Homework #7

Introduction to Engineering Computing

Due at 5:00 PM on May 8th, 2020

Introduction

Oxygen is the most well-established indicator of water quality. Dissolved oxygen is in fact essential for the survival of all aquatic organisms. Moreover, oxygen affects a vast number of other water indicators, not only biochemical but aesthetic ones like odor, clarity and taste.

When a pollutant is added to a river, then the oxygen level declines (that is, the river is in oxygen deficit). The dissolved oxygen deficit is modelled by:

$$D(t) = \frac{k_d L_0}{k_r - k_d}\left(e^{-k_d t} - e^{-k_r t}\right) + D_0 e^{-k_r t} \qquad (1)$$

$$k_r = \frac{(VS)^{0.5}}{H^{1.5}} \qquad (2)$$

Homework Description

Values for six of the seven the parameters in Eq. (1) and Eq. (2) are found in the file *PollutionData.txt*. These parameter values are for the Cedar River and are in the order listed below (all double numbers). **Please note which units you are using and convert as needed.**

- D_0 - Initial dissolved oxygen deficit (mg/l)
- L_0 - Initial concentration of pollutant added as a bolus at the point of discharge (mg/l)
- S - Diffusivity of oxygen in water (cm^2/s)
- H - Depth of flow of river (ft)
- V - Average velocity of river (ft/s)
- k_d - Oxygen use rate by natural pollution removal processes (hr^{-1})

The goal of your engineering project is to estimate how long it takes the oxygen level to return to the initial level (baseline) after a bolus injection of a pollutant is added to the river stream. The model that you will use to perform these estimates are Eq. (1) and Eq. (2). In this homework assignment you must do the following (be very careful to make sure that the units are consistent):

1) Create file named *myHW7.m*, into which you will place your main Matlab script and two functions described below.
2) Your script must access the *PollutionData.txt* and read the six data values (see *importdata* MATLAB function).
3) Create a MATLAB function called **reaeration** to estimate the reaeration constant k_r in Eq. (2). The function should be called from **your main script.** The function shall have three input parameters and return the reaeration constant. Place the **reaeration** function in your my*HW7.m* file physically after your main script as illustrated below.
4) Create a MATLAB function called **deOxygen** to calculate dissolved oxygen deficit given in Eq. (1). The function should be called from the main script in your .m file and have as input arguments the parameters needed to estimate $D(t)$. Place the ***deOxygen*** function in your *myHW7.m* file after the **reaeration** function.
   Thus your main script, the **reaeration** function, and the **deOxygen** function are in the same *myHW7.m* file, i.e., the file will contain the core matlab script, followed by the two user-defined functions.

The code in the file *myHW7.m* should then be physically structured as:

```
<main script file operations, including calls to the >
< reaeration and deOxygen functions>
    (we recommend that you don't put these main operations into a function)

function kr = reaeration(…)
   …
code
   end

function D = deOxygen(…)
   …
code
   end
```

5) Simulate, in 30-minute increments, 250 hours of dissolved oxygen at the point of discharge. Plot the result of solving Eq. (1) as a function of time as illustrated in Figure 1. The code for accomplishing this must be in the **main script** within the **myHW7.m** file.
6) Report at what time (in hours) the oxygen deficit returns to the initial deficit level.
7) Report at what time (in hours) the oxygen deficit returns to 80% of the initial deficit level.
8) Find and plot (in one figure as shown in Figure 2, Panel a) dissolved oxygen $D(t)$ (on the ordinate) versus time after pollutant added (on the abscissa) for 6 flow velocities between 100 $ft \cdot hr^{-1}$ and 250 $ft \cdot hr^{-1}$, inclusive. Use the MATLAB function `linspace(100,250,6)` to define the six flow velocities.
9) Based on the results of part 8), plot (as shown in Figure 2, Panel b) the time to return to initial (baseline) pollutant level on the ordinate versus river flow velocities on the abscissa. The plot should look exactly like what is shown in Figure 2, Panel b.

Your program should produce two and only two plots as shown below. Remember to clearly label all axes. Submit your *myHW7.m* file to the ICON dropbox for HW7.

Grading Rubric

80 points for working code

- 10 points for reading the six parameter values from *PollutionData.txt*
- 5 points for implementing the **reaeration** function
- 20 points for implementing the **deOxygen** function
- 10 points for reporting when the oxygen deficit returns to the initial deficit level
- 5 points for reporting when the oxygen deficit returns to 80% of the initial deficit level
- 30 points for correct plots

20 points for style (See the Style guide under Content on ICON):

- 5 points for indenting -- See Style Guide
- 5 points for in-line comments -- See Style Guide
- 5 points for comment blocks -- See Style Guide
- 5 points for meaningful variable names -- See Style Guide

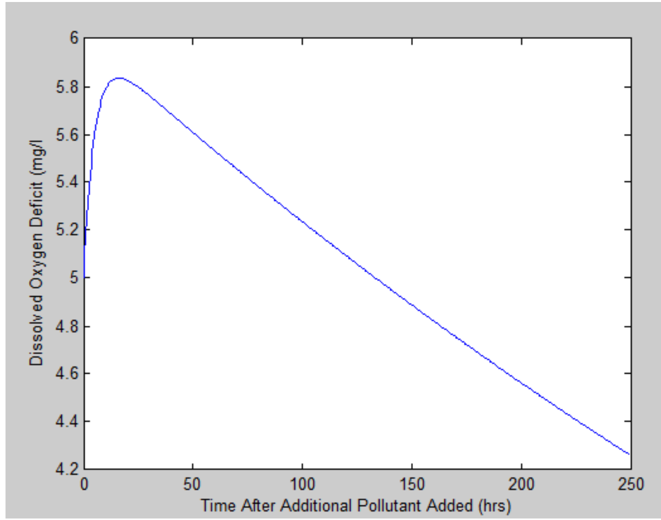

Figure 1 Example output from simulating a bolus pollutant added to a river stream as a function of time.

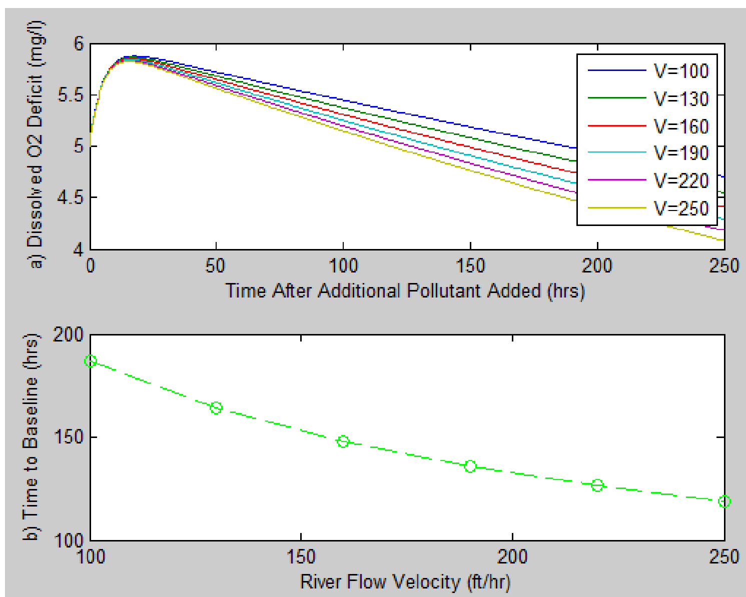

Figure 2 – Panel a) Dissolved oxygen deficit versus time for various values of river flow velocity; Panel b) Time to return to baseline pollution level versus river flow velocity.

NOTE: If you do not submit a MATLAB script file (file *myHW7.m*) you will receive a zero on your homework.

DO NOT WORK TOGETHER! Students caught working together on this assignment will receive a drop in their letter grade and a letter will be sent to the Dean's office.