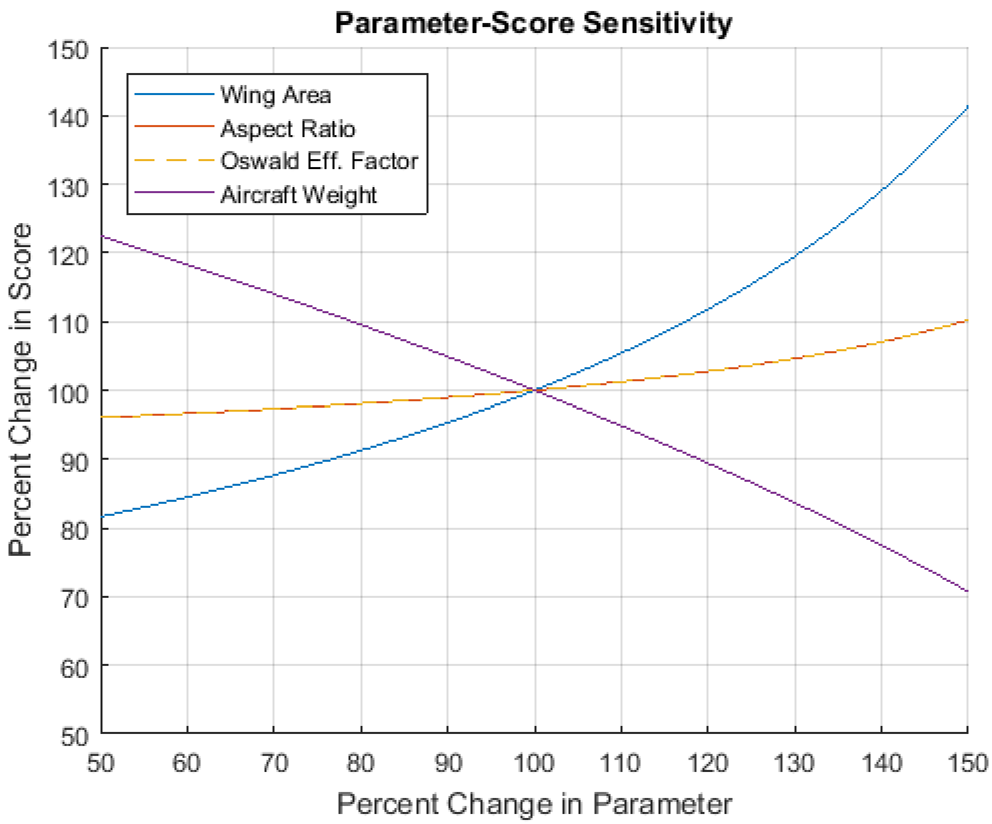
```
% Jeff Houston
% AEE 472 - Synthesis of Aerospace Systems
% Aircraft Conceptual Design
% 2/4/20

close all; clear all; clc

% References
% (1) http://airfoiltools.com/airfoil/details?airfoil=naca2408-il
% (2) https://www.engineeringtoolbox.com/air-absolute-kinematic-viscosity-d_601.html
% (3) https://en.wikipedia.org/wiki/Lifting-line_theory

% getting approximation of aircraft weight based on HBZ 3100 Kit
    W_water  = 1.5;                 % weight of water in rocket [lbs]
    W_rocket = 122/453.59;          % weight of rocket [grams] -> [lbs]
    W_rocket = W_rocket + W_water;  % weight of water & rocket [lbs]

    W_battery = 110/453.59;         % weight of battery [grams] -> [lbs]
    W_HBZ3100 = 735/453.59;         % weight of total HBZ3100 aircraft [grams] -> [lbs]

    W_aircraft = W_rocket + W_HBZ3100;

W_aircraft = 4.00;              % weight of aircraft [lbs]

% assumed characteristics
S  = 5;             % wing area [ft^2]
c  = 10./12;        % chord length [ft]
b = S./c;           % wing span [ft]
AR = b.^2./S;       % aspect ratio
e = 0.8;            % oswald eff. factor

% simulation 1 - bassline
Output = simulate(S, c, b, AR, e, W_aircraft);

P     = Output(1,:)
V_inf = Output(2,:)
score = Output(3,:)

% simulation 2 - change in WING AREA
    S_array = linspace(0.5*S,1.5*S,100);
    Output2 = simulate(S_array,c,b,AR,e,W_aircraft);
    score_2 = Output2(3,:);

parameter_change_2  = S_array./S;
sensitivity_score_2 = score_2./score;

% NOTE:  Chord length was found to leave score unchanged.
% % simulation 3 - change in CHORD LENGTH
%     c_array = linspace(0.5*c,1.5*c,100);
%     Output3 = simulate(S,c_array,b,AR,e,W_aircraft);
%     score_3 = Output3(3,:);
%
```

```
% parameter_change_3  = c_array./c;
% sensitivity_score_3 = score_3./score;

% simulation 4 - change in ASPECT RATIO
    AR_array = linspace(0.5*AR,1.5*AR,100);
    Output4 = simulate(S,c,b,AR_array,e,W_aircraft);
    score_4 = Output4(3,:);

parameter_change_4  = AR_array./AR;
sensitivity_score_4 = score_4./score;

% simulation 5 - change in OSWALD EFFICIENCY FACTOR
    e_array = linspace(0.5*e,1.5*e,100);
    Output5 = simulate(S,c,b,AR,e_array,W_aircraft);
    score_5 = Output5(3,:);

parameter_change_5  = e_array./e;
sensitivity_score_5 = score_5./score;

% simulation 6 - change in OSWALD EFFICIENCY FACTOR
    W_aircraft_array = linspace(0.5*W_aircraft,1.5*W_aircraft,100);
    Output6 = simulate(S,c,b,AR,e,W_aircraft_array);
    score_6 = Output6(3,:);

parameter_change_6  = W_aircraft_array./W_aircraft;
sensitivity_score_6 = score_6./score;

% plot sensitivity results
figure(1)
hold on
plot(parameter_change_2.*100,fliplr(sensitivity_score_2).*100)
plot(parameter_change_4.*100,fliplr(sensitivity_score_4).*100)
plot(parameter_change_5.*100,fliplr(sensitivity_score_5).*100,'--')
plot(parameter_change_6.*100,fliplr(sensitivity_score_6).*100)
hold off
grid on
axis([50 150 50 150])
title('Parameter-Score Sensitivity')
xlabel('Percent Change in Parameter')
ylabel('Percent Change in Score')
legend('location','northwest','Wing Area','Aspect Ratio','Oswald Eff. Factor','Aircraft Weight')

function Output = simulate(S, c, b, AR, e, W_aircraft)
% This function executes a simulation to output key parameters in
% determining their respective sensitivities.

% known values
    g     = 32.2;       % acceleration due to gravity [ft/s^2]
    rho_a = 2.471e-3;   % density of air @ 39ºF [slug/ft^3]
    mu_a  = 3.622e-6;   % dynamic viscosity of air @ 39ºF [lbf-s/ft^2]

% NACA 2408 Airfoil (from Ref. 1:  |CL/CD|max=37.4 at ?=5.5° for Re ~= 500,000)
%   NOTE: input different |CL/CD|max for varying Re, until Re corresponds
```

```
%   to the one in Ref. 1
    alpha     = 5.50;       % angle of attack [°]
    CL_CD_max = 37.4;       % |CL/CD|max at given alpha

    C_l_alpha = (0.65--0.30)/(4.0--4.0); % Clalpha from Ref. 1 [deg^-1]
    C_l_0     = 0.10;                    % Cl0 from Ref. 1 [deg^-1]
    C_L_alpha = C_l_alpha./(1+(57.3.*C_l_alpha./(pi.*e.*AR)));  % CLalpha [deg^-1]
    C_L_0 = C_l_0;                       % CL-alpha graph has same intercept as Cl-alpha graph
C_L = C_L_alpha.*alpha + C_L_0;          % CL

% for SLUF
V_inf = (2.*W_aircraft./(rho_a.*S.*C_L)).^0.5;   % freestream velocity [ft/s]
Re = rho_a.*V_inf.*c./mu_a;                      % Reynolds Number

    C_D_i = C_L.^2./(pi.*e.*AR);       % induced drag coeff.
    C_D_0 = C_D_i;                     % zero lift drag coeff. CD0=CDi for max range
C_D   = C_D_0 + C_D_i;                 % drag coeff.
D = 1/2*rho_a.*V_inf.^2.*S.*C_D;       % drag force [lbf]
L = W_aircraft;                        % lift force (equal to W for SLUF) [lbf]
    L_check = 1/2*rho_a.*V_inf.^2.*S.*C_L;   % double check to make sure L=W

% while it is known that P = TV, T = D for SLUF
P = D.*V_inf;       % power used during flight [ft-lb/s]
%     P = P./550      % [ft-lb/s] -> [hp]

% assuming at baseline conditions the aircraft can fly for 3 minutes
t_flight = 3*60;
% E = t_flight.*P;

d = t_flight.*V_inf;                % distance traveled [ft]
R = 15;                             % turn radius [ft]
d_lap = 30 + pi*R + 60 + pi*R +30;  % distance for approx. one lap [ft]
laps = d./d_lap;

score = laps;


Output = [P ; V_inf ; score];
end
```

```
P =

    8.5279


V_inf =

   33.5678


score =
```

28.2020



Published with MATLAB® R2017a