```python
"""
Movement along z-axis (normal to surface)
dz(t)/dt = v(t)
dv(t)/dt = -g - (A*v(t) + B*v(t)^3)/m
"""

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

z0 = 0.0       # m
v0 = 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
tm = 110.0     # sec


```

```python
def system(f, t):
    global m, g, A, B
    z = f[0]
    v = f[1]
    dzdt = v
    dvdt = -g - (A*v + B*v**3)/m
    return [dzdt, dvdt]

nt = 1000
t = np.linspace(0., tm, nt)
sol = odeint(system, [z0, v0], t)
z = sol[:, 0]
v = sol[:, 1]

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
            print("Node of landing:", i)
            print("Tflight=", Tflight)
            break

plt.plot(t, v, 'r-', linewidth=3)
plt.plot(t, [0.0]*nt, 'g-', linewidth=1)
plt.axis([0, Tflight+1, -250., 500.])
plt.grid(True)
plt.xlabel("t")
plt.ylabel("v(t)")
plt.savefig("v.pdf", dpi=300)
plt.show()

plt.plot(t, z, 'b-', linewidth=3)
plt.axis([0, Tflight+1., 0., 3500.])
plt.grid(True)
plt.xlabel("t")
plt.ylabel("z(t)")
```

```
plt.savefig("z.pdf", dpi=300)
plt.show()
```