```
"""
Start along z-axis (normal to surface)
with cross wind (along x-axis):
dx(t)/dt = Vx(t)
dVx(t)/dt = Fwind/m + Frv(V)*Vx/m
dz(t)/dt = Vz(t)
dVz(t)/dt = -g + Frv(V)*Vz/m
"""

import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

x0 = 0.0        # m
Vx0 = 0.0       # m/sec
z0 = 0.0        # m
Vz0 = 500.0     # m/sec
m = 0.009       # kg
g = 9.8         # m/sec^2
A = 1.e-5       # N*sec/m
B = 1.e-8       # N*sec^3/m^3
Fwind = 0.01    # N (force of cross wind along x-axis)
tm = 110.0      # sec


def Frv(V):
    global A, B
    # minus because of resistance force
    # in the opposite direction of velocity
    return -(A*V + B*V**3)/V


def system(f, t):
    global m, g, A, B, Fwind
    x = f[0]
    Vx = f[1]
    z = f[2]
    Vz = f[3]
    V = np.sqrt(Vx**2 + Vz**2)
    dxdt = Vx
    dVxdt = Fwind/m + Frv(V)*Vx/m
    dzdt = Vz
    dVzdt = -g + Frv(V)*Vz/m
    return [dxdt, dVxdt, dzdt, dVzdt]

nt = 1000
t = np.linspace(0., tm, nt)
```

```
sol = odeint(system, [x0, Vx0, z0, Vz0], t)
x = sol[:, 0]
Vx = sol[:, 1]
z = sol[:, 2]
Vz = sol[:, 3]

print("len(z)=", len(z))

# Simple calculation of Tflight
for i in range(len(z)):
    if z[i] < 0.0:
        Tflight = (t[i]+t[i-1])/2.0
        numnode = i
        print("Node of landing:", numnode)
        print("Tflight=", Tflight)
        break

tmax =round(Tflight+0.5)
print("tmax=", tmax)
print("t[numnode]=", t[numnode])
print("x[numnode]=", x[numnode])
print("z[numnode]=", z[numnode])
print("Vx[numnode]=", Vx[numnode])
print("Vz[numnode]=", Vz[numnode])

plt.plot(t, Vx, 'r-', linewidth=3)
plt.plot(t, [0.0]*nt, 'g-', linewidth=1)
plt.plot([Tflight], [Vx[numnode]], 'bo')
plt.axis([0, tmax, 0., 40.])
plt.grid(True)
plt.xlabel("t")
plt.ylabel("Vx(t)")
plt.savefig("Vx.pdf", dpi=300)
plt.show()

plt.plot(t, x, 'b-', linewidth=3)
plt.axis([0, tmax, 0., 1400.])
plt.grid(True)
plt.xlabel("t")
plt.ylabel("x(t)")
plt.savefig("x.pdf", dpi=300)
plt.show()

plt.plot(t, Vz, 'r-', linewidth=3)
plt.plot(t, [0.0]*nt, 'g-', linewidth=1)
plt.axis([0, tmax, -250., 500.])
plt.grid(True)
```

```
plt.xlabel("t")
plt.ylabel("Vz(t)")
plt.savefig("Vz.pdf", dpi=300)
plt.show()

plt.plot(t, z, 'b-', linewidth=3)
plt.axis([0, tmax, 0., 3500.])
plt.grid(True)
plt.xlabel("t")
plt.ylabel("z(t)")
plt.savefig("z.pdf", dpi=300)
plt.show()

xx = x[:numnode]
zz = z[:numnode]
print("len(xx)=", len(xx))

plt.plot(xx, zz, 'orangered', linewidth=5)
plt.axis([0, 1400, 0., 3500.])
plt.grid(True)
plt.title("Trajectory")
plt.xlabel("x")
plt.ylabel("z")
plt.savefig("trajectory.pdf", dpi=300)
plt.show()

```