DYMOS
DEVELOPMENT UPDATE

Rob Falck
Development Team Lead

RECENT EFFORTS INVOLVING DYMOS

NEXT GENERATION AIRCRAFT DESIGN TOOLS

- Dymos will form the basis of NASA's next generation of open-source aircraft sizing tools.

AVIARY

- Pushing Dymos development efforts in
  - Analytic phases
  - Performance
  - Training

SPACECRAFT ATTITUDE OPTIMIZATION

- Given a 3DOF reference trajectory, minimize reaction wheel inputs to
  - Maintain pointing necessary for 3DOF solution
- Using Sympy to develop vehicle 6DOF equations of motion from first principles.
- Sympy-to-OpenMDAO Component is currently a custom script
  - Would the community use a more canonical OpenMDAO solution to interface with Sympy?
- Sympy's differentiation can be inefficient

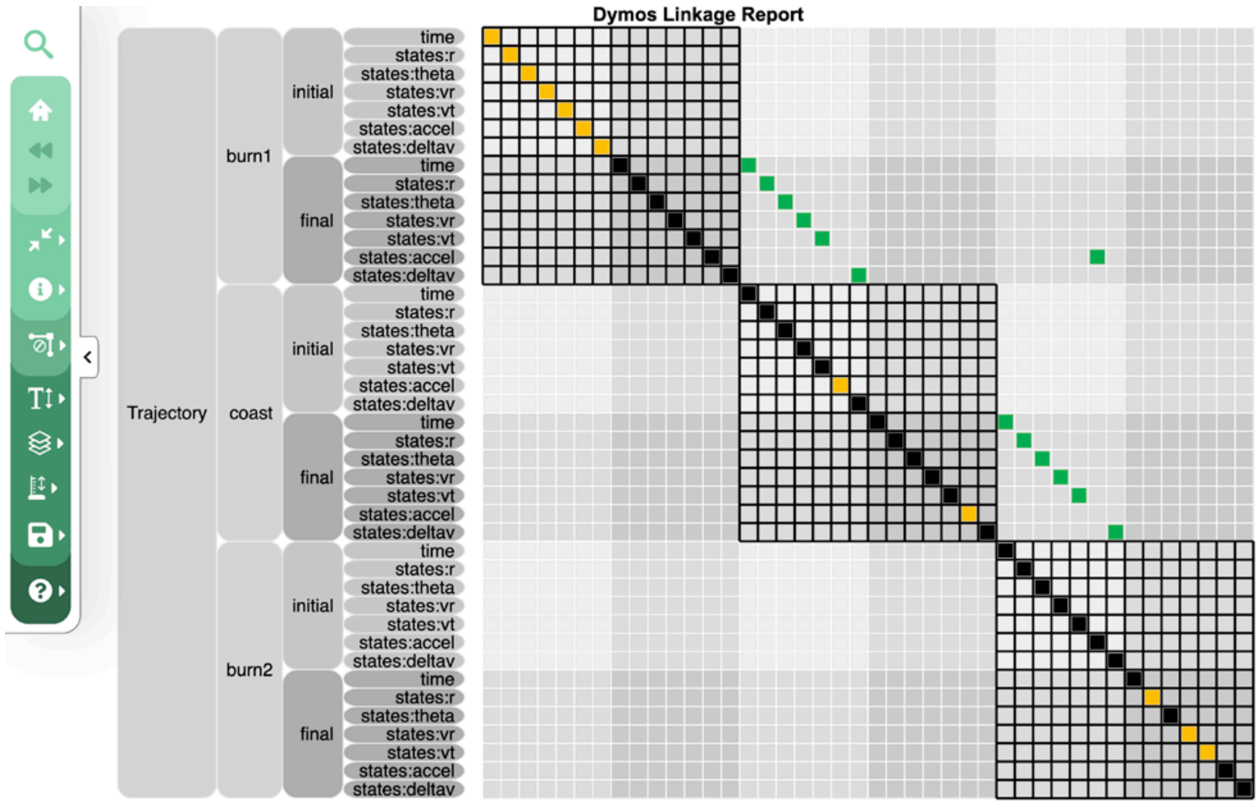
SPACECRAFT ATTITUDE OPTIMIZATION

- Current model
  - 14 state variables
  - 49 inputs
  - 52 constant inputs specified via options
  - 196 partial declarations
- Symbolic "source-code transformation" differentiation via Sympy
- Sympy's naive differentiation is inefficient
  - Some derivative involve 10,000 characters.
  - Repeated calculation of some sin & cos terms
  - Use common subexpression elimination!
    - sympy.cse

FOCUS OF RECENT DEVELOPMENT

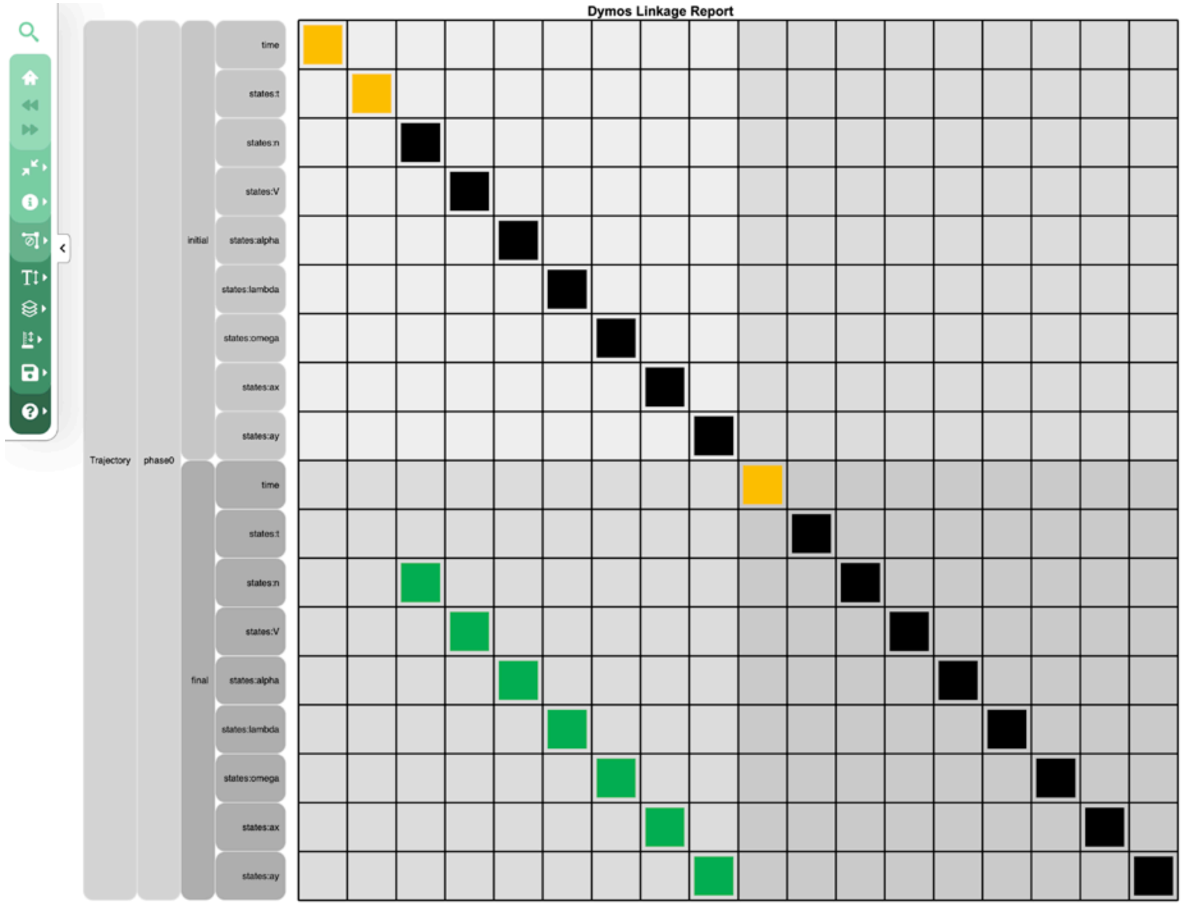
PHASE LINKAGE REPORT

- See how phases are linked via constraint or connection.
- Identify errant linkages between fixed quantities.



LINKAGE REPORT RACECAR PROBLEM

- Example of Linkages for racecar example.
- “Minimize track lap time subject to returning to the same initial point.”



TIMESERIES & CONSTRAINT EXPRESSIONS

- User-defined expressions as timeseries outputs.
  - - `phase.add_timeseries_output(‘mach = sos / v’)`
  - No need to change the ODE system
  - Evaluated by ExecComp

- Expressions can be used as constraints.

```
phase.add_boundary_constraint(‘bc0=x-p’, loc=‘final’, equals=0)
```

```
phase.add_path_constraint(‘pc0 = x - min_x’, lower=0, ref=1000)
```

SHOOTING METHODS

- We may want physical validity without optimization.
  - Get physically valid results with run_model
  - Reduce the degrees of freedom for the optimizer
- `simulate` uses `scipy.integrate.solve_ivp` to propagate trajectories
  - Does not propagate derivatives
- Option `solve_segments` can be used with pseudospectral transcriptions to solve defects with a Newton solver.
  - Subject to grid accuracy issues
  - What if the solver doesn't converge?

EXPLICIT SHOOTING

- The ExplicitShooting transcription provides fixed-step integration.
- Propagates derivatives of final state value in a segment wrt segment initial states, initial time and duration, parameters, and control values.
- No convergence issue
  - Error in a step rather than failing to converge
- Allows for enforcement of path constraints at segment bounds.
- Significantly slower than pseudospectral methods.
- Prone to getting stuck in the design space
  - Local extrema
  - Singularities
- What about grid issues…adaptive step?

ANALYTIC PHASES

- Sometimes you know the solution to your ODE through an analytic expression
  - Breguet Range
  - 2-body orbital motion
- This can be accomplished by a simple OpenMDAO system.
- Allow system with no integrated states in a trajectory.
  - Boundary & Path constraints
  - Phase linkages
  - Variable continuity plots

FUTURE DEVELOPMENT EFFORTS

RESTART IMPROVEMENTS

- Dymos methods work best with initial guesses
  - Previous optimization on a nearby solution
  - Explicit simulation
- Current restart capability interpolates design variables but not implicit outputs within the model
- Find a simple way of mapping phases in a recording file to phases in a run case when there's not a 1:1 mapping.

ADAPTIVE-STEP SHOOTING METHODS

- Use state-transition matrix approach to propagate sensitivities across each segment?
  - Errors with changing step size
  - Change in step size not accounted for in derivatives
    - *On the Accuracy of Trajectory State-Transition Matrices*, Pellegrini and Russell, August 2015
  - Adaptive-step RK methods are non-differentiable
    - $h = 0.9 \cdot h \cdot min\left(\max\left(\sqrt{\left(\frac{tol}{2|\tau_{n+1}|}\right)}, 0.3\right)\right)$
    - $h = min\left(t + h, t_f\right)$
    - Step size *h* is an implicit output of each step

TRAINING

- Develop series of videos on trajectory optimization
  - Build on John's excellent PracticalMDO series
  - Best practices
    - Difference between pseudospectral methods and simulation
    - Importance of parameterization
    - Leveraging sparsity

QUESTIONS?