

Using OpenVSP Models in the Engineering Sketch Pad

John F. Dannenhoffer, III
jfdannen@syr.edu
Aerospace Computational Methods Lab
Mechanical & Aerospace Engineering
Syracuse University

Advanced Modeling & Simulation (AMS) Seminar Series
NASA Ames Research Center
5 September 2024

Overview

- Background
  - Engineering Sketch Pad (ESP)
  - Vehicle Sketch Pad (OpenVSP)
- Static OpenVSP models in ESP
  - use case and user-facing process
  - behind the scenes
- Parametric OpenVSP models in ESP
  - use case and user-facing process
  - behind the scenes
  - sensitivities
- Summary

Engineering Sketch Pad (ESP)

- ESP **is:**
  - a geometry creation and manipulation system designed to **fully support** the analysis and design of geometrically-complex configurations
  - a stand-alone CAD-like system for the development of geometric models
  - *layer-cake* of well-crafted open-source APIs
    easily embedded into other software systems to support their geometry and process needs
- ESP **is not:**
  - a full-featured mechanical computer-aided design (**mCAD**) system
  - a system to be used for creating “drawings”
  - an MDO Framework

ESP Features — 1

- Solid modeler
  - construction process guarantees that models are realizable solids (SolidBodys)
  - supports SheetBodys and WireBodys where needed
- Feature based
  - uses standard, grown, and “user-defined” features
  - can be transformed and combined by boolean-like operations
- Parametric
  - driven by externally-accessible design and configuration parameters
  - parameters can be scalars, vectors, or matrices

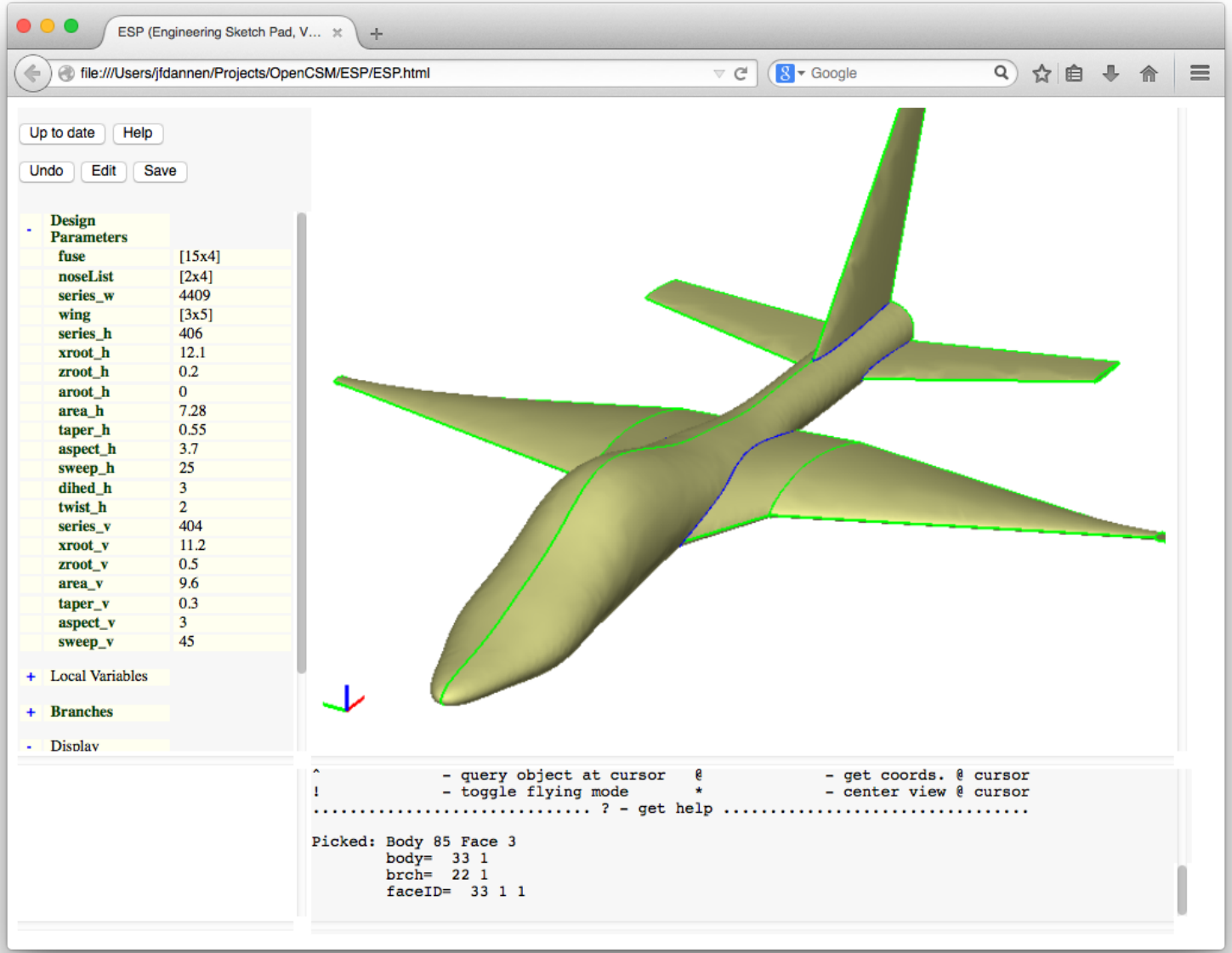
ESP Features — 2

- Differentiated
  - construction process is differentiated to provide sensitivity of any point in the model with respect to any (or multiple) design parameters
  - most operations are analytically differentiated, but finite differences are available as a fallback
- Associative
  - persistent attributes can be applied to any Body, Face, Edge, or Node at any point during the build process
  - attributes can be vector-valued or strings
  - attributes are the key to linking multi-disciplinary and multi-fidelity models

ESP Features — 3

- Extensible
  - user-defined primitives and functions (UDPs and UDFs) written in C or Fortran and linked dynamically at run time
  - user-defined components (UDCs) written as a macro in the `.csm` language
- Embeddable
  - all input/output files (such as `.csm` scripts) are simple ASCII files
  - `OpenCSM` can be called directly via its API
  - open-source license allows it to be deployed anywhere, including in an HPC environment
- Deployable
  - server back-end runs on LINUX, MAC-OS, and Windows
  - client front-end runs in all modern web browsers
- Linked to more than 20 analysis tools via `CAPS`
  - mesh generators, aerodynamic and structural analyses, optimizers

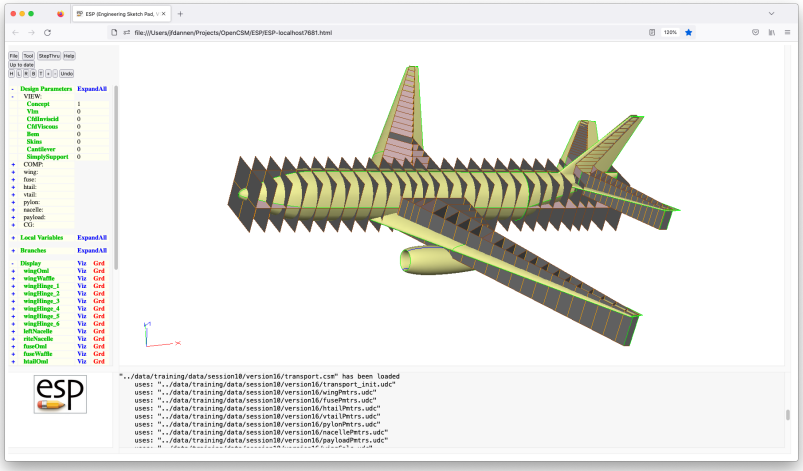
Screen Shot of ESP through serveESP



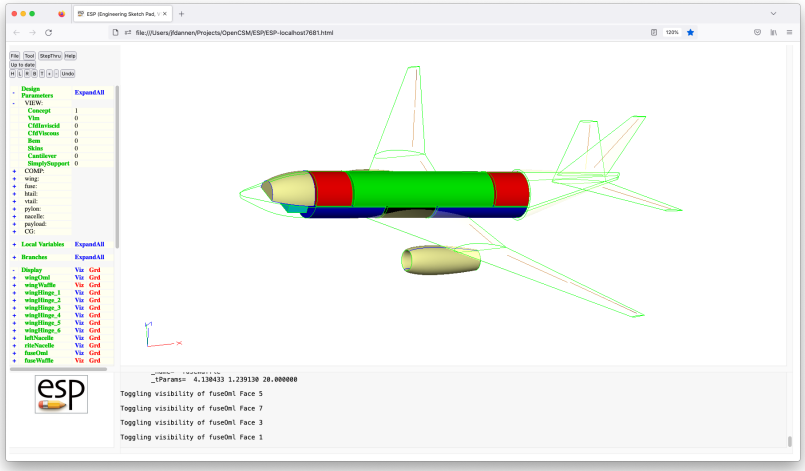
Available ESP Linkages Through CAPS

- Surface mesh generators
  - EGADS
  - AFLR4
  - AFLR2 (2d)
  - Delaundo (2d)
- Volume mesh generators
  - TetGen
  - AFLR3
  - Pointwise
- Aerodynamics
  - FRICTION
  - AWAVE
  - XFOIL
  - MSES
  - TSFOIL
  - AVL
  - CART3D
  - FUN3D
  - SU2
  - FlightStream
- Structures
  - MYSTRAN
  - NASTRAN
  - Astros
  - TACS
  - Masstran

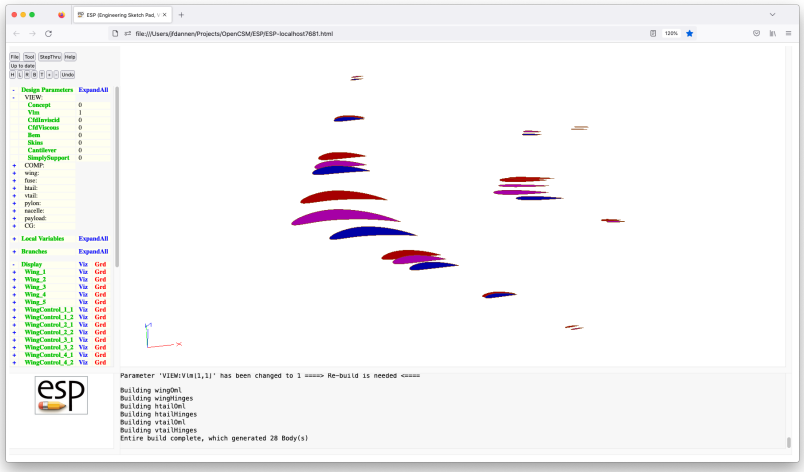
Multi-X Models in ESP — 1



concept1

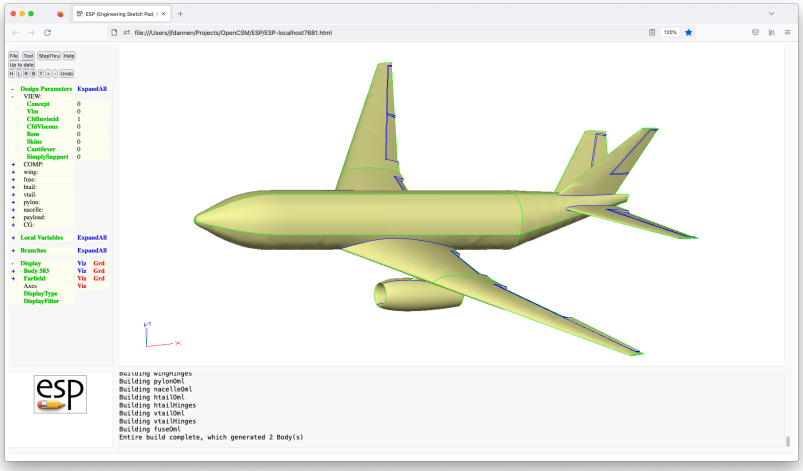


concept2

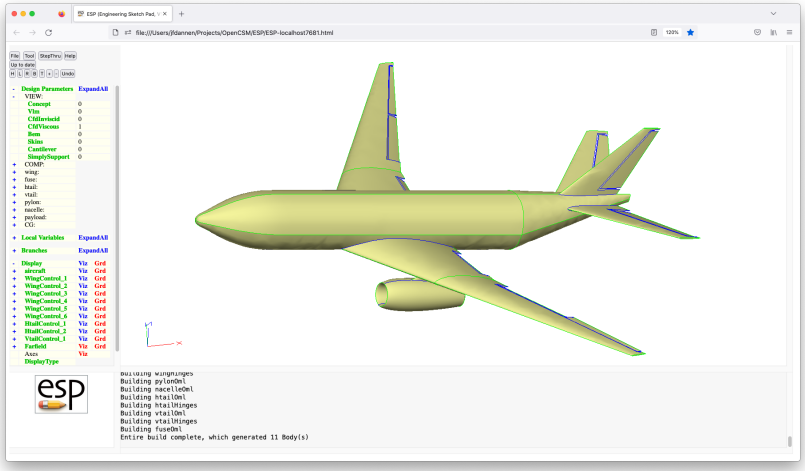


vlm

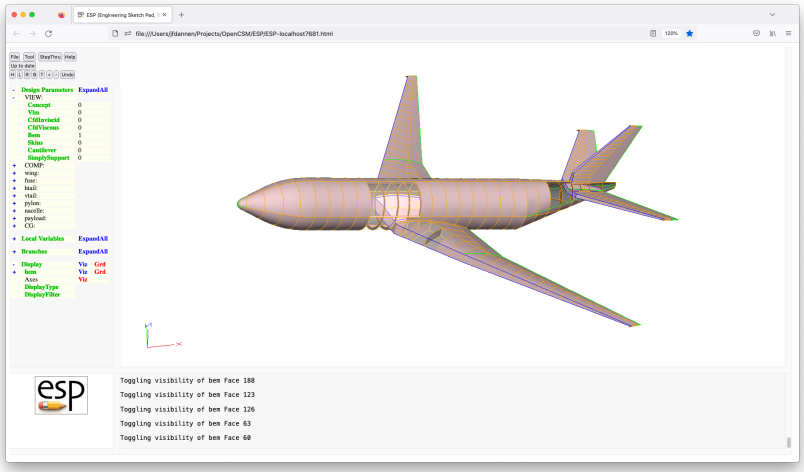
Multi-X Models in ESP — 2



cfdInviscid



cfdViscous



bem

Vehicle Sketch Pad (OpenVSP)

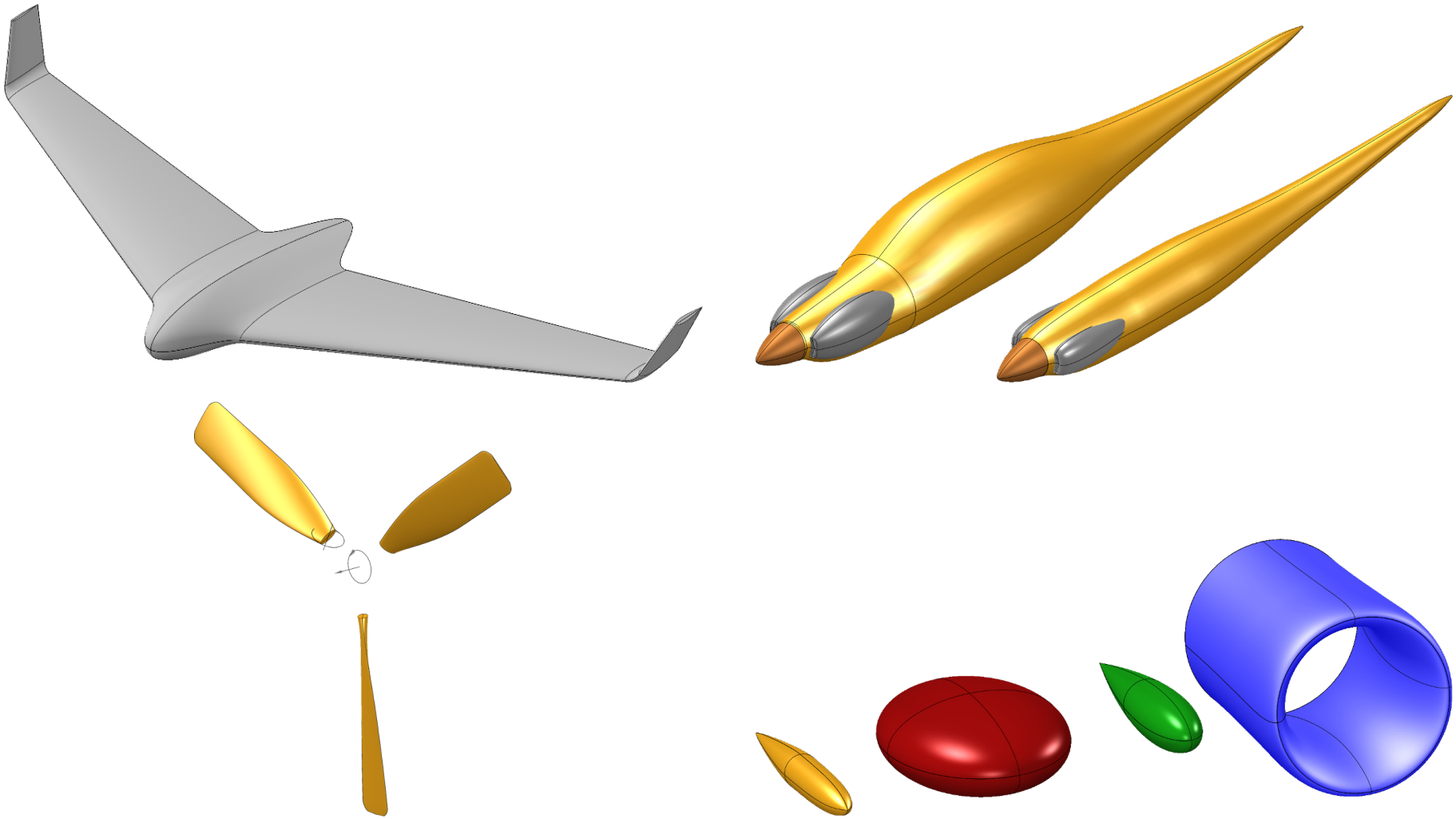
- OpenVSP is a parametric **aircraft geometry** tool
- Allows a user to create a 3D model of an aircraft defined by common engineering parameters
- OpenVSP is ideally suited to the early conceptual design phase to represent the overall form of the aircraft
- OpenVSP's models can be processed into formats suitable for engineering analysis

- Developed by J.R. Gloudemans and others for NASA since the early 1990's
- In 2012, OpenVSP was released as an open source project under the NASA Open Source Agreement (NOSA) version 1.3.

OpenVSP Features — 1

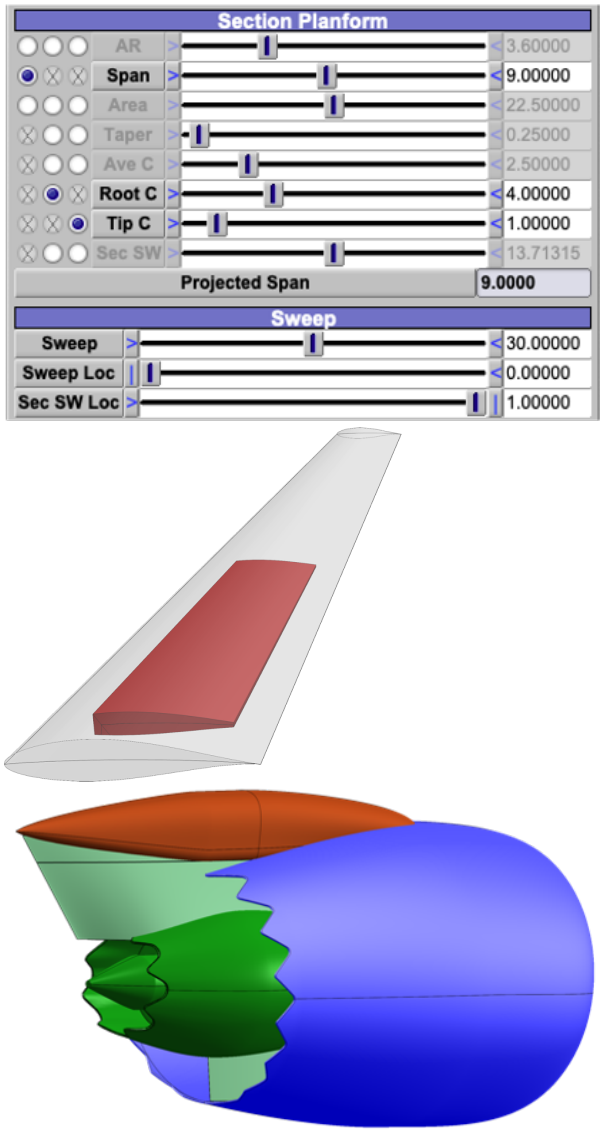
- Rich set of aircraft-specific components
  - wing
  - fuselage/stack
  - pod
  - propeller
  - body of revolution / ellipsoid
  - human
  - custom components
- Components parameterized with aircraft-specific parameters
  - aspect ratio
  - taper ratio
  - sweep
  - dihedral

OpenVSP Features — 2



OpenVSP Features — 3

- Near-instantaneous regeneration of models
  - parameters changed by sliders
  - parameters changed by values
- Background images to help configure a vehicle
- User-defined parameters and parameter linking
- Conformal components
- Specialized components



OpenVSP Features — 4

- Exports
  - CAD
  - CFD mesh
  - FEA mesh
- Integral analysis tools
  - VSPAERO, wave drag, parasite drag
  - FEA structural, aero-structural
  - FitModel
- External linkages
  - ESTECO
  - Chimera Grid Tools (CGT)
  - M4 Structures Studio
  - FlightStream
  - FLOWUnsteady

Advantages of Using Models in ESP

- Robust boolean operations
- Fillets (and Flends)
- Add non-airplane features
- Add components imported via STEP or IGES files
- Linked multi-disciplinary and multi-fidelity models
- Sensitivities
- Multi-user collaboration (“pair-programming”)
- Embedded Python interpreter (pyscript)
- Linkage to analysis tools (via CAPS)
- Incorporation of laser-scan data (via Plugs)

Static OpenVSP Models

- Use case:
  - user has an OpenVSP model and wants to combine it into a single body, apply fillets, add other components, or use any analysis supported by CAPS
  - the Bodys have fixed geometry (i.e., are not parametric)
- User process:
  - the .vsp3 file is generated by OpenVSP and must exist before the process starts
  - write a .csm file that calls the vsp UDP with the name of the .vsp file:

    ```
    UDPRIM vsp3 filename $$/sialia_r2.vsp3
    ```
  - execute serveESP on the .csm file

    ```
    serveESP sialia_r2
    ```

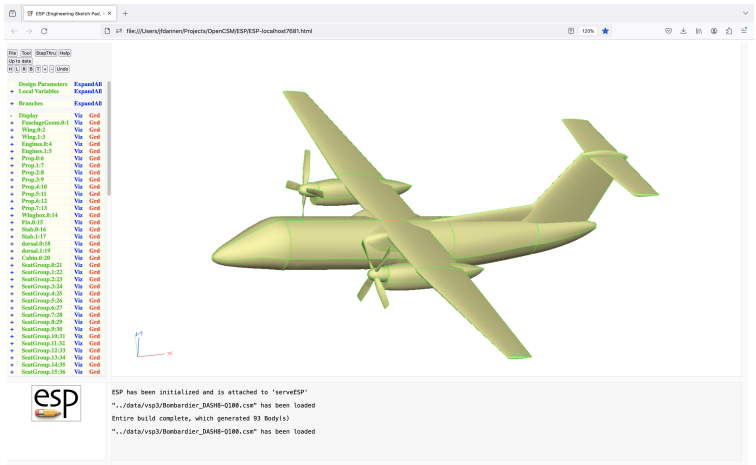
Behind the Scenes

- udpVsp3
  - writes an angelscript that directs OpenVSP to run the model and dump a STEP file that contains all (untrimmed) surfaces in the OpenVSP model
  - executes vspscript
  - imports STEP file
  - combines surfaces in each OpenVSP component into a SolidBody (or SheetBody if open)

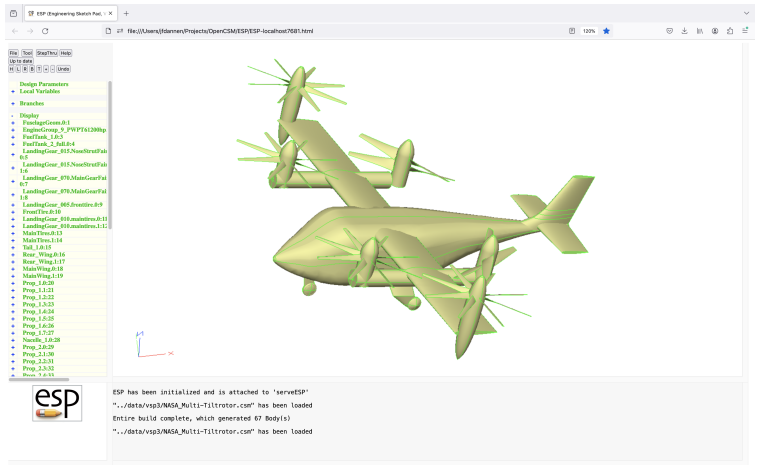
- Notes
  - process for transforming OpenVSP's input parameters to geometry is done by OpenVSP — not ESP
  - OpenVSP's piecewise Bezier surfaces are represented exactly in ESP

Static Model Examples

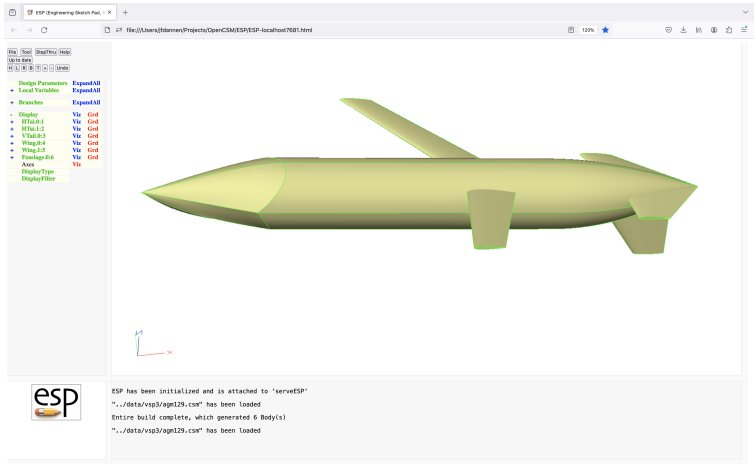
Note: one ESP Body for each OpenVSP Component



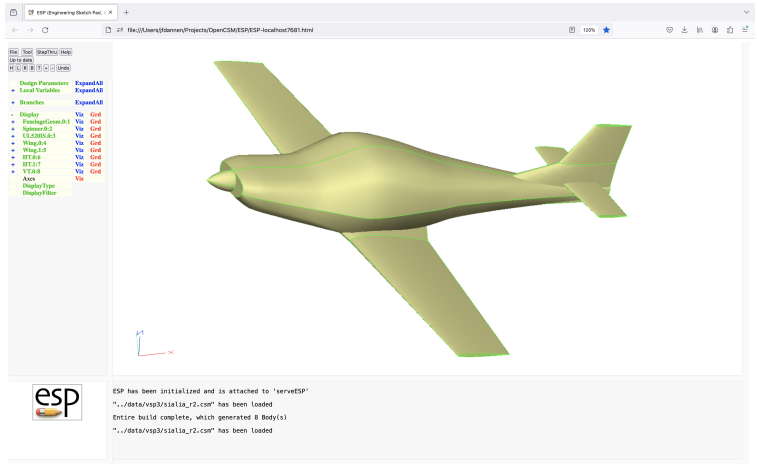
Bombardier_DASH8-Q100



NASA_Multi-Tiltrotor



agm129



sialia_r2

Static Model Demo

- Create simple `ESP` script

```
# sialia_r2
# written by John Dannenhoffer

UDPRIM    vsp3   filename $$/sialia_r2.vsp3

END
```

- Open `ESP`
  - `serveESP sialia_r2`

Parametric OpenVSP Model

- Use case:
  - user has an OpenVSP model and wants to perform parametric variations, design optimization, or sensitivity analysis
  - the OpenVSP model should have "User Parms" in the "ESP_Group"
- User process:
  - every time the user modifies the .vsp3 file:
    - in serveESP, use **Tool**$\longrightarrow$**VspSetup**, to write a .udc file that contains DESPMTR, LBOUND, and UBOUND statements from the tagged variables in the .vsp3 file, and a UDPRIM vsp3 statement to generate an updated configuration
  - write a .csm file that calls the UDC:

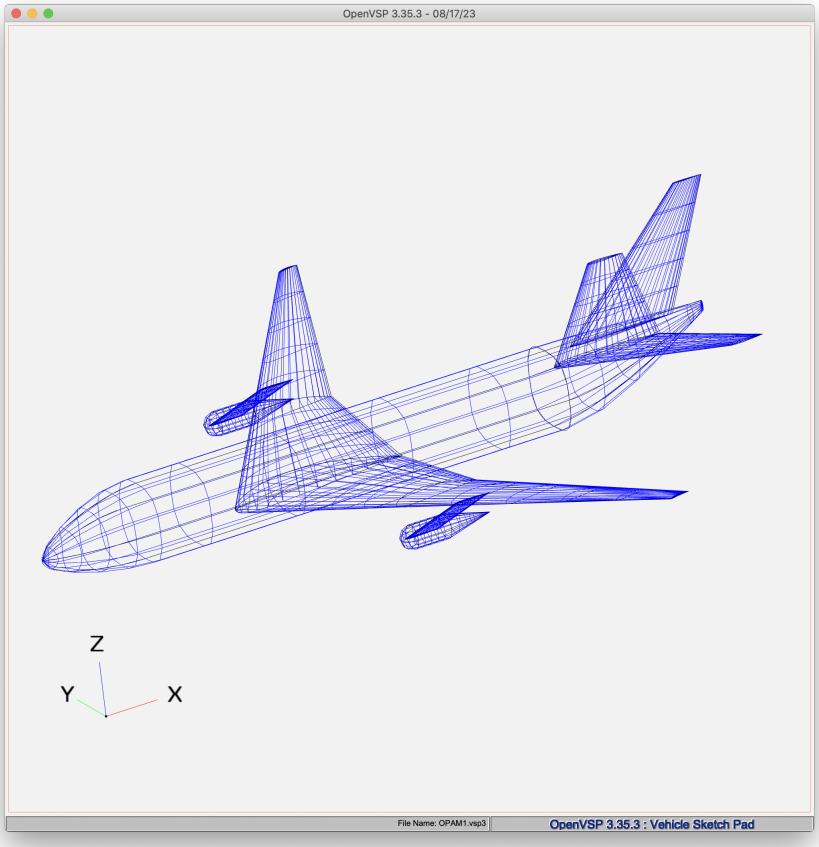
    ```
    UDPRIM /OPAM1
    ```

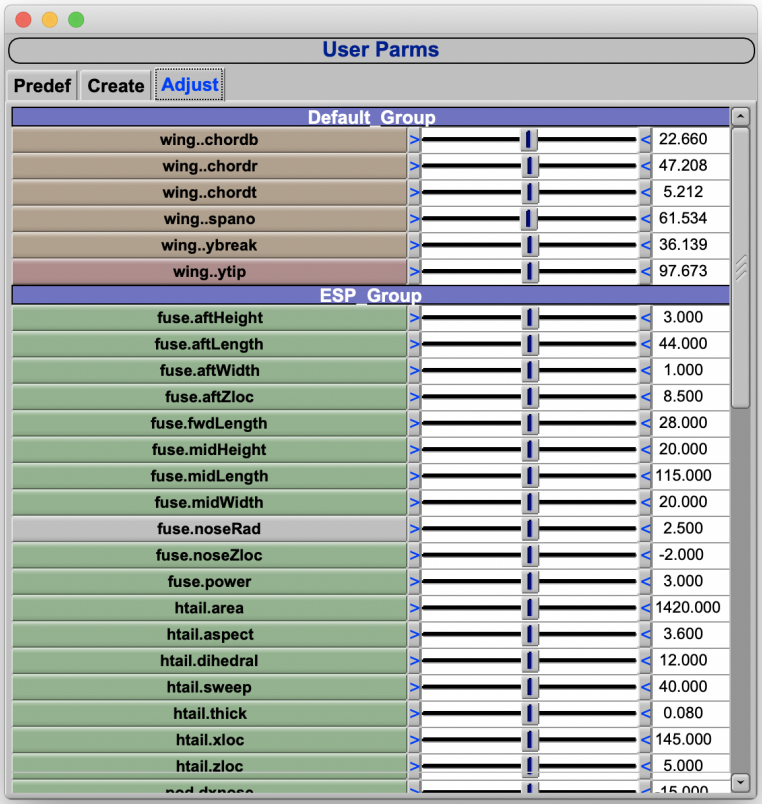
  - execute serveESP on the .csm file

    ```
    serveESP OPAM1
    ```

Parametric OpenVSP Model Demo — 1

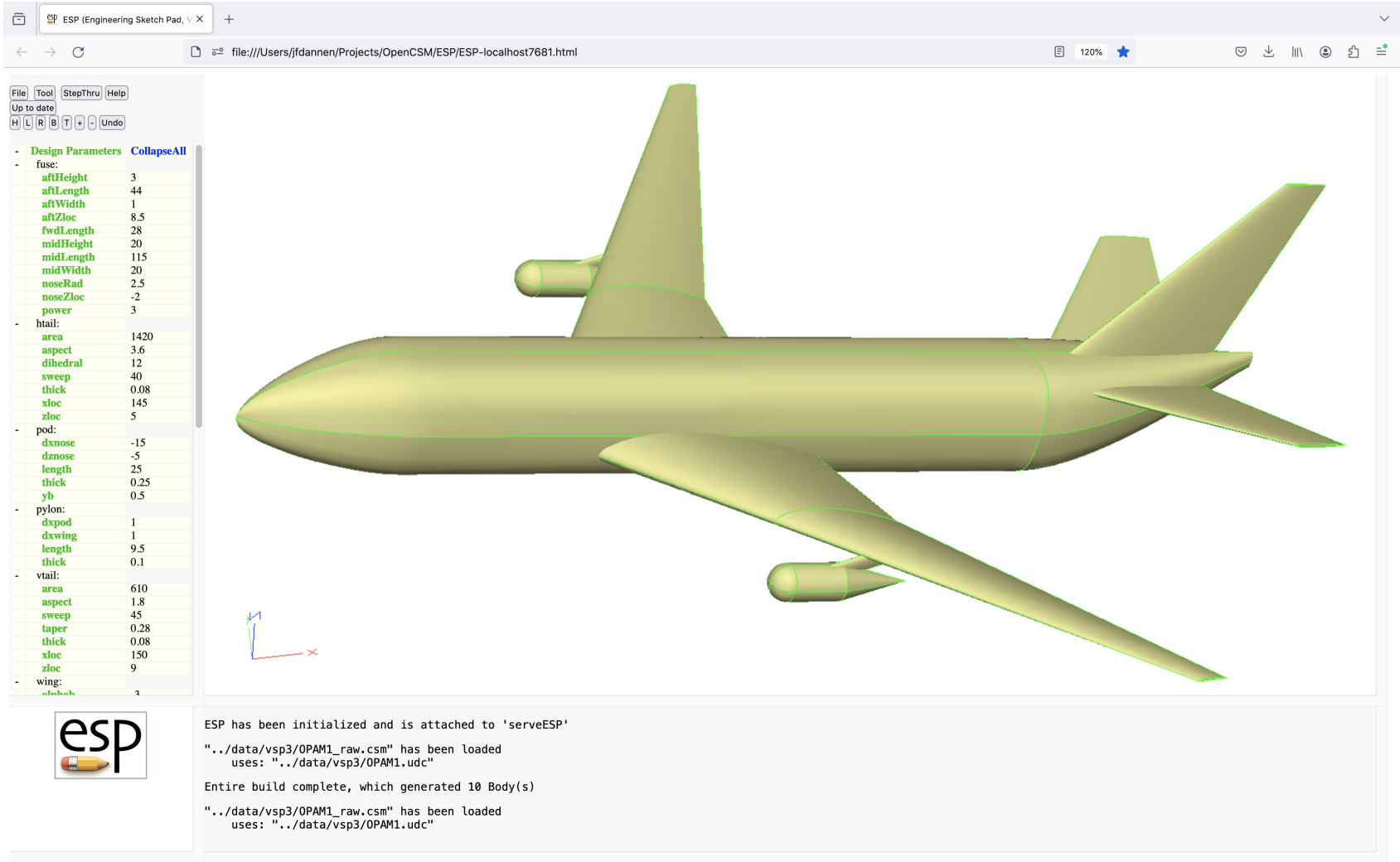
Screen-shots of OpenVSP





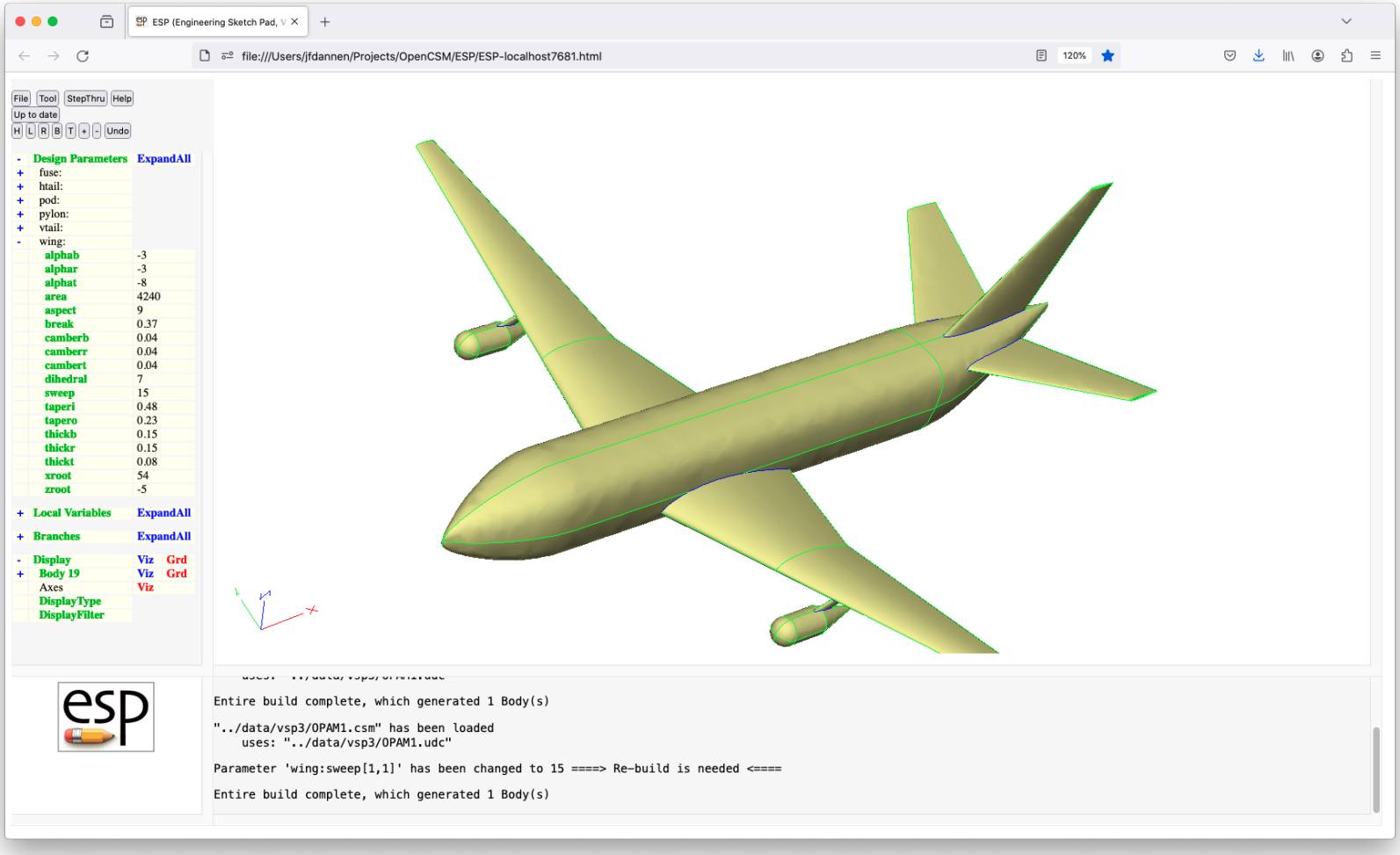
Parametric OpenVSP Model Demo — 2

Note: one ESP Body for each OpenVSP



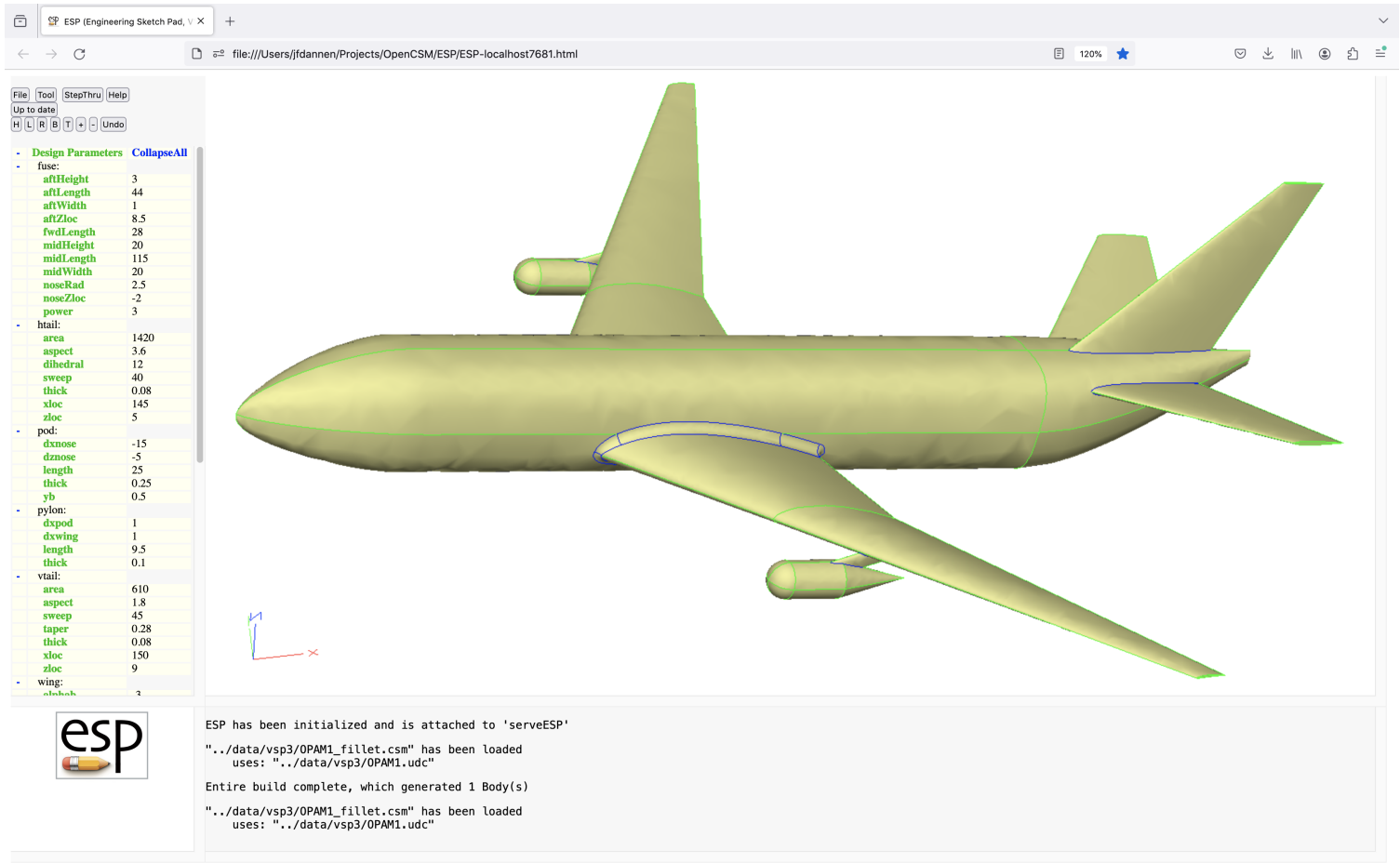
Parametric OpenVSP Model Demo — 3

Wing sweep changed to 15°



Parametric OpenVSP Model Demo — 4

Note: single Body with wing/fuselage fillets



Sensitivities

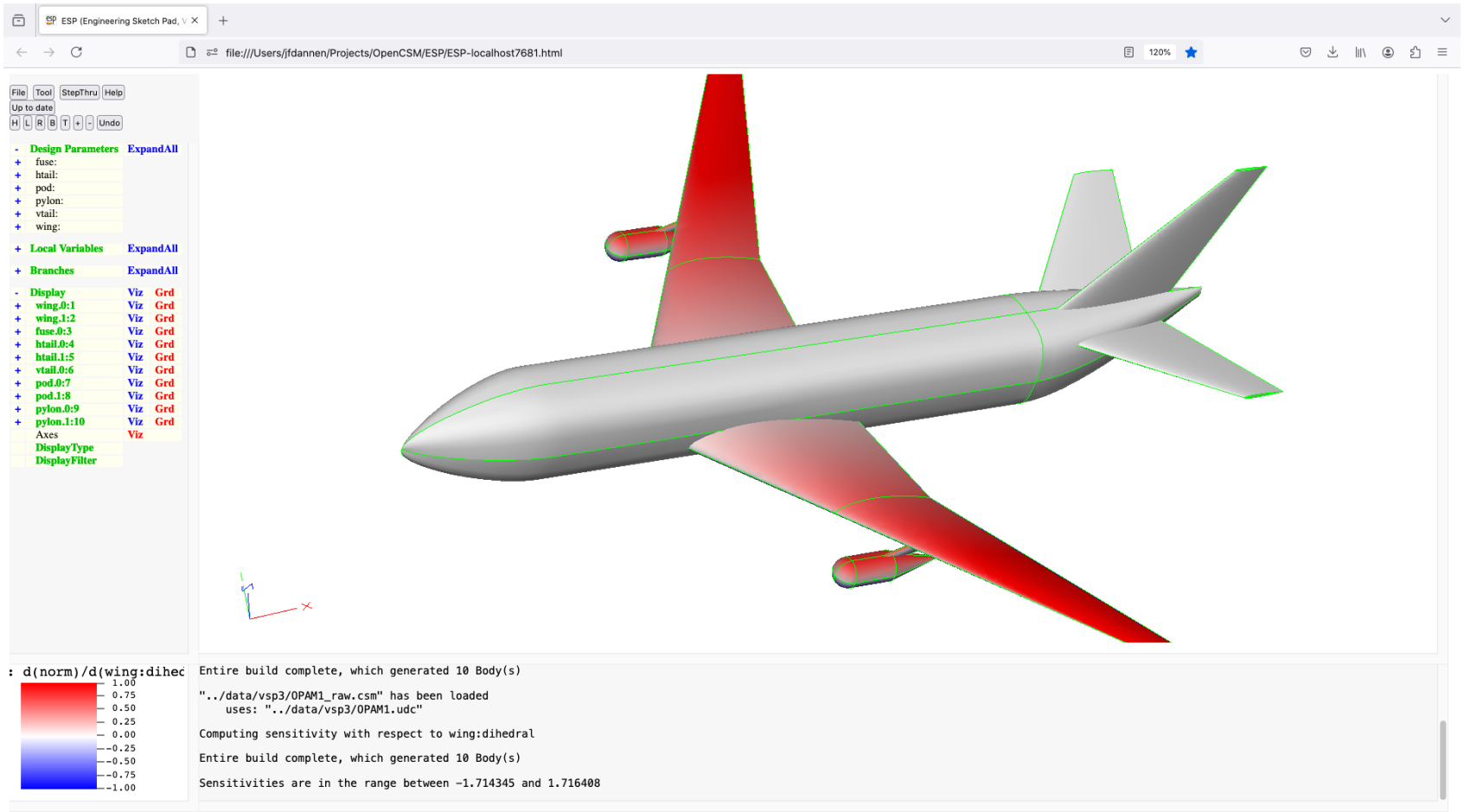
- ESP has the ability to compute two types of sensitivities:
  - configuration sensitivities tell derivative of the normal component at any point with respect to any group of design parameters (and is exact)
  - tessellation sensitivities tell derivative of tessellation point movement with respect to any group of design parameters (and is approximate since point placement algorithm is unknown to ESP)
- These sensitivities can be be applied to any to any design parameter in a parametric OpenVSP model

Behind the Scenes

- The derivative of the B-spline control point locations are computed via finite differences (using a base and perturbed model)
- The sensitivities at any location on the final model are computed analytically using `ESP`'s normal sensitivity mechanism

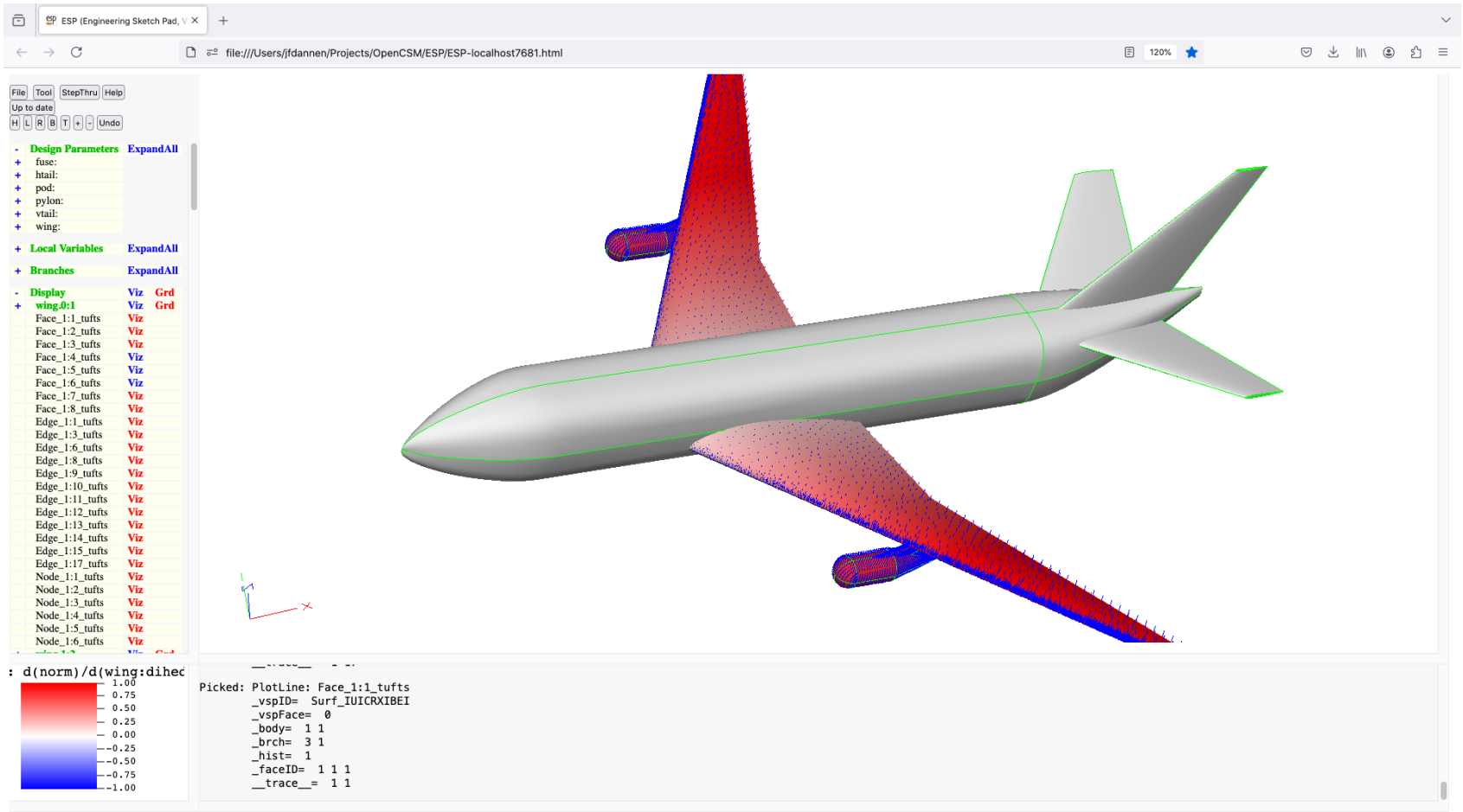
Parametric OpenVSP Model Demo — 5

Geometric Sensitivities wrt Wing Dihedral



Parametric OpenVSP Model Demo — 6

Tessellation Sensitivities wrt Wing Dihedral



Summary

- OpenVSP models can now be imported into ESP
  - static models are represented as fixed-shape Bodys
  - parametric models can be generated for any change in an ESP design parameter
- Once in ESP, users can:
  - combine the Bodys into a single Body, apply fillets, add other components, or use any analysis supported by CAPS
  - can compute sensitivities (for parametric models)
- All these capabilities are freely available in version 1.26 of ESP, which can be downloaded from acdl.mit.edu/ESP

Acknowledgments

- Special thanks to Rob McDonald for all his help in the use and understanding of `OpenVSP`
- This work was supported by the EnCAPS Project (AFRL Contract FA8650-20-2-2002): “EnCAPS: Enhanced Computational Prototype Syntheses”