Problem 2

```
function [gammaeq,Teq,alphaeq,phieq,XdotSM,YdotSM,iflagterm,niter] = ...
             solvesteadystateaircraft01(Zeq,Veq,psieq,m,S,CLalpha,...
                                        CD0,oneoverpiARe,...
                                        nitermax)
%
%  Copyright (c) 2019 Mark L. Psiaki.  All rights reserved.
%
%  This function determines for the flight-path angle,
%  thrust, angle-of-attack, and roll angle that produce
%  steady, level, straight-line motion of a point-mass
%  3-dimensional aircraft that is operating over
%  a flat non-rotating Earth (i.e., without Coriolis effects
%  due to the Earth's rotation, without centrifugal
%  effects due to the Earth's rotation other than the
%  mean centrifugal effect at the origin of the coordinate
%  system, and with a constant gravitational acceleration).
%  While it uses a flat-Earth (i.e., constant) gravity field,
%  its constant gravity takes into account the Earth's J2
%  oblatness effect at the coordinate system center and
%  it subtracts off the centrifugal acceleration of the
%  coordinate system center as caused by the Earth's rotation.
%
%  This function works by reducing the problem to a
%  single equation in the single unknown value of
%  angle of attack, alphaeq.  This single nonlinear
%  equation is solved iteratively using Newton's
%  method.  Once the angle of attack has been
%  determined, the thrust Teq can be computed directly.
%  The flight-angle gammaeq and the roll angle
%  phieq are obvious.
%
%  This function also computes the steady-motion rates
%  of change of the northward displacement X and the
%  and eastward displacement Y, which are XdotSM
%  and YdotSM.
%
%
%
%  Inputs:
%
%    Zeq                   The steady-motion vertical
%                          displacement, in meters,
%                          relative to the origin of the
%                          local level coordinate system.
%                          A positive value is down.  This
%                          coordinate system is centered
%                          at the runway of the Blacksburg,
%                          VA, airport, which is at an
%                          altitude of 649.7 m above sea
%                          level.  Therefore, (-Zeq + 649.7)
%                          is the steady-motion aircraft
%                          altitude above sea level.
%
%    Veq                   The steady-motion aircraft
%                          velocity in meters/second.
```

```
%
%   psieq                The steady-motion aircraft
%                        heading angle in meters.
%                        psieq = 0 radians is due
%                        north (i.e., along the +X
%                        local-level axis) and
%                        psieq = +pi/2 radians is
%                        due east (i.e., along the
%                        +Y axis).
%
%   m                    The aircraft mass in kg.
%
%   S                    The wing area, in meters^2, which is
%                        the aerodynamic model's reference area.
%
%   CLalpha              The lift curve slope, dCL/dalpha, which
%                        is non-dimensional.
%
%   CD0                  The drag at zero lift, which is non-
%                        dimensional.
%
%   oneoverpiARe         = 1/(pi*AR*e), where AR is the non-
%                        dimensional aspect ratio of the wing
%                        and e is the Oswald efficiency factor.
%                        This composite input quantity is non-
%                        dimensional.  It is the coefficient
%                        of CL^2 in the drag coefficient model.
%
%   nitermax             The maximum number of Newton's
%                        method iterations that will be
%                        allowed before the algorithm
%                        quits with an error condition if
%                        it has not reached convergence
%                        prior to executing this many
%                        iterations.  A conservative value
%                        for this limit is 50.
%
%  Outputs:
%
%   gammaeq              The steady-motion flight-path
%                        angle in radians.  This will
%                        equal 0.
%
%   Teq                  The steady-motion thrust, in Newtons.
%
%   alphaeq              The steady-motion angle of attack,
%                        in radians.
%
%   phieq                The steady-motion roll angle,
%                        in radians.  This will equal 0.
%
%   XdotSM               The steady-motion northward component
%                        of velocity, in meters/second.
%
%   YdotSM               The steady-motion eastward component
%                        of velocity, in meters/second.
%
%   iflagterm            A termination status flag that
%                        inticates whether the Newton's
```

```
%                          method solution for alphaeq has
%                          converged.  Its possible values and
%                          their meanings are:
%
%                            0  Normal successful termination.
%
%                            1  Failure to converge in nitermax
%                               Newton iterations.  A warning
%                               message will be sent to the
%                               display in this case.
%
%    niter                 The number of Newton iterations that
%                          have been executed in the attempt
%                          to solve for alphaeq.
%


%
%  Set up the steady-motion flight-path angle and roll angle.
%
   gammaeq = 0;
   phieq = 0;
%
%  Compute the steady-motion northward and eastward velocities
%
   Veq_cosgammaeq = Veq*cos(gammaeq);
   XdotSM = Veq_cosgammaeq * cos(psieq);
   YdotSM = Veq_cosgammaeq * sin(psieq);
%
%  Compute the air density using a decaying exponential
%  model.  This model is good to about 1500 m altitude
%  (about 5000 ft).  This model recognizes that -Zeq + 649.7
%  is the aircraft altitude above sea level in meters.
%
   rho_sealevel = 1.225; % kg/m^3
   hscale = 10230.;      % meters
   rho = rho_sealevel*exp((Zeq - 649.7)/hscale); % kg/m^3
%
%  Set the flat-Earth gravitational acceleration at the
%  Blacksburg airport minus the effects of centrifugal
%  acceleration at the Blacksburg airport due to the
%  Earth's rotation vector.
%
   g = 9.79721; % meters/second^2
%
%  Determine the dynamic pressure.
%
   qbar = 0.5*rho*Veq*Veq;
%
%  Compute the product of qbar and S.
%
   qbar_S = qbar*S;
%
%  Compute the constant term in the equation that
%  will be solved using Newton's method in order to
%  determine alphaeq.  It takes the form:
%
%    0 = f(alphaeq) = tan(alphaeq)*CD(alphaeq) + CL(alphaeq) - C0
%
%  where C0 = 2*m*g/(rho*(Veq^2)*S) = m*g/(qbar*S);
```

```
%
   C0 = m*g/qbar_S;
%
%  Initialize the guess of the steady-motion angle of attack
%  at zero.
%
   alphaeq = 0;
%
%  Initialize iflagterm at its nominal successful-case
%  value and initialize niter.
%
   iflagterm = 0;
   niter = 0;
%
%  This is the loop that performs one Newton's method
%  (also known as the Netwon-Raphson method for
%  a scalar equation in a single unknown) iteration
%  towards a better guess of alphaeq.  It also tests
%  whether the guess is close enough to stop
%  the iterations.
%
   testdone0 = 0;
   while testdone0 == 0
      niterp1 = niter + 1;
      if niterp1 > nitermax
         iflagterm = 1;
         disp('Warning in solvesteadystateaircraft01.m.: Newton''s')
         disp([' method did not converge in ',int2str(nitermax),...
               'iterations.'])
         break
      end
      niter = niterp1;
%
      CL = CLalpha*alphaeq;
      CD = CD0 + CL^2*(oneoverpiARe);
      tan_alphaeq = tan(alphaeq);
      f = tan_alphaeq*CD+CL-C0;
      dCL_dalpha = CLalpha;
      dCD_dalpha = 2*CLalpha*CLalpha*alphaeq*oneoverpiARe;
      dtan_alphaeq_dalpha = 1 + tan_alphaeq^2;
      df_dalpha = tan_alphaeq*dCD_dalpha+dtan_alphaeq_dalpha*CD+dCL_dalpha;
      deltaalphaeq = -f/df_dalpha;
      alphaeq = alphaeq + deltaalphaeq;
%
%  Test for convergence.
%
      if abs(deltaalphaeq) <= 1.e-10
         testdone0 = 1;
      end
   end
%
%  Compute the steady-motion thrust.
%
   CL = CLalpha*alphaeq;
   CD = CD0 + CL^2*(oneoverpiARe);
   D = qbar_S*CD;
   Teq = D/cos(alphaeq);
```

Published with MATLAB® R2019a

Output:

gammaeq	0
Teq	4.829888745867086e+03
alphaeq	0.052510385639473
phieq	0
XdotSM	84.264888743087582
YdotSM	-70.706637065519317