AE441A : Rocket Propulsion

DEPARTMENT OF AEROSPACE ENGINEERING
Indian Insitute of Technology Kanpur

Assignment 1

Instructor: Sathesh Mariappan

Submitted By:
Ankit Lakhiwal (180102)

ankitl@iitk.ac.in

Q. Plot the rocket trajectory (horizontal (x) vs. vertical (h) distance), rocket speed (u vs. t), rocket angle (θ vs. t) and rocket height (h vs. t) until the burn out time (t = tb). Also tabulate the burnout height (hb), burnout speed (ub), and angle of rocket at burnout (θb).

The rocket is fired from the ground (at t = 0 : x, h = 0) at an angle of 1 degree from the vertical (θ = 1 degree) with a non-zero initial vertical velocity 30 m/s. Given: constant equivalent exhaust velocity ueq = 3048 m/s, initial rocket mass (M0) = 15000 kg, propellant mass (Mp) = 12000 kg, burnout time (tb) = 100 s, assume constant mass burning rate (`m).

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
def trajectory(isDragVary,isGravityVary,isquestionE,dt):
    # initial rocket mass(kg)
    Mi = 15000
    # propellant mass(Kg)
    Mp = 12000
    # burnout mass(Kg)
    Mb = Mi - Mp
    # initial height(m)
    hi = 0
    # initial horizontal velocity(m/s)
    uxi = 0
    # initial vertical velocity(m/s)
    uyi = 30
    # initial velocity(m/s)
    ui = np.sqrt(pow(uxi,2) + pow(uyi,2))
    # equivalent velocity(m/s)
    ueq = 3048
    # initial time(sec)
    ti = 0
    # burnout time(sec)
    tb = 100
    # exhaust mass flow(Kg/s)
    mdot = -((Mb-Mi)/tb)
    # acceleration due to gravity at groung level (m/s^2)
    g0 = 9.81
    # theta from vertical
    thetai = 1*(np.pi/180)
    # earth's radius (m)
    Re = 6400000
    # # frontal cross sectional area (m2)
    Af = 1
    # gas constant (J/Kg?K)
    R = 287
    # specific heat constant
    gama = 1.4

    M_old = Mi
    h_old = hi
    u_old = ui
```

```
    theta_old = thetai
    t_old = ti
    ux_old = uxi
    uy_old = uyi
    x_old = 0

    M = []
    h = []
    u = []
    theta = []
    t = []
    ux = []
    uy = []
    x = []
    Drag = []
    G = []

    M.append(M_old)
    h.append(h_old)
    u.append(u_old)
    theta.append(theta_old)
    t.append(t_old)
    ux.append(ux_old)
    uy.append(uy_old)
    x.append(x_old)
    Drag.append(0)
    G.append(g0)

    # temprature variation with height
    def atmosTemp(h):
        # Lapse Rate (K/m)
        lamda = [-0.0065,0.0,0.0010,0.0028,0.0,-0.0028,-0.0020,0.0,0.0120]
        # for air gas constant (J/Kg/K)
        R = 287
        # Sea level temprature (Kelvin)
        T0 = 288.16
        # Sea level altitude (m)
        H0 = 0
        # radius of earth (m)
        r0 = 6400000

        if h <= 11:
            T = T0 + lamda[0]*(h - H0)
        elif h > 11 and h <= 20:
            T0 = 216.51671343
            H0 = 11
```

```
            T = T0 + lamda[1]*(h - H0)
        elif h > 20 and h <= 32:
            T0 = 216.51671343
            H0 = 20
            T = T0 + lamda[2]*(h-H0)
        elif h > 32 and h <= 47:
            T0 = 228.54076152
            H0 = 32
            T = T0 + lamda[3]*(h-H0)
        elif h > 47 and h <= 51:
            T0 = 270.62492986
            H0 = 47
            T = T0 + lamda[4]*(h-H0)
        elif h > 51 and h <= 71:
            T0 = 270.62492986
            H0 = 51
            T = T0 + lamda[5]*(h-H0)
        elif h > 71 and h <= 84:
            T0 = 214.51270541
            H0 = 71
            T = T0 + lamda[6]*(h-H0)
        elif h > 84 and h <= 91:
            T0 = 188.4606012
            H0 = 84
            T = T0 + lamda[7]*(h-H0)
        elif h > 91 and h <= 110:
            H0 = 91
            Tc = 263.1905
            A = -76.3232
            a = -19.9429
            T = Tc + A*(pow((1-pow(((h-H0)/a),2)),0.5))
        elif h > 110 and h <= 120:
            T0 = 240
            H0 = 110
            T = T0 + lamda[8]*(h-H0)
        elif h > 120:
            T0 = 360
            H0 = 120
            Tinf = 1000
            lam = 0.01875
            epsilon = (h-H0)*((r0 + H0)/(r0 + h))
            T = Tinf - (Tinf-T0)*np.exp(-lam*epsilon)
        return T

    # gravity variation with height
    def g(g0,Re,h):
```

```
    return pow((Re/(Re+h)),2)*g0

# #  dendity variation with height (kg/m3)
def rho(h):
    return 1.2*np.exp(-2.9*pow(10,-5)*pow(h,1.15))

# Cd variation with Mach Number
def CdVariation(Mn):
    if Mn <= 0.6:
        Cd = 0.208333*pow(Mn,2) - 0.25*Mn + 0.46
    elif Mn > 0.6 and Mn <= 0.8:
        Cd = 1.25*pow(Mn,3) - 2.125*pow(Mn,2) + 1.2*Mn + 0.16
    elif Mn > 0.8 and Mn <= 0.95:
        Cd = 10.37037*pow(Mn,3) - 22.88889*pow(Mn,2) + 16.9111*Mn - 3.78963
    elif Mn > 0.95 and Mn <= 1.05:
        Cd = -30*pow(Mn,3) + 88.5*pow(Mn,2) - 85.425*Mn + 27.51375
    elif Mn > 1.05 and Mn <= 1.15:
        Cd = -20*pow(Mn,3) + 60*pow(Mn,2) - 58.065*Mn + 19.245
    elif Mn > 1.15 and Mn <= 1.3:
        Cd = 11.85185*pow(Mn,3) - 44.88889*pow(Mn,2) + 56.22222*Mn - 22.58519
    elif Mn > 1.3 and Mn <= 2:
        Cd = -0.04373178*pow(Mn,3) + 0.3236152*pow(Mn,2) - 1.019679*Mn + 1.544752
    elif Mn > 2 and Mn <= 3.25:
        Cd = 0.01024*pow(Mn,3) - 0.00864*pow(Mn,2) - 0.33832*Mn + 1.08928
    elif Mn > 3.25 and Mn <= 4.5:
        Cd = -0.01408*pow(Mn,3) + 0.191688*pow(Mn,2) - 0.86976*Mn + 1.53544
    elif Mn > 4.5:
        Cd = 0.22
    return Cd

# drag
def D(Cd,Af,rho,u):
    return pow(u,2)*0.5*Cd*Af*rho(h_old)


while t_old < tb:
    if isDragVary == 0:
        drag = 0
    else:
        if isquestionE :
            a = np.sqrt(gama*R*atmosTemp(h_old))
            Mach = u_old/a
            Cd = CdVariation(Mn=Mach)
            drag = D(Cd,Af,rho,u_old)
        else:
            Cd = 0.1
```

```
            drag = D(Cd,Af,rho,u_old)

        if isGravityVary == 0:
            gravity = g0
        else:
            gravity = g(g0,Re,h_old)

        du = ((mdot*ueq/M_old)- (drag/M_old) - gravity*np.cos(theta_old))*dt
        u_new = u_old + du

        dun = gravity*np.sin(theta_old)*dt

        dur = np.sqrt(pow(du,2)+pow(dun,2))

        dtheta = np.arctan(dun/(u_new))
        theta_new = theta_old + dtheta

        dux = dur*np.sin(theta_new)
        ux_new = ux_old + dux
        duy = dur*np.cos(theta_new)
        uy_new = uy_old + duy

        dx = ux_new*dt
        x_new = x_old + dx
        dy = uy_new*dt
        h_new = h_old + dy

        dm = -mdot*dt
        M_new = M_old + dm

        t_new = t_old + dt

        t.append(t_new)
        M.append(M_new)
        u.append(u_new)
        theta.append(theta_new)
        ux.append(ux_new)
        uy.append(uy_new)
        x.append(x_new)
        h.append(h_new)
        Drag.append(drag)
        G.append(gravity)

        theta_old = theta_new
        M_old = M_new
        ux_old = ux_new
```

```
        uy_old = uy_new
        u_old = u_new
        x_old = x_new
        h_old = h_new
        t_old = t_new

    return [x,h,u,t,theta,Drag,G]
```

Q1. (a)

- constant acceleration due to gravity (g0) = 9.81 m/s2

- neglect drag (D = 0)

In [3]:

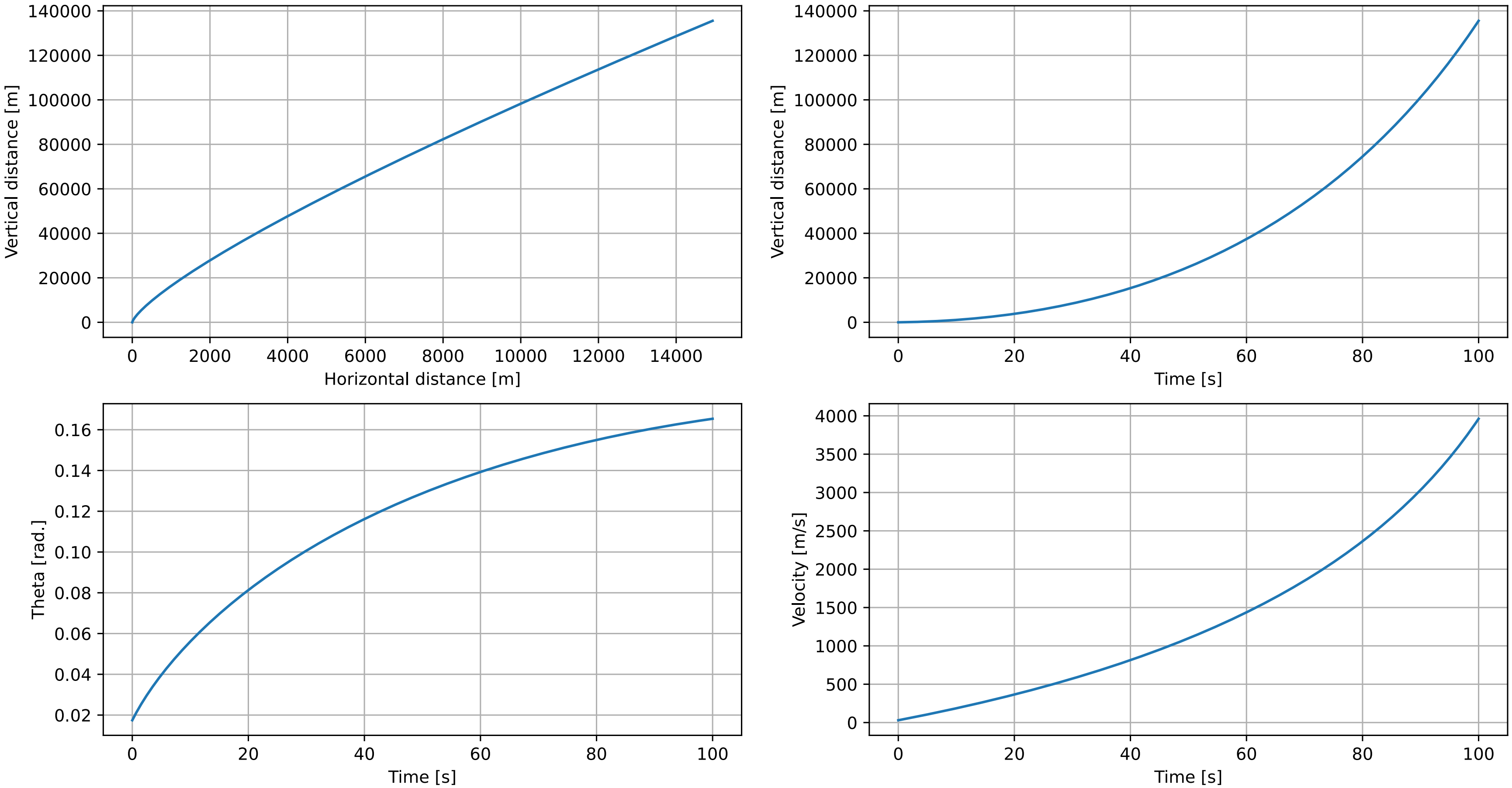
```
[xa,ha,ua,ta,thetaa,Draga,Ga] = trajectory(isDragVary=0,isGravityVary=0,isquestionE=0,dt=0.01)
```

In [4]:

```
plt.figure(figsize=(15,8));

plt.subplot(2,2,1);plt.plot(xa,ha);plt.grid();plt.xlabel("Horizontal distance [m]");plt.ylabel("Vertical distance [m]");
plt.subplot(2,2,2);plt.plot(ta,ha);plt.grid();plt.xlabel("Time [s]");plt.ylabel("Vertical distance [m]");
plt.subplot(2,2,3);plt.plot(ta,thetaa);plt.grid();plt.xlabel("Time [s]");plt.ylabel("Theta [rad.]");
plt.subplot(2,2,4);plt.plot(ta,ua);plt.grid();plt.xlabel("Time [s]");plt.ylabel("Velocity [m/s]");
```

Out[4]:

```
Text(0, 0.5, 'Velocity [m/s]')
```



Q1. (b)

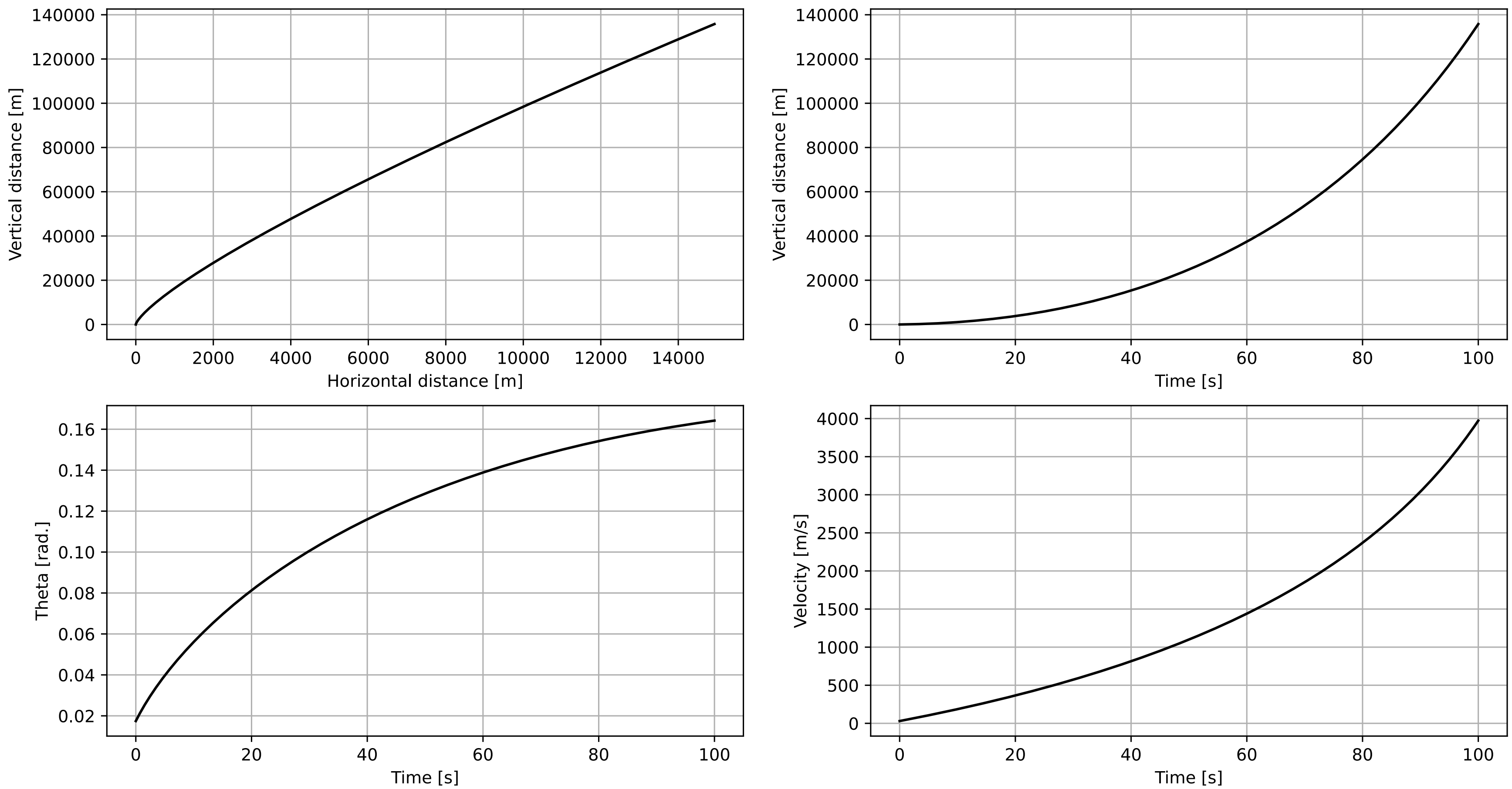
- Only acceleration due to gravity (g) varies (and D = 0):

- with height (h): g = g0 [Re/(Re + h)]2, where, Re is the earth's radius = 6,400,00 m.

In [5]:
```
[xb,hb,ub,tb,thetab,Dragb,Gb] = trajectory(isDragVary=0,isGravityVary=1,isquestionE=0,dt=0.01)
```

In [6]:

```
plt.figure(figsize=(15,8));

plt.subplot(2,2,1);plt.plot(xb,hb,"-k");plt.grid();plt.xlabel("Horizontal distance [m]");plt.ylabel("Vertical distance [m
plt.subplot(2,2,2);plt.plot(tb,hb,"-k");plt.grid();plt.xlabel("Time [s]");plt.ylabel("Vertical distance [m]");
plt.subplot(2,2,3);plt.plot(tb,thetab,"-k");plt.grid();plt.xlabel("Time [s]");plt.ylabel("Theta [rad.]");
plt.subplot(2,2,4);plt.plot(tb,ub,"-k");plt.grid();plt.xlabel("Time [s]");plt.ylabel("Velocity [m/s]");
```



Q1. (c)

Only drag (D) varies (and g = g0): with ambient gas density (ρ) and rocket velocity (u):

$$D = \frac{1}{2} * \rho * u^2 * A_f * Cd$$

, where, CD is the coefficient of drag = 0.1 (assumed constant), Af is the frontal cross-sectional area of the rocket = 1 m2.

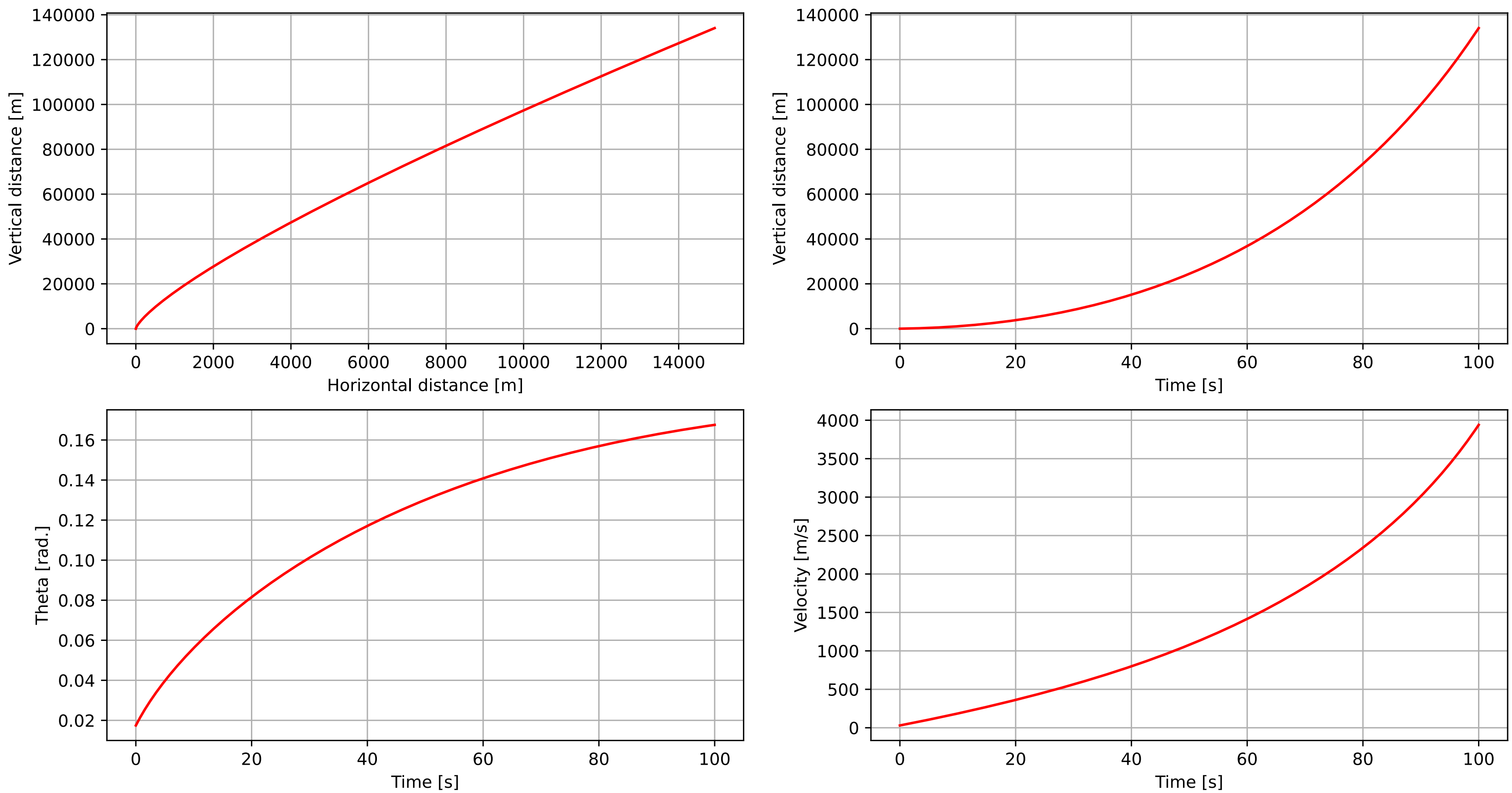
$$\rho(h) = 1.2 * e^{-2.9*10^{-5}*h^{1.15}}$$

In [7]:

```
[xc,hc,uc,tc,thetac,Dragc,Gc] = trajectory(isDragVary=1,isGravityVary=0,isquestionE=0,dt=0.01)
```

In [8]:

```
plt.figure(figsize=(15,8));

plt.subplot(2,2,1);plt.plot(xc,hc,"-r");plt.grid();plt.xlabel("Horizontal distance [m]");plt.ylabel("Vertical distance [m
plt.subplot(2,2,2);plt.plot(tc,hc,"-r");plt.grid();plt.xlabel("Time [s]");plt.ylabel("Vertical distance [m]");
plt.subplot(2,2,3);plt.plot(tc,thetac,"-r");plt.grid();plt.xlabel("Time [s]");plt.ylabel("Theta [rad.]");
plt.subplot(2,2,4);plt.plot(tc,uc,"-r");plt.grid();plt.xlabel("Time [s]");plt.ylabel("Velocity [m/s]");
```



Q1. (d)

Both g and D varies: as given in (b) and (c), respectively.

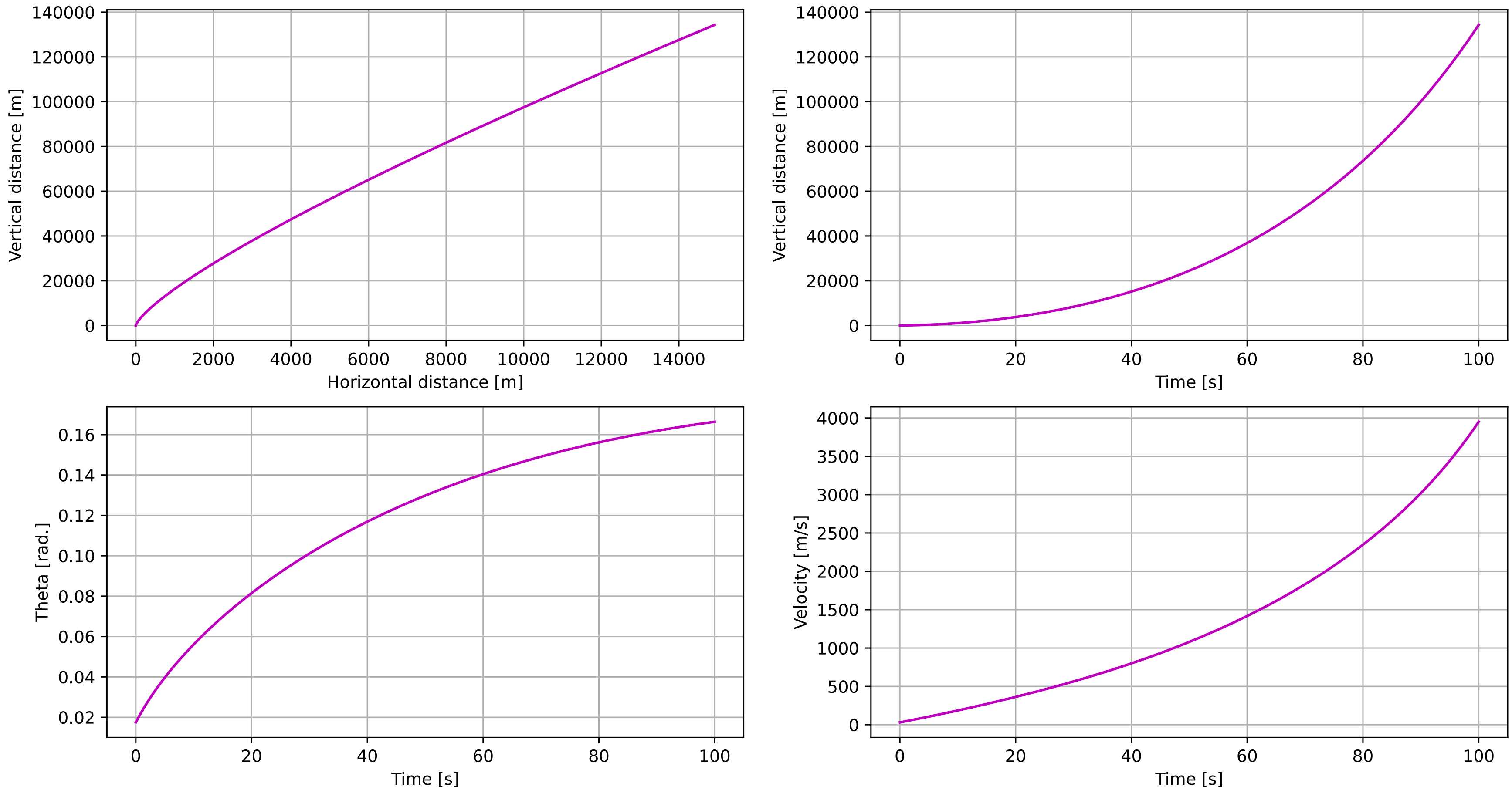
In [9]:

```
[xd,hd,ud,td,thetad,Dragd,Gd] = trajectory(isDragVary=1,isGravityVary=1,isquestionE=0,dt=0.01)
```

In [10]:

```
plt.figure(figsize=(15,8));

plt.subplot(2,2,1);plt.plot(xd,hd,"-m");plt.grid();plt.xlabel("Horizontal distance [m]");plt.ylabel("Vertical distance [m
plt.subplot(2,2,2);plt.plot(td,hd,"-m");plt.grid();plt.xlabel("Time [s]");plt.ylabel("Vertical distance [m]");
```

```
plt.subplot(2,2,3);plt.plot(td,thetad,"-m");plt.grid();plt.xlabel("Time [s]");plt.ylabel("Theta [rad.]");
plt.subplot(2,2,4);plt.plot(td,ud,"-m");plt.grid();plt.xlabel("Time [s]");plt.ylabel("Velocity [m/s]");
```



Q1. (e)

Realistic condition: CD varies with Mach number

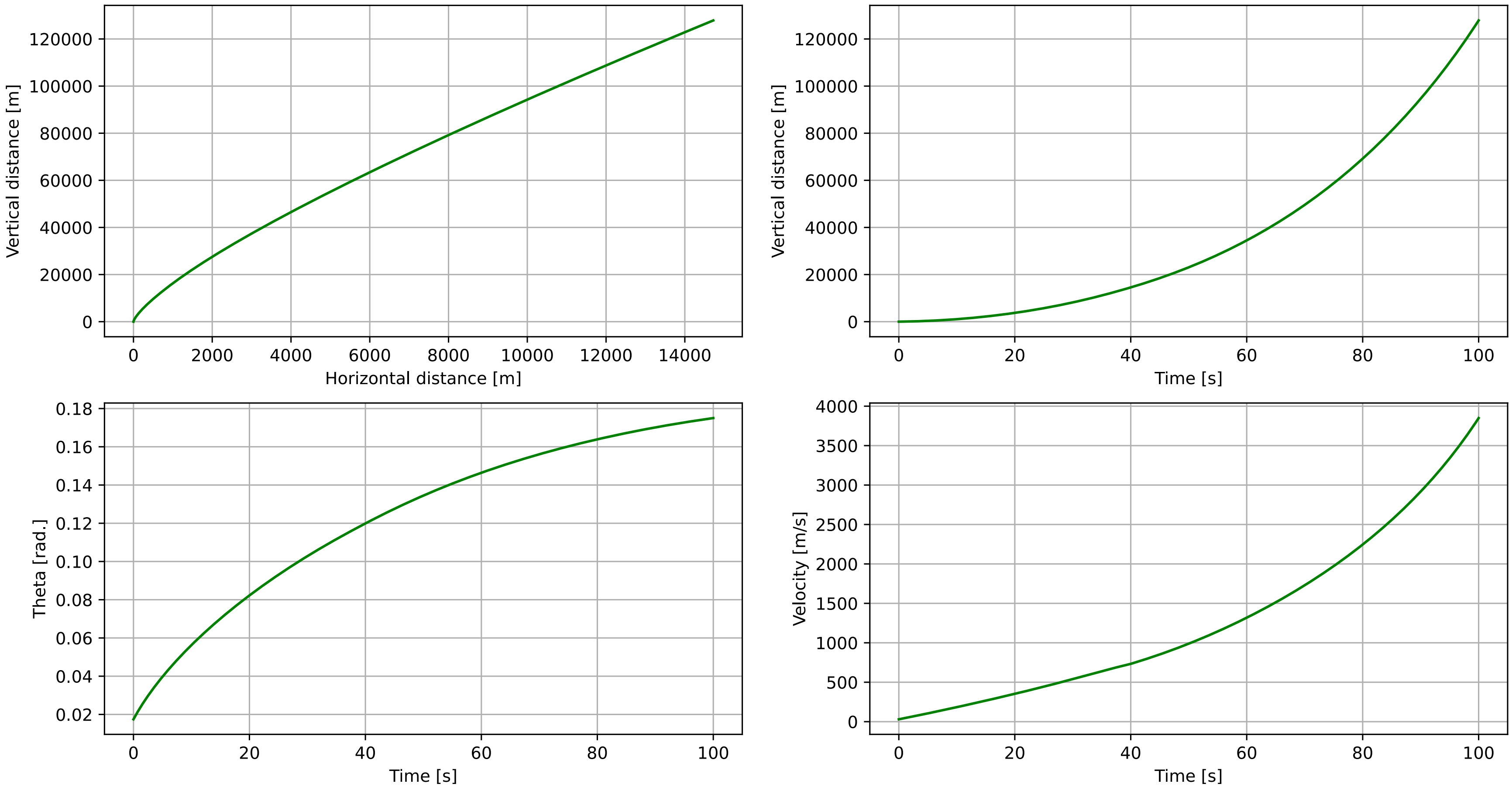
In [11]:

```
[xe,he,ue,te,thetae,Drage,Ge] = trajectory(isDragVary=1,isGravityVary=1,isquestionE=1,dt= 0.01)
```

In [17]:

```
plt.figure(figsize=(15,8));
```

```
plt.subplot(2,2,1);plt.plot(xe,he,"-g");plt.grid();plt.xlabel("Horizontal distance [m]");plt.ylabel("Vertical distance [m
plt.subplot(2,2,2);plt.plot(te,he,"-g");plt.grid();plt.xlabel("Time [s]");plt.ylabel("Vertical distance [m]");
plt.subplot(2,2,3);plt.plot(te,thetae,"-g");plt.grid();plt.xlabel("Time [s]");plt.ylabel("Theta [rad.]");
plt.subplot(2,2,4);plt.plot(te,ue,"-g");plt.grid();plt.xlabel("Time [s]");plt.ylabel("Velocity [m/s]");
```

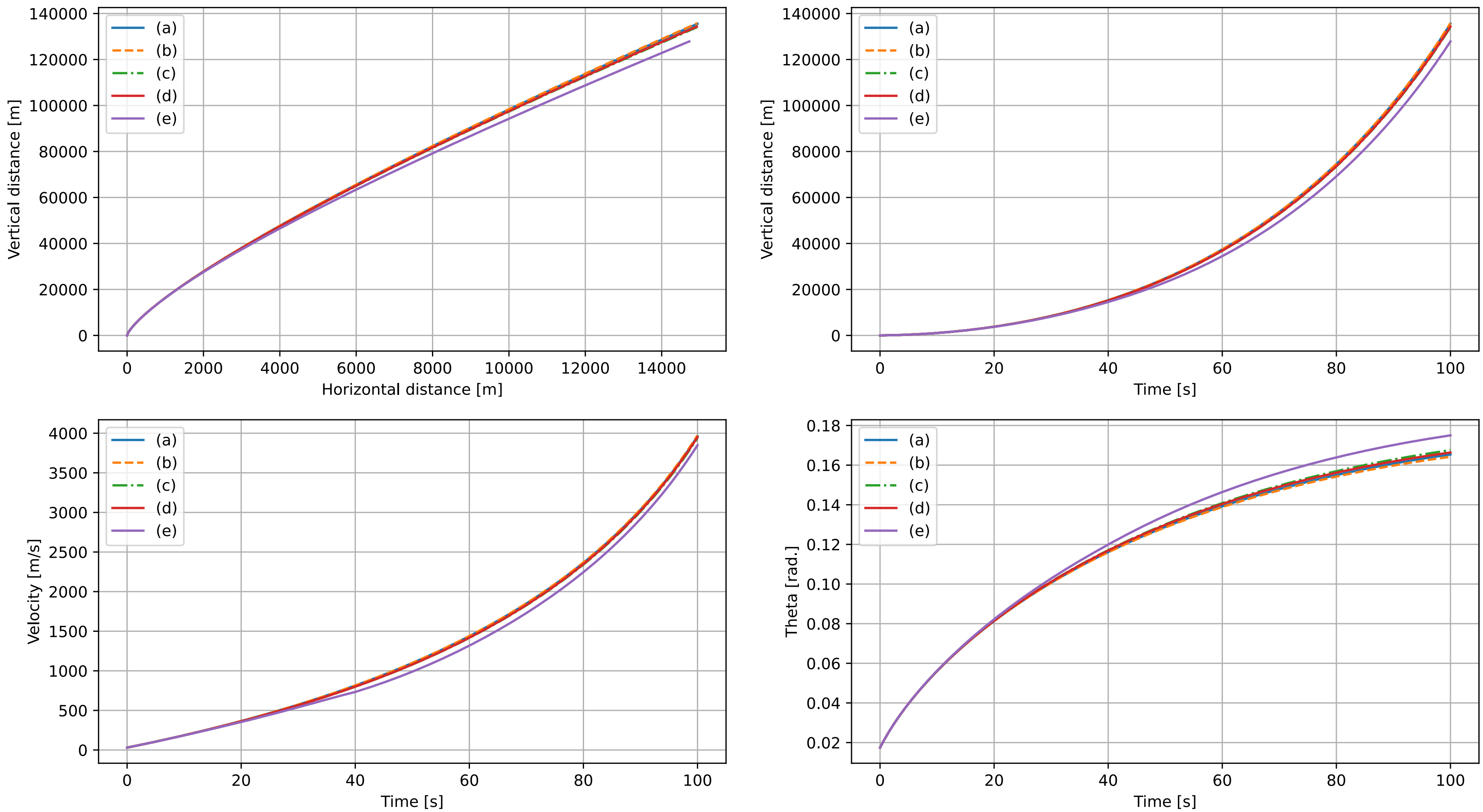


Data Comparison

In [19]:

```
plt.figure(figsize=(16,9));
plt.subplot(2,2,1);
plt.grid();
plt.xlabel("Horizontal distance [m]");plt.ylabel("Vertical distance [m]");
plt.plot(xa,ha,"-")
plt.plot(xb,hb,"--")
```

```
plt.plot(xc,hc,"-.")
plt.plot(xd,hd,"-")
plt.plot(xe,he);
plt.legend(["(a)","(b)","(c)","(d)","(e)"]);

plt.subplot(2,2,2);
plt.grid();
plt.xlabel("Time [s]");plt.ylabel("Vertical distance [m]");
plt.plot(ta,ha,"-")
plt.plot(tb,hb,"--")
plt.plot(tc,hc,"-.")
plt.plot(td,hd,"-")
plt.plot(te,he);
plt.legend(["(a)","(b)","(c)","(d)","(e)"]);

plt.subplot(2,2,3);
plt.grid();
plt.xlabel("Time [s]");plt.ylabel("Velocity [m/s]");
plt.plot(ta,ua,"-")
plt.plot(tb,ub,"--")
plt.plot(tc,uc,"-.")
plt.plot(td,ud,"-")
plt.plot(te,ue);
plt.legend(["(a)","(b)","(c)","(d)","(e)"]);

plt.subplot(2,2,4);
plt.grid();
plt.xlabel("Time [s]");plt.ylabel("Theta [rad.]");
plt.plot(ta,thetaa,"-")
plt.plot(tb,thetab,"--")
plt.plot(tc,thetac,"-.")
plt.plot(td,thetad,"-")
plt.plot(te,thetae);
plt.legend(["(a)","(b)","(c)","(d)","(e)"]);
```



Drag and Gravity Graph

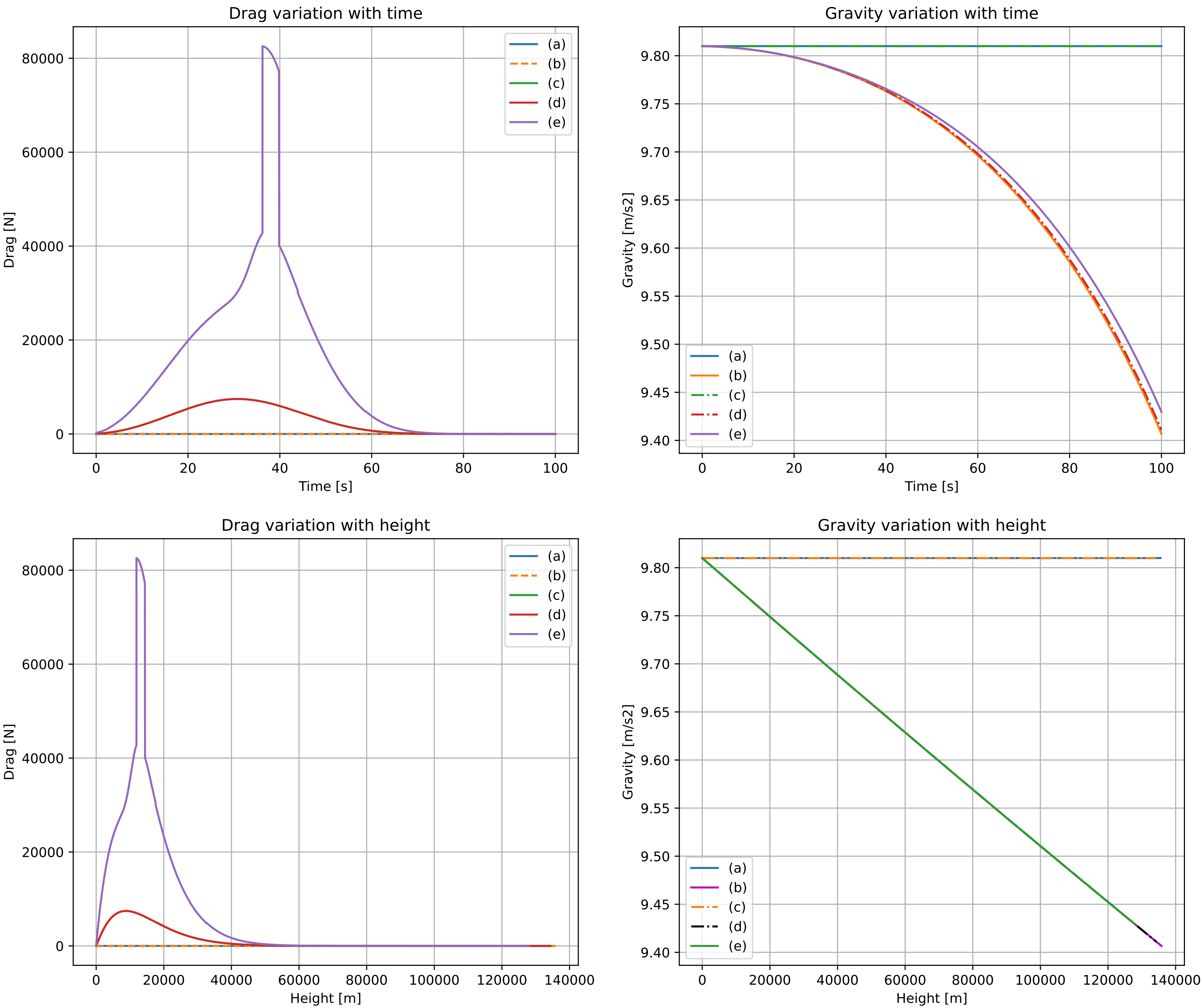
In [18]:

```
plt.figure(figsize=(15,13));

plt.subplot(2,2,1);
plt.title("Drag variation with time");
plt.grid();
plt.xlabel("Time [s]");plt.ylabel("Drag [N]");
plt.plot(ta,Draga)
plt.plot(tb,Dragb,"--")
plt.plot(tc,Dragc)
plt.plot(td,Dragd)
```

```python
plt.plot(te,Drage);
plt.legend(["(a)","(b)","(c)","(d)","(e)"]);

plt.subplot(2,2,2);
plt.title("Gravity variation with time");
plt.grid();
plt.xlabel("Time [s]");plt.ylabel("Gravity [m/s2]");
plt.plot(ta,Ga,"-")
plt.plot(tb,Gb,"-")
plt.plot(tc,Gc,"-.")
plt.plot(td,Gd,"-.")
plt.plot(te,Ge);
plt.legend(["(a)","(b)","(c)","(d)","(e)"]);

plt.subplot(2,2,3);
plt.title("Drag variation with height");
plt.grid();
plt.xlabel("Height [m]");plt.ylabel("Drag [N]");
plt.plot(ha,Draga)
plt.plot(hb,Dragb,"--")
plt.plot(hc,Dragc)
plt.plot(hd,Dragd)
plt.plot(he,Drage);
plt.legend(["(a)","(b)","(c)","(d)","(e)"]);

plt.subplot(2,2,4);
plt.title("Gravity variation with height");
plt.grid();
plt.xlabel("Height [m]");plt.ylabel("Gravity [m/s2]");
plt.plot(ha,Ga,"-")
plt.plot(hb,Gb,"-m")
plt.plot(hc,Gc,"-.")
plt.plot(hd,Gd,"-.k")
plt.plot(he,Ge);
plt.legend(["(a)","(b)","(c)","(d)","(e)"]);
```

Out[18]:

```
<matplotlib.legend.Legend at 0x11d62c40>
```


Drag variation with time
Drag [N]
Time [s]
(a)
(b)
(c)
(d)
(e)
Gravity variation with time
Gravity [m/s2]
Time [s]
(a)
(b)
(c)
(d)
(e)
Drag variation with height
Drag [N]
Height [m]
(a)
(b)
(c)
(d)
(e)
Gravity variation with height
Gravity [m/s2]
Height [m]
(a)
(b)
(c)
(d)
(e)

Burnout Data

In [15]:
```
BurnOut_data = pd.DataFrame({
    "Q1" : ("(a)","(b)","(c)","(d)","(e)"),
    "Burnout Height [m]" : (ha[-1],hb[-1],hc[-1],hd[-1],he[-1]),
    "Burnout Velocity [m/s]" : (ua[-1],ub[-1],uc[-1],ud[-1],ue[-1]),
    "Burnout Theta [rad]" : (thetaa[-1],thetab[-1],thetac[-1],thetad[-1],thetae[-1])
})
```

In [16]:
```
BurnOut_data.set_index("Q1",inplace=True)
BurnOut_data
```

Out[16]:

Q1	Burnout Height [m]	Burnout Velocity [m/s]	Burnout Theta [rad]
(a)	135534.160597	3961.668460	0.165374
(b)	135801.485837	3973.085036	0.164204
(c)	134060.341269	3939.701215	0.167559
(d)	134323.542576	3950.957755	0.166374
(e)	127861.124522	3849.269635	0.175019