P321A_Tutorial_1_Solution

September 23, 2018

1 Problem C 2.1 of Fowles and Cassiday

A parachutist of mass $m = 70$ kg jumps from a plane at an altitude of $y_0 = 32$ km above the surface of the Earth. Unfortunately, the parachute fails to open. (In the folowing parts, neglect horizontal motion and assume that the initial velocity is zero.)

(**a**) Calculate the time of fall (acurate to 1 s) until ground impact, given no air resistance and a constant value of **g**.

(**b**) Calculate the time of fall (acurate to 1 s) until ground impact, given constant g and a force of air resistance given by $F(\mathbf{v}) = -c_2\mathbf{v}|\mathbf{v}|$, where c_2 is 0.5 in SI units for a falling man and is constant.

(**c**) Calculate the time of fall (acurate to 1 s) until ground impact, given c_2 scales with atmospheric density as $c_2 = 0.5e^{-y/H}$, where $H = 8$ km is the scale height of the atmosphere and y is the height above ground.

Furthermore, assume that g is no longer constant but is given by $g = \frac{9.8}{(1+\frac{y}{R_\mathrm{e}})^2}\ \mathrm{m\,s^{-2}}$, where R_e is the radius of the Earth and is 6370 km.

(**d**) For case (c), plot the acceleration, velocity, and altitude of the parachutist as a function of time. Explain why the acceleration becomes positive as the parachutist falls.

1.0.1 The equation of motion (Newton's second law)

$\frac{d\mathbf{p}}{dt} = m\frac{d\mathbf{v}}{dt} = \mathbf{F} = \mathbf{F}_\mathrm{g} + \mathbf{F}_\mathrm{d} = m\mathbf{g} - c_2\mathbf{v}|\mathbf{v}|$

The coordinate y axis is directed vertically up and measures the height above ground. In this case, the only non-zero projection of the equation of motion on the Cartesian coordinate axes is that on the y axis, which is

$m\ddot{y} = -mg - c_2\dot{y}\,|\dot{y}|$,

or

$\ddot{y} = a + b\dot{y}\,|\dot{y}|$,

where $a = -g$ and $b = -\frac{c_2}{m}$.

The last second-order ODE can be written as a system of two first-order ODEs as

$$\dot{y_1} = y_2,\ \ \dot{y_2} = a + by_2|y_2|,$$

where $y_1 = y$, with the following initial conditions: $y_1(0) = 32000$ (m) and $y_2(0) = 0$.

In [1]:
```
# populate the interactive namespace with functions
# from the modules numpy and matplotlib
%pylab nbagg
```

```
Populating the interactive namespace from numpy and matplotlib
```

```
In [2]: # the following commands allow to produce
        # a nice pdf version of the notebook with figures:
        from matplotlib.pyplot import *
        %matplotlib inline

        from IPython.display import set_matplotlib_formats
        set_matplotlib_formats('png', 'pdf')
```

```
In [3]: # the function defining the right-hand sides (RHS) of the ODEs
        # for cases (a) and (b)
        def dy_dt(y, t, a, b):
            y1, y2 = y
            dydt = [y2, a + b*y2*abs(y2)]
            return dydt
```

```
In [4]: # the parameters of the problem that are common for all the cases
        g0 = 9.8
        c2 = 0.5
        m = 70.
        Re = 6370000.
        H = 8000.

        # the initial conditions
        y0 = [32000., 0.]
```

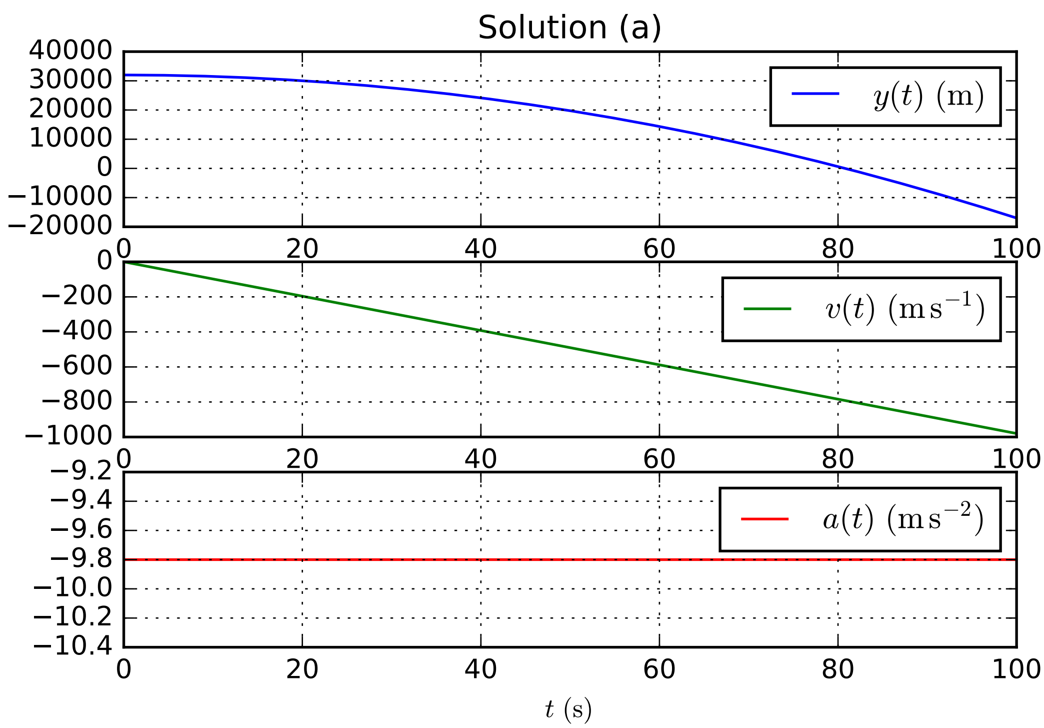
```
In [5]: # populate the interactive namespace with the function
        # that solves ODEs
        from scipy.integrate import odeint
```

1.1 Solution (a)

```
In [6]: g = g0
        a = -g
        b = 0. # c2 = 0. because there is no air resistance in this case

        # the integration time interval
        t_start = 0.
        t_end = 100. # adjust t_end, using the plot below,
        # such that parachutist's final height is negative
        nt = 10000 # the number of integration time steps

        t = linspace(t_start, t_end, nt)
```

```
In [7]: # solve the ODEs
        # the coefficients a and b are defined above
        sol = odeint(dy_dt, y0, t, args=(a, b))
```

In [8]:
```
# plot the solution
#figure(figsize=(6,8))  # this command will increase the figure size

subplot(3,1,1)
plot(t, sol[:, 0], 'b', label='$y(t)\ (\mathrm{m})$')
legend(loc='best')
xlabel('$t\ (\mathrm{s})$')
grid()
title("Solution (a)")

subplot(3,1,2)
plot(t, sol[:, 1], 'g', label='$v(t)\ (\mathrm{m\,s}^{-1})$')
legend(loc='best')
xlabel('$t\ (\mathrm{s})$')
grid()

subplot(3,1,3)
plot(t, a+b*sol[:, 1]*abs(sol[:, 1]), 'r', \
     label='$a(t)\ (\mathrm{m\,s}^{-2})$')
legend(loc='best')
xlabel('$t\ (\mathrm{s})$')
grid()

show()
```



```
In [9]: # populate the intercative namespace with the function
        # that makes 1d interpolation
        from scipy.interpolate import interp1d
```

```
In [10]: # interpolate the solution of the ODEs to find
         # its independent variable's value (the time t)
         # at the surface, where the height is sol[:,0]=0.

         f = interp1d(sol[:,0],t)
         print ("In case (a) the falling time is",f(0.),"s")
```

```
In case (a) the falling time is 80.81220341423473 s
```

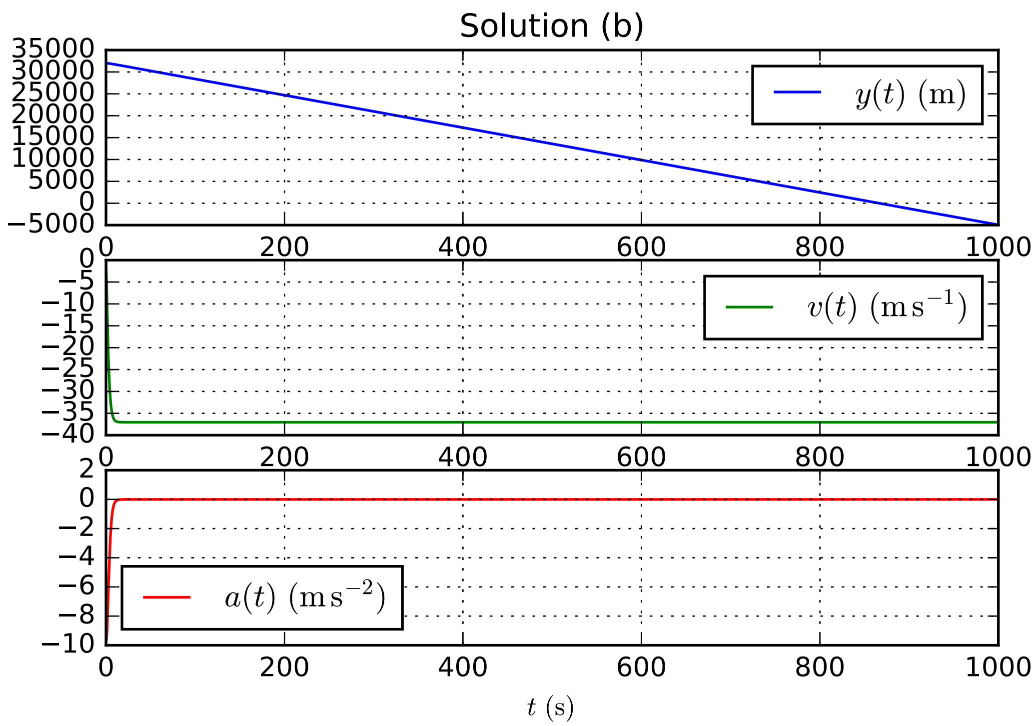
1.2 Solution (b)

```
In [11]: g = g0
         a = -g
         b = -c2/m

         # the integration time interval
         t_start = 0.
         t_end = 1000. # adjust t_end, using the plot below,
         # such that parachutist's final height is negative
         nt = 10000 # the number of integration time steps

         t = linspace(t_start, t_end, nt)
```

```
In [12]: # solve the ODEs
         # the coefficients a and b are defined above
         sol = odeint(dy_dt, y0, t, args=(a, b))
```

```
In [13]: # plot the solution
         #figure(figsize=(6,8))  # this command will increase the figure size

         subplot(3,1,1)
         plot(t, sol[:, 0], 'b', label='$y(t)\ (\mathrm{m})$')
         legend(loc='best')
         xlabel('$t\ (\mathrm{s})$')
         grid()
         title("Solution (b)")

         subplot(3,1,2)
         plot(t, sol[:, 1], 'g', label='$v(t)\ (\mathrm{m\,s}^{-1})$')
         legend(loc='best')
         xlabel('$t\ (\mathrm{s})$')
         grid()

         subplot(3,1,3)
```

```
plot(t, a+b*sol[:, 1]*abs(sol[:, 1]), 'r', \
     label='$a(t)\ (\mathrm{m\,s}^{-2})$')
legend(loc='best')
xlabel('$t\ (\mathrm{s})$')
grid()

show()
```




```
In [14]: # interpolate the solution of the ODEs to find
         # its independent variable's value (the time t)
         # at the surface, where the height is sol[:,0]=0.

         f = interp1d(sol[:,0],t)
         print ("In case (b) the falling time is",f(0.),"s")

         # this interpolation finds the numerical value of terminal
         # velocity at ground and compares it with the analytical value
         u = interp1d(sol[:,0],sol[:,1])
         print ("In case (b) the terminal velocity is",u(0.),"m/s")
         print ("and its analytical value is",-sqrt(m*g/c2),"m/s")
```

```
In case (b) the falling time is 866.5386447171986 s
In case (b) the terminal velocity is -37.04051835490428 m/s
and its analytical value is -37.0405183549 m/s
```

1.3 Solution (c)

The dependence of the gravitational acceleration on the height is

$g(y) = \frac{g(0)}{[1+(\frac{y}{R_e})]^2}$,

where $g(0) = 9.8$ and $R_e = 6370000$.

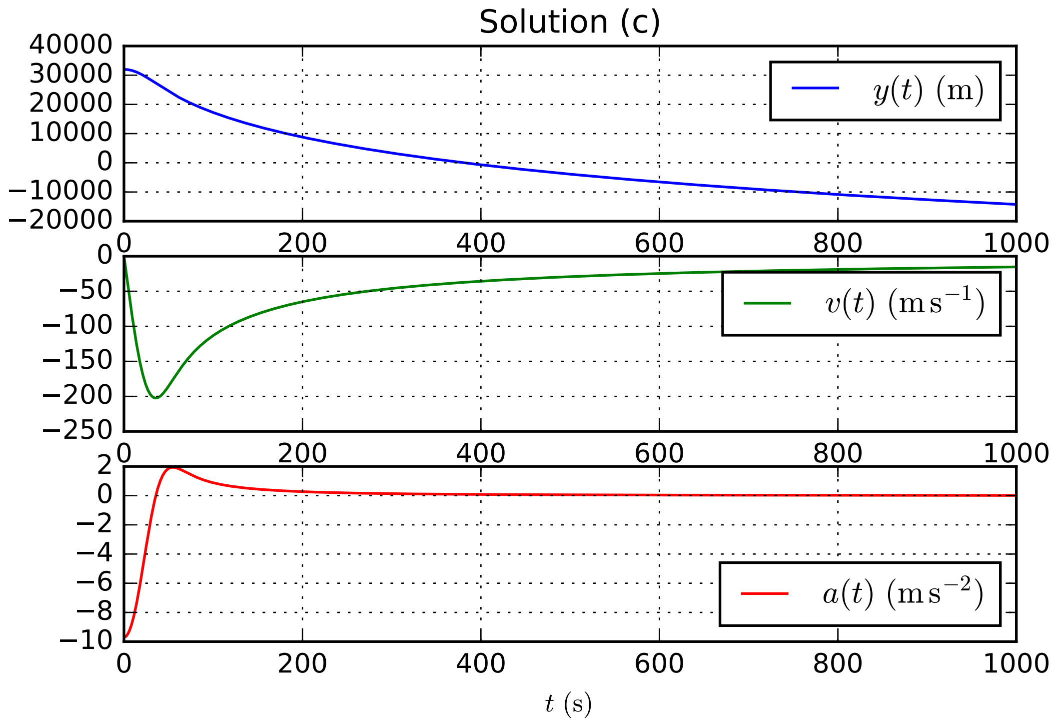
In [15]:
```
g = g0
a = -g
b = -c2/m

# the integration time interval
t_start = 0.
t_end = 1000. # adjust t_end, using the plot below,
# such that parachutist's final height is
# beneath the surface (less than 0.)
nt = 10000 # the number of integration time steps

t = linspace(t_start, t_end, nt)
```

In [16]:
```
# the right-hand sides (RHS) of the system of ODEs for case (c)
def dy_dt(y, t, a, b):
    y1, y2 = y
    # the next equation takes into account the dependence of
    # c2 on y and the dependence of g on h
    dydt = [y2, a/(1.+(y1/Re))**2 + b*exp(-y1/H)*y2*abs(y2)]
    return dydt
```

In [17]:
```
# solve the ODEs
# the coefficients a and b are defined above
sol = odeint(dy_dt, y0, t, args=(a, b))
```

In [18]:
```
# plot the solution
#figure(figsize=(6,8))  # this command will increase the figure size

subplot(3,1,1)
plot(t, sol[:, 0], 'b', label='$y(t)\ (\mathrm{m})$')
legend(loc='best')
xlabel('$t\ (\mathrm{s})$')
grid()
title("Solution (c)")

subplot(3,1,2)
plot(t, sol[:, 1], 'g', label='$v(t)\ (\mathrm{m\,s}^{-1})$')
legend(loc='best')
xlabel('$t\ (\mathrm{s})$')
grid()

subplot(3,1,3)
plot(t, a/(1.+(sol[:,0]/Re))**2 + b*exp(-sol[:,0]/H)*sol[:,1]*\
```

```
     abs(sol[:,1]),'r',label='$a(t)\ (\mathrm{m\,s}^{-2})$')
legend(loc='best')
xlabel('$t\ (\mathrm{s})$')
grid()

show()
```




```
In [19]: # interpolate the solution of the ODEs to find
         # its independent variable's value (the time t)
         # at the surface, where the height is sol[:,0]=0.

         f = interp1d(sol[:,0],t)
         print ("In case (c) the falling time is",f(0.),"s")

In case (c) the falling time is 381.19044512777765 s
```

1.4 Solution (d)

Case (c) has a surprising result: acceleration becomes positive as parachutist falls. Why? Look at Newton's 2nd law:
$\ddot{y} = -g(y) - \frac{c_2}{m}\dot{y}|\dot{y}|$.

Air resistance gives a positive contribution to the acceleration; as velocity increases, the positive contribution increases faster than $g(y)$, and as a result the overall acceleration becomes positive for some time interval. It is quite interesting that this only happens when g is position dependent.

```
In [ ]:
```