Gen IMS II Homework 2

Yiping Deng

February 15, 2018

Task 1

a)

Using physics, we have

$$V_L = L\frac{dI}{dt} \tag{1}$$

$$V_R = IR \tag{2}$$

$$V_L + V_R = V_S \tag{3}$$

Substitute (1), (2) into (3), we get the ODE

$$L\frac{dI}{dt} + IR = V_S \tag{4}$$

b)

Since it is a seperable ODE, we can solve it using seperation of variable.

$$\implies (IR - V_S)dt = -LdI \tag{5}$$

$$\implies dt = -L\frac{dI}{IR - V_S} \tag{6}$$

$$\implies t + C = -\frac{L}{R}ln(IR - V_S) \tag{7}$$

$$\implies ln(IR - V_S) = -\frac{R}{L}t + C \tag{8}$$

$$\implies IR - V_S = Ce^{-\frac{R}{L}t} \tag{9}$$

$$\implies I = \frac{V_S}{R} + Ce^{-\frac{R}{L}t} \tag{10}$$

Note that C is the constant derived from integration.
Now we can plug in the initial value that $I(0) = I_{ini}$

$$I_{ini} = \frac{V_S}{R} + C \implies \tag{11}$$

$$C = I_{ini} - \frac{V_S}{R} \tag{12}$$

Thus, this give us the solution for the differential equation.

$$I(t) = \frac{V_S}{R} + \frac{I_{ini}R - V_S}{R}e^{-\frac{R}{L}t} \tag{13}$$

c)

Take Laplace transform on both side of (4)

$$\mathcal{L}(L\frac{dI}{dt} + IR) = \mathcal{L}(V_S) \tag{14}$$

Using the linearity, we can get

$$L\mathcal{L}(I') + R\mathcal{L}(I) = V_S\mathcal{L}(1) \tag{15}$$

We use I^* to represent function in frequency domain.

$$L(sI^*(s) - I(0)) + RI^*(s) = V_S\frac{1}{s} \tag{16}$$
$$I(0) = I_{ini} \implies \tag{17}$$
$$L(sI^*(s) - I_{ini}) + RI^*(s) = V_S\frac{1}{s} \tag{18}$$
$$\text{With simplification: } (s^2L + sR)I^*(s) = V_S + I_{ini}R \tag{19}$$
$$I^*(s) = \frac{V_S + I_{ini}R}{s^2L + sR} \tag{20}$$

We factorize the right-hand side of (20), and apply inverse Laplace transform

$$I^*(s) = \frac{V_S}{R}\frac{1}{s} + \frac{I_{ini}R - V_S}{R}\frac{1}{s + \frac{R}{L}} \tag{21}$$

Apply Inverse Laplace Transform

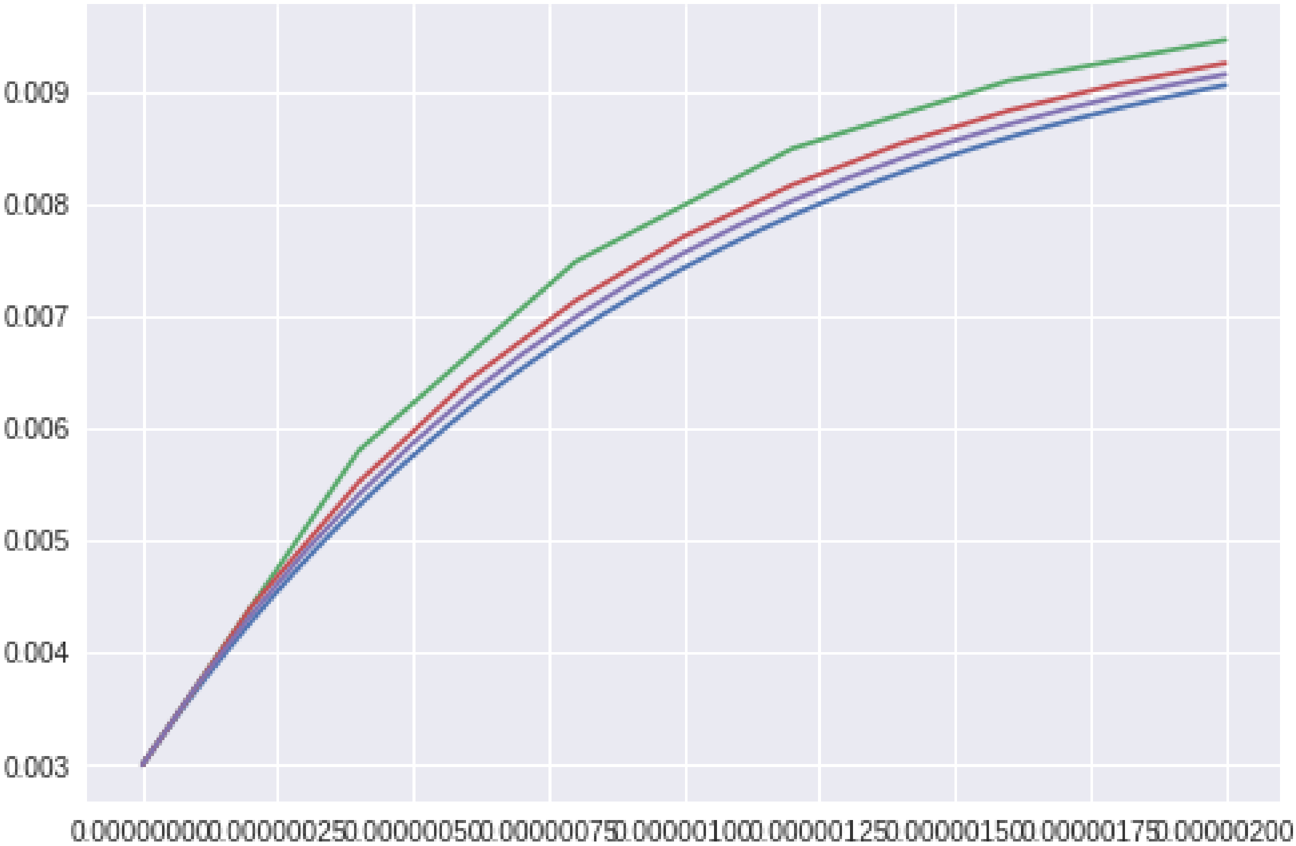
$$I(t) = \mathcal{L}(I^*(s)) = \frac{V_S}{R} + \frac{I_{ini}R - V_S}{R}e^{-\frac{R}{L}t} \tag{22}$$

c)

To evaluate $t = 0.000002$, with plot $\Delta x = 0.000002/5$, $\Delta x = 0.000002/10$, $\Delta x = 0.000002/20$

```python
# -*- coding: utf-8 -*-
"""
Author: Yiping Deng
"""

import numpy as np
import matplotlib.pyplot as plt

Vs = 10.0
R = 1000.0
L = 0.001
Iini = 0.003

def Isolved(t):
  # the solution to ODE
  return Vs / R + ((Iini * R - Vs)/R) *np.exp((-R/L) * t)

def D(y):
  # the derivative with respect to y
  return (Vs - y * R) / L

def euler(Df,f_start, start, stop, num):
  # implementation of Euler's method
  x = start
  y = f_start
  step = (stop - start) / num
  while x <= stop:
    yield y
    x += step
    y = Df(y) * step + y
```

```
def eulerX(start, stop, num):
  # the x-axis for Euler's method
  x = start
  step = (stop - start) / num
  while x <= stop:
    yield x
    x += step

# the solution plot - Blue color
x = np.linspace(0, 0.000002, 200)
y = Isolved(x)
plt.plot(x, y)
print(x, y)


# with 5 iteration - Green color
x1 = list(eulerX(0, 0.000002, 5))
y1 = list(euler(D, Iini, 0, 0.000002, 5))

plt.plot(x1,y1)
print(x1, y1)

# with 10 iteration - Red color
x2 = list(eulerX(0, 0.000002, 10))
y2 = list(euler(D, Iini, 0, 0.000002, 10))
plt.plot(x2, y2)
print(x2, y2)

# with 20 iteration - Violet color
x3 = list(eulerX(0, 0.000002, 20))
y3 = list(euler(D, Iini, 0, 0.000002, 20))
plt.plot(x3, y3)
print(x3, y3)
plt.show()
```




```
Points for 5 iteration:
(0, 0.003)
(4e-07, 0.0058)
(8e-07, 0.00748)
(1.2e-06, 0.008487999999999999)
(1.6e-06, 0.0090928)
(2e-06, 0.00945568)
Points for 10:
(0, 0.003)
(2e-07, 0.0044)
(4e-07, 0.00552)
(6e-07, 0.006416)
(8e-07, 0.0071328)
(1e-06, 0.00770624)
(1.2e-06, 0.008164992)
(1.4e-06, 0.0085319936)
(1.6e-06, 0.008825594879999999)
(1.8e-06, 0.009060475904)
(2e-06, 0.0092483807232)
Points for 20:
(0, 0.003)
(1e-07, 0.0037)
(2e-07, 0.004330000000000005)
(3e-07, 0.004897)
(4e-07, 0.0054073)
(5e-07, 0.005866570000000001)
(6e-07, 0.006279913000000001)
(7e-07, 0.006651921700000005)
(8e-07, 0.00698672953)
(9e-07, 0.007288056577)
```

```
(1e-06, 0.0075592509193)
(1.1e-06, 0.00780332582737)
(1.2e-06, 0.008022993244633)
(1.2999999999999998e-06, 0.0082206939201697)
(1.3999999999999997e-06, 0.00839862452815273)
(1.4999999999999996e-06, 0.008558762075337458)
(1.5999999999999995e-06, 0.008702885867803711)
(1.6999999999999994e-06, 0.00883259728102334)
(1.7999999999999993e-06, 0.008949337552921006)
(1.8999999999999992e-06, 0.009054403797628905)
(1.999999999999999e-06, 0.009148963417866014)
```

As you can see in the picture, the finer the Δx, the closer it approaches to the solution. This iteration scheme converges uniformly to the solution function.