CS1026a: Assignment 2

Due: Wednesday, October 28th, 9pm
Weight: 8%

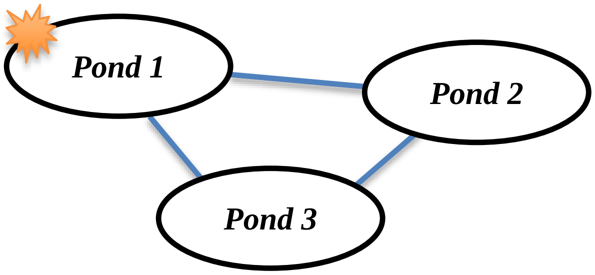
Purpose:
To gain experience with

- Loops in Python
- Use of functions in Python
- Use of Python code from other modules.

Task:

Using computers to simulate physical processes is a powerful tool in many disciplines. One of the advantages is that computers can simulate possible outcomes before they actually occur, allowing individuals to react. In this assignment, you will develop a program that will simulate a potential environmental problem.

An environmentalist has found a chemical spill in a network of ponds and wishes to understand the potential impact on the ponds if the spill continues for 24 hours (1440 minutes). The configuration of the ponds and location of the spill are depicted in the following figure:



Water flows from Pond 1 to Pond 2 and from Pond 2 to Pond 3 and from Pond 3 to Pond 1. The location of the spill is in Pond 1 (orange figure).

With some quick measurements and estimates, the environmentalist has determined the following:

1. The rate of flow between each pond is 0.005 liters per minute.
2. The amount of pollutant flowing between 2 ponds scales with the flowrate and the pollutant of the source pond. ie pollutantflow = rateofflow*pollutant.
3. The rate of pollutant into Pond 1 is 0.125 liters per minute and that there is a maximum of 40 liters of pollutant.
4. The amount of pollutant in Pond 1 at time *t* is $x1_t$ and is estimated by:

$$x1_t = x1_{t-1} + Inflow3_{t-1} - Outflow2_{t-1} + LeakRate_t$$

where:
- $Inflow3_{t-1}$ is the amount of pollutant that flowed from Pond 3 to Pond 1 at time *t-1*; it is

computed by:

$Inflow3_{t-1} = 0.005*Pollutant3_{t-1}$

where $Pollutant3_{t-1}$ is the amount of pollutant in Pond 3 at time *t-1*.

- $Outflow2_{t-1}$ is the amount of pollutant that flowed from Pond 1 to Pond 2 at time *t-1*; it is computed by:

$Outflow2_{t-1} = 0.005*Pollutant1_{t-1}$

where $Pollutant1_{t-1}$ is the amount of pollutant in Pond 1 at time *t-1*.

- $LeakRate_t$ is the amount of pollutant that leaked into Pond 1 at **time *t*** (i.e., 0.125 for as long as there pollutant left).

5. The amount of pollutant in Pond 2 at time *t* is $x2_t$ and is estimated by:

$x2_t = x2_{t-1} + Inflow1_{t-1} – Outflow3_{t-1}$

where:
- $Inflow1_{t-1}$ is the amount of pollutant that flowed from Pond 1 to Pond 2 at time *t-1*; it is computed by:

$Inflow1_{t-1} = 0.005*Pollutant1_{t-1}$

where $Pollutant1_{t-1}$ is the amount of pollutant in Pond 1 at time *t-1*.

- $Outflow3_{t-1}$ is the amount of pollutant that flowed from Pond 2 to Pond 3 at time *t-1*; it is computed by:

$Outflo3_{t-1} = 0.005*Pollutant2_{t-1}$

where $Pollutant2_{t-1}$ is the amount of pollutant in Pond 2 at time *t-1*.

6. The amount of pollutant in Pond 3 at time *t* is $x3_t$ and is estimated by:

$x3_t = x3_{t-1\ +}\ Inflow2_{t-1} – Outflow1_{t-1}$

where:
- $Inflow2_{t-1}$ is the amount of pollutant that flowed from Pond 2 to Pond 3 at time *t-1*; it is computed by:

$Inflow2_{t-1} = 0.005*Pollutant2_{t-1}$

where $Pollutant2_{t-1}$ is the amount of pollutant in Pond 2 at time *t-1*.

- $Outflow1_{t-1}$ is the amount of pollutant that flowed from Pond 3 to Pond 1 at time *t-1*; it is computed by:

  $Outflow1_{t-1} = 0.005*Pollutant3_{t-1}$

  where $Pollutant3_{t-1}$ is the amount of pollutant in Pond 3 at time *t-1*.

7. At time t0, the assumption is that the amount of pollutant in each pond is 0 (that is, $x1_0 = 0$, $x2_0= 0$, $x3_0 = 0$).

Functional Specifications:

Part A:
In this part of the assignment, you should focus on developing the code for the simulation.

1. Your program should simulate the flow and level of pollutant in the three ponds on a minute-by-minute basis; your program should use a loop.
2. You should print out the level of pollutant in each pond every hour, i.e at minutes 60,120,180,etc
3. You should print out the final level of pollutants in each pond at the end of the simulation
4. You should focus on using a *separate function* to compute the pollutant for each pond at time t given the flowrates, the level of pollutant in each pond at time t-1 and the amount of pollutant that leaked into pond 1 at t-1.
5. You are encouraged to use other functions as helpers as you see fit.

6. Once you have your simulation running well for the above example (40L, 1440 minutes, 0.125 leakRate). You should add functionality prompt the user (our environmentalist) for:
   - The maximum amount of pollutant.
   - The rate at which the pollutant is leaking.
   - The number of minutes to run the simulation; you must show at least one execution with a simulation time of 1440 minutes. You may try other times to see the effects of the pollutant over time.

This way your program can be used for more than one calculation.

Part B:
Once you have implemented your simulation and added user inputs , you can extend the program to graphical display the pollutant levels. This part of the assignment will make use of the given Python code in graphics.py and plotpoints.py; these can be found on the course web site for Assignment #2. Feel free to create a way to represent the data visually as you see fit. Use of plotpoints would be ideal however if you have a clean and creative way to display the information that is fine too. You can use the plotpoints to show the changing pollutant levels over time. That is to say, you can make a plot that shows what the pollutant value of each pond is every hour

- The Python code graphics.py is the graphics package described in the book; you can look in the text for examples (see Chapter 2, section 2.6).

- The Python code plotpoints.py does simple graph plotting using graphics.py. It makes a number of assumptions, so make sure to read the comments in the code about the parameters and limitations of the functions in plotpoints.py. It contains two functions:

- createGrid which will create a grid in a graphics window; it assumes a window that is 500-by-500.
- drawDots which will draw dots (filled circles) on a grid created by createGrid.

There is code in plotpoints.py that has been commented out – if you uncomment the lines of code, you can execute the test cases of plotpoints,py. Make sure that you put the comments back when you wish to use it with your simulation.

Non-functional Specifications:

1. Include brief comments in your code identifying yourself, describing the program, and describing key portions of the code.
2. Assignments are to be done individually and must be your own work. Software will be used to detect cheating.
3. Use Python coding conventions and good programming techniques, for example:
   i. Meaningful variable names
   ii. Conventions for naming variables and constants
   iii. Use of constants where appropriate
   iv. Readability: indentation, white space, consistency

What You Will Be Marked On:

1. Functional specifications:
   - Does the program behave according to specifications?
   - Is the output according to specifications?
2. Non-functional specifications: as described above
3. Assignment submission: via the OWL, though the assignment submission in OWL.