An Introduction to Simulation Using 42

Eric Stoneking

2017

What is 42?

- A simulation of spacecraft attitude and orbital dynamics and control
- Intended for use from concept studies through ops
  - Rapid prototyping makes it useful for MDL studies
  - Environment models support actuator sizing, performance studies
  - High-fidelity dynamics handle multi-body, flexible-body spacecraft
  - Portability (Mac, linux, Windows) minimizes infrastructure requirements
  - Clean interface aids progression from flight software “model” to dropping in actual flight software
  - Visualization aids situational awareness from concept to operations
- Designed to be powerful, but easy to get started

Features

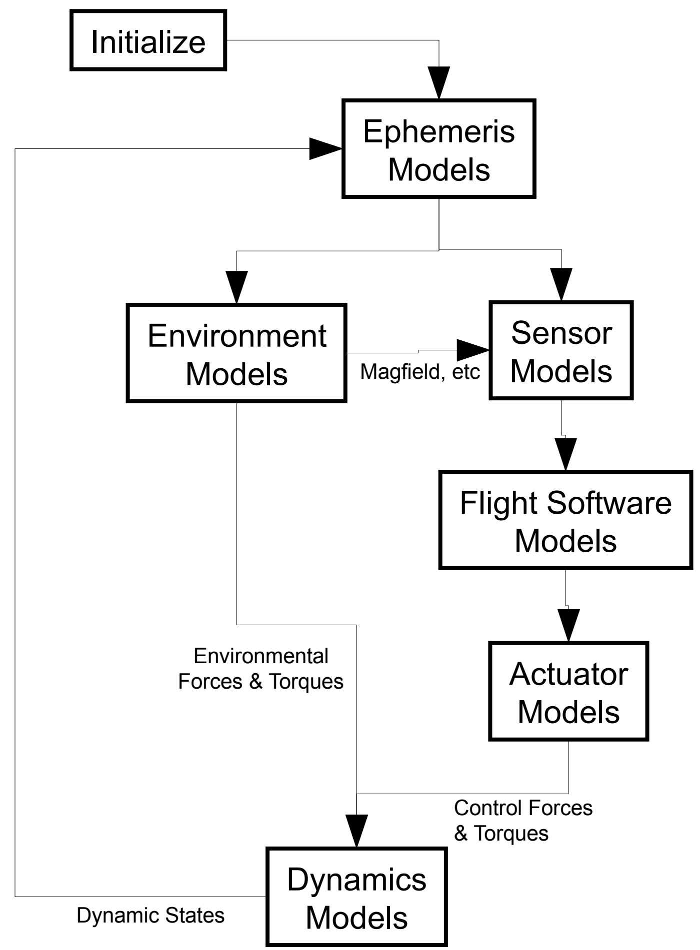
- Multiple spacecraft, anywhere in the solar system
  - Two-body, three-body orbit dynamics (with seamless transition between)
  - One sun, nine planets, 45 major moons
  - Minor bodies (comets and asteroids) added as needed
    - RQ36, Eros, Itokawa, Wirtanen, etc
- Multi-body spacecraft
  - Tree topology
  - Kane's dynamics formulation
  - Each body may be rigid or flexible
  - Flexible parameters taken (by m-file script) from Nastran output (.f06 file)
  - Joints may have any combination of rotational and translational degrees of freedom

More Features

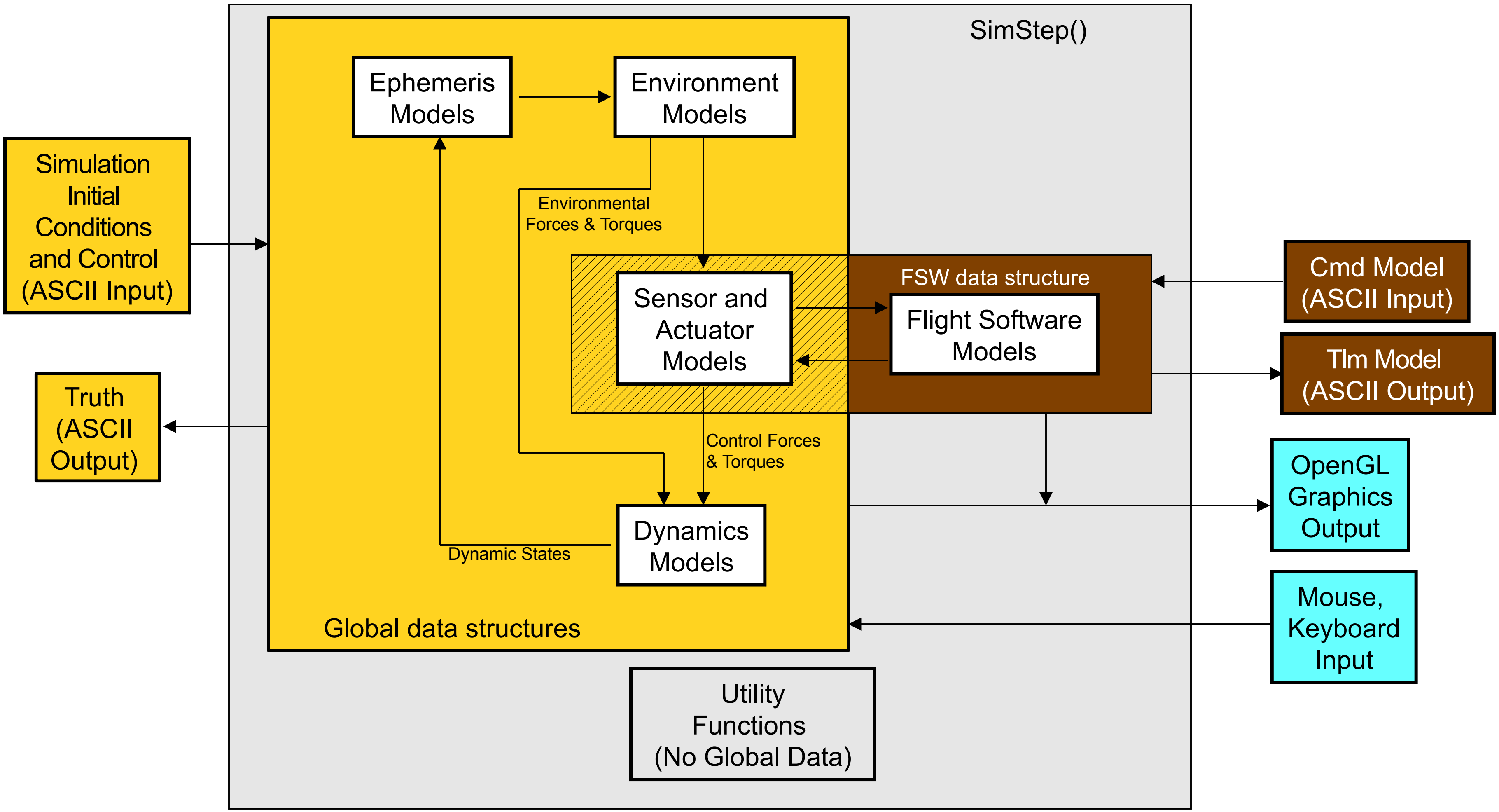
- Supports precision formation flying
  - Several S/C may be tied to a common reference orbit
  - Encke's method or Euler-Hill equations used to propagate relative orbit states
    - Precision maintained by judicious partitioning of dynamics
      - Add big things to big things, small things to small things
- Clean FSW interface facilitates FSW validation
  - Used by GLAST project for independent validation of vendor's (autocoded) GNC flight software
- Open Source, available on sourceforge and github
  - Sourceforge.net/projects/fortytwospacecraftsimulation
  - Github/ericstoneking/42

A Basic Simulation Loop

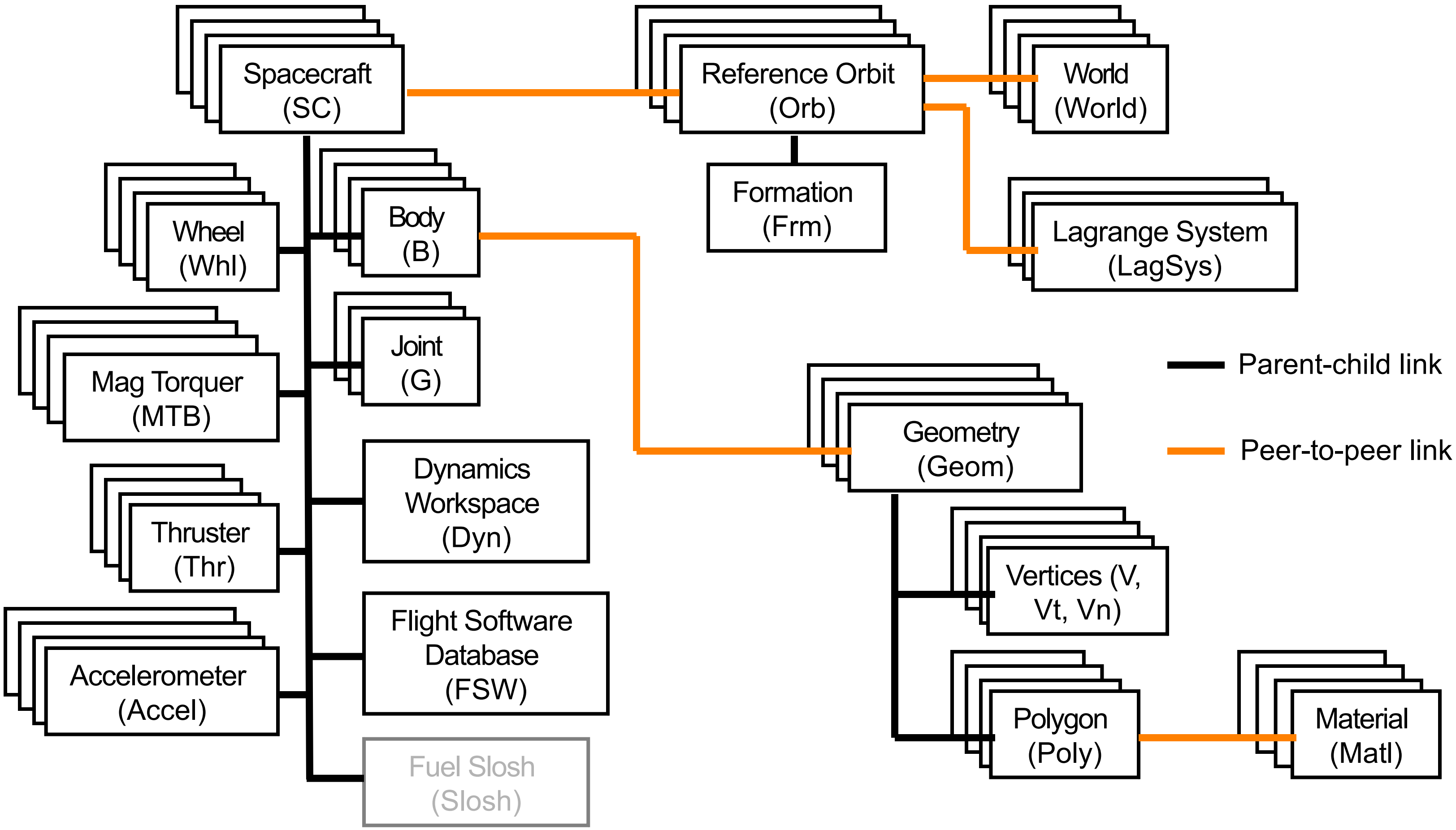
- Initialize
  - –Read user inputs
  - –Set up
- Ephemeris: Where is everything?
  - –Sun, Earth, Moon, etc
  - –Orbits
  - –Spacecraft
- Environment Models: What forces and torques exerted by the environment?
- Sensor Models
  - –Input truth
  - –Output measurements
- Flight Software Models
  - –Input Measurements
  - –Process Control Laws, etc
  - –Output Actuator Commands
- Actuator Models
  - –Input Commands
  - –Output Forces and Torques
- Dynamics: How does S/C respond to forces and torques?
  - –Integrate dynamic equations of motion over a timestep
  - –Advance time to next step



42's Architecture



Global Data Structure Relationships



42 Walkthrough

Models, Terminology, Input Files

Environment Models

- Planetary Ephemerides
  - From Meeus, “Astronomical Algorithms”
  - Good enough for GNC validation, not intended for mission planning
    - Use GMAT or ODTBX for that
- Gravity Models have coefficients up to 18th order and degree
  - Earth: EGM96
  - Mars: GMM-2B
  - Luna: GLGM2
- Planetary Magnetic Field Models
  - IGRF up to 10th order (Earth only)
  - Tilted offset dipole field
- Earth Atmospheric Density Models
  - MSIS-86 (thanks to John Downing)
  - Jacchia-Roberts Atmospheric Density Model (NASA SP-8021)
- Simple exponential Mars atmosphere density model
  - New models easily incorporated as the state of the art advances

Dynamics Models

- Full nonlinear "6DOF" (actually N-DOF) dynamics
- Attitude Dynamics
  - One or many bodies
    - Tree topology (no kinematic loops)
  - Each body may be rigid or flexible
  - Joints may combine rotational and translational DOFs
    - May be gimballed or spherical
  - Slosh may be modeled as a pendulum (lo-fi, quick to implement and run)
    - 42 may run concurrently with Star-CCM CFD software for hi-fi slosh
  - Wheels embedded in Body[0]
  - Torques from actuators, aerodynamic drag, gravity-gradient, solar radiation pressure, joint torques
- Orbit Dynamics
  - Two- or three-body orbits
  - Encke or Euler-Hill (Clohessy-Wiltshire) for relative orbit motion (good for formation flying, prox ops)
  - Forces from actuators, aerodynamic drag, non-spherical gravity, third-body gravity, solar radiation pressure

Reference Frames are Important!

- In any dynamics problem beyond the spinning top, a systematic approach to reference frames and the relationships between them is vital
- For 42, we define several fundamental reference frames, and notational conventions to keep quaternions and direction cosines sorted out

Reference Frames (1 of 2)

- Heliocentric Ecliptic (H)
  - Planet positions expressed in this frame
- Each world has an inertial (N) and rotating (W) frame
  - For Earth, N = ECI (True of date), W = ECEF
  - N is the bedrock for orbits, S/C attitude dynamics
  - Full Disclosure: Although True-of-Date <-> J2000 conversions are provided, the distinction is not always rigorously made
    - Star vectors provided in J2000 (from Skymap), converted to H
    - Planet ephemerides are assumed given in true-of-date H
    - Transformation from N to W is simple rotation, implying N is True-of-Date
    - TOD ↔ J2000 conversions in envkit.c

Reference Frames (2 of 2)

- Each reference orbit has a reference point R
  - For two-body orbit, R moves on Keplerian orbit
  - For three-body orbit, R propagates under influence of both attracting centers (as point masses)
  - S/C orbit perturbations integrated with respect to R
- Associated with each R is a LVLH frame (L) and a formation frame (F)
  - F is useful for formation-flying scenarios
  - F may be offset from R, may be fixed in N or L
- Each spacecraft has one or more Body (B) frames and one LVLH frame (L)
  - L(3) points to nadir, L(2) points to negative orbit normal
  - SC.L is distinct from Orb.L, since SC may be offset from R

Representing Attitude

- There are several ways to represent the rotation between two reference frames
  - Direction Cosines
  - Euler Angles
  - Quaternions (aka Euler Parameters)
  - and more
- They all have their strengths and weaknesses
  - Learn them all!

Strengths and Weaknesses of Attitude Representations

Representation	Strengths	Weaknesses	Best Used For
Direction Cosines	• Work well with vectors • Easy to catenate rotations • Moderately intuitive (dot products) • No singularities	• 9 params for 3 DOF	• Transforming Vectors
Quaternions	• Efficient (4 params for 3 DOF) • No singularities	• Not intuitive	• Propagating Equations of Motion
Euler Angles	• Intuitive • 3 params for 3DOF	• Singularities • 24 Variants	• Input, Output • Gimballed Joints

Notation for Quaternions, DCMs

- The rotation from frame *A* to frame *B* may be described by the direction cosine matrix

$$ {}^{B}C^{A}_{ij} = \hat{b}_i \cdot \hat{a}_j $$

- Given the components of a vector in *A*, its components in *B* may be found by the multiplication

$$ {}^{B}v = {}^{B}C^{A}\,{}^{A}v $$

- In C, we write the DCM as CBA to preserve order of superscripts, eg

```
MxV(CBA,va,vb)
```

- Quaternions are another way to describe rotations. We use a parallel notation:

```
QxV(qba,va,vb)
```

- These and similar conventions promote concise, *unambiguous* code

Interfaces to Matlab

- 42 generates ASCII output files, which may be loaded into Matlab (or whatever) for post-processing and plotting
- Using Matlab's `mcc` utility, m-files may be translated into C, and compiled and linked into 42

Matlab + 42 = Monte Carlo

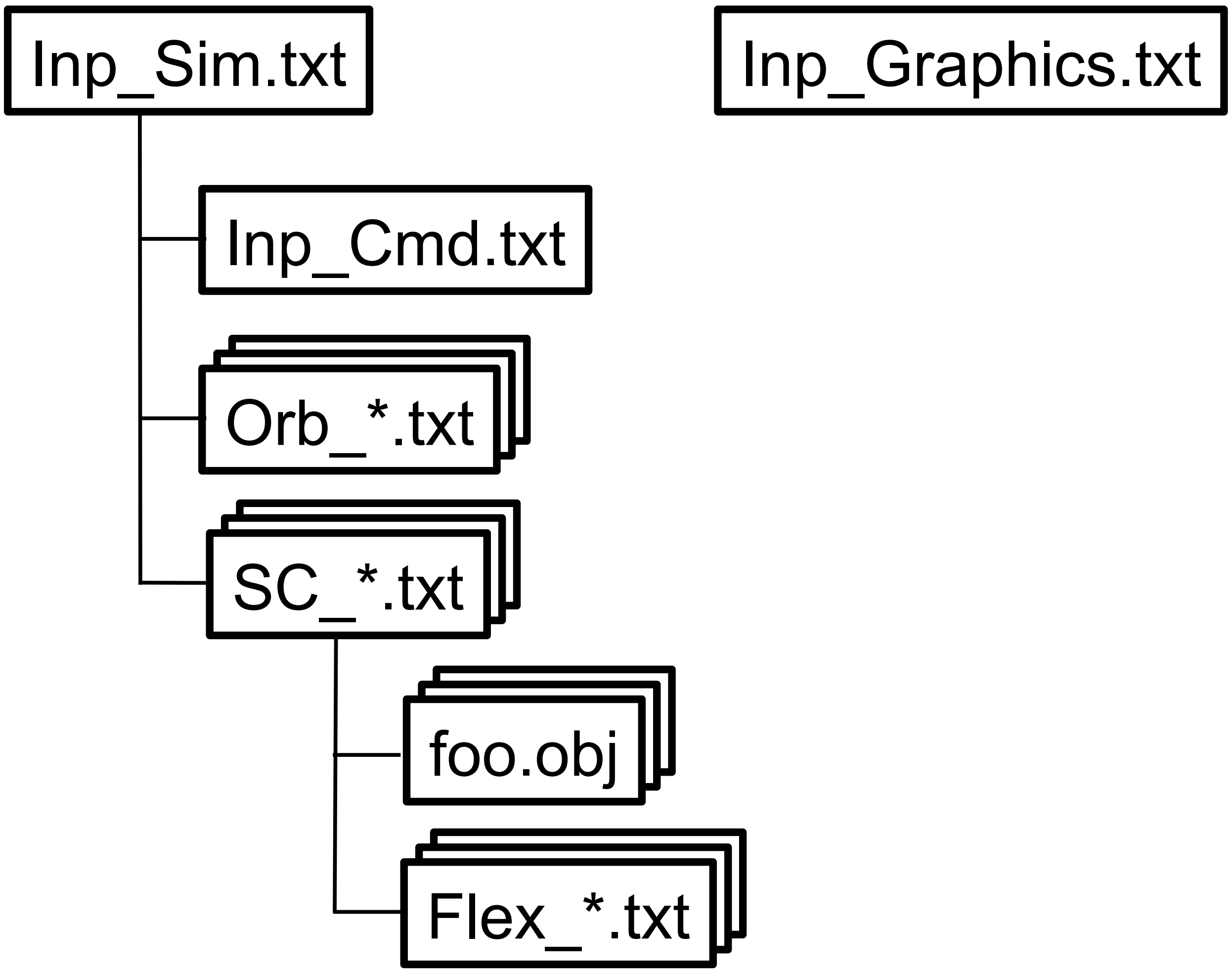
- 42 can be called from within Matlab using the `system` command
- Use Matlab as the MC executive
  - Generate initial conditions, parameters
  - Write to 42's input files
  - Run 42
  - Process and save data
  - Repeat
- Use 42 as the high-speed, high-fidelity component

Matlab/42 Example

```
for Irun=1:Nrun,

   % Compute initial attitude
   CRN = TRIAD(tvn(Irun,:),svn,[0 0 1],[1 0 0]);
   qrn = C2Q(CRN);

   % Write target to file
   Outdata = [TrgRA(Irun) TrgDec(Irun)];
   save -ascii ./MOMBIAS/TargetRaDec.inp Outdata

   % Write initial attitude to file
   line = sprintf('%f %f %f %f ! Quaternion\n', qrn(1),qrn(2),qrn(3),qrn(4));
   OverwriteLineInFile('./MOMBIAS/GLAST.inp',21,line);

   % Run 42 for three days.
   system('./42 MOMBIAS');

   % Record pointing histogram.
   load ./MOMBIAS/AngleToGo.42
   [HistCount(Irun,:),HistAng(Irun,:)] = hist(AngleToGo,20);

end
```

Input File Overview

Input File Hierarchy



Geometry Definition

- Geometry uses Wavefront "Object" format
  - ASCII, human-readable
  - Can hand-generate models *in theory*
  - In practice, you'll want mechanical help
- I use Wings3D solid modeling software
  - Free, multi-platform
- 42 can support myriad-polygon models, limited mainly by your patience
  - I get impatient at about 10,000 polys
- Note that aerodynamic and solar-pressure force and torque computations use these models
  - Interior polygons, self-shadowing are error sources

```
# Exported from Wings 3D 0.99.00b
mtllib Altair.mtl
o cylinder9
#12 vertices, 8 faces
v 2.11050391 1.21850000 -1.80666667
v 2.11050391 -1.21850000 -1.80666667
[...]
vn 0.0000000e+0 9.1113913e-17 1.00000000
vn 0.50000000 0.86602540 0.0000000e+0
[...]
g cylinder9_SHINY_WHITE
usemtl SHINY_WHITE
f 1//1 6//16 5//13 4//10 3//7 2//4
f 1//2 7//20 12//35 6//17
[...]
```

Excerpt from Altair.obj