On Unifying Geometric Representations in an MDAO Environment with Application to Aircraft Design

John F. Dannenhoffer, III

Syracuse University

Robert Haimes

Massachusetts Institute of Technology

7^{th} Research Consortium for
Multidisciplinary Systems Design Workshop
Purdue University

Overview

- Unified-geometry model
- Introduction to EGADS and OpenCSM
- Unified models for a fighter
- Future directions

Unified Geometry Model

- Start with engineering description (design intent)
  - feature tree (build script)
  - design parameters
  - attributes
- Generate *associated* models for various analyses
  - mid-surface aerodynamics (MSA)
  - outer-mold line (OML)
  - built-up element model (BEM)
  - solid structure model (SSM)

Introduction to EGADS

The Electronic Geometry Aircraft Design System (EGADS) is an open-source geometry interface to OpenCASCADE

- reduces OpenCASCADE's 17,000 methods to about 60 C calls
- supports "bottom-up" construction
  - curve: line, circle, ellipse, parabola, hyperbola, offset, bezier, Bspline
  - surface: plane, spherical, conical, cylindrical, toroidal, revolution, extrusion, offset, bezier, Bspline
  - topological: node, edge, loop, face, shell, body, (model)
- ...

Introduction to EGADS

- supports "top-down" construction
  - primitives: box, cylinder, cone, sphere, torus
  - evolved: extrude, revolve, loft, sweep
  - applied: fillet, chamfer, hollow/offset
- provides persistent user-defined attributes on all topological entities
- construction is via calls to API

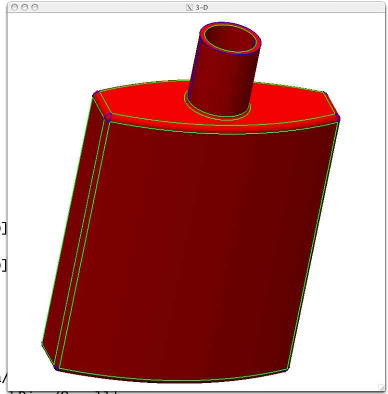
Introduction to OpenCSM

OpenCSM is an open-source constructive solid modeler

- gives user access to:
  - master model (design parameters and feature tree)
  - boundary representation (BRep composed of volumes, faces, edges, and nodes)
- built upon:
  - EGADS (simple access to OpenCASCADE)
  - CAPRI (vendor-neutral access to *Parasolid*, UniGraphics, Pro/ENGINEER, CatiaV5, SolidWorks, . . .)

OpenCSM Example

```
External Parameter(s):
  width        [  1,   1]    10.00000
  depth        [  1,   1]     4.00000
  height       [  1,   1]    15.00000
  neckDiam     [  1,   1]     2.50000
  neckHeight   [  1,   1]     3.00000
  wall         [  1,   1]     0.20000
  filRad1      [  1,   1]     0.25000
  filRad2      [  1,   1]     0.10000

Branch(es):
  Brch_000001   set      [baseHt] [height-neckHeight]
  Brch_000002   skbeg    [-width/2] [-depth/4] [0]
  Brch_000003   .cirarc  [0] [-depth/2] [0] [+width/2] [-depth/4] [0]
  Brch_000004   .linseg  [+width/2] [+depth/4] [0]
  Brch_000005   .cirarc  [0] [+depth/2] [0] [-width/2] [+depth/4] [0]
  Brch_000006   .linseg  [-width/2] [-depth/4] [0]
  Brch_000007   skend
  Brch_000008   extrude  [0] [0] [baseHt]
  Brch_000009   fillet   [filRad1] [0] [0]
  Brch_000010   set      [holeBot] [height-neckHeight/2]
  Brch_000011   cylinder [0] [0] [baseHt] [0] [0] [height] [neckDiam/
  Brch_000012   cylinder [0] [0] [holeBot] [0] [0] [height+wall] [neckDiam/2-wall]
  Brch_000013   subtract [none] [1]
  Brch_000014   union
  Brch_000015   fillet   [filRad2] [0] [0]
```



OpenCSM API

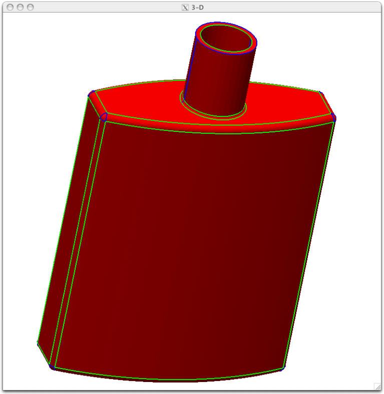
- Load a Master Model
  - ocsmLoad(filename, *modl)
- Interrogate and/or edit the Master Model
  - ocsmInfo(modl, *nbrch, *npmtr, *nbody)
  - ocsmSetBrch(modl, ibrch, actv)
  - ocsmGetPmtr(modl, ipmtr, *type, *nrow, *ncol, name[])
  - ocsmSetValu(modl, ipmtr, irow, icol, defn)
- Execute the feature tree and create a BRep
  - ocsmBuild(modl, buildTo, *builtTo, *nbody, bodyList[])
- Interrogate the BRep
  - ocsmGetBody(modl, ibody, . . . , *nnode, *nedge, *nface)
  - any of EGADS' or CAPRI's evaluators and inverse evaluators
- Note: API contains fewer than 30 calls

OpenCSM .csm File Description

- ASCII file that contains build recipe that is executed in a stack-like way
- All arguments are MATLAB-like expressions
- Primitives: box, cylinder, cone, sphere, torus
- Grown bodies: extrude, loft, revolve, (sweep)
- User-defined primitives: ellipse, freeform solid, NACA airfoil, ...
  - combines "top-down" with "bottom-up" construction
- Applied features: fillet, chamfer, hollow, offset
- Boolean operators: union, difference, intersection
- Sketches: lines, circular arcs, splines, constraints
- Transformations and Utilities: translate, rotate, scale, patterns, macros

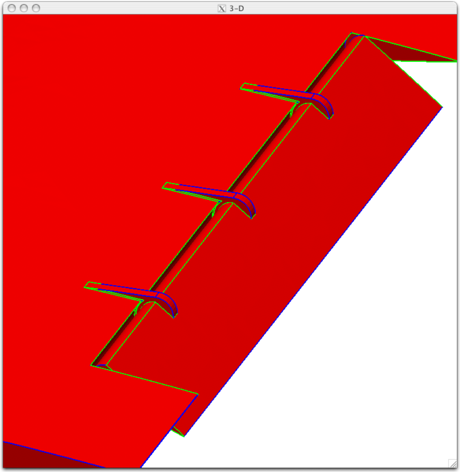
OpenCSM Example

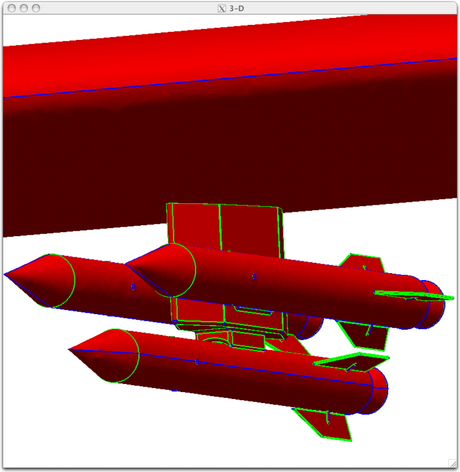
```
# design parameters
despmtr  width       10.00
despmtr  depth        4.00
despmtr  height      15.00
despmtr  neckDiam     2.50
despmtr  neckHeight   3.00
despmtr  wall         0.20
despmtr  filRad1      0.25
despmtr  filRad2      0.10

# basic bottle shape (filleted)
set         baseHt   height-neckHeight
skbeg       -width/2 -depth/4 0
   cirarc   0        -depth/2 0  +width/2 -depth/4  0
   linseg   +width/2 +depth/4 0
   cirarc   0        +depth/2 0  -width/2 +depth/4  0
skend
extrude  0        0        baseHt
fillet   filRad1  0        0

# neck with a hole
set      holeBot height-neckHeight/2
cylinder 0   0   baseHt  0 0 height      neckDiam/2
cylinder 0   0   holeBot 0 0 height+wall neckDiam/2-wall
subtract

# join the neck to the bottle and apply a fillet at the union
union
fillet   filRad2  0        0
```



Sample Configuration — Wing with stores

750 feature tree nodes

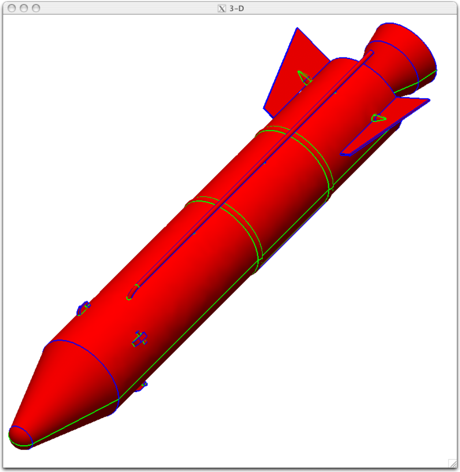


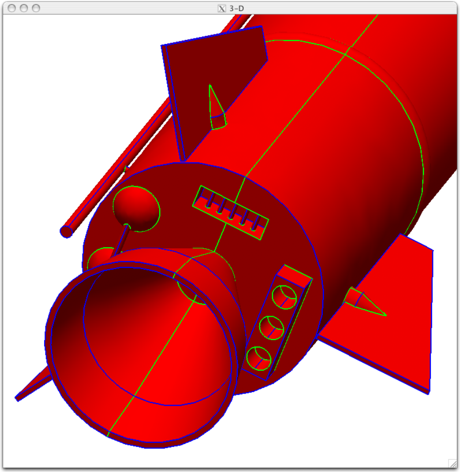


163 volumes, 715 nodes, 1229 edges, 542 faces

Sample Configuration — JMR3

378 feature tree nodes

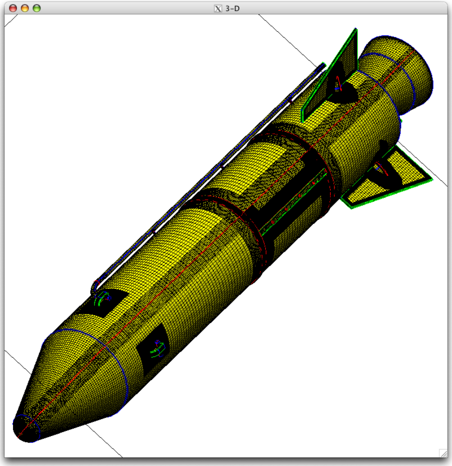


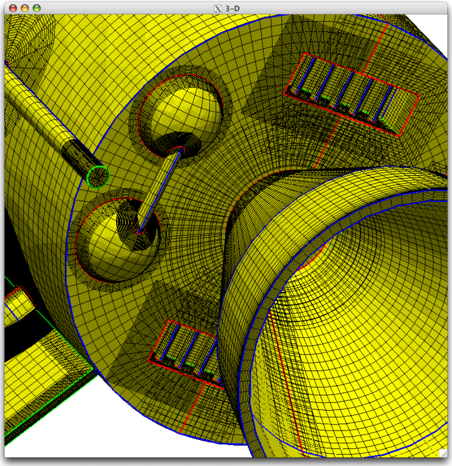


115 volumes, 296 nodes, 462 edges, 194 faces

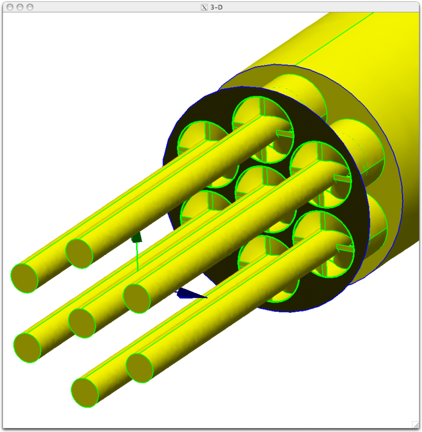
Overset Surface Grids — JMR3

76 basic grids, 152 collar grids, 1 global grid

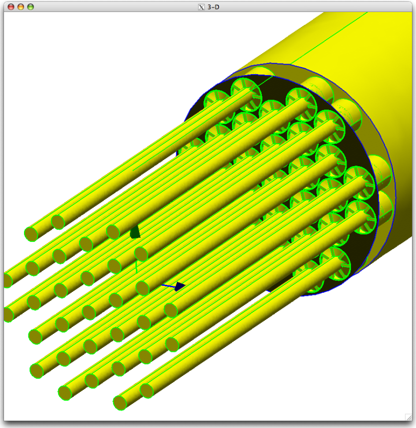




NASA's Lean-direct Injector Design



baseline
10 design variables

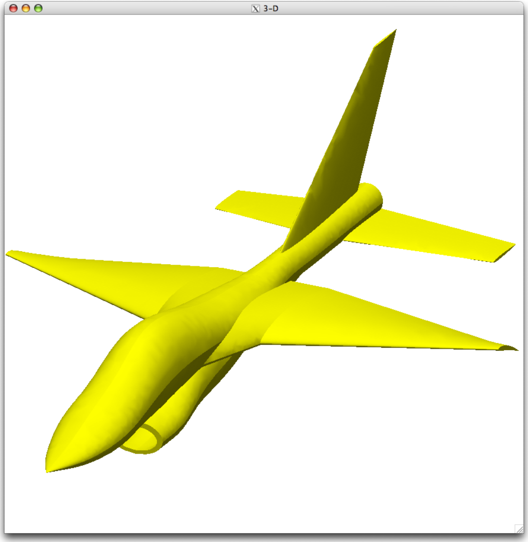


more injectors and
vanes/injector

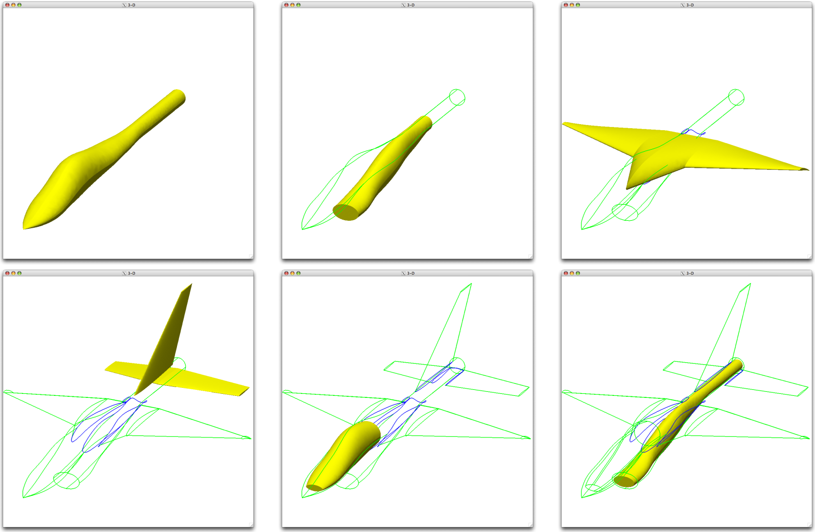
Unified Models for Fighter Configuration

- Notional description of aircraft
  - (cranked) wing
  - horizontal and vertical tails
  - fuselage
  - integrated propulsion system
- Design parameters
  - wing: series and location/chord/twist at root, break, and tip
  - tails: series and location/chord/twist at root and tip
  - fuselage:
    - OML as lofting of cross-sectional shapes
    - IMLs as lofting of cross-sectional shapes
  - propulsion system:
    - OML as lofting of cross-sectional shapes
    - IML as lofting of cross-sectional shapes

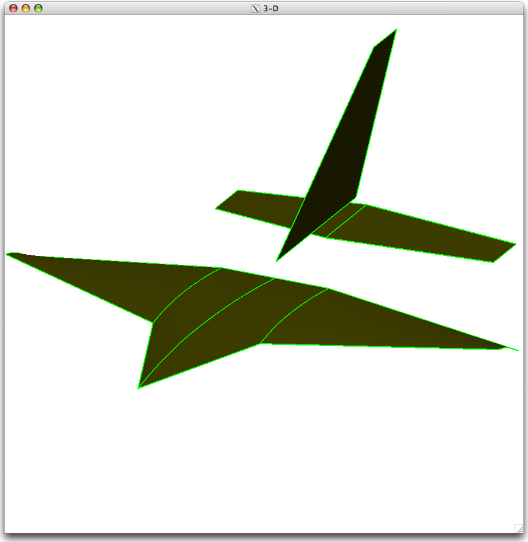
Outer-mold line (OML)



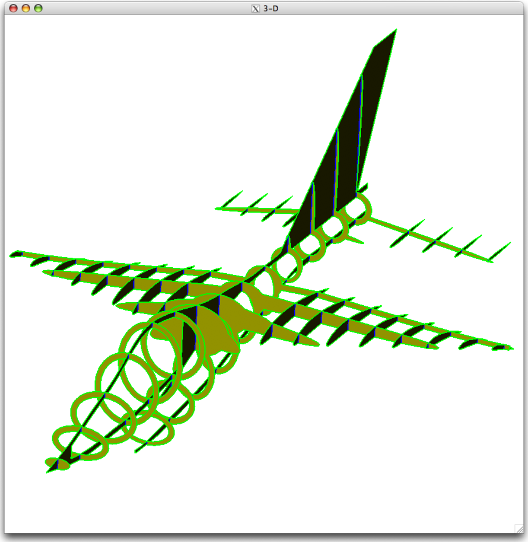
OML — Build-up Sequence



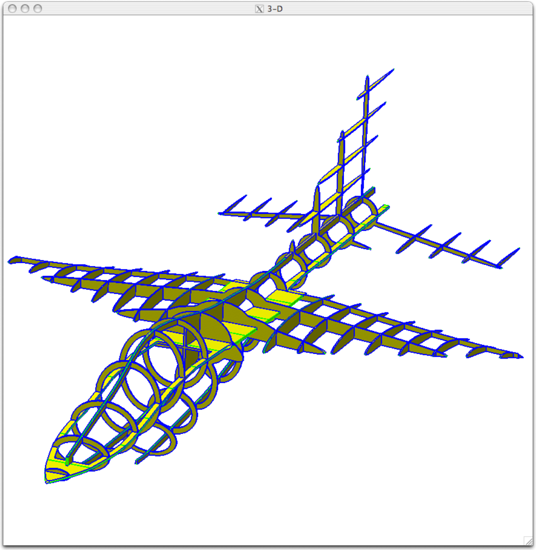
Mid-surface aerodynamics (MSA)



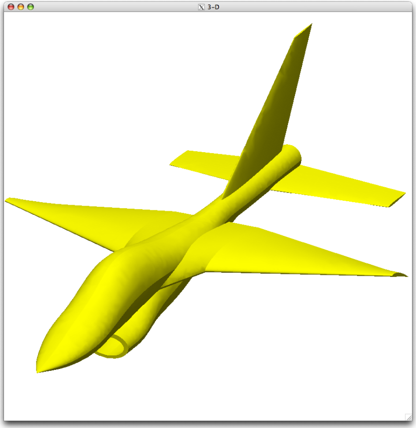
Built-up element model (BEM)



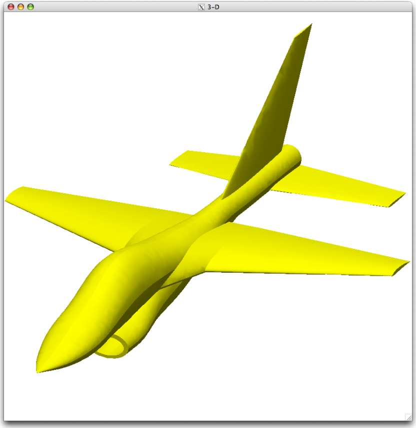
Solid structure model (SSM)



Parametric Variation 1 — Untwisted Wing

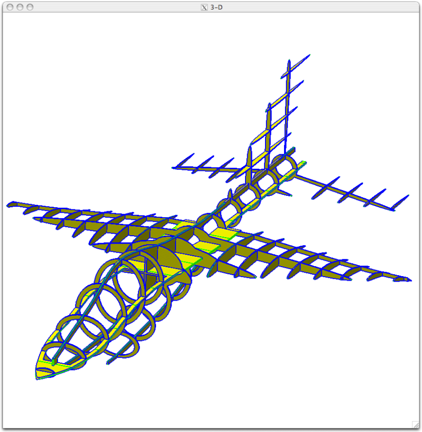


20° wing tip twist

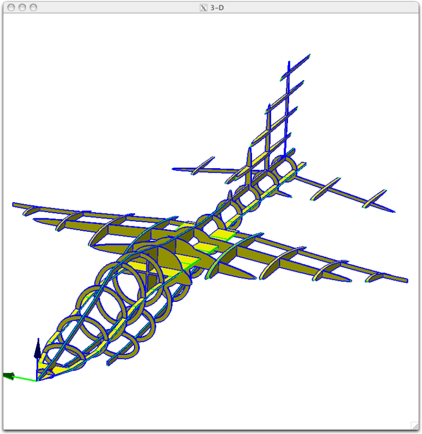


no wing tip twist

Parametric Variation 2 — Fewer Ribs



8 thin wing ribs



4 thick wing ribs

Current Status & Future Directions

- Current status
  - Availability
    - EGADS down-loadable from OpenMDAO's GitHub site
    - OpenCSM in alpha release; beta expected end of summer 2012
  - Use
    - integrated with OpenMDAO though GEM/pyRite
    - initial talks to integrate with Sorcer
- Future directions
  - multi-disciplinary and multi-fidelity coupling
  - sensitivities
  - sub-system integration
- Related work
  - automatic generation of overset grid systems

Acknowledgements

- NASA NRA
  - Chris Heath, Technical Monitor
- AFRL Collaboration
  - Ray Kolonay, Technical Contact