CS1112 Spring 2015 Project 3 Part B due Thursday 3/5 at 11pm

(Part A appears in a separate document. Both parts have the same submission deadline.)

You must work either on your own or with one partner. If you work with a partner you must first register as a group in CMS and then submit your work as a group. *Adhere to the Code of Academic Integrity.* For a group, "you" below refers to "your group." You may discuss background issues and general strategies with others, but the work that you submit must be your own. In particular, you may discuss general ideas with others but you may not work out the detailed solutions with others. It is not OK for you to see or hear another student's code and it is certainly not OK to copy code from another person or from published/Internet sources. If you feel that you cannot complete the assignment on you own, seek help from the course staff.

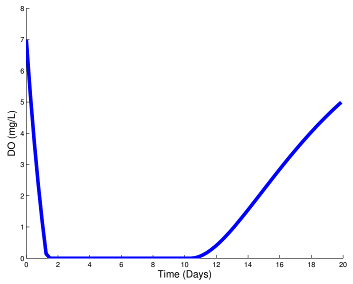
2 Expanded Dissolved Oxygen Model

In Project 2 you simulated the dissolved oxygen (DO) in rivers for the aerobic case using the Streeter-Phelps DO equation:

$$c(t) = c_s - \frac{L_0 K_d}{K_a - K_r}[\exp(-K_r t) - \exp(-K_a t)] - (c_s - c_0)\exp(-K_a t) \qquad (1)$$

where t is **t**ime since the entry of the pollutant load,
$c(t)$ is the DO **c**oncentration at time t,
c_s is the DO **s**aturation concentration (maximum possible DO given temperature and pressure),
c_0 is the initial DO concentration,
L_0 is the pollutant **L**oad into a river,
K_d is the rate of **d**ecay of pollutants through biological processes,
K_r is the rate of **r**emoval of pollutants through biological processes and settling, and
K_a is the rate of re**a**eration.

Equation (1) applies only in an aerobic environment. If DO drops to zero, an anaerobic period begins during which the pollutant continues to decompose anaerobically. Eventually DO becomes available again, returning the river to an aerobic environment. The DO profile then has the shape shown on the right.



A different set of equations apply during the anaerobic period to determine the remaining pollutant load at the start of anaerobic condition (L_i), the duration of anaerobic condition ($T_{anaerobic}$), and the pollutant load remaining at the end of the anaerobic period (L_f):

$$\begin{aligned} L_i &= L_0 \exp(-K_r t_i) && (2) \\ T_{anaerobic} &= \frac{K_d L_i - K_a c_s}{K_d K_a c_s} && (3) \\ L_f &= L_i - K_a c_s T_{anaerobic} && (4) \end{aligned}$$

where t_i is the time between the pollution event and the start of the anaerobic period.

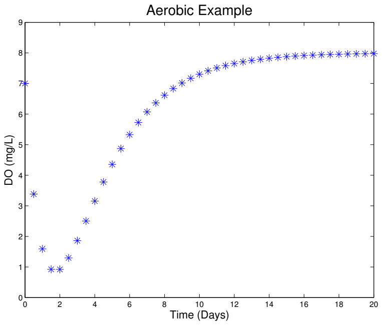
2.1 A function to simulate DO

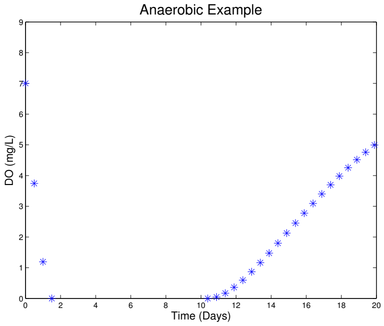
Implement the function `SPModel` to determine the DO profile following a pollution event based on the expanded DO model described:

```
function [do, time] = SPModel(dt,tf,KD,KR,KA,cs,c0,L)
% Return DO profile data based on the expanded Streeter-Phelps model.
% do: vector of DO concentration during simulation period
% time: vector of time corresponding to do vector
% dt: simulation time step FOR AEROBIC CONDITION, in days
% tf: duration of simulation, in days
% KD: decay rate, /day
% KR: removal rate, /day
% KA: reaeration rate, /day
% cs: saturation DO, mg/L
% c0: initial DO (at time of pollution event), mg/L
% L : pollutant load
```

The two returned vectors, `do` and `time`, have the same length and `do(j)` is the DO concentration at time `time(j)`, where j is a valid index for the vector. `time(1)` stores the value zero and `do(1)` stores the value c_0 (initial DO). This function only *returns* the data for the DO profile (the two vectors); this function *does not plot* the profile.

The increment between neighboring values in `time` is constant *only in the aerobic part of the profile.* Suppose the ith component of vector `time` stores the time of the beginning of the anaerobic period, then the $(i+1)$th component in `time` must be the time of the end of the anaerobic period. This is demonstrated by the figures below which plot the DO profile data using asterisks. Observe that for the anaerobic case, there are no calculated DO values during anaerobic condition (because DO is just zero—no calculation is needed).





Details and Hints

The key is to "build" the `time` and `do` vectors as the simulation proceeds. The anaerobic condition requires further thoughts . . .

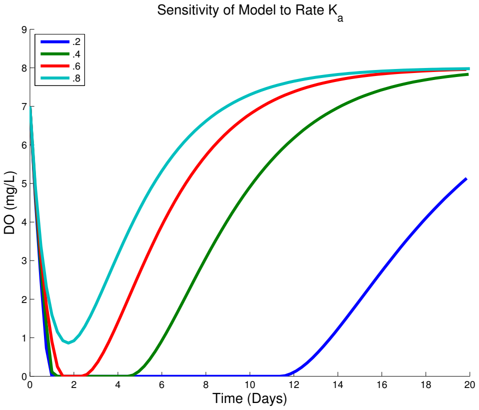
- When does it start? A zero or negative DO value calculated from equation (1) indicates the beginning of anaerobic condition. Set that DO value to zero (if it's negative) in vector `do`.
- Use equations (2) and (3) to find the duration of anaerobic condition. Then assign to the next component in vector `time` the time of the end of anaerobic condition. The corresponding component in `do` gets a zero.
- Use equation (4) to find the remaining pollutant load; this becomes the pollutant load L_0 in equation (1) when simulation under aerobic condition resumes. Don't forget to set c_0 to zero (the new initial DO). Note that t in equation (1) is the time since the entry of the pollutant load. As you have a "new" pollutant load L_0 for the resumption of aerobic condition, you need to reset t as well: the first application of equation (1) *post*-anaerobic condition uses the value `dt` (simulation time step for aerobic condition) in t.
- The above discussion points out that you need to keep track of (at least) two different kinds of "time": the simulation time (in vector `time`) and the time t associated with equation (1).
- The final time value in vector `time` should be close to `tf` (simulation duration). If there is anaerobic condition and its calculated end time is beyond `tf`, put a last value in `time` that is `tf` (and correspondingly put a last value in `do` as well).

You may write *sub*functions in `SPModel.m` if appropriate—avoid repeating blocks of similar looking code in your function. Test your function thoroughly! Simply write a script to call your function with some reasonable arguments and then make a plot with the returned vectors. For example, assuming that vectors `time` and `do` have the same length, then the command `plot(time, do)` will plot `do` versus `time` (`time` on x-axis). The two plots above were obtained based on these values: `dt` is .5, `tf` is 20, `cs` is 8, `c0` is 7, and `L` is 30. The aerobic example has these rates: `KD` and `KR` are .35, and `KA` is .8. The anaerobic example has these rates: `KD` and `KR` are .25, and `KA` is .2. You should do more testing with other values as well.

2.2 Sensitivity Analysis

In a *sensitivity analysis*, one investigates how the model result responds to systematic variations in the model's parameter values. We will call the parameter values given above for the aerobic example the base case. In a simple sensitivity analysis, we pick one parameter and vary its value over a range while holding all other parameters constant. For example, the fragment below gives a visual analysis on how sensitive the DO profile is to changes in the reaeration rate K_a. It assumes that all parameters other than `KA` are fixed at constant values. Using base case values except for `KA`, the plot on the right is produced.

```
% Sensitivity of model to changes in KA
figure
axis([0 tf 0 cs+1])
hold all
i= 0;
for KA= .2:.2:.8  % Vary KA in the range of .2 to .8
    i= i+1;
    [do,t] = SPModel(dt,tf,KD,KR,KA,cs,c0,L);
    hka(i)= plot(t,do);
end
xlabel('Time (Days)')
ylabel('DO (mg/L)')
title('Sensitivity of Model to Rate K_a')
legend(hka, '.2', '.4', '.6', '.8', 'Location', 'NorthWest')
hold off
```



The command `plot` draws a graph and can return the "handle" of that graph, as shown above. Saving the handle of a graph in a variable allows us to later access that graph (through the handle) to make changes. The variable `hka` is a vector of "handles" and we use it in the `legend` command: the label '.2' is associated with the first handle, '.4' is associated with the second handle, . . . , etc. Note that the command `hold all` instead of the usual `hold on` is used. In addition to holding each call to `plot` on the same set of axes, `hold all` keeps the graph color changing instead of resetting to the default (blue) with every call to `plot`. To experiment with other plot format options such as font size, line width, and legend positioning, refer to Appendix A in *Insight*. Of particular interest for now are Appendix sections A6 and A11.

Write a script `DOAnalysis` to investigate visually the sensitivity of the DO model to changes in K_a, K_r, and L. You can use the code given above for K_a and write similar code for K_r and L. Don't worry about redundancy in this case—you will have three blocks of code that look quite similar but it's fine here as you have to deal with text display for each case.

Your script should produce three figures. (Each time the `figure` command is used a new figure window is opened.) Before starting each analysis be sure to re-assign base case values to the parameter(s) changed in the preceding analysis. To analyze the effect of changes in K_r, set K_d to .1 and then vary K_r over the range of .1 to .5. (K_d is usually smaller than K_r.) To analyze the effect of changes in L, vary its value over the range of 20 to 80.

Submit your function file `SPModel.m` and script file `DOAnalysis.m` on CMS before the deadline.