`

VspSetup (2)

- The `.vsp3` file is generated by `OpenVSP` and must exist before the process starts
- `OpenVSP` model should have "User Parms"
  - in the "ESP_Group"
  - names include periods for grouping

VspSetup (3)


OpenVSP 3.35.3 - 08/17/23
Z
Y
X
File Name: OPAM1.vsp3
OpenVSP 3.35.3 : Vehicle Sketch Pad


User Parms
Predef
Create
Adjust
Default_Group
wing..chordb 22.660
wing..chordr 47.208
wing..chordt 5.212
wing..spano 61.534
wing..ybreak 36.139
wing..ytip 97.673
ESP_Group
fuse.aftHeight 3.000
fuse.aftLength 44.000
fuse.aftWidth 1.000
fuse.aftZloc 8.500
fuse.fwdLength 28.000
fuse.midHeight 20.000
fuse.midLength 115.000
fuse.midWidth 20.000
fuse.noseRad 2.500
fuse.noseZloc -2.000
fuse.power 3.000
htail.area 1420.000
htail.aspect 3.600
htail.dihedral 12.000
htail.sweep 40.000
htail.thick 0.080
htail.xloc 145.000
htail.zloc 5.000

VspSetup (4)

- Every time the user modifies the `.vsp3` file:
  - in `serveESP`, use **Tool⟶VspSetup**, to write a `.udc` file that contains:
    - `DESPMTR`, `LBOUND`, and `UBOUND` statements from the tagged variables in the `.vsp3` file
    - a `UDPRIM vsp3` statement to generate an updated configuration (via a call to `vspscript`)

VspSetup (5)

```
# generated by VspSetup from ../data/vsp3/OPAM1.vsp3

INTERFACE . ALL

   DESPMTR   fuse:aftHeight    3
   LBOUND    fuse:aftHeight    -100000
   UBOUND    fuse:aftHeight    100000

   <lines deleted>

   DESPMTR   wing:zroot    -5
   LBOUND    wing:zroot    -100000
   UBOUND    wing:zroot    100000

UDPRIM    vsp3    filename $$/OPAM1.vsp3

END
```

VspSetup (6)

- Write a `.csm` file that contains
  - a `UDPRIM /udcName` to include the `.udc` file

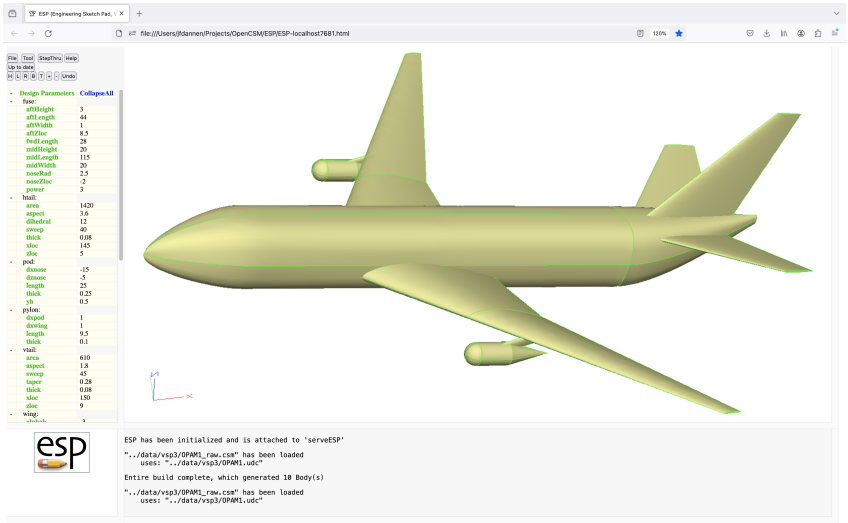
```
# import the geometry
UDPRIM $/OPAM1
```

  - any other `csm` statements, such as Boolean operations or specifying attributes

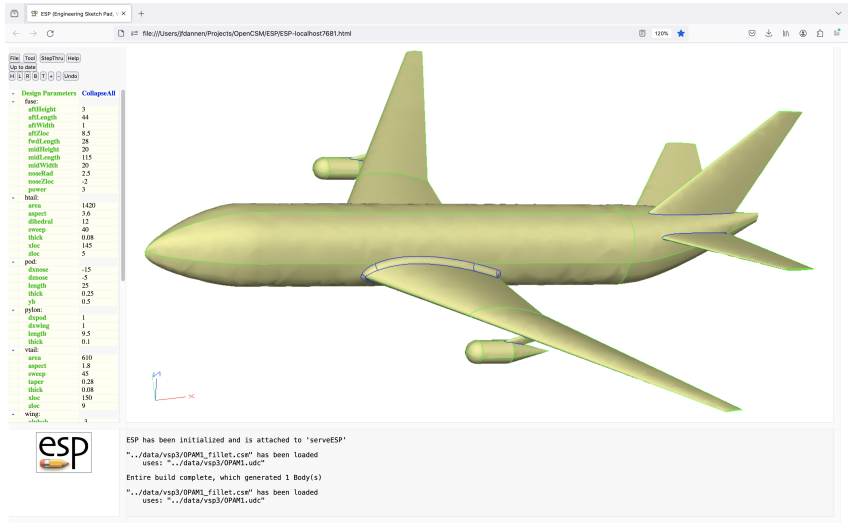
```
# make a single Body
UNION 1

# add FILLETs at the wing/fuselage junctions
DESPMTR filrad 1.5

SELECT EDGE   1 0 3 0 0
FILLET filrad @sellist 1

SELECT EDGE   2 0 3 0 0
FILLET filrad @sellist 1
```

- Changes to the `DESPMTR`s in `ESP` are reflected in changes to the geometry
- When the configuration is built in `serveESP`, geometric- and tessellation-sensitivities are available

VspSetup (7)

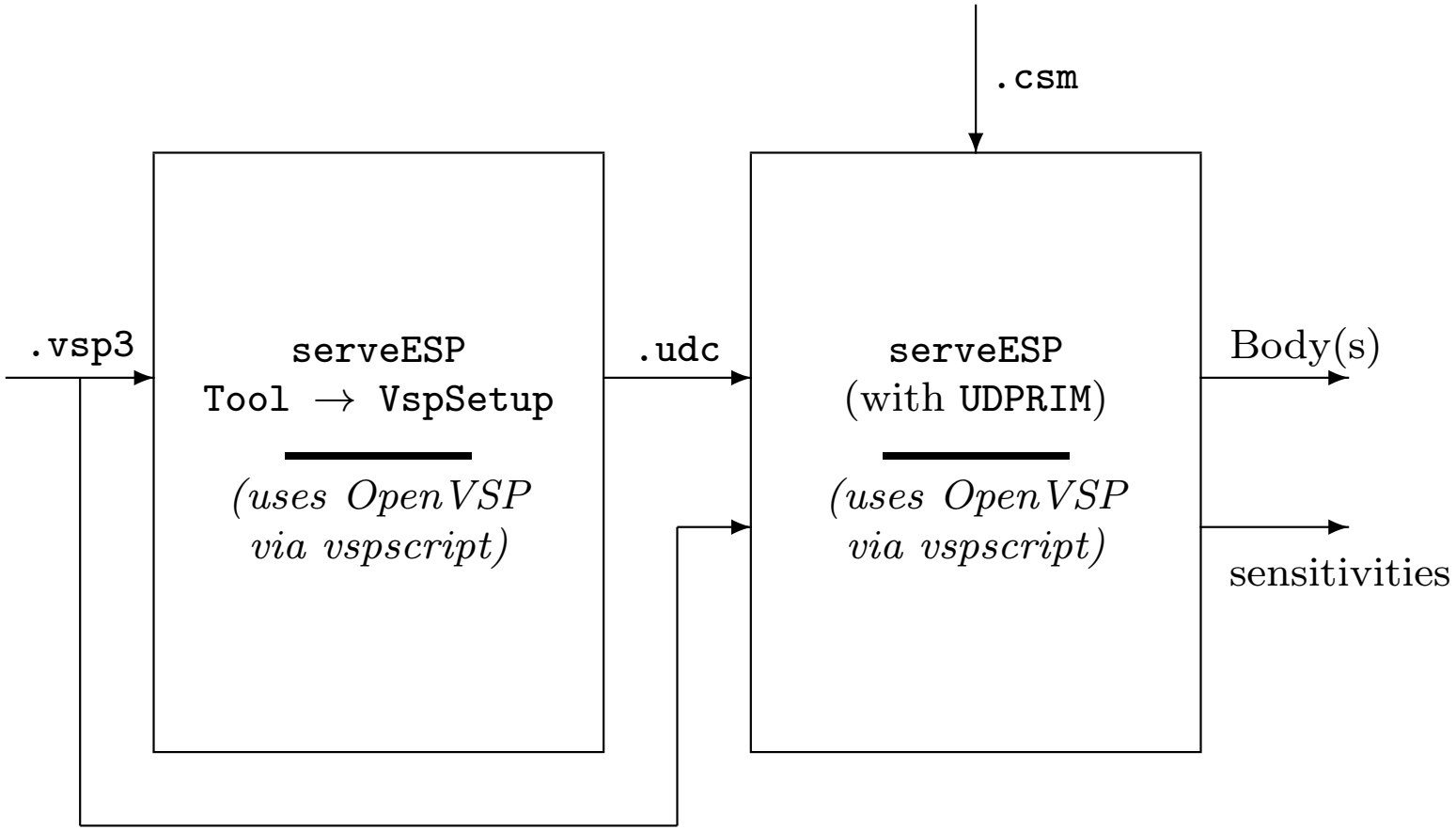


Bodys flying
in flying formation



Bodys UNIONed and
FILLETs added

VspSetup (8)


.csm
.vsp3
serveESP
Tool → VspSetup
(uses OpenVSP
via vspscript)
.udc
serveESP
(with UDPRIM)
(uses OpenVSP
via vspscript)
Body(s)
sensitivities

Homework Exercise

- Run `serveESP` on `frustrum.csm`
  - **DesignParameters→ExpandAll** to see the current values for the Design Parameters
  - Run **Tools→Plugs** on `frustrum.cloud` and see the cloud points to which we are trying to fit. You may need to make the Faces transparent to see all the points.
  - **ExecutePhase1** for 1 pass, which will make the bounding boxes the same
  - **ExecutePhase2** for (up to) 25 passes, which will adjust the Design Parameters. Watch the Design Parameters change in the TreeWindow
  - **Plugs→Save** to use these new Design Parameters
- Run `serveESP` on `frusrum.csm` with the command-line options `-despmtrs plugs.despmtrs` and `-plot frsutrum.cloud` to see the updated configuration and the original cloud points