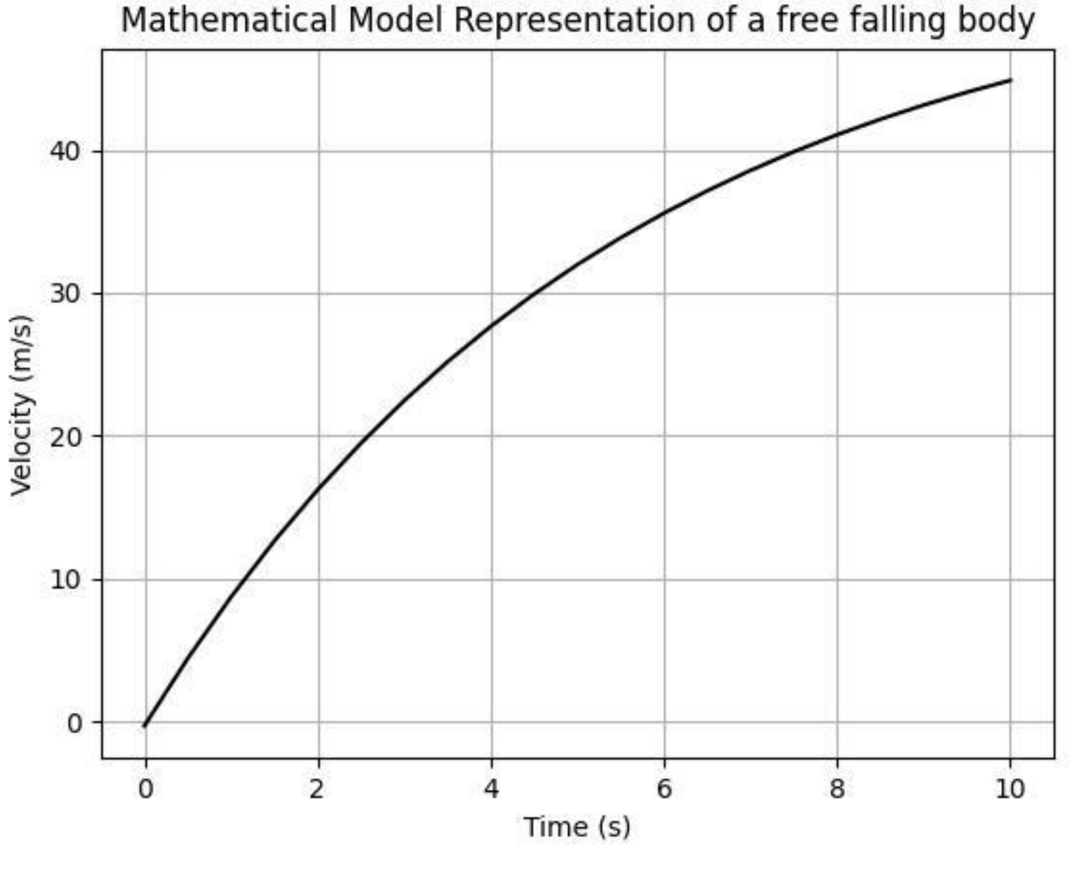
Consider the problem of falling body given in page 2 of your lecture note. Refer to exercises 1.1 (d) and 1.2 (c) in pages 4 – 5 as well. Starting with the following mathematical model for the above problem, derive necessary solutions and answer the given questions.

(a) Instead of the velocity at time $t=0$, if it is assumed that the velocity at some time $t=t_x$ was known as $v=v_x$, derive the analytical solution for the velocity

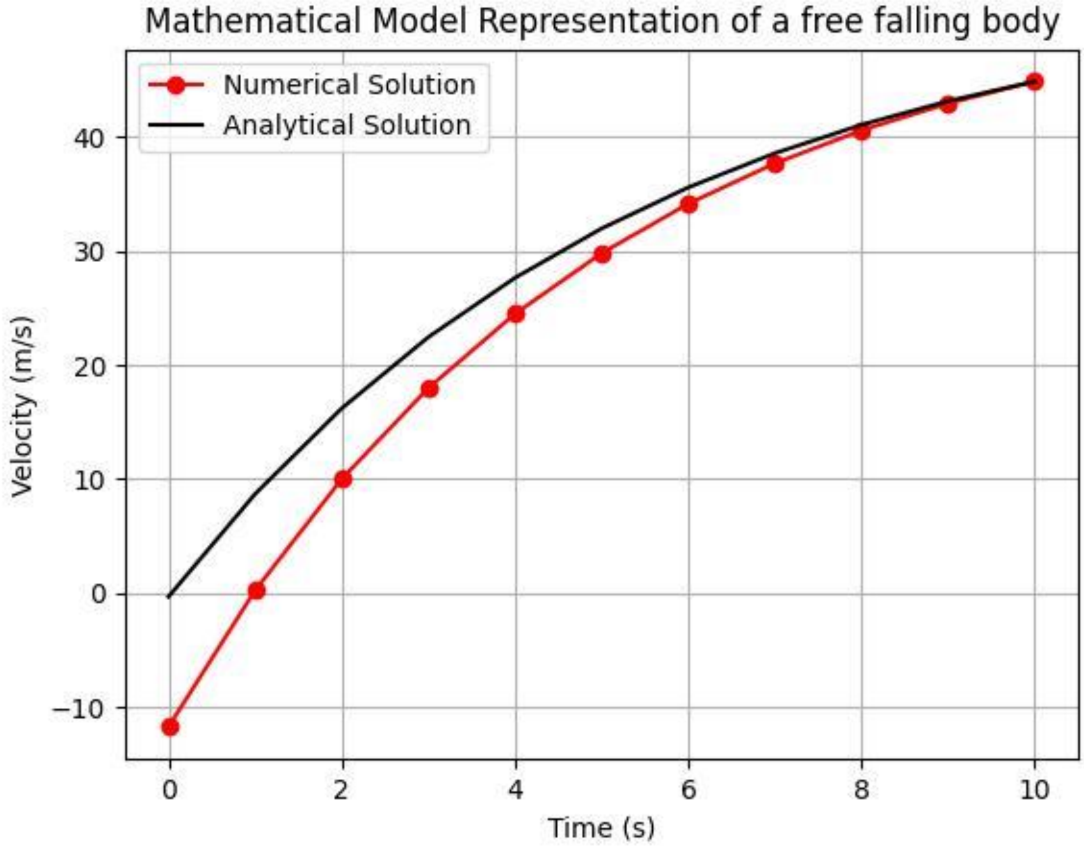
(b) If $t_x=10s$ and $v_x= 44.87$ m/s, derive a numerical scheme to calculate velocities of the body from time t = 0 – 10s

(c) Using the analytical solution you derived in part (a) above, show graphically, the variation of velocity of the body for t = 0 – 10s. Use black color for the graph

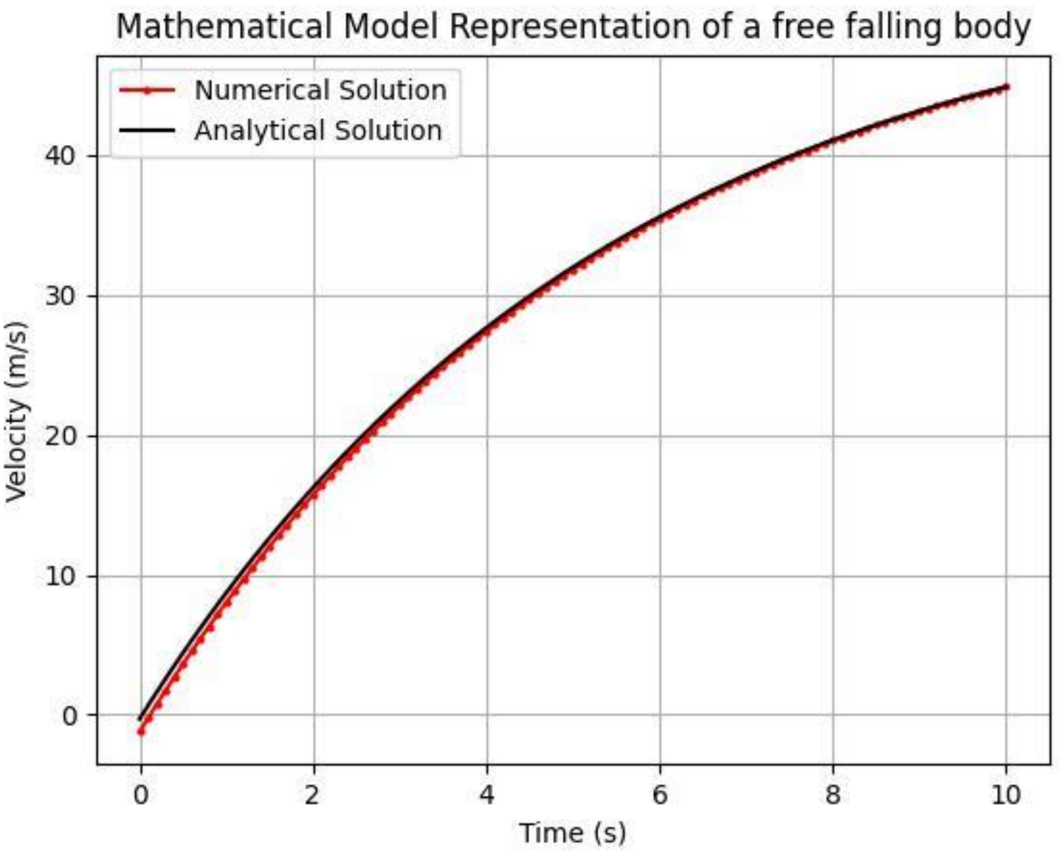


(d) Use computer to solve the numerical scheme in part (b) above to give the velocity of the body for t = 0-10s, Use red colour for the graph, and plot it on the same axes as those of (c).

Time step (dt) = 1s



Time step (dt) = 0.1s



(e) Discuss the possible reasons for any discrepancies of the two solutions (numerical (c) and analytical (d)).

It is visible that, there is a difference between analytical and numerical answers. And that difference is more visible at t=0 since the calculations begins from t=10 and then reversed.

The analytical method uses the answers received from the inputs via analytical equation. Meanwhile, the numerical method uses the numerical model's answers. Since the numerical model uses approximations, it results in these differences.

Also there is some rounding off errors when the calculations is done through the computer. It can also results in these differences.

Also when smaller timesteps are used, it also reduces errors as smaller steps results in errors that are comparably small.

D: > 4th Semester > EM215 > EM215 - Lab Assignment 1.py > ...

```python
import matplotlib.pyplot as plt
import numpy as np

deltaT = 0.1  #change this value to adjust the smoothness
m = 68.1
c = 12.5
g = 9.81
tx = 10
vx = 44.87
v0 = 0
v2=vx

numerical_velocity = [v2]
analytical_velocity = []
error_list = []

x = np.arange(0, 10+deltaT, deltaT)
y = np.arange(10, 0-deltaT, -deltaT)

#numerical equation calculation
for i in range(len(x)-1):

    v1 = (v2 - g*deltaT)/(1-c*deltaT/m)
    numerical_velocity.append(v1)
    v2 = v1

#analytical equation calculation
for i in range(len(x)):

    v = ((m*g)/c) - (m/c)*(g - ((c*vx)/m))*np.exp(-c*(x[i]-tx)/m)
    analytical_velocity.append(v)

#graph plotting
plt.plot(y, numerical_velocity,"o-r",markersize=2)
plt.plot(x, analytical_velocity,"#000000")

plt.legend(["Numerical Solution","Analytical Solution"])
plt.title("Mathematical Model Representation of a free falling body")
plt.xlabel("Time (s)")
plt.ylabel("Velocity (m/s)")
plt.grid(True)
plt.show()
```