CEE 6736: HW #1

Hanfeng Zhai
Mechanical Engineering, Cornell University
hz253@cornell.edu

September 16, 2022

```
[ ]: # !sudo apt install cm-super dvipng texlive-latex-extra texlive-latex-recommended
```

Problem 1

Please show all work (i.e. include source code, and include thoughtful, analytical discussions):

(1) Pick a population (e.g. New York City in March of 2020, or a city in the state of Florida in August 2021, etc.) and find the COVID data from the relevant health department. Assume defensible values for X and I in your populations. Use these data to test our COVID transmission model. Discuss the results.

Solution:

Recall the COVID transmission model discussed in class. First defining: $\mathbb{N}_i \equiv$ Number of infections on a given day i; $\mathbb{X} \equiv$ Expected numbers of daily contacts per infected person; $\mathbb{I} \equiv$ Probability of infection for each contact. The infection for the next day writes:

$$\mathbb{N}_{i+1} = (\mathbb{I}\mathbb{X} + 1)\mathbb{N}_i$$

This model can be further expressed as, at a certain day i, the infectious number follows:

$$\mathbb{N}_i = (1 + \mathbb{I}\mathbb{X})^i \mathbb{N}_0$$

where $\mathbb{N}_0$ denotes the initial values at $t = 0$. Promoting this to a continuous model:

$$\frac{d\mathbb{N}}{dt} = (\mathbb{I}\mathbb{X})\,\mathbb{N}$$

where $\mathbb{I}\mathbb{X}$ is identified as the $\mathcal{R}$ factor.

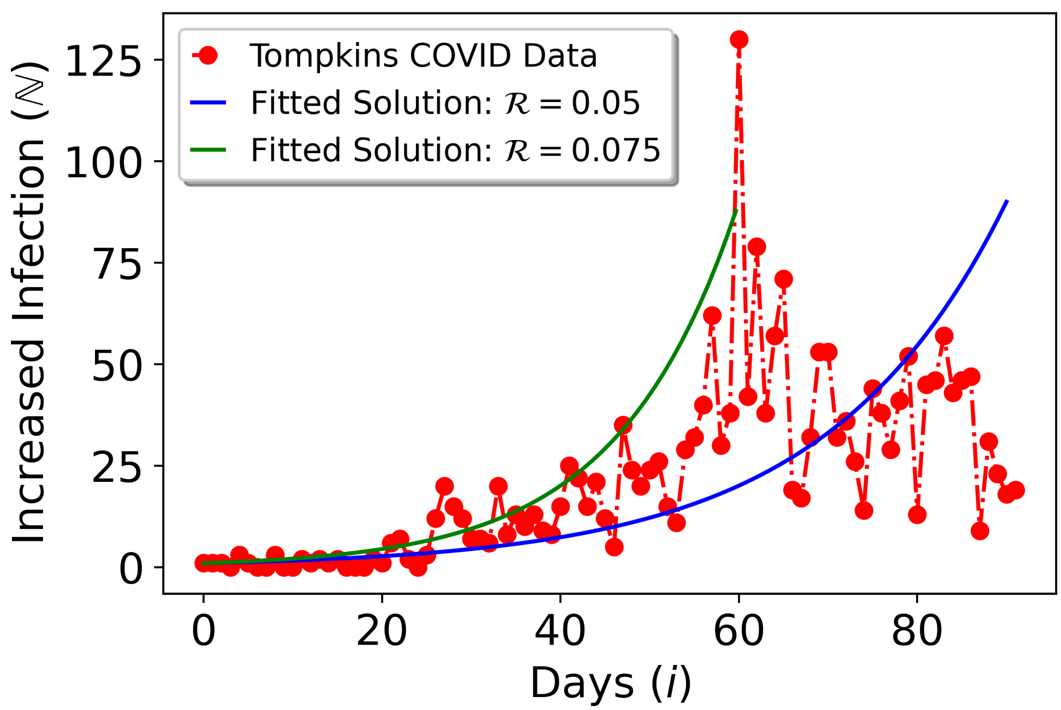
This is a special case that follows the linear ODE form when the bias term equals zero. Hence, the solution ought to follow the basic form: $\mathbb{N} = Ce^{\mathcal{R}t}$, where C is a constant. Based on the data fitting, it seems like a good fit for $\mathcal{R}$ is 0.05. Assuming an (reasonable) $\mathbb{I}$ of 0.001, the $\mathbb{X}$ is hence 50.

We now apply the data of Tompkins county (from July 2021 to Oct 2021) to test the model, starting from loading the data from `.txt` file:

```
[ ]: import numpy as np
     import matplotlib.pyplot as plt
     import matplotlib as mpl
```

```
from scipy.integrate import odeint

t_discretized = np.linspace(0,90,90)
N_real = np.exp(.05*t_discretized)
N_real_2 = np.exp(.075*t_discretized[0:60])

tompkins_data = np.loadtxt("tompkins_data.txt")
num_infec = np.flip(tompkins_data.T[2])
daily_text = np.flip(tompkins_data.T[1]);daily_text_identity = daily_text/np.
 ↪mean(daily_text)
total_infec = np.flip(tompkins_data.T[3]);total_infec_identity = total_infec/np.
 ↪mean(total_infec)
plt.plot(num_infec,'ro-.',label='Tompkins COVID Data')
plt.plot(t_discretized,N_real,'b-',label='Fitted Solution: $\mathcal{R} = 0.05$')
plt.plot(t_discretized[0:60],N_real_2,'g-',label='Fitted Solution: $\mathcal{R}␣
 ↪= 0.075$')
plt.xlabel("Days ($i$)")
plt.ylabel("Increased Infection ($\mathbb{N}$)")

plt.legend(shadow=True, handlelength=1, fontsize=12)
```

```
[ ]: <matplotlib.legend.Legend at 0x7f445026c390>
```



Problem 2

Determine if the equation of a line ($y = mx + b$) satisfies the *Axioms of Linearity*. Discuss the results.

Solution:

Let $f(x) = y = mx + b$, recall the definition of axioms of linearity,

$$cf(x_1 + x_2) = cf(x_1) + cf(x_2)$$

Substitute it into $f(x)$ we have:

LHS $= c(m(x_1 + x_2) + b) = cmx_1 + cmx_2 + cb$RHS $= c(mx_1 + b) + c(mx_2 + b) = cmx_1 + cmx_2 + 2cb$

It is observed that LHS $\neq$ RHS. Hence the line does not satisfy the axiom of linearity. What's more, it can be determined if the bias term b is eliminated, the line hence obeys the axiom of linearity.

Problem 3

Use the ODE solution method discussed in class (*i.e.* $\frac{d}{dt} ln|\cdot|$) to exactly solve the drone ODE from HW1 using mass, $m = 10kg$ and the drag coefficient, $\gamma = 2kg/sec$ and the initial condition: $v(0) = 0$. Please plot the exact solution within the vector plot from HW1 and discuss.

Solution:

Recall the governing equation of the drone free fall, the velocity writes:

$$\frac{dv(t)}{dt} = -g - \frac{\gamma}{m} v$$

where

$$\begin{aligned} \gamma &\equiv \text{drag of coefficient} \\ \gamma v &\equiv \text{wind of resistance} \\ m &\equiv \text{mass of quad copter} \end{aligned}$$

written in the standard form

$$\frac{dv(t)}{dt} + \frac{\gamma}{m} v = -g$$

applying the integration factor $\mu(t) = e^{\frac{\gamma}{m} t}$, the equation can be rewrite in the form given the fact that $\frac{d\mu(t)}{dt} = \frac{\gamma}{m}\mu$:

$$\frac{d}{dt}\left[e^{\frac{\gamma}{m}t} v(t)\right] = -e^{\frac{\gamma}{m}t} g$$

the solution further writes:

$$v(t) = e^{-\frac{\gamma}{m}t} \int_{t_0}^{t} e^{\frac{\gamma}{m}s}(-g) \cdot ds + Ce^{-\frac{\gamma}{m}t}$$

skipping the detailed derivation, we directly land in the final solution form of $v(t)$:

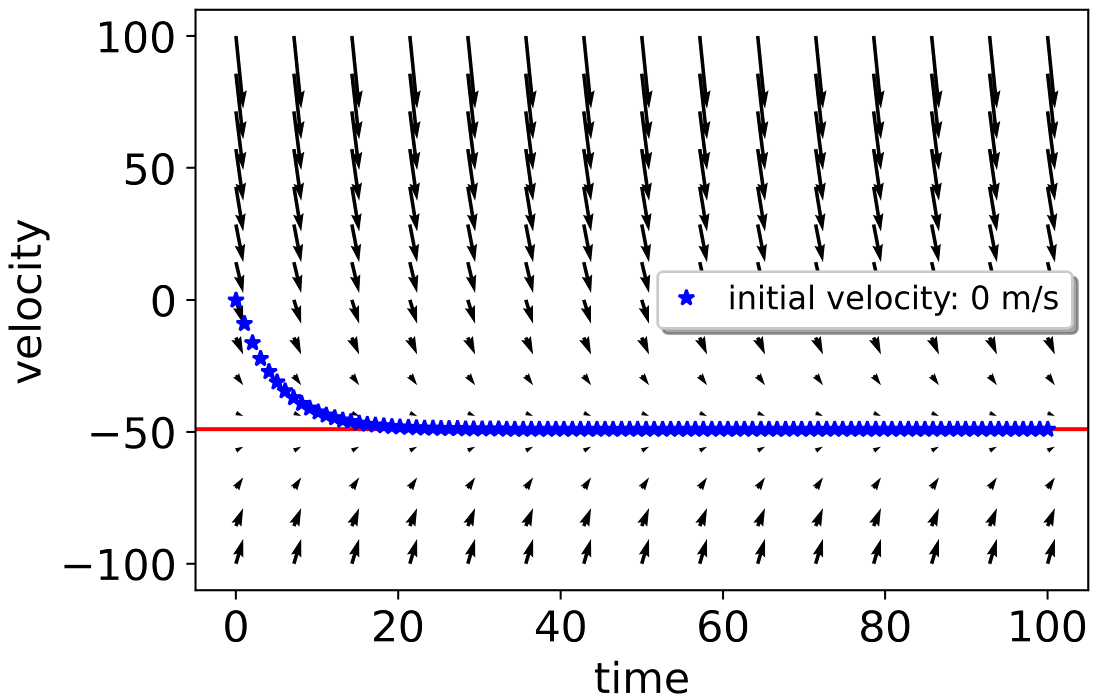
$$v(t) = \frac{1}{\mu(t)} \left[\int_{t_0}^{t} \mu(s)(-g) \cdot ds + C\right]$$

To determine the constant C, we substitute the given initial values: $v(0) = 0 \to C = 5g$. We, therefore, obtain the solution:

$$v(t) = \frac{1}{e^{\frac{\gamma}{m}t}} \left[-g \int_{t_0}^{t} e^{\frac{\gamma}{m}s} \cdot ds + 5g \right] = \frac{1}{e^{\frac{\gamma}{m}t}} \left[-g \frac{m}{\gamma} e^{\frac{\gamma}{m}t} + 5g \right] = \frac{1}{e^{0.2t}} \left[-5ge^{0.2t} + 5g \right]$$

We, therefore, plot the graph:

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
from scipy.integrate import odeint

t,v = np.meshgrid(np.linspace(0,100,15),np.linspace(-100,100,15)) # generate mesh

# define parameters
gamma = 2
m = 10

# define equations
drag = (gamma/m) * v
g = 9.8
RHS = - g - drag

sol_exact = (1/np.exp(0.2*t_dis)) * (-5*g*np.exp(0.2*t_dis) + 5*g)
# print(np.shape(sol_exact))
# solving the equation(s) using odeint
t_dis = np.linspace(0,100,100)

# # plotting
plt.quiver(t,v,3,RHS)
plt.axhline(y=-49, color='r', linestyle='-')
plt.plot(t_dis, sol_exact, 'b*', label="initial velocity: 0 m/s")
plt.xlabel('time')
mpl.rcParams.update({'font.size': 16})
plt.ylabel('velocity')
plt.legend(shadow=True, handlelength=1, fontsize=12)
plt.rcParams['figure.dpi'] = 500
plt.show()
plt.figure(figsize=(5, 3))
```



[ ]: <Figure size 2500x1500 with 0 Axes>

<Figure size 2500x1500 with 0 Axes>