MDO assignment; v. 1.0

John T. Hwang, Justin S. Gray, and John Jasa

March 21, 2016

1. **Structural optimization.** For this problem, we will use `prob1.py` as our starting point.
   (a) This script performs structural analysis and optimization of a tubular beam clamped in the middle. Run the optimization, first with uniform loading and then again with tip loads applied. What optimized thickness distributions do you see?
   Commands:
      i. run the optimization: `python prob1.py`
      ii. view the results: `python plot_all.py s`
   (b) Run the optimization with tip loads applied for a range of different mesh sizes (num_y). Plot the computation time vs num_y.
   (c) The script produces an html file, `prob1.html`, that can be useful for studying the problem structure. You can open this file in any web-browser. What is the physical interpretation of this problem? That is, what are we minimizing and subject to what constraint?

2. **Multidisciplinary analysis.** We now want to couple aerodynamics and structures together.
   (a) Open `aerostruct.html` to use a guide. Assemble the aerostructural analysis group following the layout presented there. Start with `prob2a.py`, which has the skeleton of the problem layed out for you.
   (b) Try NLGS and Newton, and Hybrid NLGS/Newton for mesh sizes (2,3), (4,5) and compare run times. Why do we put the nonlinear solver on the 'coupled' group, instead of the 'root' group?
   (c) (Bonus) Try LNGS, Krylov, Krylov-PC-GS, and direct linear solvers with the Newton nonlinear solver. Which ones can successfully converge the linear problem? Which one gives the fastest convergence for the Newton solver?

3. **Multidisciplinary optimization.** Now that you've worked with the aerostructural analysis, you're ready to try aerostructural optimization.
   (a) Compute the derivatives of the multidisciplinary system using finite differences by running `prob3a.py`.
   (b) Now compute the same derivatives using the adjoint method and compare the timings for different mesh sizes.
   (c) We will now perform aerostructural optimization. Edit `prob3c.py` by adding the following design variables:
      - 'twist', lower = -10, upper = 10, scaler = 1000
      - 'alpha', lower = -10, upper = 10, scaler = 1000
      - 't', lower = 0.003, upper = 0.025, scaler = 1000

      and the follow objective and constraints:
      - 'fuelburn'
      - 'failure', upper = 0
      - 'eq_con', equals = 0