

```

1  """
2  Start along z-axis (normal to surface)
3  with cross wind (along x-axis):
4  dx(t)/dt = Vx(t)
5  dVx(t)/dt = Fwind/m + Frv(V)*Vx/m
6  dz(t)/dt = Vz(t)
7  dVz(t)/dt = -g + Frv(V)*Vz/m
8  """
9
10 import numpy as np
11 from scipy.integrate import odeint
12 import matplotlib.pyplot as plt
13
14 x0 = 0.0 # m
15 Vx0 = 0.0 # m/sec
16 z0 = 0.0 # m
17 Vz0 = 500.0 # m/sec
18 m = 0.009 # kg
19 g = 9.8 # m/sec^2
20 A = 1.e-5 # N*sec/m
21 B = 1.e-8 # N*sec^3/m^3
22 Fwind = 0.01 # N (force of cross wind along x-axis)
23 tm = 110.0 # sec
24
25
26 def Frv(V):
27     global A, B
28     # minus because of resistance force
29     # in the opposite direction of velocity
30     return -(A*V + B*V**3)/V
31
32
33 def system(f, t):
34     global m, g, A, B, Fwind
35     x = f[0]
36     Vx = f[1]
37     z = f[2]
38     Vz = f[3]
39     V = np.sqrt(Vx**2 + Vz**2)
40     dxdt = Vx
41     dVxdt = Fwind/m + Frv(V)*Vx/m
42     dzdt = Vz
43     dVzdt = -g + Frv(V)*Vz/m
44     return [dxdt, dVxdt, dzdt, dVzdt]
45
46 nt = 1000
47 t = np.linspace(0., tm, nt)

```

```
48 sol = odeint(system, [x0, Vx0, z0, Vz0], t)
49 x = sol[:, 0]
50 Vx = sol[:, 1]
51 z = sol[:, 2]
52 Vz = sol[:, 3]
53
54 print("len(z)=", len(z))
55
56 # Simple calculation of Tflight
57 for i in range(len(z)):
58     if z[i] < 0.0:
59         Tflight = (t[i]+t[i-1])/2.0
60         numnode = i
61         print("Node of landing:", numnode)
62         print("Tflight=", Tflight)
63         break
64
65 tmax = round(Tflight+0.5)
66 print("tmax=", tmax)
67 print("t[numnode]=", t[numnode])
68 print("x[numnode]=", x[numnode])
69 print("z[numnode]=", z[numnode])
70 print("Vx[numnode]=", Vx[numnode])
71 print("Vz[numnode]=", Vz[numnode])
72
73 plt.plot(t, Vx, 'r-', linewidth=3)
74 plt.plot(t, [0.0]*nt, 'g-', linewidth=1)
75 plt.plot([Tflight], [Vx[numnode]], 'bo')
76 plt.axis([0, tmax, 0., 40.])
77 plt.grid(True)
78 plt.xlabel("t")
79 plt.ylabel("Vx(t)")
80 plt.savefig("Vx.pdf", dpi=300)
81 plt.show()
82
83 plt.plot(t, x, 'b-', linewidth=3)
84 plt.axis([0, tmax, 0., 1400.])
85 plt.grid(True)
86 plt.xlabel("t")
87 plt.ylabel("x(t)")
88 plt.savefig("x.pdf", dpi=300)
89 plt.show()
90
91 plt.plot(t, Vz, 'r-', linewidth=3)
92 plt.plot(t, [0.0]*nt, 'g-', linewidth=1)
93 plt.axis([0, tmax, -250., 500.])
94 plt.grid(True)
```

```
95 plt.xlabel("t")
96 plt.ylabel("Vz(t)")
97 plt.savefig("Vz.pdf", dpi=300)
98 plt.show()
99
100 plt.plot(t, z, 'b-', linewidth=3)
101 plt.axis([0, tmax, 0., 3500.])
102 plt.grid(True)
103 plt.xlabel("t")
104 plt.ylabel("z(t)")
105 plt.savefig("z.pdf", dpi=300)
106 plt.show()
107
108 xx = x[:numnode]
109 zz = z[:numnode]
110 print("len(xx)=", len(xx))
111
112 plt.plot(xx, zz, 'orangered', linewidth=5)
113 plt.axis([0, 1400, 0., 3500.])
114 plt.grid(True)
115 plt.title("Trajectory")
116 plt.xlabel("x")
117 plt.ylabel("z")
118 plt.savefig("trajectory.pdf", dpi=300)
119 plt.show()
120
```